SIAM J. OPTIM.
Vol. 25, No. 4, pp. 2537–2560

FEASIBLE METHOD FOR SEMI-INFINITE PROGRAMS*

SHUXIONG WANG† AND YAXIANG YUAN†

Abstract. A new numerical method is presented for semi-infinite optimization problems which guarantees that each iterate is feasible for the original problem. The basic idea is to construct concave relaxations of the lower level problem, to compute the optimal values of the relaxation problems explicitly, and to solve the resulting approximate problems with finitely many constraints. The concave relaxations are constructed by replacing the objective function of the lower level problem by its concave upper bound functions. Under mild conditions, we prove that every accumulation point of the solutions of the approximate problems is an optimal solution of the original problem. An adaptive subdivision algorithm is proposed to solve semi-infinite optimization problems. It is shown that the Karush–Kuhn–Tucker points of the approximate problems converge to a Karush–Kuhn–Tucker point of the original problem within arbitrarily given tolerances. Numerical experiments show that our algorithm is much faster than the existing adaptive convexification algorithm in computation time.

Key words. semi-infinite programs, concave optimization, adaptive subdivision algorithm

AMS subject classifications. 90C34, 90C26, 90C30

DOI. 10.1137/140982143

1. Introduction.

Semi-infinite programs refer to optimization problems with finitely many decision variables and infinitely many constraints which can be formulated as follows:

$$(P) \qquad \begin{array}{ll} \min\limits_{x\in X} & f(x) \\ \text{s.t.} & g(x,y)\le 0 \quad \forall y\in Y, \end{array}$$

where $f \in C^1(\mathbb{R}^n,\mathbb{R})$ is the objective function and $g \in C^2(\mathbb{R}^n\times\mathbb{R}^m,\mathbb{R}^p)$ is the constraint function, $X=[l,u]$ with $l<u$ is a box in $\mathbb{R}^n$, $Y=[a,b]$ with $a<b$ is a box in $\mathbb{R}^m$. In this paper, we consider the semi-infinite problem with one semi-infinite constraint and a one dimensional index set Y, that is $p=1$ and $m=1$. More general cases, that the index set is an arbitrary set in $\mathbb{R}^m$ or depends on the decision variable x, will not be discussed. However, the results proposed in this paper might be helpful to deal with these cases.

There are many applications for semi-infinite programs such as Chebyshev approximation, optimal control, robust optimization, and numerous engineering problems. More details can be found in [5, 9, 18, 24, 28, 33, 36, 38] and references therein.

Many numerical methods have been proposed to solve semi-infinite programs, which can be mainly classified into three categories: discretization methods, local reduction methods, and exchange methods. We refer to [12, 19, 27] for an overview of these methods and [29] for recent developments in semi-infinite programs. Most of these methods may not be able to guarantee the feasibility of each iterate due to infinitely many constraints. Let F be the feasible region of the semi-infinite constraint,

*Received by the editors August 12, 2014; accepted for publication (in revised form) October 14, 2015; published electronically December 10, 2015. The authors acknowledge the financial support of NSFC grants 11331012, 11321061, and 11461161005.
http://www.siam.org/journals/siopt/25-4/98214.html
†State Key Laboratory of Scientific and Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijng 100190, China (wsx@lsec.cc.ac.cn, yyx@lsec.cc.ac.cn).

namely,

$$F = \{x \in \mathbb{R}^n \mid g(x, y) \leq 0 \ \ \forall y \in Y\}.$$

Discretization methods and exchange methods approximate the feasible region F by $\bar{F}$:

$$\bar{F} = \{x \in \mathbb{R}^n \mid g(x, y) \leq 0 \ \ \forall y \in \bar{Y}\},$$

where $\bar{Y}$ is a subset of Y which contains finitely many elements. The main difference between discretization methods and exchange methods is that exchange methods drop some constraints each time $\bar{Y}$ is updated. Local reduction methods aim to obtain a Karush–Kuhn–Tucker point of problem (P). Therefore, the iterates of such a method are not necessarily feasible for the original problem, though their limit might be.

The authors in [6, 7] propose a branch-and-bound framework to generate convergent sequences of upper and lower bounds to solve semi-infinite problems. The upper bound problem is obtained by using inclusion bounds and interval arithmetic. The lower bound problem is generated by discretization. A related work is proposed in [22] where the upper bound problem is generated based on convex and linear relaxations of the lower level problem (to be defined later). The authors in [10, 32] present a feasible point method by adaptively constructing the convex relaxations of the lower level problem using the idea of the αBB optimization [1, 2]. They replace the relaxed lower level problems by their Karush–Kuhn–Tucker conditions and solve the resulting optimization problems with complementary constraints. In [31], the method proposed in [10] is extended to the case where the index set is arbitrary. A recent work [23] presents an algorithm for the global optimization of generalized semi-infinite programs which can obtain a feasible point in finite iterations. The algorithm generates convergence of lower and upper bounds which are based on deterministic global nonlinear optimization solvers. It is an extension of the work in [20] where the index set is independent of the decision variables.

The main challenge of semi-infinite programs comes from the infinitely many constraints. Given a point $\bar{x} \in \mathbb{R}^n$, to know whether it is feasible for the original problem may one require to check infinitely many constraints, or equivalently, to solve the following lower level problem $\mathrm{Q}(x)$ globally, where

$$(\mathrm{Q}(x)) \qquad \max_{y \in \mathbb{R}^m} \ g(x, y) \quad \text{s.t.} \quad y \in Y.$$

It is obvious that $\bar{x}$ is feasible if and only if the optimal value of $\mathrm{Q}(\bar{x})$ is nonpositive. If the lower level problem $(\mathrm{Q}(x))$ is convex, $(\mathrm{Q}(x))$ can be rewritten equivalently as its Karush–Kuhn–Tucker systems. Since the index set Y is convex, this only requires the concavity of $g(x, y)$ with respect to y. In the case that the lower level problem is not convex, the article [10] constructs convex relaxations of the lower level problem with ideas from the αBB method of global optimization.

In this paper, we present a new method to construct a sequence of inner approximate regions to approximate the feasible region of the original problem. The method is based on constructing concave relaxation problems of the lower level problem. More precisely, we need to construct upper bound functions of the objective function of the lower level problem.

The rest of this paper is organized as follows. In section 2 we propose our motivation and the techniques to construct upper bound functions of the constraint function. Section 3 presents our numerical method as well as the adaptive subdivision algorithm to obtain an approximate Karush–Kuhn–Tucker point of the original problem. Gener-

alizations and remarks about the method proposed in this paper are given in section 4. Numerical experiments are proposed in section 5. At last, we conclude our paper in section 6.

2. Preliminaries. In this section, we first present our motivation for dealing with the feasibility of semi-infinite programs. Then, several upper bound functions are proposed which are used to construct the inner approximate regions of the original feasible region. At last, the first order necessary conditions will be introduced.

2.1. Motivation. Let $Y_N = \{y_1, y_2, \ldots, y_N\}$ be a subset of Y which contains finitely many elements. If we replace the index set Y in the problem (P) by Y_N, the resulting feasible set F_N is an outer approximate region of the original feasible region, that is $F \subseteq F_N$, where

$$F_N = \{x \in \mathbb{R}^n \mid g(x, y_i) \leq 0, i = 1, 2, \ldots, N\}.$$

Our motivation is to make a small perturbation to F_N by introducing $\Delta_N = \{\delta_1, \delta_2, \ldots, \delta_N\}$ with $\delta_i \geq 0$, $i = 1, 2, \ldots, N$, and $\lim_{N\to\infty} \max_{1\leq i\leq N}\{\delta_i\} = 0$, in order to obtain a new set

$$F(\Delta_N) = \{x \in \mathbb{R}^n \mid g(x, y_i) + \delta_i \leq 0,\ i = 1, 2, \ldots, N\}$$

such that this set is contained in the original feasible region:

$$F(\Delta_N) \subseteq F.$$

In the following part, we will illustrate how to choose perturbation Δ_N for a given subset Y_N and refine it adaptively to approximate F by $F(\Delta_N)$.

The feasible region F can be represented by finitely many constraints if the lower level problem has some special structures. For example, it has been mentioned that if the lower level problem is convex, it can be rewritten as its Karush–Kuhn–Tucker conditions such that the upper level problem (P) can be formulated as a mathematical program with complementarity constraints [10]. The drawback of such a method is that extended Mangasarian–Fromovitz constraint qualification (EMFCQ) fails at all feasible points because of the complementarity constraints [30]. Therefore, one may obtain a stationary solution in the sense of Fritz John rather than a Karush–Kuhn–Tucker point.

Consider the cases that the constraint function $g(x, y)$ is monotone or convex with respect to y for any fixed $x \in X$. Then we know that the maximum of the lower level problem is attained on the boundary of the index set $Y = [a, b]$. This implies that the semi-infinite constraint

$$g(x, y) \leq 0\ \forall y \in Y$$

is equivalent to

$$\max\{g(x, a), g(x, b)\} \leq 0.$$

It is easy to see that the above result is true when $g(x, y)$ is monotone with respect to y for all $x \in X$. The latter case follows immediately from the following lemma with $C = Y$ and $S = \partial Y$.

LEMMA 2.1 (see [26]). *Let f be a convex function, and let $C = conv(S)$, where S is an arbitrary set of points. Then*

$$\sup\{f(x) \mid x \in C\} = \sup\{f(x) \mid x \in S\},$$

where the first supremum is attained only when the second (more restrictive) supremum is attained.

2.2. Upper bound function and interval method. Suppose that $g : [a, b] \to \mathbb{R}$ is a real-valued function; we say that $\bar{g} : [a, b] \to \mathbb{R}$ is an upper bound function of g on $[a, b]$ if

$$g(y) \leq \bar{g}(y) \ \forall y \in [a, b].$$

There are many approaches to construct an upper bound function $\bar{g}$ for a given function g. We define the upper bound function $\bar{g}$ of g on $[a, b]$ by adding a nonnegative function $\varphi(y, \alpha, a, b)$ to g, that is,

$$\bar{g}(y, \alpha, a, b) = g(y) + \varphi(y, \alpha, a, b),$$

where $\varphi(y, \alpha, a, b)$ is dependent on a parameter α and the boundary points of the interval $[a, b]$. In this paper, the term $\varphi(y, \alpha, a, b)$ is set as

$$\varphi(y, \alpha, a, b) = \frac{\alpha}{2}\left(y - \frac{a+b}{2}\right)^2.$$

In this case, the function $\bar{g}(y, \alpha, a, b)$ is twice continuously differentiable with respect to y if $g(y)$ is twice continuously differentiable. Furthermore, the second order derivative of $\bar{g}(y, \alpha, a, b)$ is

$$\nabla^2 \bar{g}(y, \alpha, a, b) = \nabla^2 g(y) + \alpha.$$

It follows that $\bar{g}(y, \alpha, a, b)$ is convex on $[a, b]$ if the parameter α satisfies

$$\alpha \geq \max_{y \in [a,b]} -\nabla^2 g(y).$$

On the other hand, if $\alpha \geq 0$, we can conclude that $\bar{g}(y, \alpha, a, b)$ is an upper bound function of $g(y)$. Consequently, if

$$\alpha \geq \max\left(0, \max_{y \in [a,b]} -\nabla^2 g(y)\right), \tag{2.1}$$

function $\bar{g}(y, \alpha, a, b)$ is a convex upper bound function of $g(y)$ on $[a, b]$. We know from Lemma 2.1 that the maximum of $\bar{g}(y, \alpha, a, b)$ must be attained on the boundary of $[a, b]$, that is,

$$\max_{y \in [a,b]} \bar{g}(y, \alpha, a, b) = \max\{g(a), g(b)\} + \frac{\alpha}{8}(b - a)^2.$$

The distance between $\bar{g}(y, \alpha, a, b)$ and $g(y)$ is defined as

$$\operatorname{dist}(\bar{g}, g) = \max_{y \in [a,b]} \{\bar{g}(y, \alpha, a, b) - g(y)\} = \frac{\alpha}{8}(b - a)^2.$$

Such a distance measures the tightness of the upper bound function. The smaller the distance, the tighter the upper bound function is. From its definition, $\operatorname{dist}(\bar{g}, g)$ decreases quadratically to zero as $|b - a|$ tends to zero.

Other choices of the additive term can be $\varphi(y, \alpha, a, b) = \alpha(y - a)$ or $\varphi(y, \alpha, a, b) = \alpha(b - y)$, where α is selected such that $\bar{g}(y, \alpha, a, b)$ is monotone. This kind of formulation is widely used in Lipschitz optimization and related areas (see, e.g., [4, 13, 17]). Furthermore, the upper bound function may have different forms. For example,

$$\bar{g}(y, \alpha, a, b) = \frac{bg(a) - ag(b)}{b - a} + \frac{g(b) - g(a)}{b - a}y + \frac{\alpha}{2}(y - a)(b - y)$$

with $\alpha \geq \max_{y\in[a,b]} |\nabla^2 g(y)|$ being an upper bound function of $g(y)$ which can be obtained directly by the Lagrangian interpolation formula. We skip the detailed discussions for these cases.

Note that the computation of the parameter α in (2.1) requires us to solve a global optimization problem. However, we can use any upper bound of the right-hand side in (2.1). Such upper bounds can be obtained by interval methods. In the next paragraph, we give a brief description of the interval analysis. The reader is referred to [4, 6, 7, 21] for more details.

Given a box $B = [b^l, b^u] \subseteq \mathbb{R}^m$, the width of the box is defined by $\omega(B) = \max_{1\leq j\leq m}\{(b_j^u - b_j^l)\}$. The range of a real-valued function $g(y)$ on B is defined by $R(g, B) = [R^l, R^u] = \{g(y) \mid y \in B\}$. An interval function G is called an inclusion function for $g(y)$ on B if

$$R(g, B) \subseteq G(B) = [G^l, G^u].$$

A natural way to derive an inclusion function G is to replace each variable $y_j \in [b_j^l, b_j^u]$ with the interval variable $B_j = [b_j^l, b_j^u]$ and to evaluate the resulting expression using interval arithmetic [21]. For functions with special structure, the inclusion may be tight, that is, $R(g, B) = G(B)$. In general, such an interval range may overestimate the true range of the original function. However, the tightness of the inclusion can be measured by Hausdorff metric $H(R(g, B), G(B))$ which is defined by

$$H(R(g, B), G(B)) = \max\{|R^l - G^l|, |R^u - G^u|\}.$$

For general nonlinear functions, we have

$$H(R(g, B), G(B)) \leq \gamma\omega(B)^p \quad \text{and} \quad \omega(G(B)) \leq \delta\omega(B)^p,$$

where $p \geq 1$, $\gamma \geq 0$, and $\delta \geq 0$ are constants depending on the expression of the function $g(y)$ and the box B. It follows that a tighter inclusion can be achieved by a subdivision of the box B.

2.3. Necessary conditions. In this subsection, we present the first order necessary optimality conditions for semi-infinite programs. For clarity, we assume that all the stationary points are contained in the interior of X.

Given a feasible point $\bar{x} \in F$, the active index set at $\bar{x}$ is defined as

$$A(\bar{x}) = \{y \in Y \mid g(\bar{x}, y) = 0\}.$$

Note that for $\bar{x} \in \partial F$, the active index set $A(\bar{x})$ is nonempty and compact. Furthermore, any point $y \in A(\bar{x})$ is a global solution of the lower level problem $\mathrm{Q}(\bar{x})$.

We say that the EMFCQ holds at $\bar{x} \in F$, if there exists a vector $d \in \mathbb{R}^n$ such that

$$d^\top \nabla_x g(\bar{x}, y) < 0 \ \forall y \in A(\bar{x}).$$

THEOREM 2.2 (see [12]). *Let x^* be a local minimizer of (P). Then we have the following:*

1. *There exist some nonnegative multipliers λ_j, $0 \leq j \leq k$, and indices $y_j \in A(x^*)$, $1 \leq j \leq k$, with $1 \leq k \leq n+1$ such that $\sum_{j=0}^k \lambda_j = 1$ and*

$$\lambda_0 \nabla f(x^*) + \sum_{j=1}^{k} \lambda_j \nabla_x g(x^*, y_j) = 0.$$

2. *If EMFCQ holds at* x^**, then there exist some nonnegative multipliers* λ_j*,* $1 \le j \le k$*, and indices* $y_j \in A(x^*)$*,* $1 \le j \le k$*, with* $1 \le k \le n$ *such that*

$$\nabla f(x^*) + \sum_{j=1}^{k} \lambda_j \nabla_x g(x^*, y_j) = 0.$$

3. Numerical method. Our numerical method consists of constructing a sequence of inner approximate regions and solving the resulting approximate problems to obtain approximate solutions to the original problem. It is achieved in two steps. First, we subdivide the index set into a union of subsets. Then we replace the lower level problem by its relaxation problems on each subset. Furthermore, we propose a refinement procedure to update the subdivisions which guarantees that the approximate regions are monotone.

DEFINITION 3.1. *Given an integer* $N \in \mathbb{N}$*,* $T = \{\tau_k \mid k \in K_0 = \{0, 1, 2, \ldots, N\}\}$ *is called a subdivision of the index set* $Y = [a, b]$ *if*

$$a = \tau_0 \le \tau_1 \le \cdots \le \tau_N = b.$$

The length of the subinterval $Y^k = [\tau_{k-1}, \tau_k]$ is defined by $|Y^k| = |\tau_k - \tau_{k-1}|$ and the length of the subdivision T is defined by

$$|T| = \max_{k \in K} |Y^k| = \max_{k \in K} |\tau_k - \tau_{k-1}|,$$

where $K = \{1, 2, \ldots, N\}$. It follows by the definition of subdivision that $Y = \cup_{k \in K} Y^k$. A trivial observation in [10] is that the original semi-infinite constraint can be formulated equivalently as the finitely many semi-infinite constraints

$$g(x, y) \le 0 \ \forall y \in Y^k, k \in K.$$

Based on such an observation, we will construct the concave relaxation problems of the lower level problem for each of these finitely many semi-infinite constraints. Then we solve the resulting approximate problems and refine the subdivisions adaptively to obtain better approximate solutions of the original problem.

3.1. Approximation. Let $X \times Y^k$ be a subset of $X \times Y$; the upper bound function g^k of g on $X \times Y^k$ is defined by

$$g^k(x, y) = g(x, y) + \frac{\alpha_k}{2} \left(y - \frac{\tau_{k-1} + \tau_k}{2} \right)^2, \tag{3.1}$$

where the parameter α_k satisfies

$$\alpha_k \ge \max \left\{ 0, \max_{(x,y) \in X \times Y^k} -\nabla_y^2 g(x, y) \right\}. \tag{3.2}$$

Some useful properties of the upper bound function g^k are summarized in the following lemma which is easily verified.

LEMMA 3.2. *Given a subdivision* $T = \{\tau_k \mid k \in K_0 = \{0, 1, 2, \ldots, N\}\}$ *of the index set* Y*, let* $g^k(x, y)$ *be defined by* (3.1) *for each* $k \in \{1, 2, \ldots, N\}$*. If the parameter* α_k *satisfies* (3.2)*, then we have the following:*

(i) $g^k(x, y)$ *is an upper bound function of* $g(x, y)$ *on* $X \times Y^k$*, that is,*

$$g(x, y) \le g^k(x, y) \ \forall (x, y) \in X \times Y^k.$$

(ii) *For any fixed $x \in X$, $g^k(x,y)$ is convex with respect to y, which implies that*

$$\max_{y\in Y^k} g^k(x,y) = \max\{g^k(x,\tau_{k-1}), g^k(x,\tau_k)\}.$$

(iii) *The distance between $g^k(x,y)$ and $g(x,y)$ on $X \times Y^k$ is*

$$\max_{(x,y)\in X\times Y^k} (g^k(x,y) - g(x,y)) = \frac{\alpha_k}{8}(\tau_k - \tau_{k-1})^2.$$

For any given subdivision $T = \{\tau_k \mid k \in K_0 = \{0,1,2,\ldots,N\}\}$ of the index set Y, with the parameters α_k, $k \in K = \{1,2,\ldots,N\}$, being selected by (3.2), the approximate region $F(\alpha,T)$ of F is defined by

$$F(\alpha,T) = \{x \in \mathbb{R}^n \mid g^k(x,y) \le 0 \ \forall y \in Y^k, k \in K\}.$$

By the first result (i) in Lemma 3.2, we know that $F(\alpha,T) \subseteq F$. Furthermore, the result (ii) in Lemma 3.2 implies that

$$F(\alpha,T) = \{x \in \mathbb{R}^n \mid g^k(x,\tau_{k-i}) \le 0, k \in K, i = 0,1\}. \tag{3.3}$$

We can see from (3.3) that the total number of constraints of $F(\alpha,T)$ are $2N$. However, not all these constraints are necessary since $F(\alpha,T)$ can be represented by $N+1$ constraints where each subdivision point in T contributes to one constraint:

$$F(\alpha,T) = \left\{ x \in \mathbb{R}^n \mid \begin{array}{l} g^1(x,\tau_0) \le 0, \\ \bar{g}^k(x,\tau_k) \le 0, k = 1,2,\ldots,N-1, \\ g^N(x,\tau_N) \le 0, \end{array} \right\} \tag{3.4}$$

where

$$\bar{g}^k(x,\tau_k) = g(x,\tau_k) + \max\left\{ \frac{\alpha_k}{2}\left(\frac{\tau_k - \tau_{k-1}}{2}\right)^2, \frac{\alpha_{k+1}}{2}\left(\frac{\tau_{k+1} - \tau_k}{2}\right)^2 \right\}.$$

Denoting $\bar{g}^0(x,\tau_0) = g^1(x,\tau_0)$ and $\bar{g}^N(x,\tau_N) = g^N(x,\tau_N)$, (3.4) can be rewritten as

$$F(\alpha,T) = \{x \in \mathbb{R}^n \mid \bar{g}^k(x,\tau_k) \le 0, k = 0,1,\ldots,N\}.$$

For clarity, we denote a series of integer indices $\bar{k}$ for $k = 1,2,\ldots,N-1$ as follows:

$$\bar{k} = \begin{cases} k & \text{if } \ \dfrac{\alpha_k}{2}\left(\dfrac{\tau_k - \tau_{k-1}}{2}\right)^2 \ge \dfrac{\alpha_{k+1}}{2}\left(\dfrac{\tau_{k+1} - \tau_k}{2}\right)^2, \\ k+1 & \text{otherwise.} \end{cases} \tag{3.5}$$

At the same time, we define $\bar{k} = 1$ if $k = 0$ and $\bar{k} = N$ if $k = N$. Then we have

$$\bar{g}^k(x,\tau_k) = g(x,\tau_k) + \frac{\alpha_{\bar{k}}}{2}\left(\frac{\tau_{\bar{k}} - \tau_{\bar{k}-1}}{2}\right)^2.$$

Consequently, we obtain an inner approximate region of the original feasible region provided that $F(\alpha,T)$ is nonempty. A subdivision T is called consistent if the corresponding approximate region $F(\alpha,T)$ is nonempty. It will be shown in the following that if the Slater condition holds for the original problem, then $F(\alpha,T)$ is nonempty provided that $|T|$ is small enough. We say that the Slater condition holds for the original problem if there exists a point $\bar{x} \in X$ such that

$$g(\bar{x},y) < 0 \ \forall y \in Y.$$

In this paper, we always assume that the Slater condition holds.

Given a consistent subdivision T, the corresponding approximate problem of the original problem is defined by

$$(P(\alpha,T)) \qquad \min_{x\in X} \quad f(x) \quad \text{s.t.} \quad x \in F(\alpha,T).$$

It is obvious that $(P(\alpha,T))$ is a nonlinear optimization problem with finitely many inequality constraints for any given consistent subdivision T. Furthermore, any feasible point of $(P(\alpha,T))$ is feasible for the original problem. Thus, if x^* is a solution of $(P(\alpha,T))$, then $f(x^*)$ provides an upper bound for the optimal value of the original problem.

In order to prove the convergence of the approximate regions to the feasible region of the original problem, we present the definition of the convergence of sets as follows. Let M be a subset of $\mathbb{R}^n$; the distance between a point $x \in \mathbb{R}^n$ and the subset M is defined by

$$Dist(x,M) = \inf\{||x-y|| \mid y \in M\},$$

where $||\cdot||$ represents the Euclidean norm.

DEFINITION 3.3 (see [3]). *Let $\{M_k\}_{k\in\mathbb{N}}$ be a sequence of subsets in $\mathbb{R}^n$; a subset M is called the limit of the sequence $\{M_k\}_{k\in\mathbb{N}}$ if*

$$M = \limsup_{k\to\infty} M_k = \liminf_{k\to\infty} M_k \triangleq \lim_{k\to\infty} M_k,$$

where $\limsup_{k\to\infty} M_k = \{x \in \mathbb{R}^n \mid \liminf_{k\to\infty} Dist(x,M_k) = 0\}$ is the outer limit of the sequence $\{M_k\}_{k\in\mathbb{N}}$ and $\liminf_{k\to\infty} M_k = \{x \in \mathbb{R}^n \mid \lim_{k\to\infty} Dist(x,M_k) = 0\}$ is the inner limit of the sequence $\{M_k\}_{k\in\mathbb{N}}$.

It is obvious that the outer and inner limits of the sequence $\{M_k\}_{k\in\mathbb{N}}$ exist and are closed. Furthermore, we have

$$\liminf_{k\to\infty} M_k \subseteq \limsup_{k\to\infty} M_k.$$

Denote by F^o the set of all the Slater points of the original problem, i.e.,

$$F^o = \{x \in \mathbb{R}^n \mid \psi(x) < 0\},$$

where $\psi(x) = \max_{y\in Y} g(x,y)$. Let $cl(F^o)$ be the closure of the set F^o.

We present in the following theorem that the sequence of the approximate regions $\{F(\alpha^k,T_k)\}_{k\in\mathbb{N}}$ converges to the original feasible set F as $|T_k|$ goes to zero under some conditions.

THEOREM 3.4. *Assume that the Slater condition holds for the original problem and $cl(F^o) = F$. A sequence of subdivisions $\{T_k \mid k \in \mathbb{N}\}$ is constructed such that T_0 is consistent and $\lim_{k\to\infty} |T_k| = 0$ with $T_k \subseteq T_{k+1}$ for all $k \in \mathbb{N}$. Then we have*

$$\lim_{k\to\infty} F(\alpha^k,T_k) = F.$$

Proof. Since the function g is continuous with respect to $(x,y) \in X \times Y$ and Y is compact, $\psi(x) = \max_{y\in Y} g(x,y)$ is continuous on X. Thus the feasible set $F = \{x \mid \psi(x) \le 0\}$ is closed.

Let $T_k = \{\tau_j \mid j \in \{0,1,2,\ldots,N_k\}\}$ and $\psi_k(x) = \max_{0\le j\le N_k} \bar{g}^j(x,\tau_j)$ for each $k \in \mathbb{N}$. The approximate regions $F(\alpha^k,T_k), k \in \mathbb{N}$, can be rewritten as

$$F(\alpha^k,T_k) = \{x \mid \psi_k(x) \le 0\}.$$

It is obvious that $\psi_k(x)$ is continuous on $\mathbb{R}^n$, thus $F(\alpha^k, T_k)$ is closed for each $k \in \mathbb{N}$. We know from Lemma 3.2 that $F(\alpha^k, T^k) \subseteq F$ for all $k \in \mathbb{N}$. It follows that $Dist(x, F) = 0$ for all $x \in F(\alpha^k, T_k), k \in \mathbb{N}$.

For any $\bar{x} \in \limsup_{k\to\infty} F(\alpha^k, T_k)$, the definition of outer limit implies that $\liminf_{k\to\infty} Dist(\bar{x}, F(\alpha^k, T_k)) = 0$. Moreover, the triangle inequality indicates that

$$Dist(\bar{x}, F) \leq Dist(\bar{x}, x) + Dist(x, F) \ \forall x \in F(\alpha^k, T_k), k \in \mathbb{N}.$$

This implies that

$$Dist(\bar{x}, F) \leq \liminf_{k\to\infty} Dist(\bar{x}, F(\alpha^k, T_k)) = 0.$$

This result, combined with the fact that F is closed, implies that $\bar{x} \in F$. Therefore, the outer limit of the sequence $\{F(\alpha^k, T_k)\}_{k\in\mathbb{N}}$ is contained in F, i.e.,

$$\limsup_{k\to\infty} F(\alpha^k, T_k) \subseteq F. \tag{3.6}$$

On the other hand, we can obtain immediately by the definitions of $\psi_k(x)$ and $\psi(x)$ that

$$\begin{aligned}\max_{x\in X} |\psi_k(x) - \psi(x)| &\leq \max_{j=1,2,\ldots,N_k} \max_{(x,y)\in X\times Y^j} |g^j(x,y) - g(x,y)| \\ &\leq \max_{j=1,2,\ldots,N_k} \frac{\alpha_j^k}{8}(\tau_j - \tau_{j-1})^2 \leq \frac{\tilde{\alpha}}{8}|T_k|^2,\end{aligned}$$

where $\tilde{\alpha}$ is the uniform bound of all the parameters $\{\alpha_j^k, j = 1, 2, \ldots, N_k, k \in \mathbb{N}\}$ which satisfies

$$\tilde{\alpha} > \max\left\{0, \max_{(x,y)\in X\times Y} -\nabla_{yy}^2 g(x,y)\right\}. \tag{3.7}$$

This implies that $\psi_k(x)$ converges to $\psi(x)$ uniformly on X as k tends to infinity. For any $\bar{x} \in F^o$, the following inequality holds:

$$\psi_k(\bar{x}) \leq \psi(\bar{x}) + \frac{\tilde{\alpha}}{8}|T_k|^2.$$

It follows immediately that $\psi_k(\bar{x}) \leq 0$ if $|T_k| \leq \sqrt{\frac{-8\psi(\bar{x})}{\tilde{\alpha}}}$. We can conclude from $\lim_{k\to\infty} |T_k| = 0$ and $T_k \subseteq T_{k+1}, k \in \mathbb{N}$ that there exists an integer $\bar{k} \in \mathbb{N}$ such that $\bar{x} \in F(\alpha^k, T_k)$ for all $k \geq \bar{k}$. Therefore, we have $Dist(\bar{x}, F(\alpha^k, T_k)) = 0$ for $k \geq \bar{k}$ which implies that $\lim_{k\to\infty} Dist(\bar{x}, F(\alpha^k, T_k)) = 0$. Thus, we have $\bar{x} \in \liminf_{k\to\infty} F(\alpha^k, T_k)$. It follows that

$$F^o \subseteq \liminf_{k\to\infty} F(\alpha^k, T_k).$$

The closeness of $\liminf_{k\to\infty} F(\alpha^k, T_k)$ implies that

$$cl(F^o) \subseteq \liminf_{k\to\infty} F(\alpha^k, T_k). \tag{3.8}$$

By (3.6), (3.8), and the assumption $cl(F^o) = F$, we can conclude that the limit of the sequence $\{F(\alpha^k, T_k)\}_{k\in\mathbb{N}}$ exists and $\lim_{k\to\infty} F(\alpha^k, T_k) = F$. This completes the proof of our theorem. □

From the proof of Theorem 3.4, we can see that $F(\alpha, T)$ is nonempty for sufficiently small $|T|$ provided that the Slater condition holds for the original problem. On the other hand, if there is a global minimizer x^* of the original problem satisfying $x^* \in F^o$, then x^* can be achieved by solving the approximate problem $P(\alpha^k, T_k)$ for k large enough.

We will prove in the following lemma that if EMFCQ is satisfied everywhere in the feasible set F, then the condition $cl(F^o) = F$ in Theorem 3.4 can be achieved.

LEMMA 3.5. *Assume that EMFCQ holds for all $x \in F$, then we have*

$$cl(F^o) = F.$$

Proof. Theorem 2.1 in [14] shows that if EMFCQ is satisfied at all points of the feasible set F, then the boundary of F is

$$\partial F = \left\{ x \in \mathbb{R}^n \mid \max_{y \in Y} g(x, y) = 0 \right\}.$$

Since the index set $Y = [a, b]$ is compact, the function $\psi(x) = \max_{y \in Y} g(x, y)$ is continuous. It follows that the feasible set $F = \{x \in \mathbb{R}^n \mid \psi(x) \leq 0\}$ is closed. Thus, we have $cl(F^o) \subseteq F$. Furthermore,

$$F = \{x \in \mathbb{R}^n \mid \psi(x) < 0\} \cup \{x \in \mathbb{R}^n \mid \psi(x) = 0\} = F^o \cup \partial F.$$

It is sufficient to prove that $\partial F \subseteq cl(F^o)$.

For any $\bar{x} \in \partial F$, since EMFCQ holds at $\bar{x}$, there exists a vector $d \in \mathbb{R}^n$ such that

$$\nabla_x g(\bar{x}, y)^\top d < 0 \ \forall y \in A(\bar{x}), \tag{3.9}$$

where $A(\bar{x}) = \{y \in Y \mid g(\bar{x}, y) = 0\}$. The compactness of Y and the fact that $g(\bar{x}, y)$ is continuous imply that $A(\bar{x})$ is also a compact set. Thus, it follows from (3.9) that there exists a constant $\bar{\delta} > 0$ such that

$$\nabla_x g(\bar{x}, y)^\top d < -\bar{\delta} \ \forall y \in A(\bar{x}).$$

The continuity of $\nabla_x g(\bar{x}, y)$ implies that there exists a sufficiently small $\hat{\delta} > 0$ such that

$$\nabla_x g(\bar{x}, y)^\top d < -\frac{\bar{\delta}}{2} \ \forall y \in A_{\hat{\delta}}(\bar{x}), \tag{3.10}$$

where $A_{\hat{\delta}}(\bar{x}) = \{y \in Y \mid g(\bar{x}, y) \geq -\hat{\delta}\}$. Hence, it follows from (3.10) that there exists $\bar{\eta} > 0$ such that

$$g(\bar{x} + \eta d, y) = g(\bar{x}, y) + \eta \nabla_x g(\bar{x}, y)^\top d + o(\eta ||d||) < 0 \tag{3.11}$$

for all $\eta \in (0, \bar{\eta})$ and $y \in A_{\hat{\delta}}(\bar{x})$. For $y \in Y \backslash A_{\hat{\delta}}(\bar{x})$, we have $g(\bar{x}, y) < -\hat{\delta}$. Thus, there exists a $\hat{\delta} > 0$ such that

$$g(\bar{x} + \eta d, y) < 0 \ \forall \eta \in (0, \hat{\eta}), y \in Y \backslash A_{\hat{\delta}}(\bar{x}). \tag{3.12}$$

Now, from (3.11) and (3.12), we have that

$$g(\bar{x} + \eta d, y) < 0 \ \forall \eta \in (0, \tilde{\eta}), y \in Y,$$

where $\tilde{\eta} = \min\{\bar{\eta}, \hat{\eta}\}$. Thus, $\bar{x} + \eta d \in F^o \ \forall \eta \in (0, \tilde{\eta})$, which implies $\bar{x} \in cl(F^o)$. Therefore, we have

$$F = F^o \cup \partial F \subseteq cl(F^o).$$

This completes our proof. □

THEOREM 3.6. *Assume that EMFCQ holds everywhere in F. Also, assume that the initial subdivision T_0 is consistent and $\lim_{k\to\infty} |T_k| = 0$ with $T_k \subseteq T_{k+1}$. Let x_k^* be a solution of the approximate problem $P(\alpha^k, T_k)$, then every accumulation point x^* of the sequence $\{x_k^*\}_{k\in\mathbb{N}}$ is a solution of the original problem.*

Proof. Since $\{x_k^*\}_{k\in\mathbb{N}} \subseteq X$ and X is compact, there exists at least an accumulation point x^* of the sequence $\{x_k^*\}_{k\in\mathbb{N}}$. It suffices to prove that x^* is an optimal solution of the original problem. Assume without loss of generality that $\lim_{k\to\infty} x_k^* = x^*$. Since $F(\alpha^k, T_k) \subseteq F$ for all $k \in \mathbb{N}$ and F is closed, x^* is feasible for the original problem. From Theorem 3.4 and Lemma 3.5, the assumption that EMFCQ holds everywhere in F implies that

$$\lim_{k\to\infty} F(\alpha^k, T_k) = F.$$

Consequently, for any $x \in F$, there exists a point $\bar{x}_k \in F(\alpha^k, T_k)$ such that $\lim_{k\to\infty} \bar{x}_k = x$. Since x_k^* is a solution of $P(\alpha^k, T_k)$, we have

$$f(x_k^*) \le f(\bar{x}_k).$$

It follows that

$$f(x^*) = \lim_{k\to\infty} f(x_k^*) \le \lim_{k\to\infty} f(\bar{x}_k) = f(x).$$

This completes our proof. □

As pointed out by an anonymous referee, the above result can be proved by epi-convergence (see [15]).

Consider a special case that the original problem is convex, that is, the objective function f is convex and the constraint function $g(\cdot, y)$ is convex with respect to the decision variable x for all $y \in Y$. In this case, it is shown in [19] that the Slater condition holds if and only if EMFCQ holds everywhere in the feasible set. This implies that the condition $cl(F^o) = F$ holds for the convex semi-infinite problem under the Slater condition.

COROLLARY 3.7. *Assume that the problem (P) is convex and the Slater condition holds. Also, assume that the initial subdivision T_0 is consistent and $\lim_{k\to\infty} |T_k| = 0$ with $T_k \subseteq T_{k+1}$. Let x_k^* be a solution of the approximate problem $P(\alpha^k, T_k)$, then every accumulation point x^* of the sequence $\{x_k^*\}_{k\in\mathbb{N}}$ is a solution of the original problem.*

3.2. Adaptive subdivision algorithm. In this section, we present an algorithm to compute a Karush–Kuhn–Tucker point of the original problem within given tolerances. At each iteration, we solve the approximate problem $(P(\alpha, T))$ to obtain a Karush–Kuhn–Tucker point. The algorithm terminates if the current iterate is a Karush–Kuhn–Tucker point of the original problem within the given tolerance. Otherwise, the current subdivision is refined adaptively in the spirit of exchange methods [12, 27] and the method proposed in [10].

We know from Theorem 2.2 that if EMFCQ holds at a local minimizer x^*, then x^* must satisfy the corresponding Karush–Kuhn–Tucker conditions. The following

lemma implies that if EMFCQ holds at $x \in F$ for the original problem, then MFCQ holds at x for $(P(\alpha, T))$ when $|T|$ is sufficiently small.

LEMMA 3.8. *Assume that EMFCQ holds for all $x \in F$, then MFCQ holds everywhere in $F(\alpha, T)$ for all T with $|T|$ small enough.*

Proof. The result follows immediately from the assumption that $g(x, y)$ is twice continuously differentiable and the compactness of the active set $A(x)$ for all $x \in F$. □

We assume throughout the following parts that EMFCQ holds everywhere in F for the original problem and that MFCQ holds for the approximate problem $(P(\alpha, T))$. The first assumption is a strong assumption. Under this condition, it is proved in Lemma 3.5 that $cl(F^o) = F$. Furthermore, this condition guarantees that the set of the Lagrangian multipliers at a stationary point of the original problem is bounded [16].

In order to obtain a Karush–Kuhn–Tucker point of the original problem within given tolerances, we solve the approximate problem $(P(\alpha, T))$ with $T = \{\tau_i \mid i = 0, 1, \ldots, N\}$. Let x^* be a Karush–Kuhn–Tucker point of the problem $(P(\alpha, T))$; we have that

$$\nabla f(x^*) + \sum_{i=1}^{k} \lambda_i \nabla_x \bar{g}^i(x^*, y_i) = 0,\ y_i \in \bar{A}(x^*), \tag{3.13}$$

where $\bar{A}(x^*) = \{\tau_i \in T \mid \bar{g}^i(x^*, \tau_i) = 0\}$. From the definition of $\bar{g}$, it follows that

$$\nabla_x \bar{g}^i(x, \tau_i) = \nabla_x g(x, \tau_i),\ \tau_i \in T.$$

Thus, the Karush–Kuhn–Tucker conditions (3.13) can be reformulated as

$$\nabla f(x^*) + \sum_{i=1}^{k} \lambda_i \nabla_x g(x^*, y_i) = 0,\ y_i \in \bar{A}(x^*). \tag{3.14}$$

This implies that if x^* is a Karush–Kuhn–Tucker point of $(P(\alpha, T))$, it is an approximate Karush–Kuhn–Tucker point of the original problem. For designing our algorithm, we first define an (ϵ, δ) Karush–Kuhn–Tucker point of the original problem as follows.

DEFINITION 3.9. *$x^* \in F$ is called an (ϵ, δ) Karush–Kuhn–Tucker point of (P) if there exist some Lagrangian multipliers $\lambda_i \geq 0, i = 1, \ldots, k, k \leq n$, such that*

$$\left\| \nabla f(x^*) + \sum_{i=1}^{k} \lambda_i \nabla g(x^*, y_i) \right\| \leq \epsilon,\ y_i \in A(x^*, \delta),$$

where $A(x^*, \delta) = \{y \in Y \mid g(x^*, y) \in [-\delta, 0]\}$.

Similarly, $x^* \in F(\alpha, T)$ is called an (ϵ, δ) Karush–Kuhn–Tucker point of $(P(\alpha, T))$ if there exist some Lagrangian multipliers $\lambda_i \geq 0, i = 1, \ldots, k, k \leq n$, such that

$$\left\| \nabla f(x^*) + \sum_{i=1}^{k} \lambda_i \nabla g(x^*, y_i) \right\| \leq \epsilon,\ y_i \in \bar{A}(x^*, \delta),$$

where $\bar{A}(x^*, \delta) = \{y_i \in T \mid \bar{g}^i(x^*, y_i) \in [-\delta, 0]\}$.

The following lemma indicates that an (ϵ, δ) Karush–Kuhn–Tucker point of the original problem can be obtained directly by solving $(P(\alpha, T))$ for $|T|$ small enough.

LEMMA 3.10. *Suppose that x^* is an $(\epsilon, \frac{\delta}{2})$ Karush–Kuhn–Tucker point of $(P(\alpha, T))$ with $T = \{\tau_k \mid k \in \{0, 1, 2, \ldots, N\}\}$, then x^* is an (ϵ, δ) Karush–Kuhn–Tucker point of the original problem if*

$$\frac{\tilde{\alpha}}{8}|T|^2 \leq \frac{\delta}{2}, \tag{3.15}$$

where $\tilde{\alpha}$ is defined by (3.7).

Proof. It suffices to prove that if $y_i \in \bar{A}(x^*, \frac{\delta}{2})$, $i = 1, 2, \ldots, k$, then $y_i \in A(x^*, \delta)$, $i = 1, 2, \ldots, k$, provided that condition (3.15) holds. For each $y_i \in T$, we have

$$\bar{g}^i(x^*, y_i) = g(x^*, y_i) + \frac{\alpha_{\bar{i}}}{2}\left(\frac{\tau_{\bar{i}} - \tau_{\bar{i}-1}}{2}\right)^2.$$

$\bar{g}^i(x^*, y_i) \in \bar{A}(x^*, \frac{\delta}{2})$ implies that $g(x^*, y_i) \leq 0$ and

$$\begin{aligned} g(x^*, y_i) &\geq -\frac{\alpha_{\bar{i}}}{8}(\tau_{\bar{i}} - \tau_{\bar{i}-1})^2 - \frac{\delta}{2} \\ &\geq -\frac{\tilde{\alpha}}{8}|T|^2 - \frac{\delta}{2}. \end{aligned}$$

It follows that $y_i \in A(\bar{x}, \delta)$, $i = 1, 2, \ldots, k$, if the condition in the lemma holds. ∎

Consider a special case that the subdivision T is obtained by subdividing the index set Y into N equal subintervals. In this case, we have $|T| = \frac{b-a}{N}$. Lemma 3.10 implies that an (ϵ, δ) Karush–Kuhn–Tucker point of (P) can be obtained by solving $(P(\alpha, T))$ for $N \geq \sqrt{\frac{\tilde{\alpha}}{\delta}}\frac{(b-a)}{2}$. Therefore, the number of the constraints of the approximate problem $(P(\alpha, T))$, which equals $N + 1$, might be large when δ is small. Such an approximate problem $(P(\alpha, T))$ might be difficult to solve. In fact, to compute an (ϵ, δ) Karush–Kuhn–Tucker point x^* of the original problem, we solve $(P(\alpha, T))$ to obtain an $(\epsilon, \frac{\delta}{2})$ Karush–Kuhn–Tucker point x^* which satisfies

$$\left\|\nabla f(x^*) + \sum_{i=1}^{k} \lambda_i \nabla g(x^*, y_i)\right\| \leq \epsilon,\ y_i \in \bar{A}\left(x^*, \frac{\delta}{2}\right), i = 1, 2, \ldots, k, 1 \leq k \leq n.$$

It is obvious that if the subinterval $Y_{\bar{i}}$, which is related to the active index y_i, is small enough, that is $\frac{\alpha_{\bar{i}}}{8}(\tau_{\bar{i}} - \tau_{\bar{i}-1})^2 \leq \frac{\delta}{2}$, then we have $y_i \in A(x^*, \delta)$. Otherwise, we need to refine the current subdivision T by adding some new points to those subintervals which are related to the active index set $\bar{A}(x^*, \frac{\delta}{2})$.

At the kth iteration, the refinement procedure is taken to generate a new subdivision T_{k+1} based on the current subdivision T_k. Thus, we obtain a sequence of the approximate regions $\{F(\alpha^k, T_k)\}_{k\in\mathbb{N}}$. We expect that the sequence is monotone in the sense that

$$F(\alpha^k, T_k) \subseteq F(\alpha^{k+1}, T_{k+1}), k \in \mathbb{N}.$$

It will be shown that the monotone property of the sequence $\{F(\alpha^k, T_k)\}_{k\in\mathbb{N}}$ can be obtained by the trisection refinement strategy. Suppose we have a subdivision $T = \{\tau_k \mid k \in \{0, 1, \ldots, N\}\}$ and $Y^k = [\tau_{k-1}, \tau_k]$ is selected to be refined for some $k, 1 \leq k \leq N$. Let $\tau_{k,1}, \tau_{k,2}$ be the trisection points of the subinterval Y^k, that is,

$$\tau_{k,1} = \tau_{k-1} + \frac{1}{3}(\tau_k - \tau_{k-1}), \quad \tau_{k,2} = \tau_{k-1} + \frac{2}{3}(\tau_k - \tau_{k-1}).$$

The convex upper bound functions of $g(x,y)$ on each subset are defined by

$$g^{k,1}(x,y) = g(x,y) + \frac{\alpha_{k,1}}{2}\left(y - \frac{\tau_{k-1}+\tau_{k,1}}{2}\right)^2,\ y \in Y^{k,1} = [\tau_k, \tau_{k,1}],$$
$$g^{k,2}(x,y) = g(x,y) + \frac{\alpha_{k,2}}{2}\left(y - \frac{\tau_{k,1}+\tau_{k,2}}{2}\right)^2,\ y \in Y^{k,2} = [\tau_{k,1}, \tau_{k,2}],$$
$$g^{k,3}(x,y) = g(x,y) + \frac{\alpha_{k,3}}{2}\left(y - \frac{\tau_{k,2}+\tau_{k}}{2}\right)^2,\ y \in Y^{k,3} = [\tau_{k,2}, \tau_{k}],$$

where the parameters $\alpha_{k,i}$, $i = 1,2,3$, are chosen such that $\alpha_k \geq \alpha_{k,i}$, $i = 1,2,3$, and

$$\alpha_{k,i} \geq \max\left\{0, \max_{(x,y)\in X\times Y^{k,i}} -\nabla^2_{yy} g(x,y)\right\}, i = 1,2,3.$$

The new region $F(\alpha, T \cup \{\tau_{k,1}, \tau_{k,2}\})$ is defined by replacing the constraint

$$g^k(x,y) \leq 0\ \forall y \in Y^k$$

in $F(\alpha, T)$ by the following constraints:

$$g^{k,i}(x,y) \leq 0\ \forall y \in Y^{k,i}, i = 1,2,3.$$

LEMMA 3.11. *Let $T = \{\tau_k \mid k \in \{0,1,2,\ldots,N\}\}$ be a consistent subdivision of the index set Y. Assume that $\tau_{k,1}, \tau_{k,2}$ are trisection points of Y^k for some $k \in \{1,2,\ldots,N\}$ and $F(\alpha, T\cup\{\tau_{k,1},\tau_{k,2}\})$ is obtained by the refinement procedure. Then we have*

$$F(\alpha, T) \subseteq F(\alpha, T \cup \{\tau_{k,1}, \tau_{k,2}\}) \subseteq F.$$

Proof. It suffices to prove that for any fixed $x \in X$, the following holds:

$$g(x,y) \leq g^{k,i}(x,y) \leq g^k(x,y)\ \forall y \in Y^{k,i}, i = 1,2,3.$$

The first inequality can be obtained directly by the definitions of $g^{k,i}(x,y), i = 1,2,3$. The second inequality is equivalent to

$$\varphi_{k,i}(y, \alpha_{k,i}, Y^{k,i}) \leq \varphi_k(y, \alpha_k, Y^k)\ \forall y \in Y^{k,i}, i = 1,2,3, \tag{3.16}$$

where $\varphi_k(y, \alpha_k, Y^k) = \frac{\alpha_k}{2}(y - \frac{\tau_{k-1}+\tau_k}{2})^2,\ y \in Y^k$, and

$$\varphi_{k,1}(y, \alpha_{k,1}, Y^{k,1}) = \frac{\alpha_{k,1}}{2}\left(y - \frac{\tau_{k-1}+\tau_{k,1}}{2}\right)^2,\ y \in Y^{k,1} = [\tau_k, \tau_{k,1}],$$
$$\varphi_{k,2}(y, \alpha_{k,2}, Y^{k,2}) = \frac{\alpha_{k,2}}{2}\left(y - \frac{\tau_{k,1}+\tau_{k,2}}{2}\right)^2,\ y \in Y^{k,2} = [\tau_{k,1}, \tau_{k,2}],$$
$$\varphi_{k,3}(y, \alpha_{k,3}, Y^{k,3}) = \frac{\alpha_{k,3}}{2}\left(y - \frac{\tau_{k,2}+\tau_{k}}{2}\right)^2,\ y \in Y^{k,3} = [\tau_{k,2}, \tau_{k}].$$

It follows from $\alpha_{k,i} \leq \alpha_k$, $i = 1,2,3$, and $\frac{\tau_{k,1}+\tau_{k,2}}{2} = \frac{\tau_{k-1}+\tau_k}{2}$ that

$$\varphi_{k,2}(y, \alpha_{k,2}, Y^{k,2}) \leq \varphi_k(y, \alpha_k, Y^k)\ \forall y \in Y^{k,2}.$$

Therefore, (3.16) holds for $i = 2$. The following relation

$$\begin{aligned}&\nabla_y(\varphi_{k,1}(y,\alpha_{k,1},Y^{k,1})-\varphi_k(y,\alpha_k,Y^k))\\&\quad=\alpha_{k,1}\left(y-\frac{\tau_{k-1}+\tau_{k,1}}{2}\right)-\alpha_k\left(y-\frac{\tau_{k-1}+\tau_k}{2}\right)\\&\quad\geq 0\ \forall y\in Y^{k,1}=[\tau_{k-1},\tau_{k,1}],\end{aligned}$$

implies that $\varphi_{k,1}(y,\alpha_{k,1},Y^{k,1})-\varphi_k(y,\alpha_k,Y^k)$ is nondecreasing on $Y^{k,1}$. We only need to prove that $\varphi_{k,1}(\tau_{k,1},\alpha_{k,1},Y^{k,1})\leq\varphi_k(\tau_{k,1},\alpha_k,Y^k)$. This is true since

$$\varphi_{k,1}(\tau_{k,1},\alpha_{k,1},Y^{k,1})=\frac{\alpha_{k,1}}{72}(\tau_k-\tau_{k-1})^2\leq\frac{\alpha_k}{72}(\tau_k-\tau_{k-1})^2=\varphi_k(\tau_{k,1},\alpha_k,Y^k).$$

Thus (3.16) holds for $i = 1$. Similarly, we can prove that (3.16) holds for $i = 3$. This completes our proof. □

Note that not all the refinement strategies can guarantee that the sequence $\{F(\alpha^k,T_k)\}_{k\in\mathbb{N}}$ is monotone. We will show by an example that the bisection refinement cannot obtain such a property.

Let $g(x,y)=x+sin(y)$ and $X\times Y=[-10,0]\times[0,\pi]$. The parameter for $g(x,y)$ on $X\times Y$ is $\alpha=1$ and the upper bound function $g^1(x,y)$ of $g(x,y)$ on $X\times Y$ is

$$g^1(x,y)=g(x,y)+\frac{\alpha}{2}\left(y-\frac{\pi}{2}\right)^2=x+\sin(y)+\frac{1}{2}\left(y-\frac{\pi}{2}\right)^2.$$

Let $Y^{1,1}=[0,\frac{\pi}{2}]$ and $Y^{1,2}=[\frac{\pi}{2},\pi]$; it follows from direct calculations that $\alpha_{1,1}=\alpha_{1,2}=1$ and

$$\begin{aligned}g^{1,1}(x,y)&=x+\sin(y)+\frac{1}{2}\left(y-\frac{\pi}{4}\right)^2,\ y\in Y^{1,1},\\g^{1,2}(x,y)&=x+\sin(y)+\frac{1}{2}\left(y-\frac{3\pi}{4}\right)^2,\ y\in Y^{1,2}.\end{aligned}$$

Let $T=\{0,\pi\}$; we have

$$\begin{aligned}F(\alpha,T)&=\{x\in X\mid g^1(x,y)\leq 0,\forall y\in Y\}=\left[-10,-\frac{\pi^2}{8}\right],\\F\left(\alpha,T\cup\left\{\frac{\pi}{2}\right\}\right)&=\{x\mid g^{1,j}(x,y)\leq 0\ \forall y\in Y^{1,j},j=1,2\}=\left[-10,-\left(1+\frac{\pi^2}{32}\right)\right].\end{aligned}$$

Thus, it follows that

$$F\left(\alpha,T\cup\left\{\frac{\pi}{2}\right\}\right)\subseteq F(\alpha,T).$$

For designing feasible algorithms to solve semi-infinite optimization problems, two issues are essential: how to maintain the feasibility of each iterate and how to solve the approximate problems. In this paper, we mainly focus on the feasibility issue. Each of the approximate problems is a standard optimization problem and can be solved by existing software.

We present an adaptive subdivision algorithm (Algorithm 1) to solve the semi-infinite optimization problem (P). In our algorithm, we solve the approximate problems and refine the current subdivision adaptively at each iteration. Note that only

those subintervals which are related to the current active index set are selected to be refined.

In Algorithm 1, the initial subdivision T_0 is obtained by applying Algorithm 1 to the following semi-infinite problem (P_0) which is also suggested in the phase I procedure in [10], where

$$(P_0) \qquad \min_{(x,t)\in X\times\mathbb{R}} t \quad \text{s.t.} \quad g(x,y) \le t \ \forall y \in Y.$$

To initialize Algorithm 1 to solve the problem (P_0), we choose the trisection subdivision $\{a, a+\frac{b-a}{3}, a+\frac{2(b-a)}{3}, b\}$ and compute the corresponding parameter $\bar{\alpha}^0$ according to (3.2). The refinement procedure is implemented until Algorithm 1 finds a solution $(\bar{x}, \bar{t})$ with $\bar{t} \le 0$. The current subdivision is selected as the initial subdivision T_0. It is obvious that $\bar{x}$ is feasible for the approximate problem $P(\alpha^0, T_0)$. Thus, $\bar{x}$ can be chosen as an initial point in Algorithm 1. Since the semi-infinite problem (P_0) might not be convex, such a procedure is not necessarily successful all the time. However, numerical experiments in section 5 show that this method is efficient and works well to find a good initial subdivision. In the case that the method does not work, one may choose different initial points or select successively smaller subdivisions to initialize the semi-infinite problem (P_0) by applying Algorithm 1 to solve it. For example, we can subdivide the index set Y into N equal subintervals and increase N if necessary. Under the Slater condition, we can easily see that a consistent initial subdivision can be achieved provided that N is large enough.

ALGORITHM 1. ADAPTIVE SUBDIVISION ALGORITHM.

1: Determine a consistent initial subdivision $T_0 = \{\tau_k \mid k \in \{0, 1, 2, \ldots, N_0\}\}$ and compute the initial parameters α_k^0, $k = 1, 2, \ldots, N_0$, by (3.2). Set tolerances ϵ and δ. Let $j = 0$.

2: At jth iteration, solve $P(\alpha^j, T_j)$ to obtain an $(\epsilon, \frac{\delta}{2})$ Karush–Kuhn–Tucker point x_j^*, the Lagrangian multipliers $\{\lambda_i\}_{i=1}^k$, $k \le n$, and the active set index $\bar{A}(x_j^*, \frac{\delta}{2})$.

3: Terminate if x_j^* is an (ϵ, δ) Karush–Kuhn–Tucker point of the original problem. Otherwise, find out the index set I_j as follows:

$$I_j = \left\{ k \mid y_k \in \bar{A}\left(x_j^*, \frac{\delta}{2}\right) \text{ and } g(x_j^*, y_k) < -\delta \right\}.$$

4: Take the refinement procedure on subsets $Y^{\bar{k}}$, $k \in I_j$, to obtain T_{j+1} and compute α^{j+1}, where $\bar{k}$ is defined in (3.5).

5: Set $j = j + 1$ and go to step 2.

3.3. Convergence analysis. In this subsection, we will show that Algorithm 1 is well defined and terminates in finitely many iterations within arbitrarily given tolerances.

LEMMA 3.12. *Let T_j, $j \in \mathbb{N}$, be generated by Algorithm* 1. *Assume that $F(\alpha^j, T_j)$, $j \in \mathbb{N}$, are constructed by the refinement procedure. Denote by ν_j^* the optimal value of the approximate problem $P(\alpha^j, T_j)$. Then the following holds:*

(i) *$T_j \subseteq T_{j+1}$ for all $j \in \mathbb{N}$. If the termination criterion in Algorithm* 1 *is not satisfied, the inclusion is strict, i.e., $I_j \neq \emptyset$.*

(ii) *The sequence ν_j^*, $j \in \mathbb{N}$, is nonincreasing: $\nu_j^* \ge \nu_{j+1}^*$ for all $j \in \mathbb{N}$. Furthermore, each ν_j^* provides an upper bound for the optimal value of the original problem.*

Proof. (i) Assume that $T_j = T_{j+1}$ for some $j \in \mathbb{N}$ (or equivalently $I_j = \emptyset$), then we know that $g(x_j^*, y_i) \geq -\delta$ for $y_i \in \bar{A}(x_j^*, \frac{\delta}{2})$ and

$$\left\| \nabla f(x_j^*) + \sum_{i=1}^{n} \lambda_i \nabla g(x_j^*, y_i) \right\| \leq \epsilon, \quad y_i \in \bar{A}\left(x_j^*, \frac{\delta}{2}\right).$$

Since x_j^* is feasible for the original problem, $g(x_j^*, y_i) \leq 0$ for all $y_i \in Y$. This result, combined with the fact that $g(x_j^*, y_i) \geq -\delta$, implies that $y_i \in A(x_j^*, \delta)$, which contradicts the assumption that the termination criterion in Algorithm 1 is not satisfied.

(ii) By Lemma 3.11 and the result in (i), we have

$$F(\alpha^j, T_j) \subseteq F(\alpha^{j+1}, T_{j+1}) \subseteq F, j \in \mathbb{N},$$

which implies that $\nu_j^* \geq \nu_{j+1}^*$ for all $j \in \mathbb{N}$. Furthermore, the ν_j^*, $j \in \mathbb{N}$, provide upper bounds for the optimal value of the original problem. ∎

The following theorem states the finite termination of Algorithm 1.

THEOREM 3.13. *For any positive tolerance* (ϵ, δ)*, Algorithm* 1 *terminates in finitely many iterations.*

Proof. Assume that Algorithm 1 does not terminate in finite iterations. Recall that the subdivision of the index set Y at the jth iteration is

$$T_j = \{\tau_k \mid k \in K_j = \{0, 1, 2, \ldots, N_j\}\}.$$

By (i) of Lemma 3.12, the index set $I_j \neq \emptyset$ for $j \in \mathbb{N}$, where

$$I_j = \left\{ k \mid y_k \in \bar{A}\left(x_j^*, \frac{\delta}{2}\right) \text{ and } g(x_j^*, y_k) < -\delta \right\}.$$

It follows that for each $j \in \mathbb{N}$, $k \in I_j$ with $1 \leq k \leq n$, we have $\bar{g}^k(x_j^*, y_k) \in [\frac{\delta}{2}, 0]$ and $g(x_j^*, y_k) < -\delta$. Therefore,

$$\frac{\alpha_{\bar{k}}}{8} |Y^{\bar{k}}|^2 > \frac{\delta}{2}, 1 \leq k \leq n, k \in I_j, j \in \mathbb{N}, \tag{3.17}$$

where $\bar{k}$ is defined in (3.5) and

$$\bar{g}^k(x_j^*, y_k) = g(x_j^*, y_k) + \frac{\alpha_{\bar{k}}}{8} |Y^{\bar{k}}|^2.$$

Without loss of generality, we set $\bar{k} = k$, then (3.17) can be reformulated as

$$\frac{\alpha_k}{8} |Y^k|^2 > \frac{\delta}{2}, 1 \leq k \leq n, k \in I_j, j \in \mathbb{N}. \tag{3.18}$$

We show in the following that the $\frac{\alpha_k}{8}|Y^k|^2, k \in I_j, j \in \mathbb{N}$, converge to zero if Algorithm 1 does not terminate. Since all parameters $\alpha_k, k \in I_j, j \in \mathbb{N}$, are bounded by $\tilde{\alpha}$ which is defined by (3.7), it is sufficient to prove that the lengths of $Y^k, k \in I_j, j \in \mathbb{N}$, tend to zero as j tends to infinity.

We know from Lemma 3.12 that at least one of the subinterval Y^k, $k \in I_j$, is subdivided into three equal subsets $Y^{k,i}$, $i = 1, 2, 3$, with respect to the subdivision T_j at the jth iteration for $j \in \mathbb{N}$. The lengths of these subsets $Y^{k,i}$, $i = 1, 2, 3$, are bounded by $\frac{\tau_k - \tau_{k-1}}{3}$. Thus, for each integer p, Algorithm 1 generates at least

one point with the length of the corresponding subinterval bounded by $\frac{b-a}{3^p}$. For each $k \in I_j, 1 \le k \le n$, all the Y^k are different from each other. Since there are a finite number of subdivision points with the lengths of the corresponding subintervals greater than $\frac{b-a}{3^p}$ for each integer $p \in \mathbb{N}$, the lengths of Y^k, $k \in I_j, j \in \mathbb{N}$, converge to zero as j tends to infinity.

This result contradicts (3.18), which implies that Algorithm 1 terminates in finitely many iterations. ∎

THEOREM 3.14. *Assume that EMFCQ is satisfied for all $x \in F$. Also, assume that x_j is generated by Algorithm 1 with the given tolerance (ϵ_j, δ_j) for $j \in \mathbb{N}$. If $\lim_{j\to\infty}(\epsilon_j, \delta_j) = (0, 0)$, then any accumulation point of the sequence $\{x_j\}_{j\in\mathbb{N}}$ is a Karush–Kuhn–Tucker point of the original problem.*

Proof. Since EMFCQ holds for all $x \in F$, the set of Lagrangian multipliers is bounded for any Karush–Kuhn–Tucker point of the original problem [11]. Lemma 3.8 implies that MFCQ holds for all $x \in F(\alpha^j, T_j)$, $j \in \mathbb{N}$, when δ_j, $j \in \mathbb{N}$, are small enough. Theorem 3.13 indicates that x_j is an (ϵ_j, δ_j) Karush–Kuhn–Tucker point of the original problem for all $j \in \mathbb{N}$. Namely, there exist some indices y_i^j, $i = 1, 2, \ldots, k, k \le n$, $j \in \mathbb{N}$, and multipliers $\lambda_i^j \ge 0$, $i = 1, 2, \ldots, k, k \le n$, $j \in \mathbb{N}$, such that

$$\left\| \nabla f(x_j) + \sum_{i=1}^{k} \lambda_i^j \nabla_x g(x_j, y_i^j) \right\| \le \epsilon_j, \tag{3.19}$$

$$-\delta_j \le g(x_j, y_i^j) \le 0. \tag{3.20}$$

Since the sequence $\{(x_j, y_1^j, \ldots, y_k^j)\}_{j\in\mathbb{N}}$, $k \le n$, is contained in the compact set $X \times Y^k$, $k \le n$, and the set of the multipliers $\{(\lambda_1^j, \ldots, \lambda_k^j)\}_{j\in\mathbb{N}}$ is bounded, there exists an accumulation point $(x^*, y_1^*, \ldots, y_k^*, \lambda_1^*, \ldots, \lambda_k^*)$ of the sequence

$$\{(x_j, y_1^j, \ldots, y_k^j, \lambda_1^j, \ldots, \lambda_k^j)\}_{j\in\mathbb{N}}.$$

The result can be obtained immediately from (3.19), (3.20), and the assumption that $\lim_{j\to\infty}(\epsilon_j, \delta_j) = (0, 0)$. ∎

From Theorem 3.14, if the original problem is convex and the Slater condition holds, then every accumulation point of the sequence $\{x_j\}_{j\in\mathbb{N}}$ is an optimal solution to the original problem.

4. Generalizations and remarks. In this section, we first discuss some possible generalizations of our method. Then we present some remarks about such a method. Although our method can be improved in various ways, our motivation was to explain the basic idea and an implementable method in its simplest form. The exact improvements and generalizations are left for future works.

4.1. Generalizations. The method proposed in this paper can be directly extended to semi-infinite programs with finitely many semi-infinite constraints.

The extension of the index set Y from an interval to an m dimensional box $Y = [a, b] \subseteq \mathbb{R}^m$ is straightforward. The upper bound function $\bar{g}$ of g on $X \times Y$ can be set as

$$\bar{g}(x, y) = g(x, y) + \sum_{i=1}^{m} \alpha_i \left(y_i - \frac{a_i + b_i}{2} \right)^2,$$

where α_i, $1 \le i \le m$, are nonnegative parameters which can be chosen in a manner similar to this paper. The extension that the index set Y is an arbitrary compact set

in $\mathbb{R}^m$ concerns two issues: the subdivisions of the index set Y and the computation of the parameters in higher dimensions. The first one can be achieved by approximating the index set with the boxes. More precisely, we can construct boxes B_i, $i \in I$, such that $Y \subseteq \cup_{i\in I} B_i$. Then the upper bound function on each subset $X \times B_i$ is reduced to the former box case. This kind of technique proposed in [31] can be used to extend the convexification algorithm in [32] from the interval index set to the arbitrary compact index set. The more general case that the index set Y is dependent on the decision variable x is outside the scope of this paper.

4.2. Weaker assumption. At the beginning of this paper, we assumed that the constraint function $g(x, y)$ is twice continuously differentiable. In fact, this assumption can be replaced by a weaker one that $g(x, y)$ is continuously differentiable with respect to y on interval Y for all $x \in X$. In this case, $\bar{g}(x, y) = g(x, y) + \frac{\alpha}{2}(y - \frac{a+b}{2})^2$ is convex with respect to y on interval $Y = [a, b]$ for any fixed $x \in X$ if and only if

$$(\nabla_y \bar{g}(x, y_1) - \nabla_y \bar{g}(x, y_2))(y_1 - y_2) \geq 0 \ \forall y_i \in Y, i = 1, 2.$$

Due to the fact that $\nabla_y \bar{g}(x, y) = \nabla_y g(x, y) + \alpha(y - \frac{a+b}{2})$, the above condition is equivalent to

$$\alpha \geq \begin{cases} 0 & \text{if } y_1 = y_2, \\ \dfrac{\nabla_y g(x, y_2) - \nabla_y g(x, y_1)}{y_1 - y_2} \ \forall y_i \in Y, i = 1, 2, & \text{otherwise.} \end{cases}$$

In order to satisfy the above inequality for all $x \in X$, we require

$$\alpha \geq \max\left\{0, \max_{x\in X, y_i \in Y, y_1 \neq y_2} \frac{\nabla_y g(x, y_2) - \nabla_y g(x, y_1)}{y_1 - y_2}, i = 1, 2\right\}. \tag{4.1}$$

If $g \in C^2(\mathbb{R}^n \times \mathbb{R}^m)$, (4.1) is equivalent to (3.2). A simple observation is that more computation effort is needed for calculating the parameter α using (4.1) than (3.2).

4.3. Comparison with existing feasible methods. The adaptive convexification method proposed in [10] aims to solve the following approximate problem (for index set $Y = [0, 1]$) at each iteration for any given subdivision $T = \{\tau_k \mid k = 0, 1, 2, \ldots, N\}$:

$$\text{(MPCC)} \qquad \begin{aligned} \min_{x, y_k, \underline{\gamma}_k, \overline{\gamma}_k} \quad & f(x) \\ \text{s.t.} \quad & x \in X, \\ & g^k(x, y_k) \leq 0, \\ & \overline{\gamma}_k(\tau_k - y_k) = 0, \\ & \underline{\gamma}_k(y_k - \tau_{k-1}) = 0, \\ & \nabla_y g^k(x, y_k) + \underline{\gamma}_k - \overline{\gamma}_k = 0, \\ & \underline{\gamma}_k, \overline{\gamma}_k, y_k - \tau_{k-1}, \tau_k - y_k \geq 0, \ k = 1, 2, \ldots, N, \end{aligned}$$

where $g^k(x, y) = g(x, y) + \frac{\alpha_k}{2}(y - \tau_k)(\tau_k - y)$ is the concave upper bound function of $g(x, y)$ on $X \times [\tau_k, \tau_k]$ and $\underline{\gamma}_k, \overline{\gamma}_k, k = 0, 1, \ldots, N$, are auxiliary augmented Lagrangian variables.

We skip the box constraint $x \in X$ which is the same for our algorithm and the adaptive convexification algorithm. Problem (MPCC) is a mathematical program with $3N$ auxiliary variables. The total number of constraints in (MPCC) is $8N$ with $2N$ complementary constraints. The authors of [10] state that the standard numerical software might not be expected to solve this approximate problem since MFCQ does not hold at all feasible points due to the complementary constraints. They can only obtain stationary points in the sense of Fritz John.

For the same subdivision T, our approximate problem $(P(\alpha, T))$ contains $N+1$ inequality constraints without any auxiliary variables or complementary constraints compared with a convexification method. This contributes to fewer constraints and less computational effort.

The recent work [23] for generalized semi-infinite programs is an extension of the method proposed in [20]. Under the assumption that there exists a near optimal generalized semi-infinite program–Slater point, the algorithm is shown to converge finitely to a feasible point which is an approximate optimal solution of the original problem. The algorithm is simple to implement. The potential disadvantage is that the subproblems must be solved globally by existing global nonlinear optimization solvers. In our algorithm, the feasibility is guaranteed if the iterates are feasible for the subproblems. Furthermore, the subproblems can be solved by existing NLP solvers.

5. Numerical experiments. We implement Algorithm 1 in MATLAB 8.1 and use $fmincon$ of *Optimization Toolbox* 6.3 with default tolerance to solve the subproblems in step 2. In fact, the tolerance on the constraint violation is 10^{-6}. All the following experiments were run on 2.67 GHz Intel(R) Core(TM)2 processor.

For computing the parameters α^j, $j \in \mathbb{N}$, at each iteration, if they can be figured out analytically we use the closed form bounds. Otherwise, we use MATLAB toolbox *Intlab* 6.0 [25] to obtain the upper bounds of the parameters α^j, $j \in \mathbb{N}$.

We consider the following examples for numerical experiments with Algorithm 1, adaptive convexification algorithm (ACA for short) in [10] and MATLAB solver $fseminf$. The code SIPSOLVER is used for the adaptive convexification algorithm which is available at http://kop.ior.kit.edu/english/downloads.php. Note that this solver was developed for the general case $m \geq 1$ and might become quicker if streamlined to the case $m = 1$.

Example 1.

$$\begin{aligned} \min_{x \in \mathbb{R}^4} \quad & x_4 \\ \text{s.t.} \quad & \pm\left(\sin(\pi y) - x_3 y^2 - x_2 y - x_1\right) \leq x_4 \ \forall y \in [0,1], \\ & -1 \leq x_1 \leq 1, 3 \leq x_2 \leq 5, -5 \leq x_3 \leq -3, -1 \leq x_4 \leq 3. \end{aligned}$$

Example 1 is taken from [10]. It is a reformulation of the Chebyshev approximation problem with extra box constraint. Clearly, it is a convex semi-infinite optimization problem and the parameters α_k^+, α_k^- for each constraint on the subinterval $[\tau_k - 1, \tau_k] \subseteq [0,1]$ are given by

$$\begin{aligned} \alpha_k^+ &= \max(0, -\pi^2 \min(\sin(\pi\tau_{k-1}), \sin(\pi\tau_k)) + 10), \\ \alpha_k^- &= \max(0, \pi^2 \theta(\tau_{k-1}, \tau_k) - 6) \end{aligned}$$

with

$$\theta(\tau_{k-1},\tau_k)=\begin{cases}\sin(\pi\tau_k), & \tau_k\le 0.5,\\ 1, & \tau_{k-1}<0.5<\tau_k,\\ \sin(\pi\tau_{k-1}), & 0.5\le\tau_{k-1}.\end{cases}$$

We choose $(1,1,1,1)$ as the initial point of Algorithm 1.

Example 2.

$$\begin{aligned}\min_{x\in\mathbb{R}^3}\quad & x_1^2+x_2^2+x_3^2\\ \text{s.t.}\quad & x_1+x_2\exp(x_3y)+\exp(2y)-2\sin(4y)\le 0\ \forall y\in[0,1],\\ & -4\le x_i\le 2, i=1,2,3.\end{aligned}$$

Example 3.

$$\begin{aligned}\min_{x\in\mathbb{R}^2}\quad & \frac{1}{3}x_1^2+\frac{1}{2}x_1+x_2^2\\ \text{s.t.}\quad & (1-x_1^2y^2)^2-x_1y^2-x_2^2+x_2\le 0\ \forall y\in[0,1],\\ & -2\le x_i\le 2, i=1,2.\end{aligned}$$

Examples 2 and 3 are taken from [35]. We add extra box constraints to these examples to put them in the form of problem (P). For Example 2, the initial point is set as $(1,1,1)$ while the initial point for Example 3 is chosen as $(-1,-1)$.

Example 4.

$$\begin{aligned}\min_{x\in\mathbb{R}^2}\quad & x_1^2+(x_2-3)^2\\ \text{s.t.}\quad & x_2-2+x_1\sin\left(\frac{y}{x_2}-0.5\right)\le 0\ \forall y\in[0,10],\\ & -1000\le x_i\le 1000, i=1,2.\end{aligned}$$

Example 4 is taken from [34] and also tested in [8] with a modified region. We set the box constraint for variables $x_i, i=1,2$, as $-1000\le x_i\le 1000, i=1,2$. The initial point is set as $(1,-1)$.

Example 5.

$$\begin{aligned}\min_{x\in\mathbb{R}^{10}}\quad & \frac{1}{2}x^Tx\\ \text{s.t.}\quad & 3+4.5\sin(4.7\pi(y-1.23)/8)-\sum_{i=1}^{10}x_iy^{i-1}\le 0\ \forall y\in[0,1],\\ & -1000\le x_i\le 10, 1\le i\le 10.\end{aligned}$$

Example 5 is taken from [37] and also tested in [8] with different dimensions. We set the box constraint for variables $x_i, i=1,2,\ldots,10$, as $-1000\le x_i\le 10, i=1,2,\ldots,10$, and initial point as $ones(1,10)$.

Example 6.

$$\begin{aligned}\min_{x\in\mathbb{R}^2}\quad & x_2\\ \text{s.t.}\quad & -(x_1-y)^2-x_2\le 0\ \forall y\in[0,1],\\ & 0\le x_1\le 1, -1000\le x_2\le 1000.\end{aligned}$$

TABLE 1
Numerical results for Example 1–6 *by using Algorithm* 1 *and adaptive convexification algorithm.*

Example	Algorithm	fval	Iter	I.time(s)	CPU(s)	Tol.nodes	$G(x^*)$
1	Algorithm 1	0.028	15	0.04	0.54	63	−6.5628e-08
	ACA	0.028	7	0.4368	7.3476	-	−2.9354e-13
2	Algorithm 1	5.3347	10	0.0723	0.4324	28	−1.3404e-06
	ACA	5.3347	6	0.4836	163.32	-	9.7581e-05
3	Algorithm 1	0.1945	17	0.0512	0.6084	42	−1.8814e-06
	ACA	0.1945	34	0.7644	205.97	-	−3.3307e-16
4	Algorithm 1	1.0000	1	0.0752	0.1658	10	2.1817e-11
	ACA	-	-	-	-	-	-
5	Algorithm 1	0.0657	11	0.2367	0.7919	38	−6.3701e-06
	ACA	-	-	-	-	-	-
6	Algorithm 1	4.7042e-07	5	0.0407	0.1721	16	−4.7042e-07
	ACA	-	-	-	-	-	-

TABLE 2
Numerical results for Example 1–6 *by using Algorithm* 1 *and MATLAB Toobox* $fseminf$.

Example	Algorithm	fval	Iter	I.time(s)	CPU(s)	Tol.nodes	$G(x^*)$
1	Algorithm 1	0.028	15	0.04	0.54	63	−6.5628e-08
	$fseminf$	0.028	6	-	0.07	-	3.4461e-11
2	Algorithm 1	5.3347	10	0.0723	0.4324	28	−1.3404e-06
	$fseminf$	5.3128	44	-	0.2065	-	0.0535
3	Algorithm 1	0.1945	17	0.0512	0.6084	42	−1.8814e-06
	$fseminf$	0.1945	4	-	0.0289	-	−1.0264e-06
4	Algorithm 1	1.0000	1	0.0752	0.1658	10	2.1817e-11
	$fseminf$	1.0000	23	-	0.1914	-	6.1561e-08
5	Algorithm 1	0.0657	11	0.2367	0.7919	38	−6.3701e-06
	$fseminf$	0.0656	2	-	0.1198	-	9.1898e-04
6	Algorithm 1	4.7042e-07	5	0.0407	0.1721	16	−4.7042e-07
	$fseminf$	0	2	-	0.0435	-	0

Example 6 is taken from [12] and also tested in [7]. The initial point for this problem is $(1, 1)$.

We summarize in Table 1 the numerical results of Algorithm 1 and the adaptive convexification algorithm. The numerical results of Algorithm 1 and $fseminf$ are presented in Table 2. In both tables, $Iter$ represents the total number of iterations, $fval$ is the objective function value when the algorithms terminate, $I.time$ is the CPU time cost in finding an initial consistent subdivision while CPU is the total CPU time, $Tol.nodes$ is the number of nodes in the final subdivision when the algorithms terminate, $G(x^*) = \max_{y\in\bar{Y}} g(x^*, y)$ measures the feasibility of x^* where $\bar{Y} = a : 10^{-6} : b$. In MATLAB toolbox $fseminf$, we set the initial sampling interval as $a : 0.001 : b$.

Table 1 shows a comparison of the numerical results between Algorithm 1 and the adaptive convexification algorithm. For the first three examples, we can see that both algorithms obtain the same objective function values. For finding an initial subdivision, our algorithm use less time compared with the adaptive convexification algorithm. CPU in the 6th column shows that our algorithm is much faster than the adaptive convexification algorithm. SIPSOLVER, solver of the adaptive convexification algorithm, fails to solve the last three examples.

Consider the numerical performance of Algorithm 1 and $fseminf$ in Table 2. We can see that the objective function values for all the examples are almost the same. The last column in Table 2 shows that our algorithm computes a feasible Karush–Kuhn–Tucker point for each problem which coincides with the theoretical

results. The feasibility of Example 4 violates because of the numerical accuracy. On the other hand, $fseminf$ is faster than Algorithm 1 in most cases. However, it cannot guarantee the feasibility of the iterates when it terminates.

We can see from Tables 1 and 2 that the total number of nodes in column 7 is not too large which indicates that our adaptive refinement procedure works well. It follows that a small number of the subdivision nodes are needed to obtain an Karush–Kuhn–Tucker point within the given tolerance.

6. Conclusions and future work. A novel numerical method to solve semi-infinite programs with interval index set is presented. Based on the concave relaxations of the lower level problem, the inner approximate regions consisted of finitely many constraints constructed to approximate the original feasible region. The approximation guarantees that each iterate is feasible for the original problem. An adaptive subdivision algorithm is proposed to solve the semi-infinite optimization problems. The trisection refinement procedure is taken adaptively to guarantee that the sequence of the approximate regions is monotone.

Various methods might be considered for the acceleration of the method proposed in this paper. On of them is to drop some subdivision points when the subdivisions are updated using the idea of exchange method. The main issue of such a method is that the resulting iterates obtained by solving the approximate problems might not be feasible for the original problem. Another way to accelerate the algorithm is to compute tighter parameters on the subsets of the domain $X \times Y$. This requires the additional computation effort. We leave these questions for future research.

Acknowledgment. The authors are grateful to two anonymous referees for their detailed and valuable comments and suggestions which helped a lot in improving our paper.

REFERENCES

[1] C. S. Adjiman, I. P. Androulakis, and C. A. Floudas, *A global optimization method, αBB, for general twice-differentiable constrained NLPs–I. Theoretical advances*, Comput. Chem. Eng., 22 (1998), pp. 1137–1158.

[2] C. S. Adjiman, I. P. Androulakis, and C. A. Floudas, *A global optimization method, αBB, for general twice-differentiable constrained NLPs–II. Implementation and computational results*, Comput. Chem. Eng., 22 (1998), pp. 1159–1179.

[3] J. P. Aubin and H. Frankowska, *Set-Valued Analysis*, Birkhäuser, Boston, 2009.

[4] G. Alefeld and G. Mayer, *Interval analysis: Theory and applications*, J. Comput. Appl. Math., 121 (2000), pp. 421–464.

[5] D. Bertsimas, D. B. Brown, and C. Caramanis, *Theory and applications of robust optimization*, SIAM Rev., 53 (2011), pp. 464–501.

[6] B. Bhattacharjee, W. H. Green, and P. I. Barton, *Interval methods for semi-infinite programs*, Comput. Optim. Appl., 30 (2005), pp. 63–93.

[7] B. Bhattacharjee, P. Lemonidis, W. H. Green, and P. I. Barton, *Global solution of semi-infinite programs*, Math. Program., 103 (2005), pp. 283–307.

[8] L. Chen, Q. Ni, L. Qi, and S. Y. Wu, *A new smoothing Newton-type algorithm for semi-infinite programming*, J. Global Optim., 47 (2010), pp. 133–159.

[9] S. Daum, and R. Werner, *A novel feasible discretization method for linear semi-infinite programming applied to basket option pricing*, Optimization, 60 (2011), pp. 1379–1398.

[10] C. A. Floudas and O. Stein, *The adaptive convexification algorithm: A feasible point method for semi-infinite programming*, SIAM J. Optim., 18 (2008), pp. 1187–1208.

[11] J. Gauvin, *A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming*, Math. Program., 12 (1977), pp. 136–138.

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

[13] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, *Lipschitzian optimization without the Lipschitz constant*, J. Optim. Theory Appl., 79 (1993), pp. 157–181.
[14] H. T. Jongen, F. Twilt, and G. W. Weber, *Semi-infinite optimization: Structure and stability of the feasible set*, J. Optim. Theory Appl., 72 (1992), pp. 529–552.
[15] P. Kall, *Approximation to optimization problems: An elementary review*, Math. Oper. Res., 11 (1986), pp. 9–18.
[16] D. Klatte and R. Henrion, *Regularity and stability in nonlinear semi-infinite optimization*, Semi-Infinite Programming, Springer, Boston, 1998, pp. 69–102.
[17] R. Krawczyk and A. Neumaier, *Interval slopes for rational functions and associated centered forms*, SIAM J. Numer. Anal., 22 (1985), pp. 604–616.
[18] Q. Lin, R. Loxton, and K. L. Teo, *Optimal control of nonlinear switched systems: Computational methods and applications*, J. Oper. Res. Soc. China, 1 (2013), pp. 275–311.
[19] M. López and G. Still, *Semi-infinite programming*, European J. Oper. Res., 180 (2007), pp. 491–518.
[20] A. Mitsos, *Global optimization of semi-infinite programs via restriction of the right-hand side*, Optimization, 60 (2011), pp. 1291–1308.
[21] R. Moore, *Methods and applications of interval analysis*, SIAM Stud. Appl. Math. 2, SIAM, Philadelphia, 1979.
[22] A. Mitsos, P. Lemonidis, C. K. Lee, and P. I. Barton, *Relaxation-based bounds for semi-infinite programs*, SIAM J. Optim., 19 (2008), pp. 77–113.
[23] A. Mitsos and A. Tsoukalas, *Global optimization of generalized semi-infinite programs via restriction of the right hand side*, J. Global Optim., 2014, pp. 1–17.
[24] A. Potchinkov and R. Reemsten, *The design of FIR filter in the complex plane by convex optimization*, Signal Process, 46 (1995), pp. 127–146.
[25] S. M. Rump, *INTLAB-INTerval LABoratory*, http://www.ti3.tu-harburg.de/rump/intlab (1999).
[26] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[27] R. Reemtsen and S. Görner, *Numerical methods for semi-infinite programming: A survey*, in Semi-Infinite Programming, R. Reemtsen and J.-J. Rückmann, eds., Kluwer, Boston, 1998, pp. 195–275.
[28] O. Stein, *Bi-level Strategies in Semi-infinite Programming*, Kluwer, Boston, 2003.
[29] O. Stein, *How to solve a semi-infinite optimization problem*, European J. Oper. Res., 223 (2012), pp. 312–320.
[30] H. Scheel and S. Scholtes, *Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity*, Math. Oper. Res., 25 (2000), pp. 1–22.
[31] O. Stein and P. Steuermann, *The adaptive convexification algorithm for semi-infinite programming with arbitrary index sets*, Math. Program., 136 (2012), pp. 183–207.
[32] O. Stein and A. Winterfeld, *Feasible method for generalized semi-infinite programming*, J. Optim. Theory Appl., 146 (2010), pp. 419–443.
[33] R. Tichatschke and V. Nebeling, *A cutting plane method for quadratic semi-infinite programming problems*, Optimization, 19 (1988), pp. 803–817.
[34] K. L. Teo, X. Q. Yang, and L. S. Jennings, *Computational discretization algorithms for functional inequality constrained optimization*, Ann. Oper. Res., 98 (2000), pp. 215–234.
[35] G. A. Watson, *Numerical experiments with globally convergent methods for semi-infinite programming problems*, in Semi-infinite Programming and Applications. Springer, Berlin, 1983, pp. 193–205.
[36] R. Werner, *Consistency of robust optimization with application to portfolio optimization*, eprint, optimization-online:2010/09/2747, 2010.
[37] S. Y. Wu, D. H. Li, L. Qi, and G. Zhou, *An iterative method for solving KKT system of the semi-infinite programming*, Optim. Methods Softw., 20 (2005), pp. 629–643.
[38] L. Zhang, S.-Y. Wu, and M. A. López, *A new exchange method for convex semi-infinite programming*, SIAM J. Optim., 20 (2010), pp. 2959–2977.