Comput Optim Appl (2012) 53:91–113
DOI 10.1007/s10589-011-9452-9

Relaxed cutting plane method with convexification for solving nonlinear semi-infinite programming problems

Ting-Jang Shiu · Soon-Yi Wu

Received: 28 March 2011 / Published online: 13 January 2012

Abstract In this paper, we present an algorithm to solve nonlinear semi-infinite programming (NSIP) problems. To deal with the nonlinear constraint, Floudas and Stein (SIAM J. Optim. 18:1187–1208, 2007) suggest an adaptive convexification relaxation to approximate the nonlinear constraint function. The αBB method, used widely in global optimization, is applied to construct the convexification relaxation. We then combine the idea of the cutting plane method with the convexification relaxation to propose a new algorithm to solve NSIP problems. With some given tolerances, our algorithm terminates in a finite number of iterations and obtains an approximate stationary point of the NSIP problems. In addition, some NSIP application examples are implemented by the method proposed in this paper, such as the proportional-integral-derivative controller design problem and the nonlinear finite impulse response filter design problem. Based on our numerical experience, we demonstrate that our algorithm enhances the computational speed for solving NSIP problems.

Keywords Nonlinear semi-infinite programming · Convexification approximation · Cutting plane method · Control design problem

1 Introduction

The semi-infinite programming problem is an optimization problem with finite-dimensional decision variables $x \in \mathbb{R}^n$ and infinite number of constraints. We are

T.-J. Shiu · S.-Y. Wu (✉)
Department of Mathematics, National Cheng Kung University, Tainan, Taiwan
e-mail: soonyi@mail.ncku.edu.tw

T.-J. Shiu
e-mail: tjshiu@gmail.com

S.-Y. Wu
Taiwan and National Center for Theoretical Sciences, Tainan, Taiwan

concerned about the following nonlinear semi-infinite programming problem (NSIP) in this paper:

$$\text{(NSIP):} \quad \begin{aligned} &\min_{x \in X} f(x) \\ &\text{subject to} \quad g(x, y) \leq 0, \quad y \in Y, \end{aligned}$$

where X is a box constraint set in $\mathbb{R}^n$ with $x^L \leq x \leq x^U$. Both x^L and x^U are n-dimensional vectors with $x^L < x^U$; the vector inequalities are understood as componentwise. The objective function $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable on X; set Y is a fixed nonempty compact subset of $\mathbb{R}^m$; and the constraint g is a twice differentiable function with respect to x and y. A more general case where the x-dependent index set $Y(x)$ substitutes for the fixed index set Y is called generalized semi-infinite programming problems (GSIP). In this paper, we focus on the case of $m = 1$ for the sake of convenience. Moreover, without loss of generality, we let $Y = [0, 1]$ throughout this paper.

The optimality condition and duality theorem for semi-infinite programming problems (SIP) have been studied since the 1960s. A complete survey on the optimality condition and duality theory for linear, convex, and smooth SIP problems is clarified in [22]. Many applied problems can be treated as SIP problems, such as approximation theory, economic problems, and numerous engineering problems such as optimal control, robot trajectory planning, digital filter design, air pollution control, and production planning. We refer to [12], an excellent review paper on SIP problem including the theory, algorithms, and some applications. Furthermore, [19, 21] summarize the existing and typically used algorithms for solving SIP problems.

The main difficulty in solving SIP problems is the infinite number of constraints. In the last two decades, many algorithms have been proposed to solve SIP problems. Among these, the most popular method for solving SIP problems is the discretization method (cf. [19, 23, 25]). Another common approach, the reduction-based method, is reduced locally to a finite-dimensional nonlinear programming problem (see, e.g. [11, 18, 24]). In these two methods, the discretization method costs much computational time when the cardinality of the finite set of Y grows. Moreover, although the reduction-based method, which utilizes an exact penalty function, yields global convergence, it usually requires stronger assumptions and is implemented in a rather simple form. In addition to the foregoing methods, the Karush-Kuhn-Tucker (KKT) system of the problem (NSIP) is reformulated into a system of semi-smooth equations (cf. [15, 20, 26]). The authors then apply the smoothing, semi-smoothing Newton method, or Newton-type method to solve the KKT system. In the recent studies, a branch-and-bound technique is used to generate convergent sequences of upper and lower bounds on the SIP solution value in these articles [5, 6]. The authors focus on finding the global solution of SIP problems and present the algorithm with feasible iterates for SIP problems.

In addition, one of the important methods, the exchange method, is proposed in [4, 14]. The exchange method, also called the cutting plane method, provides an outer approximation of the feasible set of SIP problems. For each iterate, this algorithm reduces the infinite index set Y into a finite index subset of Y. However, these authors need to evaluate a problem called lower level optimization problem at every iteration:

$$\max g(\bar{x}, y) \quad \text{subject to} \quad y \in Y, \tag{1}$$

for a given point $\bar{x} \in \mathbb{R}^n$. Determining the global maximizer of problem (1) numerically may be difficult, especially in the case where $g(\bar{x}, \cdot)$ is non-convex with respect to the variable y. In fact, the standard solver for (1) can only be expected to obtain a local maximizer, not the global one. Nevertheless, to solve the SIP problem, where both $f(x)$ and $g(x, \cdot)$ are convex functions, Zhang et al. [28] propose a new exchange method. This method only requires finding some points with a certain computationally easy criterion instead of finding the global optimal solution of the problem (1).

Recently, Floudas and Stein [8] propose an adaptive convex relaxation of the lower level problem (1). The authors split Y into a finite number of subintervals, and substituted an overestimated concave function regarded as the approximate constraint for each subinterval. They produced feasible iterates for the SIP problems. Although the adaptive convexification method provides an idea to solve the global maximum of the lower level problem (1) efficiently, we find that the computation may be costly when the number of subintervals increases. Therefore, we intend to integrate the idea of the exchange method into the adaptive convexification method, and then propose a modified algorithm in this paper. We now denote that $\mathcal{F}$ is the feasible set of the problem (NSIP) as follows:

$$\mathcal{F} = \{x \in X \mid g(x, y) \leq 0, \text{ for all } y \in Y\}.$$

Here, we provide the first order optimality condition for the problem (NSIP) from [22].

Theorem 1 *Let $\tilde{x} \in \mathcal{F}$ be a local minimizer of (NSIP) and $Y_{\text{act}}(\tilde{x}) := \{y \in Y | g(\tilde{x}, y) = 0\}$ be a nonempty set. There exists $\tilde{y}_k \in Y_{\text{act}}(\tilde{x})$, for $1 \leq k \leq n+1$, and $(\lambda_0, \lambda_1, \ldots, \lambda_{n+1}) \in \mathbb{R} \times \mathbb{R}^{n+1}$, where $\lambda_0, \lambda_1, \ldots, \lambda_{n+1} \geq 0$ such that*

$$\lambda_0 \nabla f(\tilde{x}) + \sum_{k=1}^{n+1} \lambda_k \nabla_x g(\tilde{x}, \tilde{y}_k) = 0, \tag{2}$$

$$\lambda_k \cdot g(\tilde{x}, \tilde{y}_k) = 0, \quad 1 \leq k \leq n+1, \tag{3}$$

$$g(\tilde{x}, y) \leq 0, \quad \forall y \in Y; \tag{4}$$

hold. Furthermore, a point $\tilde{x} \in \mathcal{F}$ is said to satisfy the extended Mangasarian-Fromovitz constraint qualification (EMFCQ) if there exits a direction $d \in \mathbb{R}^n$ such that $\nabla_x g(\tilde{x}, y)^\top d < 0$ for all $y \in Y_{\text{act}}(\tilde{x})$. If EMFCQ is valid at $\tilde{x}$, we can set $\lambda_0 = 1$ in (2).

The point $\tilde{x}$ that satisfies (2)–(4) is called the *stationary point* for the problem (NSIP). Throughout this paper, we assume that *EMFCQ* holds at every stationary point of the problem (NSIP). Under *EMFCQ*, (2)–(4) with $\lambda_0 = 1$ are called KKT optimality conditions. Regarding the result in [22], the set $\{\lambda_k\}$ is nonempty and bounded when the *EMFCQ* is fulfilled.

We organize this paper as follows. In Sect. 2, we introduce one kind of convex relaxed methods: the αBB method. The algorithm used to solve the generated subproblem is also mentioned in this section. We propose a modified adaptive convexification algorithm in Sect. 3 and show the convergence properties in Sect. 4. Finally, we apply our algorithm to some examples and present the computational efficiency of the algorithm.

2 Preliminaries

2.1 The αBB method

Based on the convergence properties of the existing global optimization algorithms, these algorithms can be categorized into deterministic and stochastic. One of the deterministic global optimization algorithms, the αBB (α-based Branch and Bound) method, offers mathematical guarantees for convergence to find a point arbitrarily close to the global minimum for the twice-differentiable nonlinear programming problems. In the αBB method, a convex underestimator of a nonconvex function is constructed by decomposing it into a numerous sum of nonconvex terms (cf. [1, 2, 7]). We mention in Sect. 1 that obtaining the global maximizer of the lower level optimization problem (1) may be difficult. Therefore, the concept of the αBB method is used to construct the concave overestimator for the nonconvex constraint $g(x, y)$. We now divide Y into N subintervals with $Y = \bigcup_{j=1}^{N}[\eta_j^L, \eta_j^U]$, where $\eta_1^L = 0$, $\eta_N^U = 1$ and $\eta_j^L < \eta_j^U = \eta_{j+1}^L$, for $j = 1, 2, \ldots, N - 1$. The single semi-infinite constraint, $g(x, y) \le 0,\ y \in Y$, can be considered N semi-infinite constraints under the discretization

$$g(x, y) \le 0, \quad y \in Y_j, \quad \text{for } j = 1, 2, \ldots, N,$$

where $Y_j = [\eta_j^L, \eta_j^U]$. With the obvious modification, we construct a concave overestimator by adding a quadratic relaxation function ψ for a nonconcave function $g : [\eta_j^L, \eta_j^U] \to \mathbb{R}$, a C^2-function on an open neighborhood of $[\eta_j^L, \eta_j^U]$. Set

$$\psi(y : \alpha_j, \eta_j^L, \eta_j^U) = \frac{\alpha_j}{2}(y - \eta_j^L)(\eta_j^U - y),$$

and define $g_j(x, y)$, for $j = 1, 2, \ldots, N$, by

$$g_j(x, y) = g(x, y) + \psi(y : \alpha_j, \eta_j^L, \eta_j^U), \quad \text{for } y \in Y_j. \tag{5}$$

Note that if $\alpha_j > 0$, the function $g_j(x, y)$ is greater than $g(x, y)$ except for the boundary points η_j^L and η_j^U, i.e., $g(x, y) \le g_j(x, y)$ for $x \in X,\ y \in Y_j$. To ensure that $g_j(x, y)$ is a concave function with respect to y on the subinterval Y_j, α_j must satisfy the following condition:

$$\alpha_j > \max_{(x,y)\in X\times Y_j} \frac{\partial^2}{\partial y^2} g(x, y),$$

and α_j must be positive. Thus,

$$\alpha_j > \max\left(0, \max_{(x,y)\in X\times Y_j} \frac{\partial^2}{\partial y^2} g(x, y)\right). \tag{6}$$

We call α_j, which satisfies (6), as the *convexification parameter* corresponding to Y_j, and denote the vector of the convexification parameters by $\alpha = (\alpha_1, \ldots, \alpha_N)$. The maximal separation distance between $g(x, y)$ and $g_j(x, y)$ on $X \times Y_j$ is defined by the maximum of the difference of these functions; i.e., for any $x \in X$

$$d_{\alpha\mathrm{BB}}(y; \alpha_j) := \max_{y \in Y_j} \big(g_j(x, y) - g(x, y)\big) = \frac{\alpha_j}{8}(\eta_j^U - \eta_j^L)^2. \tag{7}$$

Clearly, the distance will converge to zero as $(\eta_j^U - \eta_j^L) \to 0$.

Next, we define some notation. Let $E = \{Y_j = [\eta_j^L, \eta_j^U] \,|\, j = 1, \ldots, N\}$ be the *set of collections of subintervals* and the vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N)$ be the *corresponding convexification parameters on* E, where α_j, defined in (6), is the j-th component of α, i.e., α_j is the convexification parameter corresponding to Y_j, for $j = 1, 2, \ldots, N$. The approximate feasible set $\mathcal{F}_{\alpha\mathrm{BB}}(E, \alpha)$ is then defined as

$$\mathcal{F}_{\alpha\mathrm{BB}}(E, \alpha) = \{x \in X \,|\, g_j(x, y) \le 0, \text{ for } y \in Y_j, \ Y_j \in E\},$$

where $g_j(x, y) = g(x, y) + \frac{\alpha_j}{2}(y - \eta_j^L)(\eta_j^U - y)$ is an approximate concave function on the subinterval Y_j. Obviously, $\mathcal{F}_{\alpha\mathrm{BB}}(E, \alpha)$ is a subset of $\mathcal{F}$. Furthermore, we replace the single semi-infinite constraint $g(x, y) \le 0$ by N semi-infinite constraints,

$$g_j(x, y) \le 0, \quad y \in Y_j, \quad \text{for } j = 1, 2, \ldots, N.$$

With the substitution of the single semi-infinite constraint, we obtain an approximate problem $(\mathrm{NSIP}_{\alpha\mathrm{BB}}(E, \alpha))$ for problem (NSIP):

$$\begin{aligned} (\mathrm{NSIP}_{\alpha\mathrm{BB}}(E, \alpha))\colon \quad & \min_{x \in X} f(x) \\ & \text{subject to} \quad g_j(x, y) \le 0, \quad y \in Y_j, \ j = 1, 2, \ldots, N. \\ & \phantom{\text{subject to}} \quad Y_j \in E. \end{aligned}$$

The inequality (6) is strict, and thus the problem $(\mathrm{NSIP}_{\alpha\mathrm{BB}}(E, \alpha))$ has a finite number of strict concave constraints $g_j(\cdot, y)$. Therefore, for any fixed point $\bar{x} \in X$ and each concave constraint function $g_j(\bar{x}, y)$, $j = 1, 2, \ldots, N$, the lower level problem

$$\max g_j(x, y) \quad \text{subject to} \quad y \in Y_j,$$

has an unique global optimal solution $\bar{y}_j$. We can obtain $\bar{y}_j$ without difficulty. Next, we will present a method to solve the problem $(\mathrm{NSIP}_{\alpha\mathrm{BB}}(E, \alpha))$.

2.2 The cutting plane method

In this subsection, we introduce the cutting plane method to solve the problem $(\mathrm{NSIP}_{\alpha\mathrm{BB}}(E, \alpha))$. The cutting plane method is commonly used when solving SIP problems. Let $J = \{1, 2, \ldots, N\}$ be the index set of the collection of subintervals, where N is the number of subintervals in E. We present the cutting plane method to solve the problem $(\mathrm{NSIP}_{\alpha\mathrm{BB}}(E, \alpha))$ in the following algorithm.

Algorithm 1 The cutting plane method for solving the problem $(\mathrm{NSIP}_{\alpha\mathrm{BB}}(E,\alpha))$

Step 1: For each $Y_j \in E$, choose a finite set $T_j^1 = \{y_{j,1}, \dots, y_{j,k_j^1}\}$ such that $T_j^1 \subset Y_j$. Let $T^1 = \bigcup_{j\in J} T_j^1$ and $p = 1$.

Step 2: Solve the nonlinear programming problem $(\mathrm{NSIP}_{\alpha\mathrm{BB}}(T^p))$ to obtain the optimal solution $\tilde{x}^p$, where

$$(\mathrm{NSIP}_{\alpha\mathrm{BB}}(T^p)):\quad \min_{x\in X} f(x) \quad \text{subject to} \quad g_j(x, y_{j,i}) \le 0, \quad \text{for } i = 1, 2, \dots, k_j^p,\ j \in J.$$

Step 3: For each $j \in J$, find the optimal solution $\tilde{y}_j$ for the following problem:

$$\max g_j(\tilde{x}^p, y) \quad \text{subject to} \quad y \in Y_j.$$

If $g_j(\tilde{x}^p, \tilde{y}_j) \le 0$ for all $j \in J$, we stop the algorithm and output $\tilde{x}^p$ as the optimal solution. Otherwise, define the set T_j^{p+1} by

$$T_j^{p+1} = \begin{cases} T_j^p \cup \{\tilde{y}_j\} & \text{if } g_j(\tilde{x}^p, \tilde{y}_j) > 0; \\ T_j^p & \text{if } g_j(\tilde{x}^p, \tilde{y}_j) \le 0. \end{cases}$$

Let $k_j^{p+1} = |T_j^{p+1}|$ be the cardinal number of reference set on Y_j at $(p+1)$-iteration for $j \in J$. Set $p := p + 1$, and go to Step 2.

Associating with each finite subset T^p at p-iteration, the problem $(\mathrm{NSIP}_{\alpha\mathrm{BB}}(E,\alpha))$ is reduced to a finite dimensional nonlinear programming problem. Therefore, we can solve it using a common NLP solver to obtain the optimal solution $\tilde{x}^p$. We subsequently check the feasibility for each constraint $g_j(\tilde{x}^p, y)$ on Y_j, for $Y_j \in E$. Next, we add the worst infeasible point to T_j^p if there is an infeasible point on Y_j. The algorithm terminates when no infeasible point exists in all subintervals in E. The convergence proof for Algorithm 1 can be found in [19, 27]. The convergence proof in [19, 27] is satisfied for every point in a fixed interval, and thus the convergence result of Algorithm 1 naturally holds. Therefore, we omit the proof here.

3 Modified convexification algorithm

The main idea of our algorithm is to modify the adaptive convexification algorithm proposed by Floudas and Stein [8]. They solved the problem $(\mathrm{NSIP}_{\alpha\mathrm{BB}}(E,\alpha))$ and refined some of the subintervals to approach the stationary point of the problem (NSIP). Let any given predefined set E^0 be the collection of some subintervals in Y defined by

$$E^0 = \{Y_k^0 \,|\, k = 1, 2, \dots, m_0\},$$

where Y_k^0 is a subinterval of Y with lower bound $\eta_k^{L,0}$ and upper bound $\eta_k^{U,0}$, for $k = 1, 2, \dots, m_0$, and $m_0 := |E^0|$ denotes the cardinal number of E^0. Note that

E^0 only collects some of the subintervals of Y, not all of the subintervals. That is, $\bigcup_{k=1}^{m_0} Y_k^0 \subsetneqq Y$. The convexification parameter α_k^0 on Y_k^0 is defined in (6):

$$\alpha_k^0 > \max\left(0, \max_{(x,y)\in X\times Y_k^0} \frac{\partial^2}{\partial y^2} g(x,y)\right)$$

Let $\alpha^0 = (\alpha_1^0, \ldots, \alpha_{m_0}^0)$ be the corresponding convexification parameter on E^0. We substitute E^0 for E and propose a modified subproblem $(\text{NSIP}_{\alpha\text{BB}}(E^0, \alpha^0))$:

$$\begin{aligned}(\text{NSIP}_{\alpha\text{BB}}(E^0,\alpha^0)):\quad & \min_{x\in X} f(x) \\ & \text{subject to} \quad g_k^0(x,y) \le 0, \qquad y \in Y_k^0,\ \text{for } k = 1,2,\ldots,m_0, \\ & \qquad\qquad\quad\ Y_k^0 \in E^0,\end{aligned}$$

where $g_k^0(x,y) = g(x,y) + \frac{\alpha_k^0}{2}(y - \eta_k^{L,0})(\eta_k^{U,0} - y)$. We solve the problem $(\text{NSIP}_{\alpha\text{BB}}(E^0,\alpha^0))$ to obtain the stationary point x^0 and verify whether x^0 is a stationary point of the problem (NSIP). If x^0 is not a stationary point of the problem (NSIP), we apply the refinement step to formulate a new subproblem until x^0 is the stationary point of the problem (NSIP). Before moving on to discuss the refinement step, y_k^0 denotes the optimal solution to the following lower level optimization problem

$$\max g_k^0(x^0, y) \quad \text{subject to} \quad y \in Y_k^0,$$

and the existence of y_k^0 is unique because of the strict concavity of $g_k^0(\cdot, y)$ on Y_k^0.

Recall the definition that the point $y_k^0 \in Y_k^0$ is called an active point at x^0 if $g_k^0(x^0, y_k^0) = 0$. Furthermore, we say that the subinterval Y_k^0 is an *active interval at* x^0 if $g_k^0(x^0, y_k^0) = 0$ for $y_k^0 \in Y_k^0$. Define

$$E_{\text{act}}(x^0) = \{Y_k^0 \in E^0 | Y_k^0 \text{ is an active interval at } x^0\}, \tag{8}$$

and

$$Y_{\text{act}}^{\alpha\text{BB}}(x^0) = \{y | g_k^0(x^0, y) = 0,\ \text{for } y \in Y_k^0 \text{ and } Y_k^0 \in E_{\text{act}}(x^0)\}. \tag{9}$$

Clearly, $E_{\text{act}}(x^0)$ is a subset of E^0, and $Y_{\text{act}}^{\alpha\text{BB}}(x^0)$ collects the active points in $\bigcup_{k=1}^{m_0} Y_k^0$ at x^0 with respect to the αBB convexification. We can then conclude that the cardinal numbers of $E_{\text{act}}(x^0)$ coincides with that of $Y_{\text{act}}^{\alpha\text{BB}}(x^0)$ because $g_k^0(x^0, y)$ is strictly concave with respect to y on Y_k^0. The purpose of the refinement step is to divide the active intervals in $E_{\text{act}}(x^0)$. For the numerical technique, we introduce the definition of $\varepsilon_\cup$-*active interval* within a given tolerance. The definition of $\varepsilon_\cup$-active interval and the refinement step are modified from the ideas in [8].

Definition 1 Given $\varepsilon_\cup > 0$, Y_k^0 is an $\varepsilon_\cup$-active interval if Y_k^0 is an active interval and

$$\min\{y_k^0 - \eta_k^{L,0}, \eta_k^{U,0} - y_k^0\} > \varepsilon_\cup \cdot (\eta_k^{U,0} - \eta_k^{L,0}) \tag{10}$$

holds. Moreover, the active point $y_k^0 \in Y_k^0$ is an $\varepsilon_\cup$-active point if (10) holds.

Here, we specify the refinement step for the $\varepsilon_\cup$-active intervals. The refinement step checks all the subintervals in E^0. We have two conditions when Y_k^0 is an active interval. One is that Y_k^0 is the $\varepsilon_\cup$-active interval, the refinement step divides Y_k^0 into two subintervals, $Y_{k,1}^0$ and $Y_{k,2}^0$; and also generates a set $\tilde{E}^0 = \{Y_1^0, \dots, Y_{k,1}^0, Y_{k,2}^0, Y_{k+1}^0, \dots, Y_{m_0}^0\}$. We then check the interval Y_{k+1}^0 at the next iteration of the refinement step. Otherwise, Definition 1 shows that, the active interval Y_k^0 is not an $\varepsilon_\cup$-active interval if either the length of $Y_{k,1}^0$ or that of $Y_{k,2}^0$ is too small. Under this condition, we do not divide the active interval since the active point is sufficiently close to one of the endpoints. In our algorithm, this definition will assist us to save computational time when we solve the problem (NSIP). Moreover, due to the property in [7], we have $g_{k,i}^0(x, y) \le g_k^0(x, y)$ on $X \times Y_{k,i}^0$ and $\alpha_{k,i}^0 \le \alpha_k^0$ for $i = 1, 2$. Therefore, the new concave functions $g_{k,i}^0(x, y)$ are closer to the constraint function $g(x, y)$ on Y_k^0. Now, we summarize the steps of the refinement step as the following algorithm.

Algorithm 2 The refinement step for the $\varepsilon_\cup$-active intervals

Step 0 Input $E^0 = \{Y_1^0, Y_2^0, \dots, Y_{m_0}^0\}$, α^0, and $Y_{\text{act}}^{\alpha\text{BB}}(x^0)$, where $m_0 = |E^0|$. Set $k = 1$, $\tilde{E}^0 = E^0$, and $\tilde{\alpha}^0 = \alpha^0$.

Step 1 If $Y_k^0 = [\eta_k^{L,0}, \eta_k^{U,0}]$ is an $\varepsilon_\cup$-active interval and $y_k^0 \in Y_{\text{act}}^{\alpha\text{BB}}(x^0)$ is the respective $\varepsilon_\cup$-active point on Y_k^0, go to the Inner-Step 1 of inner loop; otherwise, go to Step 2.

Inner-Step 1: Define $Y_{k,1}^0 = [\eta_k^{L,0}, y_k^0]$ and $Y_{k,2}^0 = [y_k^0, \eta_k^{U,0}]$. Find the corresponding convexification parameters on $Y_{k,1}^0$ and $Y_{k,2}^0$, respectively, $\alpha_{k,1}^0$ and $\alpha_{k,2}^0$, which satisfy

$$\alpha_{k,1}^0 > \max\left(0, \max_{(x,y)\in X\times Y_{k,1}^0} \frac{\partial^2}{\partial y^2} g(x, y)\right)$$

and

$$\alpha_{k,2}^0 > \max\left(0, \max_{(x,y)\in X\times Y_{k,2}^0} \frac{\partial^2}{\partial y^2} g(x, y)\right).$$

Moreover, define

$$g_{k,1}^0(x, y) = g(x, y) + \frac{\alpha_{k,1}^0}{2}(y - \eta_k^{L,0})(y_k^0 - y)$$

$$g_{k,2}^0(x, y) = g(x, y) + \frac{\alpha_{k,2}^0}{2}(y - y_k^0)(\eta_k^{U,0} - y)$$

as the concave function on $Y_{k,1}^0$ and $Y_{k,2}^0$, respectively.

Inner-Step 2: Denote that $\tilde{E}^0$ is the new collection of subintervals by substituting Y_k^0 as $Y_{k,1}^0$ and $Y_{k,2}^0$, i.e., $\tilde{E}^0 = \{E^0 \setminus \{Y_k^0\}\} \cup \{Y_{k,1}^0\} \cup \{Y_{k,2}^0\}$, and replace the constraint

$$g_k^0(x, y) \leq 0, \quad \text{for } y \in Y_k^0,$$

by the two new constraints

$$g_{k,i}^0(x, y) \leq 0, \quad \text{for all } y \in Y_{k,i}^0,\ i = 1, 2.$$

Furthermore, let $\tilde{\alpha}^0$ be the corresponding convexification parameter on $\tilde{E}^0$, where $\tilde{\alpha}^0$ is the vector by replacing the entry α_k^0 of α^0 by two new entries $\alpha_{k,1}^0$ and $\alpha_{k,2}^0$. Go to Step 2.

Step 2 Terminate the algorithm when $k = m_0$ and output $\tilde{E}^0$, and $\tilde{\alpha}^0$. Otherwise, let $k = k + 1$, $E^0 = \tilde{E}^0$, and $\alpha^0 = \tilde{\alpha}^0$. Go to Step 1.

Referring to Theorem 1, we can conclude that the stationary point x^0 for the problem (NSIP$_{\alpha\text{BB}}(E^0, \alpha^0)$) satisfies

$$\nabla f(x^0) + \sum_{k=1}^{m_0} \lambda_k^0 \nabla_x g_k^0(x^0, y_k^0) = 0, \tag{11}$$

$$\lambda_k^0 \cdot g_k^0(x^0, y_k^0) = 0, \quad k = 1, 2, \ldots, m_0, \tag{12}$$

$$g_k^0(x^0, y) \leq 0, \quad \text{for } y \in Y_k^0 \text{ and } k = 1, 2, \ldots, m_0, \tag{13}$$

where $\lambda_1^0, \ldots, \lambda_{m_0}^0$ are the corresponding Lagrange multipliers. Let $y^0 = (y_1^0, \ldots, y_{m_0}^0)$ and $\lambda^0 = (\lambda_1^0, \ldots, \lambda_{m_0}^0)$. Thus, (x^0, y^0, λ^0) satisfies (11)–(13). Clearly, (2) also holds at (x^0, y^0, λ^0) since $\nabla_x g_k^0(x, y) = \nabla_x g(x, y)$ for all $(x, y) \in X \times Y_k^0$. However, (x^0, y^0, λ^0) may not satisfy (3) because

$$\lambda_k^0 g(x^0, y_k^0) = -\frac{\lambda_k^0 \alpha_k^0}{2}(y_k^0 - \eta_k^{L,0})(\eta_k^{U,0} - y_k^0)$$

for $k = 1, 2, \ldots, m_0$. Therefore, when $\lambda_k^0 > 0$, $\lambda_k^0 g(x^0, y_k^0) = 0$ only happens in the case of either $y_k^0 - \eta_k^{L,0} = 0$ or $\eta_k^{U,0} - y_k^0 = 0$. However, achieving this condition is difficult. Hence, relaxing the complementarity slackness condition for the numerical algorithm is crucial. We make the following definition of the given tolerance ε.

Definition 2 The point $x^0 \in \mathcal{F}$ is called *stationary with ε-complementarity* for the problem (NSIP) if there exists y_k^0 such that (11) and

$$-\lambda_k^0 \cdot \varepsilon \leq \lambda_k^0 g(x^0, y_k^0) \leq 0. \tag{14}$$

are fulfilled simultaneously.

We emphasize that even though $g(x, y) \leq g_k^0(x, y)$ for all $(x, y) \in X \times Y_k^0$, a feasible point x^0 of the problem (NSIP$_{\alpha\text{BB}}(E^0, \alpha^0)$) cannot imply $x^0 \in \mathcal{F}$. For this reason, the feasibility on the unconsidered subintervals of Y may not hold at x^0. That is, let Y_c be an arbitrary unconsidered subinterval of Y. Then we have $Y_c \notin E^0$. The concave function $g_c(x^0, y)$ with (5) on Y_c may not be less than or equal to zero for all y. To develop an algorithm to find the stationary point with ε-complementarity of the problem (NSIP), we check the feasibility on the unconsidered subintervals and refine the $\varepsilon_{\cup}$-active intervals simultaneously to ensure that the feasibility and the ε-complementarity slackness condition (14) hold in the end. Moreover, for a given small number $\gamma > 0$, we demonstrate that the point x^0 is γ-infeasible if there exists an index c, such that $g_c(x^0, y) > \gamma$ for some $y \in Y_c$. The aforementioned index c is called the index of γ-infeasibility. Furthermore, denote c_0 as the *index of maximal γ-infeasibility* at x^0 if

$$\max_{y\in Y_{c_0}} g_{c_0}(x^0, y) \geq \max_{y\in Y_c} g_c(x^0, y),$$

for all indices of γ-infeasibility c. Therefore, we conclude our idea in the following algorithm.

Algorithm 3 The relaxed cutting plane method with convexification

Step 0: Split Y into N disjoint subintervals $Y_1, Y_2, \dots, Y_N$ which $Y = \bigcup_{j=1}^{N} Y_j$ and $|Y_j| = \frac{1}{N}$. Let the set $E := \{Y_1, Y_2, \dots, Y_N\}$ be the correction of the subintervals. Determine the convexification parameter α with (6) on E, and define the approximate concave functions g_j on Y_j with (5). Let $\bar{\alpha} := \max_i\{\alpha_i\}$.

Step 1: Set $\nu = 1$. Choose arbitrary subintervals, $Y_1^1, Y_2^1, \dots, Y_{m_1}^1$, in E with $m_1 < N$. Let $E^1 = \{Y_1^1, Y_2^1, \dots, Y_{m_1}^1\}$ and $\alpha^1 = \{\alpha_1^1, \dots, \alpha_{m_1}^1\}$ be the corresponding convexification parameter on E^1. Give adaptive small real numbers $\varepsilon, \gamma > 0$, and let $\varepsilon_{\cup} := 2\varepsilon \cdot \min\{1, 1/\bar{\alpha}\}$. Let $E_1^c = E \setminus E^1$ be the collection of remain subintervals.

Step 2: Solve the problem $(\mathrm{NSIP}_{\alpha\mathrm{BB}}(E^\nu, \alpha^\nu))$ using Algorithm 1 to obtain the stationary point x^ν and the corresponding active point set $Y_{\mathrm{act}}^{\alpha\mathrm{BB}}(x^\nu)$.

Step 3: Evaluate the values of

$$\max_{y\in Y_j} g_j(x^\nu, y),$$

for all $Y_j \in E_\nu^c$. Denote c_ν as the index of maximal γ-infeasibility, and go to Step 4 if c_ν exists. If c_ν does not exist, and for every k, the inequalities $-\lambda_k^\nu \cdot \varepsilon \leq \lambda_k^\nu g(x^\nu, y_k^\nu) \leq 0$ hold, and then STOP and output x^ν as an approximate stationary point of the problem (NSIP). Otherwise, go to Step 5.

Step 4: If there exists at least one $\varepsilon_{\cup}$-active point $y_k^\nu \in Y_{\mathrm{act}}^{\alpha\mathrm{BB}}(x^\nu)$, apply Algorithm 2 to E^ν, α^ν and $Y_{\mathrm{act}}^{\alpha\mathrm{BB}}(x^\nu)$. Define $E^{\nu+1} = \tilde{E}^\nu \cup \{Y_{c_\nu}\}$, $\alpha^{\nu+1} = (\tilde{\alpha}^\nu, \alpha_{c_\nu})$, and $E_{\nu+1}^c = E_\nu^c \setminus \{Y_{c_\nu}\}$, where $\tilde{E}^\nu$ and $\tilde{\alpha}^\nu$ are generated by Algorithm 2. Otherwise, in the case of no $\varepsilon_{\cup}$-active point, define $E^{\nu+1} = E^\nu \cup \{Y_{c_\nu}\}$, $\alpha^{\nu+1} = (\alpha^\nu, \alpha_{c_\nu})$, and $E_{\nu+1}^c = E_\nu^c \setminus \{Y_{c_\nu}\}$. Set $m_{\nu+1} = |E^{\nu+1}|$ and $\nu = \nu + 1$ and go to Step 2.

Step 5: Apply Algorithm 2 to E^ν, α^ν, and $Y_{\mathrm{act}}^{\alpha\mathrm{BB}}(x^\nu)$. Define $E^{\nu+1} = \tilde{E}^\nu$, $\alpha^{\nu+1} = \tilde{\alpha}^\nu$, and $E_{\nu+1}^c = E_\nu^c$. Set $m_{\nu+1} = |E^{\nu+1}|$. Let $\nu = \nu + 1$, and go to Step 2.

Note here, $E_{\mathrm{act}}(x^{\nu})$ and $Y_{\mathrm{act}}^{\alpha\mathrm{BB}}(x^{\nu})$ are defined in a similar way by substituting the index 0 for ν in (8) and (9). This shows that $E_{\nu}^{c} \subset E$ for every ν. However, E^{ν} is not a subset of E when Algorithm 2 is applied once in the previous iteration. More remarks on Algorithm 3 are listed as follows.

Remark 1

(i) Notice that in Step 2 of Algorithm 3, we check the γ-feasibility on the subintervals in E_{ν}^{c} and also determine whether x^{ν} is the stationary point with ε-complementarity of the problem (NSIP). If there exists a γ-infeasible subinterval $Y_{c_{\nu}}$, we add a new concave constraint $g_{c_{\nu}}(x, y)$ to the problem $(\mathrm{NSIP}_{\alpha\mathrm{BB}}(E^{\nu}, \alpha^{\nu}))$ in Step 4. Consequently, Algorithm 3 outputs a stationary point with ε-complementarity and γ-feasibility.

(ii) The γ-feasibility check on E_{ν}^{c} is an easy task since the functions $g_j(\cdot, y)$ are concave on Y_j for $Y_j \in E_{\nu}^{c}$. Therefore, there are various numerical methods that can find the value of $\max g_j(x^{\nu}, y)$ for $y \in Y_j$ quickly.

(iii) In Step 3, we refine the $\varepsilon_{\cup}$-active intervals and substitute two new approximate concave constraints for the original concave constraint on $\varepsilon_{\cup}$-active interval. The constraint functions on the rest of the subintervals in E^{ν} are maintained. Certainly, the $\varepsilon_{\cup}$-active interval may not exist. If both $\varepsilon_{\cup}$-active interval and c_{ν} do not exist, according to Lemma 2 described in next section, Algorithm 3 will satisfy the stopping criteria.

(iv) To avoid confusion, we rearrange the indices of the subintervals in $E^{\nu+1}$ before go to Step 2. For the sake of convenience, let Y_1^{ν} be the unique $\varepsilon_{\cup}$-active interval and the index of maximal γ-infeasibility c_{ν} exist at ν-th iterate. Then $E^{\nu+1} = \tilde{E}^{\nu} \cup \{Y_{c_{\nu}}\}$ is defined in Step 4, where $\tilde{E}^{\nu} = \{Y_{1,1}^{\nu}, Y_{1,2}^{\nu}, Y_2^{\nu}, \ldots, Y_{m_{\nu}}^{\nu}\}$. Therefore, $E^{\nu+1} = \{Y_{1,1}^{\nu}, Y_{1,2}^{\nu}, Y_2^{\nu}, \ldots, Y_{m_{\nu}}^{\nu}, Y_{c_{\nu}}\}$. We then rearrange the indices of the subintervals in $E^{\nu+1}$ as

$$E^{\nu+1} = \{Y_1^{\nu+1}, Y_2^{\nu+1}, \ldots, Y_{m_{\nu+1}}^{\nu+1}\},$$

by letting $Y_1^{\nu+1} = Y_{1,1}^{\nu}$, $Y_2^{\nu+1} = Y_{1,2}^{\nu}, \ldots, Y_{m_{\nu+1}}^{\nu+1} = Y_{c_{\nu}}$. It is similar to the rearrangement for the indices of $\alpha^{\nu+1}$. We can find that $Y_k^{\nu+1}$ may not equal Y_k^{ν} after the rearrangement.

Observe that we have two stopping criteria in Algorithm 3. Generally, we anticipate that the feasibility on E_{ν}^{c} will hold after a few iterations. In other words, the number of the remaining subintervals will not be zero when Algorithm 3 terminates. We verify this result by applying some numerical examples in Sect. 5.

4 Convergence properties for algorithm

We demonstrate the finite termination and the error bound properties of Algorithm 3 in this section. Let L_{ν} be defined as

$$L_{\nu} := \max\left\{ \|Y_k^{\nu}\| \;\middle|\; Y_k^{\nu} \text{ is the } \varepsilon_{\cup}\text{-active interval in } E_{\mathrm{act}}(x^{\nu}) \right\}, \tag{15}$$

where $\|Y_k^\nu\| := \eta_k^{U,\nu} - \eta_k^{L,\nu}$ is the length of Y_k^ν. In the following lemma, we discuss the relationship between $\|x^{\nu+1} - x^\nu\|$ and the largest length of $\varepsilon_\cup$-active interval, L_ν. This lemma will help us analyze the convergence property of Algorithm 3. Apply the fact that there is a point $\bar{x}^\nu$ in the open segment connecting x^ν and $x^{\nu+1}$ such that

$$g(x^{\nu+1}, y) - g(x^\nu, y) = \nabla_x g(\bar{x}^\nu, y)^\top (x^{\nu+1} - x^\nu), \tag{16}$$

for all $y \in Y$. Additionally, we make the following assumption.

Assumption 1 There exists an index $\bar{\nu}$ such that for each $\varepsilon_\cup$-active point $y_k^\nu \in Y_{\text{act}}^{\alpha\text{BB}}(x^\nu)$, there is a small positive number δ that satisfies the following assumptions for all $\nu \geq \bar{\nu}$:

(a) $\|\nabla_x g(\bar{x}^\nu, y)\| \geq \delta$ for all y, where $\bar{x}^\nu$ is the point in (16), and
(b) $|\nabla_x g(\bar{x}^\nu, y_k^\nu)^\top (x^{\nu+1} - x^\nu)| \geq \delta \|x^{\nu+1} - x^\nu\|$.

We note that Assumption 1 can be checked in each iteration of the algorithm. Clearly, Assumption 1(a) can be achieved easily. Assumption 1(b) states that the angle between $\nabla_x g(\bar{x}^\nu, y_k^\nu)$ and $x^{\nu+1} - x^\nu$ are not too close to being orthogonal. It is known that $\nabla_x g(\bar{x}^\nu, y_k^\nu) \nabla_x g(\bar{x}^\nu, y_k^\nu)^\top$ is a positive semi-definite matrix. Therefore, if the smallest eigenvalue of $\nabla_x g(\bar{x}^\nu, y_k^\nu) \nabla_x g(\bar{x}^\nu, y_k^\nu)^\top$ is greater than δ^2, then Assumption 1(b) naturally holds. Next, we present the following lemma, which is important in our convergence proof.

Lemma 1 *Assume that, for each $\nu \geq \bar{\nu}$, there is at least one $\varepsilon_\cup$-active interval. At least one of the subintervals of $\varepsilon_\cup$-active interval is an active interval in the next iteration. Suppose that Assumption 1 holds at $\varepsilon_\cup$-active points. Then, for each $\nu \geq \bar{\nu}$, there exists a bounded real number b_ν such that*

$$\|x^{\nu+1} - x^\nu\| \leq b_\nu L_\nu^2.$$

Proof Let Y_k^ν be the $\varepsilon_\cup$-active interval that satisfies the assumption in this lemma and $y_k^\nu \in Y_{\text{act}}^{\alpha\text{BB}}(x^\nu)$ be the $\varepsilon_\cup$-active point. Hence, we have $g_k^\nu(x^\nu, y_k^\nu) = 0$ and

$$g(x^\nu, y_k^\nu) = -\frac{\alpha_k^\nu}{2}(y_k^\nu - \eta_k^{L,\nu})(\eta_k^{U,\nu} - y_k^\nu) < 0.$$

As Y_k^ν is the $\varepsilon_\cup$-active interval, Y_k^ν will be divided into $Y_{k,1}^\nu = [\eta_k^{L,\nu}, y_k^\nu]$ and $Y_{k,2}^\nu = [y_k^\nu, \eta_k^{U,\nu}]$ in the next iteration. Without loss of generality, we assume that the subinterval $Y_{k,1}^\nu = [\eta_k^{L,\nu}, y_k^\nu]$ is the active interval in the next iteration, $\nu + 1$. Therefore, $g_{k,1}^\nu(x^{\nu+1}, y_{k,1}^{\nu+1}) = 0$, where $y_{k,1}^{\nu+1}$ is the maximizer in the following optimization problem:

$$\max g_{k,1}^\nu(x^{\nu+1}, y) \quad \text{subject to} \quad y \in Y_{k,1}^\nu.$$

The function $g_{k,1}^\nu(x, y)$ is defined by

$$g_{k,1}^\nu(x, y) = g(x, y) + \frac{\alpha_{k,1}^\nu}{2}(y_k^\nu - y)(y - \eta_k^{L,\nu}). \tag{17}$$

Clearly, according to (17),

$$\begin{aligned} g(x^{\nu+1}, y_k^\nu) &= g_{k,1}^\nu(x^{\nu+1}, y_k^\nu) \\ &\le g_{k,1}^\nu(x^{\nu+1}, y_{k,1}^{\nu+1}) = 0. \end{aligned}$$

Thus, we have

$$\begin{aligned} &g(x^{\nu+1}, y_k^\nu) - g(x^\nu, y_k^\nu) \\ &\quad < -g(x^\nu, y_k^\nu) = \frac{\alpha_k^\nu}{2}(y_k^\nu - \eta_k^{L,\nu})(\eta_k^{U,\nu} - y_k^\nu) \\ &\quad \le \frac{\bar{\alpha}}{8}(\eta_k^{U,\nu} - \eta_k^{L,\nu})^2, \end{aligned} \tag{18}$$

since $\alpha_k^\nu \le \bar{\alpha}$ for all k. We know that both $g_{k,1}^\nu(x^{\nu+1}, y_{k,1}^{\nu+1})$ and $\frac{\partial}{\partial y} g_{k,1}^\nu(x^{\nu+1}, y_{k,1}^{\nu+1})$ are equal to zero; therefore,

$$\begin{aligned} &g(x^{\nu+1}, y_k^\nu) - g(x^\nu, y_k^\nu) \\ &\quad \ge g_{k,1}^\nu(x^{\nu+1}, y_k^\nu) - g_{k,1}^\nu(x^{\nu+1}, y_{k,1}^{\nu+1}) \\ &\quad \ge \frac{\partial}{\partial y} g_{k,1}^\nu(x^{\nu+1}, y_k^\nu)(y_k^\nu - y_{k,1}^{\nu+1}) \quad \text{(since } g_{k,1}^\nu(x, y) \text{ is concave on } Y_{k,1}^\nu\text{)} \\ &\quad = \frac{\partial}{\partial y} g_{k,1}^\nu(x^{\nu+1}, y_k^\nu)(y_k^\nu - y_{k,1}^{\nu+1}) - \frac{\partial}{\partial y} g_{k,1}^\nu(x^{\nu+1}, y_{k,1}^{\nu+1})(y_k^\nu - y_{k,1}^{\nu+1}) \\ &\quad = \frac{\partial^2}{\partial y^2} g_{k,1}^\nu(x^{\nu+1}, \bar{y}_{k,1}^{\nu+1})(y_k^\nu - y_{k,1}^{\nu+1})^2, \end{aligned} \tag{19}$$

where $y_{k,1}^{\nu+1} \le \bar{y}_{k,1}^{\nu+1} \le y_k^\nu$. However, the value of $\frac{\partial^2}{\partial y^2} g_{k,1}^\nu(x^{\nu+1}, \bar{y}_{k,1}^{\nu+1})$ is bounded on $X \times Y_{k,1}^\nu$ for all ν. Hence, we have a positive number b^* such that $\frac{\partial^2}{\partial y^2} g_{k,1}^\nu(x, y) \ge -b^*$ for all ν. Equation (19) implies that $g(x^{\nu+1}, y_k^\nu) - g(x^\nu, y_k^\nu) > -b^*(\eta_k^{U,\nu} - \eta_k^{L,\nu})^2$ since $\eta_k^{U,\nu} - \eta_k^{L,\nu} > y_k^\nu - y_{k,1}^{\nu+1}$. Combining (18) and (19), and then applying the mean value theorem, we obtain

$$-b^*(\eta_k^{U,\nu} - \eta_k^{L,\nu})^2 \le \nabla_x g(\bar{x}^\nu, y_k^\nu)^\top (x^{\nu+1} - x^\nu) \le \frac{\bar{\alpha}}{8}(\eta_k^{U,\nu} - \eta_k^{L,\nu})^2, \tag{20}$$

where $\bar{x}^\nu$ is the point on the line segment between x^ν and $x^{\nu+1}$. Consequently, we have the following result from both (20) and Assumption 1(b)

$$\begin{aligned} \delta^2 \|x^{\nu+1} - x^\nu\|^2 &< (x^{\nu+1} - x^\nu)^\top \nabla_x g(\bar{x}^\nu, y_k^\nu) \nabla_x g(\bar{x}^\nu, y_k^\nu)^\top (x^{\nu+1} - x^\nu) \\ &\le \max\left(\frac{\bar{\alpha}^2}{64}, (b^*)^2\right)(\eta_k^{U,\nu} - \eta_k^{L,\nu})^4. \end{aligned}$$

As the inequality holds for the $\varepsilon_{\cup}$-active subinterval, this implies

$$\delta^2\|x^{\nu+1}-x^{\nu}\|^2 \leq \max\left(\frac{\bar{\alpha}^2}{64}, (b^*)^2\right)L_{\nu}^4.$$

Hence, $\|x^{\nu+1}-x^{\nu}\| \leq b_{\nu}L_{\nu}^2$ when setting $b_{\nu}=\max(\frac{\bar{\alpha}}{8}, b^*)/\delta$. Here, b_{ν} is bounded by a constant for all $\nu \geq \bar{\nu}$. □

Lemma 1 shows that the value of $\|d^{\nu}\| := \|x^{\nu+1}-x^{\nu}\|$ is concerned with the length of $\varepsilon_{\cup}$-active intervals. We present a lemma modified directly from [8].

Lemma 2 *If no γ-infeasible subinterval and no $\varepsilon_{\cup}$-active interval exist, Algorithm 3 terminates.*

Proof According to the result of Lemma 4.5 in [8], if there is no $\varepsilon_{\cup}$-active interval, Algorithm 3 generates a stationary point with ε-complementarity. Combined with the fact of having no γ-infeasible subinterval, Algorithm 3 terminates because of the satisfaction of these two stopping criteria. □

The next theorem presents that the maximum length of $\varepsilon_{\cup}$-active interval tends to 0.

Lemma 3 *Let $\{L_{\nu}\}$ be the sequence defined by* (15), *and suppose that Algorithm 3 does not terminate. Then,*

$$\lim_{\nu\to\infty} L_{\nu} = 0.$$

Proof In Algorithm 3, we divide interval Y into N subintervals $Y_1,\dots,Y_N$ in Step 0, and choose m_1 subintervals to start our algorithm. Let I be the set of the index of iterations for Algorithm 3, and let $I_{\gamma}=\{\nu^1,\nu^2,\dots,\nu^l\}\subset I$ be the collection for the index of the iteration with γ-infeasibility, where $l\in\mathbb{N}$. This indicates that, for $i=1,2,\dots,l$, c_{ν^i} exists at the ν^i-th iteration. Since the number of unconsidered subintervals is bounded by $N-m_1$, we have $0\leq l\leq N-m_1$. Moreover, from Lemma 2 and for each $\nu\in I\setminus I_{\gamma}$, at least two new subintervals, $Y_{k,1}^{\nu}$ and $Y_{k,2}^{\nu}$, are generated from Y_k^{ν} for some k. Therefore, if Y_k^{ν} is an $\varepsilon_{\cup}$-active interval, then the result

$$\max\{\|Y_{k,1}^{\nu}\|, \|Y_{k,2}^{\nu}\|\} \leq (1-\varepsilon_{\cup})\|Y_k^{\nu}\| \leq (1-\varepsilon_{\cup})\cdot\frac{1}{N}$$

follows from the definition of $\varepsilon_{\cup}$-active interval. Note that all the subintervals in E^{ν} are at most divided once for each ν, and the length of $\varepsilon_{\cup}$-active interval is monotonically decreasing. Hence, given any integer $p\in\mathbb{N}$, there exists a corresponding sufficiently large iterate $\tilde{\nu}\in I\setminus I_{\gamma}$ such that L_{ν} is bounded by $(1-\varepsilon_{\cup})^p\cdot\frac{1}{N}$ for $\nu>\tilde{\nu}$ because Algorithm 3 does not terminate. Thus, we obtain $\lim_{\nu\to\infty}L_{\nu}=0$ by letting $p\to\infty$. □

The result in Lemma 3 directly implies that $\|d^{\nu}\|$ approaches 0 as ν goes to infinity. The next theorem shows that Algorithm 3 converges to a stationary point of the problem (NSIP) with ε-complementarity and γ-feasibility after finite iterations.

Theorem 2 *Suppose that for each iteration* ν, *the corresponding Lagrange multipliers are uniformly bounded by a constant* M. *Moreover, the assumptions in Lemma 1 hold. Let* $\{x^{\nu}\}$ *be a sequence generated by Algorithm 3. Algorithm 3 will then generate a stationary point of the problem (NSIP) with* ε*-complementarity and* γ*-feasibility after finite iterations.*

Proof The KKT condition for the problem $(\mathrm{NSIP}_{\alpha\mathrm{BB}}(E^{\nu_s}, \alpha^{\nu_s}))$ can be reformulated as

$$\nabla f(x^{\nu_s}) + \int_Y \nabla_x g(x^{\nu_s}, y)\, d\mu^{\nu_s} = 0, \tag{21}$$

$$\sum_{k=1}^{m_{\nu_s}} \int_{Y_k^{\nu_s}} g_k^{\nu_s}(x^{\nu_s}, y)\, d\mu^{\nu_s} = 0, \tag{22}$$

$$g_k^{\nu}(x^{\nu}, y) \le 0, \quad \text{for all } y \in Y_k^{\nu},\ k = 1, 2, \ldots, m_{\nu}, \tag{23}$$

where $\mu^{\nu_s}(y)$ is a discrete measure defined on Y by

$$\mu^{\nu_s}(y) = \begin{cases} \lambda_k^{\nu_s} & \text{if } y = y_k^{\nu_s}; \\ 0 & \text{elsewhere.} \end{cases}$$

As X and Y are compact sets, there exists a subsequence $\{x^{\nu_s}\}$ converging to the accumulation point x^*, and $\{\mu^{\nu_s}(y)\}$ converges to $\mu^*(y)$ weakly, as $s \to \infty$. Moreover, by continuity of $\nabla f(x)$ and $\nabla_x g(x, y)$, (21) is satisfied for all s. These facts imply that $\|\nabla f(x^*) + \int_Y \nabla_x g(x^*, y)\, d\mu^*\| = 0$ directly.

Next, we show that x^* is a stationary point with ε-complementarity. We claim that for $k = 1, 2, \ldots, m_{\nu_s}$, $-\lambda_k^{\nu_s} \cdot \varepsilon \le \lambda_k^{\nu_s} g(x^{\nu_s}, y_k^{\nu_s}) \le 0$ when s is sufficiently large. The case for $\lambda_k^{\nu_s} = 0$ is trivial; therefore, we focus our attention on $\lambda_k^{\nu_s} > 0$. Even though the number of subintervals m_{ν_s} increases monotonically, in view of Carathéodory's theorem, the number of positive Lagrange multipliers is at most $n + 1$, where n is the dimension of x. In other words, without loss of generality, let the Lagrange multipliers $\lambda_1^{\nu_s}, \lambda_2^{\nu_s}, \ldots, \lambda_{m'}^{\nu_s}$ be positive, where $m' \le n + 1$. Hence, the corresponding subintervals $Y_k^{\nu_s}$, with respect to $\lambda_k^{\nu_s}$ for $k = 1, 2, \ldots, m'$, are active intervals. However, we have two cases. First, if the subinterval is an $\varepsilon_{\cup}$-active interval, then

$$\frac{1}{2}\|\alpha_k^{\nu_s}(y_k^{\nu_s} - \eta_k^{L,\nu_s})(\eta_k^{U,\nu_s} - y_k^{\nu_s})\| \le \frac{1}{8}\alpha_k^{\nu_s}\|\eta_k^{U,\nu_s} - \eta_k^{L,\nu_s}\|^2 \le \frac{1}{8}\bar{\alpha} L_{\nu_s}^2,$$

where $\bar{\alpha}$ is a constant defined in Step 0 of Algorithm 3. Since $L_{\nu_s} \to 0$, we can find an integer s_2 and the inequality $\frac{1}{8}\bar{\alpha} L_{\nu_s}^2 \le \varepsilon$ is valid for $s > s_2$. Otherwise, if $Y_k^{\nu_s}$ is not an $\varepsilon_{\cup}$-active interval. From the definition of $\varepsilon_{\cup} = 2\varepsilon \cdot \min\{1, 1/\bar{\alpha}\}$, we obtain

$$\frac{1}{2}\|\alpha_k^{\nu_s}(y_k^{\nu_s} - \eta_k^{L,\nu_s})(\eta_k^{U,\nu_s} - y_k^{\nu_s})\| \le \varepsilon_{\cup}\bar{\alpha}\|\eta_k^{U,\nu_s} - \eta_k^{L,\nu_s}\|^2 \le \frac{\varepsilon}{N^2} < \varepsilon,$$

where $1/N$ is the length of the initial discretized subinterval in Algorithm 3. Moreover, $g(x^{\nu_s}, y_k^{\nu_s}) = -(\alpha_k^{\nu_s}/2)(y_k^{\nu_s} - \eta_k^{L,\nu_s})(\eta_k^{U,\nu_s} - y_k^{\nu_s})$. Consequently, for all $s > s_2$, we can obtain $-\lambda_k^{\nu_s} \cdot \varepsilon \le \lambda_k^{\nu_s} g(x^{\nu_s}, y_k^{\nu_s}) \le 0$.

Finally, we present that $g(x^*, y) \le 0$ for $y \in Y$ is true. Suppose that the collection of the remaining subinterval $E_{\nu_s}^c$ is nonempty for $s = 1, 2, \ldots$. Otherwise, the proof is achieved if $E_{\nu_s}^c$ is empty for some s. Let Y_r be a subinterval in $E_{\nu_s}^c$ for all s. We denote that

$$y_r^* \in \arg\max_{y \in Y_r} g_r(x^*, y) \quad \text{and} \quad y_r^{\nu_s} \in \arg\max_{y \in Y_r} g_r(x^{\nu_s}, y),$$

where $g_r(x, y)$ is the approximate concave function with (5) on Y_r. Assume that, by contradiction, $g_r(x^*, y_r^*) \ge \kappa$, for some $\kappa > 0$. Since g_r is a continuous differentiable function with respect to variable x, there is an index $s_\kappa \in \mathbb{N}$ such that $g_r(x^{\nu_s}, y_r^*) \ge \kappa/2$, for all $s \ge s_\kappa$. This concludes that

$$g_r(x^{\nu_s}, y_r^{\nu_s}) \ge g_r(x^{\nu_s}, y_r^*) \ge \kappa/2, \quad \text{for all } s \ge s_\kappa.$$

Since $Y_r \in E_{\nu_s}^c$ for all $s \in \mathbb{N}$, there must exist an index of maximal γ-infeasibility c_{ν_s} and the corresponding subinterval $Y_{c_{\nu_s}} \in E_{\nu_s}^c$ such that

$$g_{c_{\nu_s}}(x^{\nu_s}, y_{c_{\nu_s}}) > g_r(x^{\nu_s}, y_r^{\nu_s}) \ge \kappa/2,$$

where $y_{c_{\nu_s}} \in \arg\max_{y \in Y_{c_{\nu_s}}} g_{c_{\nu_s}}(x^{\nu_s}, y)$ and $g_{c_{\nu_s}}(x, y)$ is the concave function with (5) on $Y_{c_{\nu_s}}$. Hence, we conclude that $g_{c_{\nu_s}}(x, y)$ is the new member of the constraints in the problem ($\text{NSIP}_{\alpha\text{BB}}(E^{\nu_s+1}, \alpha^{\nu_s+1})$). This implies that

$$g_{c_{\nu_s}}(x^{\nu_s+1}, y) \le 0 \quad \text{for } y \in Y_{c_{\nu_s}}$$

where x^{ν_s+1} is the stationary point of the problem ($\text{NSIP}_{\alpha\text{BB}}(E^{\nu_s+1}, \alpha^{\nu_s+1})$). Therefore, we have

$$g_{c_{\nu_s}}(x^{\nu_s}, y_{c_{\nu_s}}) - g_{c_{\nu_s}}(x^{\nu_s+1}, y_{c_{\nu_s}}) \ge \kappa/2.$$

Thus, $\|\nabla_x g_{c_{\nu_s}}(\bar{x}^{\nu_s}, y_{c_{\nu_s}})\| \|d^{\nu_s}\| \ge \kappa/2$ for some point $\bar{x}^{\nu_s}$ lies between the segment generated by x^{ν_s} and x^{ν_s+1}. From Lemma 1, we know that $\|d^{\nu_s}\| \to 0$. This contradicts the inequality. Consequently, we obtain

$$\lim_{s \to \infty} g_r(x^{\nu_s}, y) = g_r(x^*, y) \le 0,$$

for all $y \in Y_r$ Therefore, given any $\gamma > 0$, there is $s_3 \in \mathbb{N}$ such that $g_r(x^{\nu_s}, y) \le \gamma$ for $y \in Y_r$ and for all $Y_r \in E_{\nu_s}^c$ when $s > s_3$. Let $\bar{s} = \max\{s_1, s_2, s_3\}$. Then $x^{\nu_{\bar{s}}}$ is a stationary point with ε-complementarity and γ-feasibility. Therefore, our proof is complete. □

We choose ε_{U} to be a small positive number for the numerical implementation in Algorithm 3. In the end, we make our numerical solution satisfy the ε-complementarity slackness condition since the complementarity slackness condition

(3) is difficult to satisfy. In Theorem 2, $-\lambda_k^\nu \cdot \varepsilon \le \lambda_k^\nu g(x^\nu, y_k^\nu) \le 0$ holds if the active point y_k^ν is not an $\varepsilon_\cup$-active point. Hence, the existence of the $\varepsilon_\cup$-active interval helps us avoid dividing the unnecessary active intervals. In conclusion, the $\varepsilon_\cup$-active interval plays an important role in the numerical experiment.

We prove that Algorithm 3 terminates in a finite number of iterations for any given tolerances $\varepsilon > 0$ and $\gamma > 0$. Let the point $x^{\nu^*(\varepsilon,\gamma)}$ be generated by Algorithm 3 when the algorithm terminates with ε-complementarity and γ-feasibility. In the following theorem, we show that any accumulation point of $x^{\nu^*(\varepsilon,\gamma)}$ as $\varepsilon, \gamma \to 0$ is the stationary point of the problem (NSIP).

Theorem 3 *Let $x^{\nu^*(\varepsilon,\gamma)}$ be the stationary point with ε-complementarity and γ-feasibility of the problem* (*NSIP*) *generated by Algorithm* 3. *Then,*

(a) *For a fixed tolerance $\varepsilon > 0$, every accumulation point $x^{\nu^*(\varepsilon)}$ of $x^{\nu^*(\varepsilon,\gamma)}$ as $\gamma \to 0$ is a stationary point with ε-complementarity of the problem* (*NSIP*).
(b) *Every accumulation point x^{ν^*} of $x^{\nu^*(\varepsilon)}$ as $\varepsilon \to 0$ is a stationary point of the problem* (*NSIP*).

Proof We first show (a). Since X is a compact subset of $\mathbb{R}^n$, there exists a point $x^{\nu^*(\varepsilon,\gamma)} \in X$ such that $x^{\nu^*(\varepsilon,\gamma)} \to x^{\nu^*(\varepsilon)}$ as $\gamma \to 0$. Furthermore, according to the proof in Theorem 2, the result in (a) follows immediately. Next, we show (b). Following the result in (a), we obtain $x^{\nu^*(\varepsilon)} \in \mathcal{F}$. As $\mathcal{F}$ is a compact subset of $\mathbb{R}^n$, there exists an accumulation point $x^{\nu^*} \in \mathcal{F}$ of $x^{\nu^*(\varepsilon)}$ as $\varepsilon \to 0$. Let $y_1^{\nu^*(\varepsilon)}, y_2^{\nu^*(\varepsilon)}, \dots, y_{m''}^{\nu^*(\varepsilon)}$, $m'' \le n+1$, be the points satisfying the KKT conditions (11)–(13) with positive Lagrange multipliers $\lambda_1^{\nu^*(\varepsilon)}, \lambda_2^{\nu^*(\varepsilon)}, \dots, \lambda_{m''}^{\nu^*(\varepsilon)}$ at $x^{\nu^*(\varepsilon)}$. Thus, we have

$$\nabla f(x^{\nu^*(\varepsilon)}) + \sum_{k=1}^{m''} \lambda_k^{\nu^*(\varepsilon)} \nabla_x g(x^{\nu^*(\varepsilon)}, y_k^{\nu^*(\varepsilon)}) = 0, \tag{24}$$

for all $\varepsilon > 0$. Hence, (24) converges to (11) when $\varepsilon \to 0$. Moreover, define

$$\delta^{\nu^*(\varepsilon)} := \max_{1 \le k \le m''} \{-g(x^{\nu^*(\varepsilon)}, y_k^{\nu^*(\varepsilon)})\}.$$

Therefore, $0 \le \delta^{\nu^*(\varepsilon)} \le \varepsilon$ is obtained according to the stopping criterion of Algorithm 3 and $\lim_{\varepsilon \to 0} \delta^{\nu^*(\varepsilon)} = 0$. Let $\varepsilon \to 0$. Due to the continuity of function g, we obtain the result (b).

□

Remark 2 Since $\varepsilon_\cup \to 0$ as $\varepsilon \to 0$, Lemma 3 may not hold. Hence, this fact leads to the failure of the convergence proof in Theorem 2. We can not show the result of (b) in Theorem 3 by letting $\varepsilon \to 0$ and applying Theorem 2.

5 Numerical experience

In this section, we apply Algorithm 3 to some examples. We implement Algorithm 3 in Matlab 7.0 and use the routine *fmincon* in Matlab, with default tolerances as the black box NLP solver in Step 2 of Algorithm 1. In particular, the tolerance for constraint violation is $1e-6$. All the experiments were run on a personal computer with AMD(R) Phenom II(R) CPU 3.0 GHz and 2 GB RAM.

Example 1 Among all control design techniques, the *Proportional-Integral-Derivative* (PID) controller design problem is the most widely used. The PID controller design problem is formulated as an SIP problem in [9]. More recently, the authors in [3] applied the semi-infinite programming technique to solve the optimal worst-case H_2 design of the PID controller for uncertain nonminimum-phase plants. The first example was taken from [9] and was implemented in [13, 25]. The PID controller design problem minimizes the cost function given by

$$f(x) = \frac{x_2(122 + 17x_1 + 6x_3 - 5x_2 + x_1x_3) + 180x_3 - 36x_1 + 1224}{x_2(408 + 56x_1 - 50x_2 + 60x_3 + 10x_1x_3 - 2x_1^2)},$$

and subject to the constraint

$$g(x, y) = \Im(T(x, y)) - 3.33(\Re(T(x, y)))^2 + 1 \le 0, \quad \text{for all } y \in Y,$$

where

$$x = (x_1, x_2, x_3)^\top \in X, \quad \text{for } X = [0, 100] \times [0.1, 100] \times [0, 100] \subset \mathbb{R}^3; \quad \text{and}$$

$$T(x, y) = 1 + H(x, yj)G(yj); \qquad H(x, s) = x_1 + \frac{x_2}{s} + x_3 s;$$

$$G(s) = \frac{1}{(s + 3)(s^2 + 2s + 2)},$$

with $Y = [10^{-6}, 30]$, $j = \sqrt{-1}$. $\Im(a)$ and $\Re(a)$ represent the imaginary and real part of complex number a, respectively. To start our algorithm, we have to determine the value of α first. Generally, the estimation of α is not easy. Here, we evaluate the explicit form of the second derivative of the constraint function and then set $\alpha = 3.4763$. The calculation of α costs time; thus we let $\alpha_k^\nu = 3.4763$ be a constant for all k and ν. We initialize our algorithm with the tolerances $\varepsilon = 1e - 4$ and $\gamma = 1e - 6$. For comparison, we implement different values of m_1, where m_1 denotes the number of chosen subintervals in the initial step (Step 0) of Algorithm 3. Create a modified Algorithm 3, denoted Algorithm 3.a, by setting $m_1 = N$ in Step 0 of the Algorithm 3. That is, we select all the discretized subintervals in the initial step and need not to check the γ-feasibility in each iteration. We also use Algorithm 1 to solve the subproblem generated by Algorithm 3.a. Moreover, the subroutine for solving the SIP problem in Matlab called *fseminf* is also implemented for comparison with our results. The following table lists the numerical results for Example 1.

Table 1 shows **opt-val** as the computed optimal value, **cpu-t** as the computation time, **#iter** as the iteration number taken in two algorithms, and **#re-int** as the number of remaining subintervals in Algorithm 3. Moreover, max g denotes the optimal value of the constraint function g at the terminated solution over Y. In Table 1, the number of infeasible subintervals is few, and we add five infeasible subintervals to both cases. Since the number of constraints in Algorithm 3 is less than that in Algorithm 3.a,

Table 1 Numerical results for Example 1

Algorithm	N	m_1	opt-val	cpu-t	#iter	#re-int	max g
3.a	100	100	0.174759	8.0525	18	–	−4.0823e-8
3	100	10	0.174711	4.2261	29	85	−3.1748e-6
3.a	300	300	0.174640	18.4632	11	–	−1.1776e-5
3	300	30	0.174632	6.8447	21	265	−4.9648e-6
fseminf	–	–	0.174627	106.8047	–	–	1.8152e-9

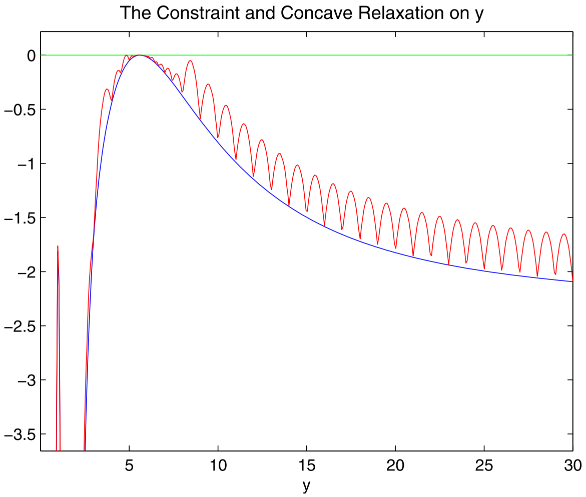

Fig. 1 Constraint and concave approximation at the terminated step in Example 1

Algorithm 3 at least saves half of the time in computation. Although the number of iterations for Algorithm 3 is greater than that in Algorithm 3.a, the total CPU time spent is less than that in Algorithm 3.a. In the case of $N = 100$, the average CPU time of Algorithm 3 for each iteration, 0.5538 s, is much less than the average time of Algorithm 3.a, 0.1735 s. Therefore, selecting some of the subintervals instead of involving all the subintervals makes the computation more efficient. We can also find in Fig. 1 that the subintervals we refine concentrate on the neighborhood of the active point. Furthermore, the terminated solution of Algorithm 3 is feasible to this example. As $g(x, y) \leq g_j(x, y)$ and $g_j(x, y) = g(x, y) + \frac{\alpha_j}{2}(y - \eta_j^L)(\eta_j^U - y) \leq \gamma$ for all $y \in Y_j$ and for all $Y_j \in E_\nu^c$ at each iteration ν, the terminated solution generated by Algorithm 3 solution may be feasible.

Example 2 Let $Y = Y^P \cup Y^S$ be the union of disjoint closed design frequency interval of a filter, where $Y^P = [0, y^p]$ and $Y^S = [y^s, \pi]$ represent the passband and stopband, respectively. $0 < y^p < y^s < \pi$ are well-defined frequencies. A desired prescription for the frequency response of the filter is given by

$$D(y) = \begin{cases} e^{-jy\tau_0}, & \text{for } y \in Y^P; \\ 0, & \text{for } y \in Y^S, \end{cases} \tag{25}$$

where τ_0 is the given constraint group delay and $j = \sqrt{-1}$. The *Finite Impulse Response* (FIR) filter design problem considers the best approximation of a given magnitude response by that of a FIR filter. The most recent approach consists of solving the complex Chebyshev approximation problem [10, 17]

$$\rho_{CA}^{*} := \min_{x\in\mathbb{R}^n} \max_{y\in Y} W(y)\big||D(y)| - |H(x, y)|\big| \tag{26}$$

where $x := (x_1, \ldots, x_n)^\top \in \mathbb{R}^n$ is the vector containing the coefficients of the unit impulse response, and

$$H(x, y) := \sum_{k=1}^{n} x_k e^{-jy(k-1)}.$$

Problem (26) is equivalent to the nonlinear SIP problem

$$\begin{aligned}&\max_{(x,\delta)\in\mathbb{R}^{n+1}} \quad \delta\\ &\text{subject to} \quad W(y)\big||D(y)| - |H(x, y)|\big| - \delta \le 0, \quad \text{for all } y \in Y.\end{aligned}$$

In [16, 17], the authors suggest some new algorithms for the solution of problem (26) in the case that problem (26) is a convex SIP problem. More general formulations of the design approximation problems in the frequency domain are presented in [10]. The authors in [10] proposed serval non-convex SIP models for the FIR filter design and solved serval specific design problems. In Example 2, we implement one of the non-convex SIP models in [10].

Let $n = 91$, $Y^P = [0, \pi/5]$, and $Y^S = [\pi/4, \pi]$. The desired frequency response is given by $\tau_0 = 120$ in (25). This example approximates $|D(y)| = 1$ by the composed magnitude response $M_Q(y)M(x, y)$ of both filters in Y^P and bounds $M_Q(y)M(x, y)$ by a constant $\delta_S := 3.16227e - 4$, where the functions are given by

$$\begin{aligned}&M(x, y) = |H(x, y)|, \quad \text{and} \quad M_Q(y) = |Q(jy)|;\\ &Q(y) := \frac{1}{1 + 1.3202(y/\omega_g) + 0.86175(y/\omega_g)^2 + 0.2887(y/\omega_g)^3},\end{aligned} \tag{27}$$

where $\omega_g := \pi/7.02832$ is a constant. From [10], we know that $M_Q(y)$ is positive for all $y \in Y$. Hence, the error function $|1 - M_Q(y)M(x, y)|$ on Y^P can be rewritten as

$$M_Q(y)\left(\frac{1}{M_Q(y)} - M(x, y)\right), \quad \text{for } y \in Y^P,$$

and the other error function on Y^S is

$$|0 - M_Q(y)M(x, y)| = M_Q(y)M(x, y) \le \delta_S, \quad \text{for } y \in Y^S.$$

Thus, this problem can be interpreted as the problem (26) with the weight function $W(y) = M_Q(y)$ and the desired magnitude response $1/M_Q(y)$ in Y^P. Due to the

Table 2 Numerical results for Example 2

Algorithm	N	m_1	opt-val	cpu-t	#iter	#re-int	max g
3.a	120	120	0.007810	46757.9496	150*	–	−2.0261e-6
3	120	60	0.007756	23419.6712	92	23	6.1378e-8

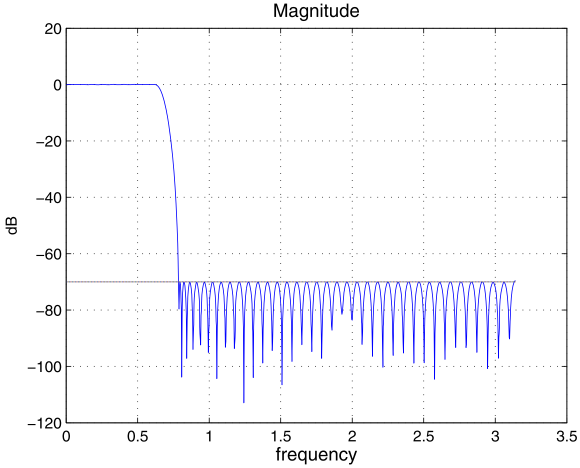

Fig. 2 Magnitude in Example 2

result in [10], the problem has a solution that can be computed by the following non-convex SIP problem:

$$
\begin{aligned}
(\mathrm{SIP}_{FIR}) \quad & \max_{(x,\delta)\in X} \ \delta \\
& \text{subject to} \quad +1 - M_Q(y)M(x,y) - \delta \le 0, \quad \text{for } y \in Y^P, \\
& \qquad\qquad\quad -1 + M_Q(y)M(x,y) - \delta \le 0, \quad \text{for } y \in Y^P, \\
& \qquad\qquad\quad M_Q(y)M(x,y) - \delta_S \le 0, \quad \text{for } y \in Y^S,
\end{aligned}
$$

where we restrict $X = [-0.3, 0.3]^{n+1}$. We discretize Y^P into 20 subintervals and Y^S into 100 subintervals to implement the (SIP_{FIR}) problem. The parameters are set by $\varepsilon = 1e-4$ and $\gamma = 1e-6$. After calculating, we let $\alpha = 1.901$ to start the two algorithms. Table 2 shows our results and Fig. 2 represents the magnitude $M_Q(y)M(x,y)$ for $y \in [0,\pi]$.

The symbol * denotes that Algorithm 3.a reaches the maximum iteration number we set. Therefore, Algorithm 3.a here runs for 150 iterations and does not meet the stopping criteria. Note that *fseminf* failed on this example. For each iteration, we find that the NLP solver in Matlab does not compute efficiently. Moreover, the improvement for the optimal value is very torpid. Hence, if the number of initial discretization is too small, it costs more CPU time to run this example. However, Algorithm 3 is obviously much better than Algorithm 3.a in the FIR filter design example.

6 Conclusion

We propose a new algorithm and show the convergence properties of our algorithm in this paper. We improve the algorithm proposed in [8] by combining it with the concept of the cutting plane method. The convexification relaxation on the variable y for the constraint function g can make the computation for the lower level optimization problem easier. This method has an advantage in solving non-convex SIP problems, especially when the constraint function g is complicated. Based on the numerical results in the numerical section, Algorithm 3 implements with better efficiency in Examples 1 and 2 than the other two methods.

Acknowledgements The authors would like to thank the associate editor and the two anonymous referees for their helpful comments and suggestions that have enabled us to improve the preliminary version of this paper.

References

1. Adjiman, C.S., Androulakis, I.P., Floudas, C.A.: A global optimization method, αBB, for general twice-differentiable constrained NLPs—I: Theoretical advances. Comput. Chem. Eng. **22**, 1137–1158 (1998)
2. Adjiman, C.S., Androulakis, I.P., Floudas, C.A.: A global optimization method, αBB, for general twice-differentiable constrained NLPs—II: Implementation and computational results. Comput. Chem. Eng. **22**, 1159–1179 (1998)
3. Bianco, C.G.L., Piazzi, A.: A hybrid algorithm for infinitely constrained optimization. Int. J. Syst. Sci. **32**, 91–102 (2001)
4. Betrò, B.: An accelerated central cutting plane algorithm for linear semi-infinite programming. Math. Program. **101**, 479–495 (2004)
5. Bhattacharjee, B., Green, W.H. Jr., Barton, P.I.: Interval methods for semi-infinite programs. Comput. Optim. Appl. **30**, 63–93 (2005)
6. Bhattacharjee, B., Lemonidis, P., Green, W.H.Jr., Barton, P.I.: Global solution of semi-infinite programs. Math. Program. **103**, 283–307 (2005)
7. Floudas, C.A.: Deterministic Global Optimization, Theory, Methods and Applications. Kluwer Academic, Dordrecht (2000)
8. Floudas, C.A., Stein, O.: The adaptive convexification algorithm: a feasible point method for semi-infinite programming. SIAM J. Optim. **18**, 1187–1208 (2007)
9. Gonzaga, G., Polak, E., Trahan, R.: An improved algorithm for optimization problems with functional inequality constraints. IEEE Trans. Autom. Control **AC25**, 49–54 (1980)
10. Görner, S., Potchinkov, A., Reemtsen, R.: The direct solution of nonconvex nonlinear FIR filter design problems by a SIP method. Optim. Eng. **1**, 123–154 (2000)
11. Gramlich, G., Hettich, R., Sachs, E.W.: Local convergence of SQP methods in semi-infinite programming. SIAM J. Optim. **5**, 641–658 (1995)
12. Hettich, R., Kortanek, K.: Semi-infinite programming: theory, methods and applications. SIAM Rev. **35**, 380–429 (1993)
13. Jennings, L.S., Teo, K.L.: A computational algorithm for functional inequality constrained optimization problems. Automatica **26**, 371–375 (1990)
14. Kortanek, K., No, H.: A central cutting plane algorithm for convex semi-infinite programming problems. SIAM J. Optim. **3**, 901–918 (1993)
15. Ni, Q., Ling, C., Qi, L., Teo, K.L.: A truncated projected Newton-type algorithm for large-scale semi-infinite programming. SIAM J. Optim. **16**, 1137–1154 (2006)
16. Potchinkov, A., Reemtsen, R.: FIR filter design in the complex domain by a semi-infinite programming technique. I. The method. Arch. Elektron. Übertrag.tech. **48**, 35–144 (1994). II. Examples: 200–209
17. Potchinkov, A., Reemtsen, R.: The simultaneous approximation of magnitude and phase by FIR digital filters. Part 1. A new approach. Int. J. Circuit Theory Appl. **25**, 167–177 (1997). Part 2. Methods and examples: 179–197

18. Price, C.J., Coope, C.J.: Numerical experiments in semi-infinite programming. Comput. Optim. Appl. **6**, 169–189 (1996)
19. Polak, E.: Optimization, Algorithms and Consistent Approximations. Springer, Berlin (1997)
20. Qi, L., Ling, C., Tong, X.J., Zhou, G.: A smoothing projected Newton-type algorithm for semi-infinite programming. Comput. Optim. Appl. **42**, 1–30 (2009)
21. Reemsten, R., Görner, S.: Numerical methods for semi-infinite programming: a survey. In: Reemsten, R., Rückmann, J. (eds.) Semi-infinite Programming, pp. 195–275. Kluwer Academic, Boston (1998)
22. Shapiro, A.: Semi-infinite programming, duality, discretization and optimality condition. Optimization **58**, 133–161 (2009)
23. Still, G.: Discretization in semi-infinite programming: the rate of convergence. Math. Program. **91**, 53–69 (2001)
24. Tanaka, Y., Fukushima, M., Ibaraki, T.: A global convergent SQP method for semi-infinite nonlinear optimization. J. Comput. Appl. Math. **23**, 141–153 (1988)
25. Teo, K.L., Yang, X.Q., Jennings, L.S.: Computational discretization algorithms for functional inequality constrained optimization. Ann. Oper. Res. **28**, 215–234 (2000)
26. Wu, S.Y., Li, D.H., Qi, L., Zhou, G.: An iterative method for solving KKT system of the semi-infinite programming. Optim. Methods Softw. **20**, 629–643 (2005)
27. Zhang, L.P., Hayashi, S., Wu, S.Y.: On the finite termination of an exchange method for convex and nonlinear semi-infinite programming problems. Technique report (2010)
28. Zhang, L.P., Wu, S.Y., López, M.A.: A new exchange method for convex semi-infinite programming. SIAM J. Optim. **20**, 2959–2977 (2010)