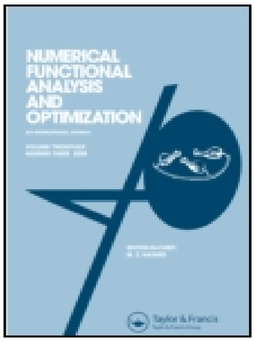

Numerical Functional Analysis and Optimization

ISSN: 0163-0563 (Print) 1532-2467 (Online) Journal homepage: http://www.tandfonline.com/loi/lnfa20

An Efficient Algorithm for Min-Max Convex Semi-Infinite Programming Problems

Liping Zhang & Soon-Yi Wu

To cite this article: Liping Zhang & Soon-Yi Wu (2016) An Efficient Algorithm for Min-Max Convex Semi-Infinite Programming Problems, Numerical Functional Analysis and Optimization, 37:8, 1037-1053, DOI: 10.1080/01630563.2016.1191033

To link to this article: http://dx.doi.org/10.1080/01630563.2016.1191033

Accepted author version posted online: 26 May 2016.
Published online: 26 May 2016.

NUMERICAL FUNCTIONAL ANALYSIS AND OPTIMIZATION
2016, VOL. 37, NO. 8, 1037–1053
http://dx.doi.org/10.1080/01630563.2016.1191033

Taylor & Francis
Taylor & Francis Group

An Efficient Algorithm for Min-Max Convex Semi-Infinite Programming Problems

Liping Zhang[a] and Soon-Yi Wu[b]

[a]Department of Mathematical Sciences, Tsinghua University, Beijing, China; [b]Department of Mathematics, National Cheng Kung University, Tainan, Taiwan

ABSTRACT
In this article, we consider the convex min-max problem with infinite constraints. We propose an exchange method to solve the problem by using efficient inactive constraint dropping rules. There is no need to solve the maximization problem over the metric space, as the algorithm has merely to find some points in the metric space such that a certain criterion is satisfied at each iteration. Under some mild assumptions, the proposed algorithm is shown to terminate in a finite number of iterations and to provide an approximate solution to the original problem. Preliminary numerical results with the algorithm are promising. To our knowledge, this article is the first one conceived to apply explicit exchange methods for solving nonlinear semi-infinite convex min-max problems.

ARTICLE HISTORY
Received 3 November 2014
Revised 16 March 2016
Accepted 14 May 2016

KEYWORDS
Convex programming; exchange method; semi-infinite programming

MATHEMATICS SUBJECT CLASSIFICATION
90C30; 49D39; 65K05; 65D15

1. Introduction

In this article, we consider the following nonlinear min-max problem with infinite constraints:

$$(P) \qquad \min_{\boldsymbol{x}\in\mathbb{R}^n} \ f(\boldsymbol{x}) := \max\{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_\ell(\boldsymbol{x})\}$$
$$s.t. \ \ g(\boldsymbol{x}, \boldsymbol{\omega}) \le 0, \forall \boldsymbol{\omega} \in \Omega,$$

which is called the semi-infinite min-max problem. Throughout this study, the following assumptions about the data in (P) are made.

Assumption 1.1.
(a) Ω is a compact and nonempty subset of $\mathbb{R}^m$;
(b) $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, \ell$ are convex and continuously differentiable on $\mathbb{R}^n$ and are not equal to each other;
(c) $g : \mathbb{R}^n \times \Omega \to \mathbb{R}$ is continuous on $\mathbb{R}^n \times \Omega$, $g(\cdot, \boldsymbol{\omega})$ is convex for all $\boldsymbol{\omega} \in \Omega$, and $\nabla_x g(\boldsymbol{x}, \boldsymbol{\omega})$ exists and is continuous on $\mathbb{R}^n \times \Omega$;
(d) There is a Slater point $\hat{\boldsymbol{x}} \in \mathbb{R}^n$ such that $g(\hat{\boldsymbol{x}}, \boldsymbol{\omega}) < 0$ for all $\boldsymbol{\omega} \in \Omega$.

CONTACT Liping Zhang lzhang@math.tsinghua.edu.cn Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China.

We call problems in the form of (P) with Assumption 1.1 min-max convex semi-infinite programming problems. In the particular case that $\ell = 1$, (P) is an ordinary convex semi-infinite programming (CSIP) problem. Notice that although the function f is piecewise smooth and locally Lipschitz continuous, it is not differentiable. Hence, it is important to find an approach to solving the min-max CSIP problem (P). See [9] for more information on min-max optimization problems.

Min-max semi-infinite programming problems arise in various engineering applications. For example, in civil engineering, electronic circuit design, and optimal control in robot path planning (see, e.g., [2–4, 6, 7, 10, 12]). A lot of literature deals with solution methods for solving these types of problems. For example, Polak et al. [11, 12] proposed algorithms with smoothing techniques for solving finite and semi-infinite min-max problems. Auslender et al. [1] proposed penalty and smoothing methods for solving min-max CSIP in the form of (P) where the number of $f_i(x)$ is infinite. Obviously, an ordinary CSIP problem is the special case of (P). Many solution methods were presented for solving semi-infinite programming problems (SIPs) [7, 13]. The main idea used in solution methods is to replace (P) by a sequence of finite programming problems, i.e., problems with only a finite number of constraints. According to the way that finite problems are generated, there are three important types of numerical methods for solving SIPs including discretization methods, local reduction based methods, and exchange methods (see [13]). Discretization methods (see, e.g., [15, 16]) have the advantage of internally working with finite subsets of Ω only. However, they are computationally costly, and the cost per iteration increases dramatically as the cardinality of the auxiliary problem grows. Globally convergent reduction based methods (see, e.g., [5, 17]), on the other hand, require strong assumptions and are often conceptual methods which can merely be implemented in a rather simplified form. The exchange method (see, e.g., [8, 18]) is one of the important methods beyond discretization and reduction based methods. For linear SIP, Lai and Wu [8] proposed an explicit algorithm in which they solved a linear programming with a finite feasible set Ω_k. They drop out redundant points in Ω_k at each iteration and only keep active points. Hence, the algorithm is very efficient in saving computational time. Similar to the exchange method, an iterative method was recently proposed in [14] for solving the KKT system of SIP in which some redundant points were dropped at certain iterations. Recently, Zhang, Wu and López [19] proposed a new exchange method for solving convex SIP based on the algorithm in [8]. It is shown that the algorithm provides an approximate solution after a finite number of iterations under weaker conditions.

In this article, motivated by the ideas in [8, 14, 19], we design an exchange method for solving the min-max CSIP (P). However, our analysis techniques are quite different from the ones used in [8, 14], because the objective function is of the min-max form, the infinite constraints functions are nonlinear, and the nonnegative constraint does not exist on which the analysis techniques in [8] mainly depended. We note that our algorithm introduces a relaxed scheme which does not require solving the global maximization problem with respect to $\boldsymbol{\omega} \in \Omega$ at each iteration. Our algorithm has merely to find some $\boldsymbol{\omega} \in \Omega$ such that a certain criterion with small scalar $\rho > 0$ is satisfied. We prove that the algorithm terminates in a finite number of iterations and the output is an approximate optimal solution of (P). Namely, we show that the obtained solution converges to the solution of (P) as ρ tends to zero.

This article is organized as follows. We present the algorithm in Section 2 and give a convergence analysis in Section 3. In Section 4, we give some numerical results. We conclude the article with some remarks in Section 5.

2. Algorithm description

In this section, motivated by ideas in [8, 14], we propose an exchange algorithm to solve the min-max CSIP problem (P).

Although (P) is also regarded as an ordinary CSIP problem satisfying the Slater constraint qualification, it is not easy to solve by directly using algorithms for CSIP because $f(x)$ is not continuously differentiable. Introducing an artificial variable $\boldsymbol{x}_{n+1} \in \mathbb{R}$, we equivalently reformulate (P) as the following $n+1$-dimensional minimization problem:

$$\begin{array}{lll} P[\Omega] & \min & \boldsymbol{x}_{n+1} \\ & s.t. & f_i(\boldsymbol{x}) \leq \boldsymbol{x}_{n+1}, \quad i = 1, 2, \ldots, \ell, \\ & & g(\boldsymbol{x}, \boldsymbol{\omega}) \leq 0, \forall \boldsymbol{\omega} \in \Omega. \end{array}$$

Obviously, problem $P[\Omega]$ is an ordinarily smoothing CSIP problem. This fact allows us to develop iterative methods based on problem $P[\Omega]$ to solve the original problem (P) without facing the non-differentiability of $f(\boldsymbol{x})$.

We now present an efficient exchange algorithm based on the auxiliary problem $P[\Omega]$. The exchange algorithm solves a finitely constrained convex programming at each iteration. Associated with each finite subset $B \subset \Omega$, we define the finitely constrained convex program by

$$\begin{array}{lll} P[B]: & \min & \boldsymbol{x}_{n+1} \\ & s.t. & f_i(\boldsymbol{x}) \leq \boldsymbol{x}_{n+1}, \quad i = 1, 2, \ldots, \ell, \\ & & g(\boldsymbol{x}, \boldsymbol{\omega}) \leq 0, \quad \forall \boldsymbol{\omega} \in B. \end{array}$$

We can easily establish the KKT conditions for $P[B]$, which is written as

$$\begin{aligned} &\sum_{i=1}^{\ell} \lambda_i = 1, \\ &\sum_{i=1}^{\ell} \lambda_i \nabla f_i(\boldsymbol{x}) + \sum_{\boldsymbol{\omega} \in B} \nu(\boldsymbol{\omega}) \nabla_x g(\boldsymbol{x}, \boldsymbol{\omega}) = 0, \end{aligned} \tag{1}$$

$$\lambda_i(f_i(\boldsymbol{x}) - \boldsymbol{x}_{n+1}) = 0, \quad \lambda_i \geq 0, \quad f_i(\boldsymbol{x}) \leq \boldsymbol{x}_{n+1}, \quad i = 1, \ldots, \ell,$$
$$\nu(\boldsymbol{\omega})g(\boldsymbol{x}, \boldsymbol{\omega}) = 0, \quad \nu(\boldsymbol{\omega}) \geq 0, \quad g(\boldsymbol{x}, \boldsymbol{\omega}) \leq 0, \quad \forall \boldsymbol{\omega} \in B.$$

Here, $\nu(\boldsymbol{\omega})$, $\boldsymbol{\omega} \in B$, and $\lambda_i, i = 1, \ldots, \ell$ can be regarded as the Lagrange multipliers of problem P[B].

Let $\hat{\boldsymbol{x}} \in \mathbb{R}^n$ be the Slater point in Assumption 1.1(d). Choose a scalar $\hat{\boldsymbol{x}}_{n+1}$ such that $\hat{\boldsymbol{x}}_{n+1} > \max\{f_1(\hat{\boldsymbol{x}}), \ldots, f_\ell(\hat{\boldsymbol{x}})\}$. Then we have $f_i(\hat{\boldsymbol{x}}) < \hat{\boldsymbol{x}}_{n+1}, i = 1, \ldots, \ell$. Hence, under Assumption 1.1(d) there is a Slater point for problem $P[\Omega]$. Consequently, conditions (1) turn out to be necessary and sufficient optimality conditions for problem $P[B]$.

Theorem 2.1. *Let $(\boldsymbol{x}^*, \boldsymbol{x}^*_{n+1}) \in \mathbb{R}^{n+1}$ be a feasible solution for problem $P[B]$ and Assumption 1.1 be satisfied. Then, $(\boldsymbol{x}^*, \boldsymbol{x}^*_{n+1})$ is optimal if and only if there exist Lagrange multipliers $\{\nu^*(\boldsymbol{\omega}) | \boldsymbol{\omega} \in B\}$ and $\{\lambda^* | \ i = 1, \ldots, \ell\}$ such that conditions (1) are satisfied.*

The details of the algorithm are described as follows.

Algorithm 2.1 (An efficient exchange method).

Step 0. Choose finite points $\Omega_0 = \{\boldsymbol{\omega}^0_1, \ldots, \boldsymbol{\omega}^0_{m_0}\}$ and a sufficiently small number $\rho > 0$. Solve problem $P[\Omega_0]$ to obtain its optimum $(\boldsymbol{x}^0, \boldsymbol{x}^0_{n+1})$. Set $k := 0$.

Step 1. Find a $\boldsymbol{\omega}^k_{new} \in \Omega$ such that

$$g(\boldsymbol{x}^k, \boldsymbol{\omega}^k_{new}) > \rho. \tag{2}$$

If such a $\boldsymbol{\omega}^k_{new}$ does not exist, then stop. Otherwise, let

$$\bar{\Omega}_{k+1} := \Omega_k \cup \{\boldsymbol{\omega}^k_{new}\}.$$

Step 2. Solve problem $P[\bar{\Omega}_{k+1}]$ to obtain its optimum $(\boldsymbol{x}^{k+1}, \boldsymbol{x}^{k+1}_{n+1})$ and the corresponding Lagrange multipliers $\{\nu^{k+1}(\boldsymbol{\omega}) | \boldsymbol{\omega} \in \bar{\Omega}_{k+1}\}$ and $\{\lambda^{k+1}_i | \ i = 1, \ldots, \ell\}$.

Step 3. Let

$$\Omega_{k+1} := \{\boldsymbol{\omega} \in \bar{\Omega}_{k+1} | \nu^{k+1}(\boldsymbol{\omega}) > 0\}.$$

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

There are some remarks for Algorithm 2.1:

(a) Steps 1–3 are the main differences from the algorithms of [8, 14].

(b) In Step 1, it is also possible to choose multiple elements satisfying (2). Although we merely deal with the single-point exchange scheme in the following analysis, the obtained results are also applicable to multiple exchange type algorithms.

(c) In Step 2, the optimal solution $(\boldsymbol{x}^{k+1}, \boldsymbol{x}_{n+1}^{k+1})$ of problem $P[\bar{\Omega}_{k+1}]$ also solves problem $P[\Omega_{k+1}]$.

(d) In Step 3, all inactive constraints at the optimum $(\boldsymbol{x}^{k+1}, \boldsymbol{x}_{n+1}^{k+1})$ are removed since the KKT conditions of problem $P[\bar{\Omega}_{k+1}]$ implies that $\nu^{k+1}(\boldsymbol{\omega}) = 0$ for any $\boldsymbol{\omega} \in \bar{\Omega}_{k+1}$ with $g(\boldsymbol{x}^{k+1}, \boldsymbol{\omega}) < 0$.

We now define some notations for convenience in analyzing the convergence properties of Algorithm 2.1. Let v^* denote the optimal value of problem $P[\Omega]$, and let $\{(\boldsymbol{x}^k, \boldsymbol{x}_{n+1}^k)\}$, $\{\nu^k(\boldsymbol{\omega}) | \boldsymbol{\omega} \in \Omega_k\}$, and $\{\lambda_i^k | i = 1, \dots, \ell\}$ be the sequences of optimal solution and corresponding Lagrangian multipliers generated by Algorithm 2.1. For $k = 1, 2, \dots$, we define

$$\boldsymbol{d}^k := \boldsymbol{x}^{k+1} - \boldsymbol{x}^k, \quad \theta_i^k := \boldsymbol{x}_{n+1}^k - f_i(\boldsymbol{x}^k), \quad i = 1, \dots, \ell. \tag{3}$$

Then, by Assumption 1.1(d) and KKT conditions (1) for problem $P[\Omega_k]$, we have

$$\begin{aligned}
&\sum_{i=1}^{\ell} \lambda_i^k = 1, \\
&\sum_{i=1}^{\ell} \lambda_i^k \nabla f_i(\boldsymbol{x}^k) + \sum_{\boldsymbol{\omega} \in \Omega_k} \nu^k(\boldsymbol{\omega}) \nabla_x g(\boldsymbol{x}^k, \boldsymbol{\omega}) = 0, \\
&\lambda_i^k \theta_i^k = 0, \quad \lambda_i^k \geq 0, \quad \theta_i^k \geq 0, \quad i = 1, \dots, \ell, \\
&\nu^k(\boldsymbol{\omega}) g(\boldsymbol{x}^k, \boldsymbol{\omega}) = 0, \quad \nu^k(\boldsymbol{\omega}) \geq 0, \quad g(\boldsymbol{x}^k, \boldsymbol{\omega}) \leq 0, \quad \forall \boldsymbol{\omega} \in \Omega_k.
\end{aligned} \tag{4}$$

We also define

$$\begin{aligned}
D_i^k &:= f_i(\boldsymbol{x}^{k+1}) - f_i(\boldsymbol{x}^k) - \nabla f_i(\boldsymbol{x}^k)^T \boldsymbol{d}^k, \quad i = 1, \dots, \ell, \\
G_i^k &:= f_i(\boldsymbol{x}^k) - f_i(\boldsymbol{x}^{k+1}) + \nabla f_i(\boldsymbol{x}^{k+1})^T \boldsymbol{d}^k, \quad i = 1, \dots, \ell, \\
S^k(\boldsymbol{\omega}) &:= g(\boldsymbol{x}^{k+1}, \boldsymbol{\omega}) - g(\boldsymbol{x}^k, \boldsymbol{\omega}) - \nabla_x g(\boldsymbol{x}^k, \boldsymbol{\omega})^T \boldsymbol{d}^k, \\
H^k(\boldsymbol{\omega}) &:= g(\boldsymbol{x}^k, \boldsymbol{\omega}) - g(\boldsymbol{x}^{k+1}, \boldsymbol{\omega}) + \nabla_x g(\boldsymbol{x}^{k+1}, \boldsymbol{\omega})^T \boldsymbol{d}^k.
\end{aligned} \tag{5}$$

Then, by using the Taylor expansion and the convexity of f_i and $g(\cdot, \boldsymbol{\omega})$ for $\boldsymbol{\omega} \in \Omega$, we obtain

$$\begin{aligned}
&0 \leq D_i^k = o(\|\boldsymbol{d}^k\|), \quad 0 \leq G_i^k = o(\|\boldsymbol{d}^k\|), \quad i = 1, \dots, \ell, \\
&0 \leq S^k(\boldsymbol{\omega}) = o(\|\boldsymbol{d}^k\|), \quad 0 \leq H^k(\boldsymbol{\omega}) = o(\|\boldsymbol{d}^k\|).
\end{aligned} \tag{6}$$

Let $v(\Omega_k)$ denote the optimal value of problem $P[\Omega_k]$. From Step 1 of Algorithm 2.1, it is easy to see that

$$\bar{\Omega}_{k+1} \supseteq \Omega_k,$$

which implies that the feasible region of problem $P[\bar{\Omega}_{k+1}]$ is contained in that of problem $P[\Omega_k]$. Hence, from **(c)** we have

$$v(\Omega_{k+1}) = v(\bar{\Omega}_{k+1}) \geq v(\Omega_k).$$

Consequently, we immediately obtain the following theorem.

Theorem 2.2. *The sequence of optimal values* $\{v(\Omega_k)\}$ *of* $\{P[\Omega_k]\}$ *is nondecreasing, i.e.,*

$$v(\Omega_{k+1}) \geq v(\Omega_k) \quad \text{for } k = 1, 2, \ldots.$$

The following theorem is very important for analyzing the convergence properties of Algorithm 2.1, because it evaluates the increment of the optimal value $v(\Omega_k)$ of problem $P[\Omega_k]$ at each iteration.

Theorem 2.3. *For* $k = 1, 2, \ldots$, *we have*

$$\begin{aligned} v(\Omega_{k+1}) - v(\Omega_k) &= \sum_{i=1}^{\ell} \lambda_i^k \left(\theta_i^{k+1} + D_i^k\right) + \sum_{\omega \in \Omega_k} \nu^k(\omega) \left(S^k(\omega) - g(x^{k+1}, \omega)\right) \\ &= g(x^k, \omega_{new}^k)\nu^{k+1}(\omega_{new}^k) - \sum_{\omega \in \Omega_{k+1}} \nu^{k+1}(\omega) H^k(\omega) \\ &\quad - \sum_{i=1}^{\ell} \lambda_i^{k+1}(G_i^k + \theta_i^k). \end{aligned} \tag{7}$$

Proof. First, we prove that the first equality in (7) is satisfied. From (4) we have

$$\begin{aligned} v(\Omega_{k+1}) - v(\Omega_k) &= \sum_{i=1}^{\ell} \lambda_i^k x_{n+1}^{k+1} - \sum_{i=1}^{\ell} \lambda_i^k f_i(x^k) \\ &= \sum_{i=1}^{\ell} \lambda_i^k \theta_i^{k+1} + \sum_{i=1}^{\ell} \lambda_i^k (D_i^k + \nabla f_i(x^k)^T d^k) \\ &= \sum_{i=1}^{\ell} \lambda_i^k (\theta_i^{k+1} + D_i^k) - \sum_{\omega \in \Omega_k} \nu^k(\omega) \nabla_x g(x^k, \omega)^T d^k \\ &= \sum_{i=1}^{\ell} \lambda_i^k (\theta_i^{k+1} + D_i^k) + \sum_{\omega \in \Omega_k} \nu^k(\omega) \left(S^k(\omega) - g(x^{k+1}, \omega)\right), \end{aligned} \tag{8}$$

where the first and the third equalities follow from (4), the second one follows from (3) and (5), and the last one holds due to (5) and $g(x^k, \omega) = 0$ for $\omega \in \Omega_k$. Hence, the first equality in (7) holds.

Next, we show the validity of the second equality in (7). By $g(x^k, \omega) = 0$ for $\omega \in \Omega_k$ and $\bar{\Omega}_{k+1} = \Omega_k \cup \{\omega_{new}^k\}$, we have

$$\begin{aligned} \sum_{\omega \in \bar{\Omega}_{k+1}} \nu^{k+1}(\omega) g(x^k, \omega) &= \sum_{\omega \in \Omega_k} \nu^{k+1}(\omega) g(x^k, \omega) + \nu^{k+1}(\omega_{new}^k) g(x^k, \omega_{new}^k) \\ &= \nu^{k+1}(\omega_{new}^k) g(x^k, \omega_{new}^k). \end{aligned} \tag{9}$$

On the other hand, it follows from $\Omega_{k+1} = \{\boldsymbol{\omega} \in \bar{\Omega}_{k+1} | \nu^{k+1}(\boldsymbol{\omega}) > 0\}$ that

$$
\begin{aligned}
&\sum_{\boldsymbol{\omega}\in\bar{\Omega}_{k+1}} \nu^{k+1}(\boldsymbol{\omega})g(\boldsymbol{x}^k,\boldsymbol{\omega}) \\
&\quad = \sum_{\boldsymbol{\omega}\in\Omega_{k+1}} \nu^{k+1}(\boldsymbol{\omega})(g(\boldsymbol{x}^{k+1},\boldsymbol{\omega}) + H^k(\boldsymbol{\omega}) - \nabla_x g(\boldsymbol{x}^{k+1},\boldsymbol{\omega})^T \boldsymbol{d}^k) \\
&\quad = \sum_{\boldsymbol{\omega}\in\Omega_{k+1}} \nu^{k+1}(\boldsymbol{\omega})H^k(\boldsymbol{\omega}) + \sum_{i=1}^{\ell} \lambda_i^{k+1} \nabla f_i(\boldsymbol{x}^{k+1})^T \boldsymbol{d}^k \\
&\quad = \sum_{\boldsymbol{\omega}\in\Omega_{k+1}} \nu^{k+1}(\boldsymbol{\omega})H^k(\boldsymbol{\omega}) + \sum_{i=1}^{\ell} \lambda_i^{k+1}(G_i^k + f_i(\boldsymbol{x}^{k+1}) - f_i(\boldsymbol{x}^k)) \\
&\quad = \boldsymbol{x}_{n+1}^{k+1} - \boldsymbol{x}_{n+1}^{k} + \sum_{\boldsymbol{\omega}\in\Omega_{k+1}} \nu^{k+1}(\boldsymbol{\omega})H^k(\boldsymbol{\omega}) + \sum_{i=1}^{\ell} \lambda_i^{k+1}(G_i^k + \theta_i^k),
\end{aligned} \tag{10}
$$

where the first and the third equalities follow from (5), the second one holds due to (4) and $g(\boldsymbol{x}^{k+1},\boldsymbol{\omega}) = 0$ for $\boldsymbol{\omega} \in \Omega_{k+1}$, and the last one is satisfied since

$$
\begin{aligned}
\sum_{i=1}^{\ell} \lambda_i^{k+1}(f_i(\boldsymbol{x}^{k+1}) - f_i(\boldsymbol{x}^k)) &= \sum_{i=1}^{\ell} \lambda_i^{k+1} \boldsymbol{x}_{n+1}^{k+1} + \sum_{i=1}^{\ell} \lambda_i^{k+1}(\theta_i^k - \boldsymbol{x}_{n+1}^k) \\
&= \boldsymbol{x}_{n+1}^{k+1} - \boldsymbol{x}_{n+1}^{k} + \sum_{i=1}^{\ell} \lambda_i^{k+1}\theta_i^k \\
&= v(\Omega_{k+1}) - v(\Omega_k) + \sum_{i=1}^{\ell} \lambda_i^{k+1}\theta_i^k,
\end{aligned}
$$

where the second equality follows from $\sum_{i=1}^{\ell} \lambda_i^{k+1} = 1$. Thus, equalities (9) and (10) imply that the second equality in (7) holds. □

3. Finite termination convergence analysis

In this section, we show that Algorithm 2.1 terminates in a finite number of iterations under some mild conditions. Furthermore, we prove that the output at the final iteration is sufficiently close to the optimal solution of (P) if the criterion value ρ is sufficiently close to zero.

Lemma 3.1. *For any given $k \in \{1, 2, \ldots\}$, we have*

$$v(\Omega_{k+1}) > v(\Omega_k), \quad \boldsymbol{\omega}_{new}^k \in \Omega_{k+1},$$

hold if $(\boldsymbol{x}^k, \boldsymbol{x}_{n+1}^k)$ is the unique optimal solution to problem $P[\Omega_k]$.

Proof. For any given $k \geq 1$, it follows from Theorem 2.2 that

$$\nu(\Omega_{k+1}) \geq \nu(\Omega_k).$$

Suppose, reasoning by contradiction, that there exists k_0 such that

$$\nu(\Omega_{k_0+1}) = \nu(\Omega_{k_0}). \tag{11}$$

Let $\mathcal{F}_{k_0}$ and $\bar{\mathcal{F}}_{k_0+1}$ be the feasible regions of $P[\Omega_{k_0}]$ and $P[\bar{\Omega}_{k_0+1}]$, respectively. Then, from Step 1 of Algorithm 2.1, we have

$$\mathcal{F}^{k_0} \supseteq \bar{\mathcal{F}}^{k_0+1},$$

which together with (11) implies that $(\boldsymbol{x}^{k_0+1}, \boldsymbol{x}_{n+1}^{k_0+1})$ is also an optimal solution to problem $P[\Omega_{k_0}]$. Since $(\boldsymbol{x}^{k_0}, \boldsymbol{x}_{n+1}^{k_0})$ is the unique optimal solution to problem $P[\Omega_{k_0}]$, it follows that $\boldsymbol{x}^{k_0+1} = \boldsymbol{x}^{k_0}$. Therefore, by (2) and Step 2 of Algorithm 2.1, we obtain the following contradiction:

$$0 \geq g(\boldsymbol{x}^{k_0+1}, \boldsymbol{\omega}_{new}^{k_0}) = g(\boldsymbol{x}^{k_0}, \boldsymbol{\omega}_{new}^{k_0}) > \rho > 0.$$

Thus, $\nu(\Omega_{k+1}) > \nu(\Omega_k)$.

To prove the fact that $\boldsymbol{\omega}_{new}^k \in \Omega_{k+1}$, it suffices to show $\nu^{k+1}(\omega_{new}^k) > 0$. It follows from (4) that $\nu^{k+1}(\omega_{new}^k) \geq 0$. Suppose, reasoning by contradiction, that there exists an integer $k_1 \geq 0$ such that $\nu^{k_1+1}(\boldsymbol{\omega}_{new}^{k_1}) = 0$. Then, the second equality in (7) and $\nu(\Omega_{k_1+1}) > \nu(\Omega_{k_1})$ imply that

$$-\sum_{\boldsymbol{\omega} \in \Omega_{k_1+1}} \nu^{k_1+1}(\boldsymbol{\omega}) H^{k_1}(\boldsymbol{\omega}) - \sum_{i=1}^{\ell} \lambda_i^{k_1+1}(G_i^{k_1} + \theta_i^{k_1}) > 0. \tag{12}$$

On the other hand, combining (4) and (6), we have

$$\nu^{k_1+1}(\boldsymbol{\omega}) > 0, \quad H^{k_1}(\boldsymbol{\omega}) \geq 0, \quad \forall \boldsymbol{\omega} \in \Omega_{k_1+1},$$
$$\lambda_i^{k_1+1} \geq 0, \quad G_i^{k_1} \geq 0, \quad \theta_i^{k_1} \geq 0, \quad i = 1, \ldots, \ell,$$

which contradicts (12). This completes the proof. □

There are sufficient conditions for the assumption in Lemma 3.1.

Lemma 3.2. *If either $f_i, i = 1, \ldots, \ell$ are strictly convex, or $g(\cdot, \boldsymbol{\omega}), \boldsymbol{\omega} \in \Omega$ is strictly convex, then the sequence $\{P[\Omega_k]\}$ of finite minimization subproblems generated by Algorithm 2.1 has the unique optimal solution $\{(\boldsymbol{x}^k, \boldsymbol{x}_{n+1}^k)\}$.*

Proof. For any given $k \geq 1$, let $(\boldsymbol{x}^k, \boldsymbol{x}_{n+1}^k)$ and $(\hat{\boldsymbol{x}}^k, \hat{\boldsymbol{x}}_{n+1}^k)$ be optimal solutions to problem $P[\Omega_k]$. Then,

$$\hat{\boldsymbol{x}}_{n+1}^k = \boldsymbol{x}_{n+1}^k. \tag{13}$$

Define $\boldsymbol{y}^k := \hat{\boldsymbol{x}}^k - \boldsymbol{x}^k$,

$$Q_i^k := f_i(\hat{\boldsymbol{x}}^k) - f_i(\boldsymbol{x}^k) - \nabla f_i(\boldsymbol{x}^k)^T \boldsymbol{y}^k, \quad i = 1, \ldots, \ell,$$

and

$$W^k(\boldsymbol{\omega}) := g(\hat{\boldsymbol{x}}^k, \boldsymbol{\omega}) - g(\boldsymbol{x}^k, \boldsymbol{\omega}) - \nabla_x g(\boldsymbol{x}^k, \boldsymbol{\omega})^T \boldsymbol{y}^k, \quad \boldsymbol{\omega} \in \Omega_k.$$

Then,

$$Q_i^k \geq 0,\ i = 1, \ldots, \ell, \quad W_i^k(\boldsymbol{\omega}) \geq 0,\ \forall \boldsymbol{\omega} \in \Omega_k. \tag{14}$$

Using a similar argument as (8), we obtain

$$\hat{\boldsymbol{x}}_{n+1}^k - \boldsymbol{x}_{n+1}^k = \sum_{i=1}^{\ell} \lambda_i^k \left(\hat{\theta}_i^k + Q_i^k \right) + \sum_{\boldsymbol{\omega} \in \Omega_k} \nu^k(\boldsymbol{\omega}) \left(W^k(\boldsymbol{\omega}) - g(\hat{\boldsymbol{x}}^k, \boldsymbol{\omega}) \right). \tag{15}$$

Since $\lambda_i^k \geq 0$, $\hat{\theta}_i^k := \hat{\boldsymbol{x}}_{n+1}^k - f_i(\hat{\boldsymbol{x}}^k) \geq 0$, $i = 1, \ldots, \ell$, and $g(\hat{\boldsymbol{x}}^k, \boldsymbol{\omega}) \leq 0$ for all $\boldsymbol{\omega} \in \Omega_k$, it follows from (13), (14), and (15) that

$$\sum_{i=1}^{\ell} \lambda_i^k Q_i^k = 0, \quad \sum_{\boldsymbol{\omega} \in \Omega_k} \nu^k(\boldsymbol{\omega}) W^k(\boldsymbol{\omega}) = 0,$$

which together with

$$\sum_{i=1}^{\ell} \lambda_i^k = 1,\ \lambda_i^k \geq 0,\ i = 1, \ldots, \ell, \quad \nu^k(\boldsymbol{\omega}) > 0,\ \forall \boldsymbol{\omega} \in \Omega_k,$$

implies that there exists i_0 such that

$$Q_{i_0}^k = 0, \quad W^k(\boldsymbol{\omega}) = 0,\ \forall \boldsymbol{\omega} \in \Omega_k. \tag{16}$$

Since either f_{i_0} or $g(\cdot, \boldsymbol{\omega})$ is strictly convex, it follows from (16) that $\boldsymbol{y}^k = 0$, i.e., $\boldsymbol{x}^k = \hat{\boldsymbol{x}}^k$. Namely, $(\boldsymbol{x}^k, \boldsymbol{x}_{n+1}^k)$ is the unique optimal solution to problem $P[\Omega_k]$. □

Theorem 3.1. *Suppose that either $f_i, i = 1, \ldots, \ell$ are strictly convex, or $g(\cdot, \boldsymbol{\omega})$ is strictly convex and there exist $k_0 > 0$ and $\delta > 0$ such that $\nu^k(\boldsymbol{\omega}) \geq \delta$ for all $\boldsymbol{\omega} \in \Omega_k$ and for $k \geq k_0$. If $\{\boldsymbol{x}^k\}$ is bounded, then Algorithm 2.1 terminates in a finite number of iterations.*

Proof. Suppose, reasoning by contradiction, that Algorithm 2.1 does not finitely stop. Then, by Theorem 2.2, we have

$$v^* \geq \boldsymbol{x}_{n+1}^{k+1} \geq \boldsymbol{x}_{n+1}^k, \quad k = 1, 2, \ldots,$$

which implies that

$$\lim_{k \to \infty} (\boldsymbol{x}_{n+1}^{k+1} - \boldsymbol{x}_{n+1}^k) = 0. \tag{17}$$

Since $\{\boldsymbol{x}^k\}$ is bounded, $\{\boldsymbol{\omega}_{new}^k\} \subset \Omega$ and Ω is compact, and

$$\lambda_i^k \geq 0,\ i = 1, \ldots, \ell, \quad \sum_{i=1}^{\ell} \lambda_i^k = 1,$$

we can assume, without loss of generality, that there exist $\bar{\boldsymbol{x}} \in \mathbb{R}^n$, $\bar{\boldsymbol{\omega}} \in \Omega$, $\bar{\boldsymbol{d}} \in \mathbb{R}^n$, and $0 \le \bar{\lambda}_i \le 1, i = 1, \ldots, \ell$ such that

$$\lim_{k\to\infty} \lambda_i^k = \bar{\lambda}_i, \quad \sum_{i=1}^{\ell} \bar{\lambda}_i = 1, \tag{18}$$

$$\lim_{k\to\infty} (\boldsymbol{x}^k, \boldsymbol{\omega}_{new}^k) = (\bar{\boldsymbol{x}}, \bar{\boldsymbol{\omega}}), \quad \lim_{k\to\infty} \boldsymbol{d}^k = \bar{\boldsymbol{d}}.$$

Hence, it follows from (2) that

$$\lim_{k\to\infty} g(\boldsymbol{x}^k, \boldsymbol{\omega}_{new}^k) = g(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\omega}}) \ge \rho > 0. \tag{19}$$

By (18), the first equality in (7), and (17), there exist i_0 with $\bar{\lambda}_{i_0} > 0$ such that

$$f_{i_0}(\bar{\boldsymbol{x}} + \bar{\boldsymbol{d}}) - f_{i_0}(\bar{\boldsymbol{x}}) - \nabla f_{i_0}(\bar{\boldsymbol{x}})^T \bar{\boldsymbol{d}} = 0. \tag{20}$$

On the other hand, if there exists $k_0 > 0$ and $\delta > 0$ such that $\nu^k(\boldsymbol{\omega}) \ge \delta$ for all $\boldsymbol{\omega} \in \Omega_k$ and for $k \ge k_0$, then it follows from (7) and (17) that there exists a point $\hat{\boldsymbol{\omega}} \in \Omega$ such that

$$g(\bar{\boldsymbol{x}} + \bar{\boldsymbol{d}}, \hat{\boldsymbol{\omega}}) - g(\bar{\boldsymbol{x}}, \hat{\boldsymbol{\omega}}) - \nabla_x g(\bar{\boldsymbol{x}}, \hat{\boldsymbol{\omega}})^T \bar{\boldsymbol{d}} = 0. \tag{21}$$

Therefore, either from strict convexity of f_{i_0} and (20) or from strict convexity of $g(\cdot, \hat{\boldsymbol{\omega}})$ and (21), we can obtain

$$\lim_{k\to\infty} \boldsymbol{d}^k = \bar{\boldsymbol{d}} = 0.$$

Thus for any sufficiently small $\varepsilon > 0$, there exists a large positive integer $N \ge k_0$ such that

$$\|\boldsymbol{x}^{N+1} - \boldsymbol{x}^N\| = \|\boldsymbol{d}^N\| < \varepsilon^2. \tag{22}$$

Lemma 3.1 and Lemma 3.2 imply that $\boldsymbol{\omega}_{new}^N \in \Omega_{N+1}$. This, together with (4), yields $g(\boldsymbol{x}^{N+1}, \boldsymbol{\omega}_{new}^N) = 0$. Since $\{\boldsymbol{x}^k\}$ is bounded and $\nabla_x g(\boldsymbol{x}, \boldsymbol{\omega})$ is continuous on $\mathbb{R}^n \times \Omega$, there exists a constant $c_0 > 0$ such that

$$\|\nabla_x g(\boldsymbol{x}^N, \boldsymbol{\omega}_{new}^N)\| \le c_0.$$

Hence, we obtain

$$\begin{aligned} g(\boldsymbol{x}^N, \boldsymbol{\omega}_{new}^N) &= g(\boldsymbol{x}^N, \boldsymbol{\omega}_{new}^N) - g(\boldsymbol{x}^{N+1}, \boldsymbol{\omega}_{new}^N) \\ &\le \nabla_x g(\boldsymbol{x}^N, \boldsymbol{\omega}_{new}^N)^T (\boldsymbol{x}^N - \boldsymbol{x}^{N+1}) \\ &\le \|\nabla_x g(\boldsymbol{x}^N, \boldsymbol{\omega}_{new}^N)\| \|\boldsymbol{x}^{N+1} - \boldsymbol{x}^N\| \\ &\le c_0 \varepsilon^2 \to 0, \quad \text{as } \varepsilon \to 0, \end{aligned}$$

where the first inequality holds due to convexity of $g(\cdot, \boldsymbol{\omega}_{new}^N)$, and the last one follows from (22). This contradicts (19). Thus Algorithm 2.1 terminates in a finite number of iterations. □

For any given $k \geq 1$, define index sets

$$I := \{1, 2, \ldots, \ell\}, \quad I_+^k := \{i \in I \mid \lambda_i^k > 0\}.$$

Let us also define a matrix A^k of $|I_+^k| \times n$:

$$A^k := (\cdots \nabla f_i(\boldsymbol{x}^k)^T \cdots)_{i \in I_+^k}^T.$$

We introduce the following assumption for discussing the convergence of Algorithm 2.1.

Assumption 3.1. *There exists an integer k_0 large enough, scalars $\sigma_1 > 0$ and $\sigma_2 > 0$ such that $\sigma_1 \geq \sigma_2$ and the following statements hold for all $k \geq k_0$.*
(a) $\{\boldsymbol{x}^k\}$ is bounded, and $\sigma_2 \leq \nu^k(\boldsymbol{\omega}) \leq \sigma_1$ for $\boldsymbol{\omega} \in \Omega_k$.
(b) $\lambda_i^k \geq \sigma_2$ for $i \in I_+^k$.
(c) $\mu_{min}((A^k)^T A^k) \geq \sigma_2$, where μ_{min} denotes the minimum eigenvalue.
(d) $\boldsymbol{\omega}_{new}^k \in \Omega_{k+1}$.

Note that, Assumption 3.1(a)–(c) are regularity conditions. Assumption 3.1(d) is also mild because the conditions in Lemma 3.1 and Lemma 3.2 all ensure (d) is satisfied.

Theorem 3.2. *Algorithm 2.1 finitely terminates if Assumption 3.1 holds.*

Proof. Suppose, reasoning by contradiction, that Algorithm 2.1 does not finitely terminate. Then, by Theorem 2.2 we have

$$\nu(\Omega_k) \leq \nu(\Omega_{k+1}) \leq \nu^*, \quad k = 1, 2, \ldots,$$

which implies that

$$\lim_{k \to \infty} (\nu(\Omega_{k+1}) - \nu(\Omega_k)) = 0.$$

Since $\{\boldsymbol{x}^k\}$ is bounded and $\{\boldsymbol{\omega}_{new}^k\} \subset \Omega$ and Ω is compact, there exists a convergent subsequence. For the sake of convenience, we can assume, without loss of generality, that there exist $\boldsymbol{x}^* \in \mathbb{R}^n$ and $\boldsymbol{\omega}^* \in \Omega$ such that

$$\lim_{k \to \infty} (\boldsymbol{x}^k, \boldsymbol{\omega}_{new}^k) = (\boldsymbol{x}^*, \boldsymbol{\omega}^*),$$

which together with (2) yields

$$\lim_{k \to \infty} g(\boldsymbol{x}^k, \boldsymbol{\omega}_{new}^k) = g(\boldsymbol{x}^*, \boldsymbol{\omega}^*) \geq \rho > 0. \tag{23}$$

For any sufficiently small $\varepsilon \in (0, min\{\sigma_2, \sigma_2^2\})$, we can find a sufficiently large integer $N > k_0$ such that

$$0 \leq \boldsymbol{x}_{n+1}^{N+1} - \boldsymbol{x}_{n+1}^N < \varepsilon^2, \quad |g(\boldsymbol{x}^N, \boldsymbol{\omega}_{new}^N) - g(\boldsymbol{x}^*, \boldsymbol{\omega}^*)| < \varepsilon^2. \tag{24}$$

From Theorem 2.3, the first formulation in (24) and Assumption 3.1(d), we have

$$0 < \sum_{i=1}^{\ell} \lambda_i^N(\theta_i^{N+1} + D_i^N) + \sum_{\boldsymbol{\omega}\in\Omega_N} \nu^N(\boldsymbol{\omega})(S^N(\boldsymbol{\omega}) - g(\boldsymbol{x}^{N+1}, \boldsymbol{\omega})) < \varepsilon^2, \quad (25)$$

and

$$g(\boldsymbol{x}^N, \boldsymbol{\omega}_{new}^N) \le \frac{\varepsilon^2 + \sum_{\boldsymbol{\omega}\in\Omega_{N+1}} \nu^{N+1}(\boldsymbol{\omega})H^N(\boldsymbol{\omega}) + \sum_{i=1}^{\ell} \lambda_i^{N+1}(G_i^N + \theta_i^N)}{\nu^{N+1}(\boldsymbol{\omega}_{new}^N)}. \quad (26)$$

Inequality (25) implies that

$$0 \le \lambda_i^N \theta_i^{N+1} < \varepsilon^2, \quad \text{for } i \in I_+^N,$$

which together with Assumption 3.1(b) yields

$$\theta_i^{N+1} < \varepsilon^2/\sigma_2, \quad \text{for } i \in I_+^N. \quad (27)$$

Since $f_i(\boldsymbol{x}^N) = \boldsymbol{x}_{n+1}^N \le \boldsymbol{x}_{n+1}^{N+1}$ for $i \in I_+^N$, it follows from (27) that

$$A^N(\boldsymbol{x}^{N+1} - \boldsymbol{x}^N) + o(\|\boldsymbol{x}^{N+1} - \boldsymbol{x}^N\|) = O(\varepsilon^2/\sigma_2).$$

Hence, by Assumption 3.1(c) we have

$$\begin{aligned} O(\varepsilon^4/\sigma_2^2) &= (\boldsymbol{x}^{N+1} - \boldsymbol{x}^N)^T(A^N)^T A^N(\boldsymbol{x}^{N+1} - \boldsymbol{x}^N) \\ &\ge \mu_{min}\|\boldsymbol{x}^{N+1} - \boldsymbol{x}^N\|^2 \ge \sigma_2\|\boldsymbol{x}^{N+1} - \boldsymbol{x}^N\|^2. \end{aligned} \quad (28)$$

Since $0 < \varepsilon < \sigma_2^2$, it follows from (28) that

$$\|\boldsymbol{x}^{N+1} - \boldsymbol{x}^N\| = o(\varepsilon).$$

Hence, we have

$$G_i^N = o(\|\boldsymbol{d}^N\|) = o(\varepsilon), \quad H^N(\boldsymbol{\omega}) = o(\|\boldsymbol{d}^N\|) = o(\varepsilon), \quad \forall \boldsymbol{\omega} \in \Omega_{N+1},$$

and

$$\begin{aligned} \sum_{i=1}^{\ell} \lambda_i^{N+1}\theta_i^N &= \sum_{i=1}^{\ell} \lambda_i^{N+1}[(\boldsymbol{x}_{n+1}^N - \boldsymbol{x}_{n+1}^{N+1}) + (f_i(\boldsymbol{x}^{N+1}) - f_i(\boldsymbol{x}^N))] \\ &= O(\varepsilon^2) + \sum_{i=1}^{\ell} \lambda_i^{N+1}\nabla f_i(\boldsymbol{x}^N)^T(\boldsymbol{x}^{N+1} - \boldsymbol{x}^N) + o(\varepsilon) \\ &= o(\varepsilon), \end{aligned}$$

where the second equality follows from (24) and Taylor expansion, and the last one holds since $\{\boldsymbol{x}^k\}$ is bounded and $\nabla f_i(\boldsymbol{x})$ is continuous on $\mathbb{R}^n$. There exists a constant $M > 0$ such that

$$\|\nabla f_i(\boldsymbol{x}^N)\| \le M, \quad \forall i \in I,$$

which together with $\sum_{i=1}^{\ell} \lambda_i^{N+1} = 1$ yields

$$\sum_{i=1}^{\ell} \lambda_i^{N+1} \nabla f_i(\boldsymbol{x}^N)^T (\boldsymbol{x}^{N+1} - \boldsymbol{x}^N) \leq M \|\boldsymbol{x}^{N+1} - \boldsymbol{x}^N\|.$$

Consequently, by the second formulation in (24), (26), and Assumption 3.1(a), we have

$$|g(\boldsymbol{x}^*, \boldsymbol{\omega}^*)| \leq |g(\boldsymbol{x}^N, \boldsymbol{\omega}_{new}^N)| + \varepsilon^2 < \varepsilon^2 + \frac{o(\varepsilon) + \varepsilon^2}{\sigma_2}. \tag{29}$$

Since $\sigma_1 > \sigma_2 > \varepsilon$, (29) yields

$$|g(\boldsymbol{x}^*, \boldsymbol{\omega}^*)| \to 0 \quad \text{as } \varepsilon \to 0,$$

which contradicts (23). Hence, Algorithm 2.1 terminates in a finite number of iterations. □

Until now, we have shown the finite termination property of Algorithm 2.1. Nevertheless, the previous theorems would be meaningless if the obtained solution were far from the optimum of (P). Hence, we give the following theorem, which indicates that Algorithm 2.1 can yield an approximate optimal solution of (P) in a finite number of iterations.

Theorem 3.3. *Suppose that Algorithm 2.1 terminates in a finite number of iterations, and let $k^*(\rho)$ be the number of iterations in which Algorithm 2.1 terminates. If there exists $\rho_0 > 0$ such that the set*

$$\mathcal{F}_\rho := \{(\boldsymbol{x}, \boldsymbol{x}_{n+1}) \in \mathbb{R}^{n+1} | f_i(\boldsymbol{x}) \leq \boldsymbol{x}_{n+1}, i = 1, \ldots, \ell, \; g(\boldsymbol{x}, \boldsymbol{\omega}) \leq \rho, \; \forall \boldsymbol{\omega} \in \Omega\}$$

is bounded when $\rho = \rho_0$, then the optimal value of problem $P[\Omega_{k^*(\rho)}]$ is sufficiently close to the optimal value v^* of problem $P[\Omega]$ if ρ is tends to zero, i.e.,

$$\lim_{\rho \to 0} v(\Omega_{k^*(\rho)}) = v^*.$$

Moreover, if (P) has a unique optimal solution $\boldsymbol{x}^*$, then $\lim_{\rho \to 0} \boldsymbol{x}^{k^*(\rho)} = \boldsymbol{x}^*$.

Proof. Let $\mathcal{F}$ be the feasible region of problem $P[\Omega]$. It is clear that

$$(\boldsymbol{x}^{k^*(\rho)}, \boldsymbol{x}_{n+1}^{k^*(\rho)}) \in \mathcal{F}_\rho, \quad \mathcal{F} \subseteq \mathcal{F}_\rho.$$

Since there exists $\rho_0 > 0$ such that the set $\mathcal{F}_{\rho_0}$ is bounded, we further have

$$\lim_{\rho \to 0} dist(\mathcal{F}, \mathcal{F}_\rho) = 0, \tag{30}$$

where $dist(\cdot,\cdot)$ is the Hausdorff distance between two sets. Let $(\boldsymbol{x}^{Pr}, \boldsymbol{x}^{Pr}_{n+1})$ be the projection of $(\boldsymbol{x}^{k^*(\rho)}, \boldsymbol{x}^{k^*(\rho)}_{n+1})$ onto $\mathcal{F}$. Then, we have

$$\begin{aligned}
0 &\le v^* - v(\Omega_{k^*(\rho)}) \\
&= v^* - \boldsymbol{x}^{Pr}_{n+1} + \boldsymbol{x}^{Pr}_{n+1} - \boldsymbol{x}^{k^*(\rho)}_{n+1} \\
&\le \boldsymbol{x}^{Pr}_{n+1} - \boldsymbol{x}^{k^*(\rho)}_{n+1} \\
&\le \|(\boldsymbol{x}^{Pr}, \boldsymbol{x}^{Pr}_{n+1}) - (\boldsymbol{x}^{k^*(\rho)}, \boldsymbol{x}^{k^*(\rho)}_{n+1})\| \\
&\le dist(\mathcal{F}, \mathcal{F}_\rho),
\end{aligned}$$

which, together with (30), imply that

$$\lim_{\rho\to 0} v(\Omega_{k^*(\rho)}) = v^*.$$

Consequently, we can easily show the second conclusion of this theorem by using the boundedness of $\{\boldsymbol{x}^{k^*(\rho)}\}$. □

Notice that in Step 1 of Algorithm 2.1 we may also simultaneously choose q different points $\{\boldsymbol{\omega}^k_1, \dots, \boldsymbol{\omega}^k_q\}$ such that

$$g(\boldsymbol{x}^k, \boldsymbol{\omega}^k_i) > \rho \quad \text{for } i = 1, \dots, q,$$

and let

$$\bar{\Omega}_{k+1} := \Omega_k \cup \{\boldsymbol{\omega}^k_1, \dots, \boldsymbol{\omega}^k_q\}.$$

For such a multiple explicit exchange method, Theorem 2.3 and Theorems 3.1–3.3 can be shown by using analogous techniques.

4. Implementation and numerical examples

In this section, we report some preliminary numerical results for Algorithm 2.1. We implement Algorithm 2.1 in MATLAB 7.8.0 (R2009a) and run experiments on a personal computer with Pentium(R) CPU 1.73GHz and RAM of 512MB. For all examples, we choose the vector of ones as the starting point. We implement Algorithm 2.1 with multiple exchange, and we apply nonlinear programming solver **fmincon** from MATLAB toolbox to solve each subproblem. If $\Omega = [a, b]$, Ω_0 and $\bar{\Omega}$ are set to be

$$\Omega_0 = \{a + i(b-a)/(N_0 - 1) \mid i = 0, 1, \dots, N_0 - 1\}, \quad \text{where } N_0 = 10,$$

and

$$\bar{\Omega} = \{a + i(b-a)/(N - 1) \mid i = 0, 1, \dots, N - 1\}, \quad \text{where } N = 10.$$

In Step 1 of Algorithm 2.1, we find an $\boldsymbol{\omega}^k_{new} \in \Omega$ such that $g(\boldsymbol{x}^k, \boldsymbol{\omega}^k_{new}) > \rho$ in the following way. We test each point in $\bar{\Omega}$ to see whether there is a point satisfying $g(\boldsymbol{x}^k, \boldsymbol{\omega}^k_{new}) > \rho$. If all points fail, then we set $N = 100$ and test each point in the new $\bar{\Omega}$ to find a point satisfying $g(\boldsymbol{x}^k, \boldsymbol{\omega}^k_{new}) > \rho$. If this fails again, then we set $N = 1000$ and test each point in the new $\bar{\Omega}$ to seek a point satisfying $g(\boldsymbol{x}^k, \boldsymbol{\omega}^k_{new}) > \rho$. If this fails again, then we set $N = 10000$, etc. In Step 3, we relax the criterion $\nu^{k+1}(\boldsymbol{\omega}) > 0$ to $\nu^{k+1}(\boldsymbol{\omega}) > 10^{-6}$. We stop the iteration of Algorithm 2.1 when $max\{g(\boldsymbol{x}^k, \boldsymbol{\omega}) | \boldsymbol{\omega} \in \bar{\Omega}\} \leq \rho$, where $\bar{\Omega}$ is defined with $N = 100000$.

We implement Algorithm 2.1 on the following three problems with setting $\rho := 10^{-6}$. The numerical results are summarized in Table 1, where *EXAM* denotes the experimented problems, *NIT*(*NIFC*) denotes the number of iterations and the cardinality of set Ω_k at the final iteration, *CPU*(*s*) is the CPU time (in seconds) for solving each problem, *FVAL* denotes the final value of the objective function, and *MAXG* denotes the final value of the function $\max\{g(\boldsymbol{x}, \boldsymbol{\omega}) | \, \boldsymbol{\omega} \in \bar{\Omega}\}$, where $\bar{\Omega}$ is defined with $N = 100000$.

For comparison, we also apply **fseminf**, that is a solver for SIP based on an implementation of the discretization SQP method in MATLAB toolbox, to solve the three examples in the form $P[\Omega]$. For the solver **fseminf**, we use all the default values.

Problem 1

$$\begin{aligned}
&f_1(\boldsymbol{x}) = \boldsymbol{x}_1^2 + \boldsymbol{x}_2^4,\\
&f_2(\boldsymbol{x}) = (\boldsymbol{x}_1 - 2)^2 + (\boldsymbol{x}_2 - 2)^2,\\
&g(\boldsymbol{x}, \boldsymbol{\omega}) = 5\boldsymbol{x}_1^2 sin(\pi\sqrt{\boldsymbol{\omega}})/(1 + \boldsymbol{\omega}^2) - \boldsymbol{x}_2 \leq 0,\\
&\Omega = [0, 1].
\end{aligned}$$

After 4 iterations, we find the optimal solution

$$\boldsymbol{x}_1 = 0.514474445040588, \quad \boldsymbol{x}_2 = 1.256745063664707.$$

Problem 2

$$\begin{aligned}
&f_1(\boldsymbol{x}) = \boldsymbol{x}_1^2 + \boldsymbol{x}_2^2 + \boldsymbol{x}_3^2 + \boldsymbol{x}_4^2 - 2\boldsymbol{x}_1 - 5\boldsymbol{x}_2 - 36\boldsymbol{x}_3 + 7\boldsymbol{x}_4,\\
&f_2(\boldsymbol{x}) = 11\boldsymbol{x}_1^2 + 11\boldsymbol{x}_2^2 + 12\boldsymbol{x}_3^2 + 11\boldsymbol{x}_4^2 + 5\boldsymbol{x}_1 - 15\boldsymbol{x}_2 - 11\boldsymbol{x}_3 - 3\boldsymbol{x}_4 - 80,\\
&f_3(\boldsymbol{x}) = 11\boldsymbol{x}_1^2 + 21\boldsymbol{x}_2^2 + 12\boldsymbol{x}_3^2 + 21\boldsymbol{x}_4^2 - 15\boldsymbol{x}_1 - 5\boldsymbol{x}_2 - 21\boldsymbol{x}_3 - 3\boldsymbol{x}_4 - 100,\\
&g(\boldsymbol{x}, \boldsymbol{\omega}) = (1 + \boldsymbol{\omega}^2)^2 - \boldsymbol{x}_1 - \boldsymbol{x}_2\boldsymbol{\omega} - \boldsymbol{x}_3\boldsymbol{\omega}^2 - \boldsymbol{x}_4\boldsymbol{\omega}^3,\\
&\Omega = [0, 1].
\end{aligned}$$

Table 1. Test results for Problems 1–3.

Algorithm	EXAM	NIT(NIFC)	CPU (s)	FVAL	MAXG
Algorithm 2.1	Problem 1	3(1)	0.14	2.759214074824113	1.88e-007
	Problem 2	6(2)	0.63	-55.468813235577016	1.11e-016
	Problem 3	4(1)	0.23	-24.637013595823785	9.58e-008
fseminf	Problem 1	4	1.09	2.759214129908945	4.04e-008
	Problem 2	8698	217.83	-55.469542809472998	1.30e-004
	Problem 3	19	1.34	-24.637012132368071	-2.12e-013

After 6 iterations, we find the optimal solution

$$x_1 = 1.00000000, \quad x_2 = 1.1328729785, \quad x_3 = 1.5256254664,$$
$$x_4 = 0.3415015551.$$

Problem 3

$$f_1(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4,$$
$$f_2(x) = 11x_1^2 + 11x_2^2 + 12x_3^2 + 11x_4^2 + 5x_1 - 15x_2 - 11x_3 - 3x_4 - 80,$$
$$f_3(x) = 11x_1^2 + 21x_2^2 + 12x_3^2 + 21x_4^2 - 15x_1 - 5x_2 - 21x_3 - 3x_4 - 100,$$
$$f_4(x) = 11x_1^2 + 211x_2^2 + 12x_3^2 + 15x_1 - 15x_2 - 21x_3 - 3x_4 - 50,$$
$$g(x, \omega) = exp(\omega) - x_1 - x_2\omega - x_3\omega^2 - x_4\omega^3,$$
$$\Omega = [0, 1].$$

After 4 iterations, we find the optimal solution

$$x_1 = 1.1808339962, \quad x_2 = 0.0756637247, \quad x_3 = 1.4436594485,$$
$$x_4 = 0.8178610799.$$

The numerical results reported in Table 1 show that Algorithm 2.1 performs very well for all the tested problems and can give the optimal solutions. In particular, from the last columns of NIT, CPU (s), and MAXG in Table 1, it is easy to see that Algorithm 2.1 is effective. The solver **fseminf** is good for Problem 1 and Problem 3, but it is not good for Problem 2.

5. Concluding remarks

In this article, we present an exchange method for solving convex min-max problems with infinite constraints. The algorithm is given based on a sequence of auxiliary subproblems. Under reasonable assumptions we prove that the proposed algorithm has finite termination property and the approximate optimal solution of the original problem can be derived from the optimal solution of the subproblem at final iteration. Numerical results show that the algorithm is efficient.

Funding

This work is supported by the National Natural Science Foundation of China (Grant No. 11271221).

References

1. A. Auslender, M. A. Goberna, and M. A. López (2009). Penalty and smoothing methods for convex semi-infinite programming. *Mathematics of Operations Research* 34:303–319.
2. J. W. Bandler, P. C. Liu, and H. Tromp (1976). A nonlinear programming approach to optimal design centering, tolerancing, and tunning. *IEEE Transactions on Circuits and Systems* 23:155–165.

3. T. Chen and M. K. H. Fan (1998). On convex formulation of the floor-plan area minimization problem. In: *Proceedings of the International Symposium on Physical Design*, California: Monterey, pp. 124–128.
4. A. V. Fiacco and K. O. Kortanek (1983). Semi-infinite programming and applications. *Lecture notes in Economics and Mathematical Systems* 215, Springer-Verlag, Berlin, Germany.
5. G. Gramlich, R. Hettich, and E. W. Sachs (1995). Local convergence of SQP methods in semi-infinite programming. *SIAM Journal on Optimization* 5:641–658.
6. A. A. Kassim and B. V. K. Vijaya Kumar (1997). Path planning autonomuous robots using neural networks. *Journal of Intelligent Systems* 7:33–55.
7. R. Hettich and K. O. Kortanek (1993). Semi-infinite programming: Theorey, methods and applications. *SIAM Review* 35:380–429.
8. H.-C. Lai and S.-Y. Wu (1992). On linear semi-infinite programming problems–An algorithm. *Numerical Functional Analysis and Optimization* 13:287–304.
9. E. Polak (1997). *Optimization: Algorithms and Consistent Approximations*. Springer-Verlag, New York, NY.
10. E. Polak (1993). On the use of consistent approximations in the solution of semi-infinite optimization and optimal control problems. *Mathematical Programming* 62:385–414.
11. E. Polak and J. O. Royset (2005). On the use of augmented Lagrangians in the solution of generalized semi-infinite min-max problems. *Computational Optimization and Applications* 31:173–192.
12. E. Polak and J. O. Royset (2003). Algorithms for finite and semi-infinite min-max-min problems using adaptive smoothing techniques. *Journal of Optimization Theory and Applications* 119:421–457.
13. R. Reemsten and S. Görner (1998). Numerical methods for semi-infinite programming: A survey. In: *Semi-Infinite Programming* (R. Reemsten and J. Rückmann, eds.) Kluwer Academic Publishers, Boston, MA, pp. 195–275.
14. S.-Y. Wu, D. Li, L. Qi, and G. Zhou (2005). An iterative method for solving KKT system of the semi-infinite programming. *Optimization Methods and Software* 20:629–643.
15. R. Reemsten (1991). Discretization methods for solution of semi-infinite programming problems. *Journal of Optimization Theory and Applications* 71:85–103.
16. G. Still (2001). Discretization in semi-infinite programming: The rate of convergence. *Mathematical Programming* 91:53–69.
17. Y. Tanaka, M. Fukushima, and T. Ibaraki (1988). A globally convergent SQP method for semi-infinite nonlinear optimization. *Journal of Computational and Applied Mathematics* 23: 141–153.
18. G. A. Watson (1975). A multiple exchange algorithm for multivariate Chebyshev approximation. *SIAM Journal on Numerical Analysis* 12:46–52.
19. L. P. Zhang, S.-Y. Wu, and M. A. López (2010). A new exchange method for convex semi-infinite programming. *SIAM Journal on Optimization* 20:2959–2977.