Manuscript submitted to
AIMS' Journals
Volume , Number **0**,

Website: http://AIMsciences.org

pp. –

A SMOOTHING APPROACH FOR SEMI-INFINITE PROGRAMMING WITH PROJECTED NEWTON-TYPE ALGORITHM

Zhi Guo Feng

College of Mathematics and Computer Science, Chongqing Normal University
Chongqing, P.R.C.

Kok Lay Teo

Department of Mathematics and Statistics, Curtin University of Technology
Perth, W.A., Australia

Volker Rehbock

Department of Mathematics and Statistics, Curtin University of Technology
Perth, W.A., Australia

(Communicated by the associate editor name)

Abstract. In this paper we apply the projected Newton-type algorithm to solve semi-infinite programming problems. The infinite constraints are replaced by an equivalent nonsmooth function which is then approximated by a smoothing function. The KKT system is formulated as a nonsmooth equation. We then apply the projected Newton-type algorithm to solve this equation and show that the accumulation point satisfies the KKT system. Some numerical results are presented for illustration.

1. **Introduction.** Semi-infinite programming problems (SIP) have wide applications, such as the approximation theory, optimal control, eigenvalue computation, statistical design and other engineering problems. There are many papers in the literature dedicating to SIP problems. These include survey papers, such as [2, 11]. In [4], a constraint transcription technique is developed to solve a general class of SIP problems, where a smoothing parameter is required to be adjusted. In [13], the constraint transcription technique is used in conjunction with the concept of the penalty function for solving a general SIP problem.

Newton method is an effective tool in the numerical computation of nonlinear equations ([7, 9, 12, 14]). In [5, 6, 10, 15, 16], Newton method was used to develop numerical algorithms for solving SIP problems, where the KKT system is formulated as a nonsmooth equation. This method is very efficient, but its shortcoming is also quite obvious. More specifically, the number of Lagrange multipliers in KKT system is not known. In fact, even if the number of Lagrange multipliers is assumed given, it is still a demanding task to determine the respective attainers.

2000 *Mathematics Subject Classification.* Primary: ; Secondary:

Key words and phrases. KKT system, Projected Newton-type algorithm, Smoothing function.

This project is supported by a research grant from the Australian Research Council, and National Natural Science Foundation of China under Grant 60704003.

To overcome these difficulties, we apply the Newton method to solve the SIP problems through a different formulation of the KKT system. More specifically, the infinite constraints are transformed into an equivalent nonsmooth function. We then approximate this nonsmooth function by a smoothing function with a positive smoothing variable. For a given smoothing variable, we derive the corresponding KKT system for the smoothing function. The Newton method is used to develop a computational algorithm for solving this KKT system, where the smoothing variable is kept positive during computation. In this way, the task of determining the number of Lagrange multipliers and to the respective attainers is not necessary.

We organize the paper as follows. In Section 2, we formulate the KKT system of this problem by introducing a smoothing function. In Section 3, some sufficient conditions are obtained. In Section 4, approximate smoothing functions are introduced. In Section 5, a projected Newton-type algorithm is developed. For illustration, some numerical results are showed in Section 6.

2. **Problem Formulation.**

We consider the semi-infinite programming (SIP) problem

$$\min f(x) \tag{2.1a}$$

$$s.t.\ \ g(x,v) \le 0, \forall v \in V, \tag{2.1b}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ are twice continuously differentiable functions, and V is a bounded compact subset in $\mathbb{R}^m$.

An efficient method to solve (2.1) is to formulate its KKT system as an equation and then the Newton method is used. However, the KKT system so formulated contains p multipliers together with the corresponding points $v_i, i = 1, \ldots, p$, at each of which $g(x, v_i) = 0$. These points are called attainers. However, the number p is not known, and so are the attainers.

The determination of v_i is a serious task. For example, suppose $v_1, v_2 \in V, v_1 \neq v_2$, such that $g(x, v_i) = 0, i = 1, 2$. For each $i = 1, 2$, v_i is solved by the necessary condition $\nabla_v g(x, v_i) = 0$. Since both v_1 and v_2 satisfy this condition, we may end up with a solution of $v_1 = v_2$ during the computation. This means that we have missed finding one of the two attainers. Consequently, it also gives a wrong indication on the number p.

Because of these deficiencies, we propose an alternative approach. Clearly, the infinite inequality constraints (2.1b) are equivalent to

$$G(x) = \max_{v\in V} g(x, v) \le 0. \tag{2.2}$$

However, $G(x)$ is nonsmooth. Suppose there exists a function $G_s(t, x)$, defined in $\{(t, x)|t \ge 0, x \in \mathbb{R}^n\}$, such that the following properties are satisfied.

Property 1 $(G_s(t, x))$**.**

i. $G_s(t, x)$ *is twice differentiable when* $t > 0$.

ii. $G_s(0, x) = G(x), \quad \forall x \in \mathbb{R}^n$.

iii. $\lim\limits_{t\to 0+, z\to x} G_s(t, z) = G(x), \quad \forall x \in \mathbb{R}^n$.

iv. $\nabla_t G_s(t, x) \ge 0, \forall x \in \mathbb{R}^n, t > 0$.

$G_s(t, x)$ is a smoothing function of $G(x)$. For properties of smoothing functions, see [8]. In Section 4, we will give a smoothing function which satisfies the properties mentioned above.

For each $t > 0$, we obtain an approximate problem of (2.1) given below.

$$\min f(x) \tag{2.3a}$$

$$s.t.\ \ G_s(t,x) \le 0. \tag{2.3b}$$

Then, we aim to solve (2.1) via solving (2.3) as $t \to 0+$.

For this approximate problem, we assume that the constraint qualification holds at the optimal solution $x(t)$ when $t > 0$ is sufficiently small. Clearly, the KKT system of the approximate problem (2.3) is given by

$$\left\{\begin{array}{c} \nabla f(x(t)) + \mu(t)\nabla_x G_s(t,x(t)) = 0 \\ \mu(t)G_s(t,x(t)) = 0 \\ \mu(t) \ge 0,\ G_s(t,x(t)) \le 0, \end{array}\right. \tag{2.4}$$

where $\mu(t)$ is the Lagrange multiplier with respect to $x(t)$.

Suppose that $x(t)$ has an accumulation point $x(0)$ as $t \to 0+$, and the corresponding Lagrange multiplier vector $\mu(t)$ also has an accumulation point $\mu(0)$ as $t \to 0+$. Then, the pair $(x(0),\mu(0))$ satisfies

$$\left\{\begin{array}{c} \nabla f(x(0)) + \mu(0)\nabla_x G_s(0,x(0)) = 0 \\ \mu(0)G_s(0,x(0)) = 0 \\ \mu(0) \ge 0,\ G_s(0,x(0)) \le 0, \end{array}\right. \tag{2.5}$$

where $\nabla_x G_s(0,x(0)) = \lim\limits_{t\to 0+} \nabla_x G_s(t,x(0))$.

Adding a new positive variable t into (x,μ), we see that $(0,x(0),\mu(0))$ is a solution of

$$\left\{\begin{array}{c} t = 0 \\ \nabla f(x(t)) + \mu(t)\nabla_x G_s(t,x(t)) = 0 \\ \mu(t)G_s(t,x(t)) = 0 \\ \mu(t) \ge 0,\ G_s(t,x(t)) \le 0. \end{array}\right. \tag{2.6}$$

Next, we consider the constraint $G_s(t,x) \le 0$. It is equivalent to

$$\max\{0, G_s(t,x)\} = 0.$$

The function $\max(0,y)$ is nonsmooth. It can be approximated by many kinds of smoothing functions. Here, we suppose that a smoothing function $\varphi_t(y)$, which satisfies the following properties, is given.

Property 2 ($\varphi_t(y)$)**.**

i. $\varphi_t(y)$ *is differentiable when* $t > 0$.
ii. $\varphi_t(y) = \max(0,y), \quad \forall y \in \mathbb{R}$.
iii. $\lim\limits_{t\to 0+, z\to y} \varphi_t(z) = \max(0,y), \quad \forall y \in \mathbb{R}$.

The function $\max(0, G_s(t,x)) = 0$ can be approximated by

$$\tilde{G}(t,x) = \varphi_t(G_s(t,x)) = 0. \tag{2.7}$$

Then, by denoting $\omega = (t,x,\mu)$ and letting $\Omega = \{(t,x,\mu)| \in \mathbb{R}^{n+2}, t \ge 0, \mu \ge 0\}$ be the set of all ω, $(0,x(0),\mu(0))$ is the solution of the following problem.

Problem 1. *Find a* $\omega \in \Omega$*, such that*

$$\Upsilon(\omega) = \left(\begin{array}{c} t \\ W(t,x,\mu) \end{array}\right) = 0, \tag{2.8}$$

where

$$W(t,x,\mu) = \begin{pmatrix} \nabla f(x) + \mu \nabla_x G_s(t,x) \\ \tilde{G}(t,x) \\ \mu G_s(t,x) \end{pmatrix}. \tag{2.9}$$

For this problem, $\nabla_x G_s(t,x)$ is not continuous when $t = 0$. Thus, it is important to keep $t > 0$ during the numerical computation. In Section 5, we will propose an algorithm by keeping $t > 0$ during the iteration and find the accumulation point as $t \to 0+$.

3. **Approximation Analysis.** In this section, we introduce some sufficient conditions and show that the optimal solution $x = x(0)$ of (2.1) together with $t = 0$ and the corresponding multiplier vector $\mu = \mu(0)$ is the accumulation point of Problem 1. On this basis, we can solve problem (2.1) by finding the accumulation point of Problem 1.

Suppose that the smoothing function $G_s(t,x)$, which satisfies the Property 1, is given. Define

$$X_t = \{x | G_s(t,x) \le 0\}, \quad X_0 = \{x | G(x) \le 0\}. \tag{3.1}$$

Clearly, X_t and X_0 denote the feasible fields of the approximate problem (2.3) and problem (2.1), respectively. Since $\nabla_t G_s(t,x) \ge 0$, $G_s(t,x)$ is monotonically increasing with respect to t. That is,

$$G_s(t_1,x) \ge G_s(t_2,x) \ge G(x), \quad \text{if } t_1 > t_2 > 0. \tag{3.2}$$

Then, we have

$$X_{t_1} \subseteq X_{t_2} \subseteq X_0, \quad \text{if } t_1 > t_2 > 0. \tag{3.3}$$

Consequently,

$$\overline{\bigcup_{t>0} X_t} \subseteq X_0. \tag{3.4}$$

In this paper, we assume that the following condition is satisfied.

$$X_0 = \overline{\bigcup_{t>0} X_t}. \tag{3.5}$$

Some sufficient conditions for the validity of (3.5), are given in the following theorem.

Theorem 3.1. *Define $X_0' = \{x | x \in X_0, G(x) = 0\}$. If X_0' does not contain any open sets in X_0 (i.e., $(\mathcal{O} \cap X_0) \nsubseteq X_0'$, for any open set $\mathcal{O} \subset \mathbb{R}^n$). Then, (3.5) is satisfied.*

Proof. Suppose that (3.5) is not satisfied, that is, $\exists x' \in X_0$ but $x' \notin \overline{\bigcup_{t>0} X_t}$. Then, we must have $G(x') = 0$. This is due to the fact that if $G_s(0,x') = G(x') < 0$, then, by the continuity of $G_s(t,x')$ with respect to t, there exists a $t_1 > 0$, such that $G_s(t,x') \le 0$ for all t, $0 \le t < t_1$. Thus, $x' \in \overline{\bigcup_{t>0} X_t}$. This is a contradiction. Therefore, $x' \in X_0'$. To continue, we note that $\overline{\bigcup_{t>0} X_t}$ is a closed set in X_0 and $x' \notin \overline{\bigcup_{t>0} X_t}$. Thus, there exists a neighborhood $\mathcal{N}(x') \subset \mathbb{R}^n$, such that $\mathcal{N}(x') \cap \overline{\bigcup_{t>0} X_t} = \emptyset$. Since $X_0 = X_0' \cup \overline{\bigcup_{t>0} X_t}$, we have

$$(\mathcal{N}(x') \cap X_0) = (\mathcal{N}(x') \cap X_0') \subseteq X_0'.$$

This is a contradiction. Thus, (3.5) is satisfied. □

On the basis of Theorem 3.1, we have the following corollary.

Corollary 1. *Suppose that the extended Mangasarian-Fromovitz constraint qualification (EMFCQ) holds at* X_0'. *That is, for any* $x \in X_0'$, *there exists a vector* $d \in \mathbb{R}^n$ *such that*

$$(\nabla_x g(x, v))^\intercal d < 0, \quad \forall v \in V(x), \tag{3.6}$$

where

$$V(x) = \{v \in V : g(x, v) = 0\}.$$

Then, (3.5) is satisfied.

Proof. For any $x \in X_0'$ and any neighborhood $\mathcal{N}(x)$ of x in $\mathbb{R}^n$, it follows from (3.6) that there exists a point $x' \in \mathcal{N}(x)$ such that $G(x') < G(x) = 0$. This means that X_0' does not contain any open set in $\mathbb{R}^n$. Consequently, X_0' does not contain any open set in X_0. Therefore, by virtue of Theorem 3.1, (3.5) is satisfied. □

We have the following theorem.

Theorem 3.2. *Suppose that the condition (3.5) is satisfied. Let* $x(t)$ *be the optimal solution of the approximate problem (2.3) and let* $x(t)$ *admit at least an accumulation point* $x(0)$ *as* $t \to 0+$. *Then,* $x(0)$ *is an optimal solution of (2.1).*

Proof. Note that

$$X_{t_1} \subseteq X_{t_2} \subseteq X_0, \quad \text{if } t_1 > t_2 > 0.$$

We have

$$f(x(t_1)) \geq f(x(t_2)) \geq f(x(0)).$$

Consequently, $\{f(x(t)\}$ has a limit when $t \to 0+$. Since $x(0)$ is an accumulation point, it follows that, by the continuity of f, $f(x(0))$ is a limit of $\{f(x(t))\}$ as $t \to 0+$.

Let f^* be the optimal value of (2.1). Since $x(0)$ is the accumulation point of $x(t)$, we have $x(0) \in X_0$ and hence $f(x(0)) \geq f^*$.

Suppose that $f(x(0)) > f^*$. Then, by the continuity of f over X_0, there exists a point $x_1 \in X_0$ such that $f^* < f(x_1) < f(x(0))$. By (3.5), there exists a sequence $\{x_1(t) : x_1(t) \in X_t\}$, such that $\lim_{t \to 0+} x_1(t) = x_1$. Thus, we have $f(x_1(t)) < f(x(0)) \leq f(x(t))$, provided t is sufficiently small. This contradicts to the fact that $x(t)$ is the optimal solution of (2.3). Thus, we have $f(x(0)) = f^*$ and $x(0)$ is an optimal solution of (2.1). The proof is complete. □

4. **Smoothing Functions.** In this section, we give the appropriate smoothing functions $G_s(t, x)$ and $\varphi_t(y)$, and then discuss their respective properties.

In a numerical computation, it is impossible to calculate the exact maximum of the function $g(x, v)$ over the compact set V. We shall discretized the compact set V into $V_N = \{v_i \in V, i = 1, \ldots, N\}$. Then, we find the maximum of the function $g(x, v)$ over the discretized set V_N. In this way, we obtain an approximate function for $G(x)$ as given below.

$$G_N(x) = \max_{v \in V_N} g(x, v) = \max_i g(x, v_i). \tag{4.1}$$

For $G_N(x)$, we introduce the following smoothing function.

$$G_s(t,x) = \begin{cases} A + t\ln(\sum\limits_{i=1}^{N} e^{(g(x,v_i)-A)/t}), & \text{if } t > 0, \\ G_N(x), & \text{if } t = 0, \end{cases} \tag{4.2a}$$

where A is a constant which satisfies

$$A \geq \max_{v\in V_N} g(x,v). \tag{4.2b}$$

Remark 1. *The smoothing function (4.2) is obtained by modifying the one given in* [15]*, which is:*

$$G_s(t,x) = \begin{cases} t\ln(\sum\limits_{i=1}^{N} e^{g(x,v_i)/t}), & \textit{if } t > 0, \\ G_N(x), & \textit{if } t = 0, \end{cases} . \tag{4.3}$$

The original smoothing function (4.3) is unsatisfactory in the sense that it will lead to overflow in the numerical computation when $g(x,v_i)$ is positive and t is small. Thus, we have constructed the smoothing function (4.2). With this smoothing function, the deficiency associated with (4.3) is overcome.

We need to verify that $G_s(t,x)$ satisfies the properties given in Section 2. The first two properties are easy to verify. For the third property, it follows from the continuity of $G(x)$ and the approximation property given by

$$0 \leq G_s(t,x) - G_N(x) \leq t\ln(N), \qquad \forall x \in \mathbb{R}^n. \tag{4.4}$$

For the fourth property, we first calculate $\nabla_t G_s(t,x)$ directly, giving

$$\nabla_t G_s(t,x) = \ln(\sum_{i=1}^{N}(e^{(g(x,v_i)-A)/t})) - \frac{1}{t}\frac{\sum\limits_{i=1}^{N}[e^{(g(x,v_i)-A)/t}\cdot(g(x,v_i)-A)]}{\sum\limits_{i=1}^{N} e^{(g(x,v_i)-A)/t}}. \tag{4.5}$$

Since there exists at least one integer $i' \in \{1,\dots,N\}$ such that $g(x,v_{i'}) = A$, it follows that we have $\sum\limits_{i=1}^{N} e^{(g(x,v_i)-A)/t} \geq 1$ and consequently

$$\ln(\sum_{i=1}^{N} e^{(g(x,v_i)-A)/t}) \geq 0.$$

Furthermore, when $t > 0$, by virtue of the facts that $e^{(g(x,v_i)-A)/t} \geq 0$ and $g(x,v_i) - A \leq 0$, it follows that

$$-\frac{1}{t}\frac{\sum\limits_{i=1}^{N}[e^{(g(x,v_i)-A)/t}\cdot(g(x,v_i)-A)]}{\sum\limits_{i=1}^{N} e^{(g(x,v_i)-A)/t}} \geq 0.$$

Thus, the fourth property in Section 2 is also satisfied. Therefore, we conclude that $G_s(t,x)$ is a well-defined smoothing function.

To proceed further, we note that $\nabla_x G_s(t,x)$ can be calculated directly as

$$\nabla_x G_s(t,x) = \frac{\sum_{i=1}^{N} (e^{(g(x,v_i)-A)/t} \cdot g_x(x,v_i))}{\sum_{i=1}^{N} e^{(g(x,v_i)-A)/t}}, \quad t > 0. \tag{4.6}$$

On this basis, suppose that

$$\lim_{t\to 0+} \nabla_x G_s(t,x) = \sum_{i\in I(x)} g_x(x,v_i)/n_x \neq 0 \tag{4.7}$$

holds on $x(0)$, where $I(x) = \{i \in \{1,\dots,N\} : g(x,v_i) = A\}$ and that n_x is the number of $I(x)$. Then, (2.4) is satisfied.

For $\max\{0,y\}$, there are many smoothing functions. We use the Chen-Harker-Kanzow-Smale smoothing function given by

$$\varphi_t(y) = \frac{\sqrt{y^2+4t^2}+y}{2}. \tag{4.8}$$

The approximation property for this smoothing function is given by

$$0 \le \varphi_t(y) - \max(0,y) \le t, \qquad \forall y \in \mathbb{R}. \tag{4.9}$$

Thus, by the choices of $G_s(t,x)$ and $\tilde{G}(t,x)$, we can formulate the SIP problem (2.1) in the form of Problem 1.

5. **A Smoothing Projected Newton-Type Algorithm.** In this section, we propose an algorithm To solve Problem 1, which keeps $t > 0$ during the iterations of numerical computation. It is based on the smoothing projected Newton-type algorithm proposed in [12]. Details are given as follows.

Define a merit function Ψ by

$$\Psi(\omega) = \frac{1}{2} \parallel \Upsilon(\omega) \parallel^2 . \tag{5.1}$$

Then, Problem 1 is equivalent to

Problem 2. *Find a $\omega \in \Omega$, such that*

$$\min \Psi(\omega) = 0. \tag{5.2}$$

It is not difficult to see that $\Psi(\omega)$ is continuously differentiable when $t > 0$, and the gradient is given by

$$\nabla\Psi(\omega) = \nabla\Upsilon(\omega)\Upsilon(\omega). \tag{5.3}$$

A common stopping rule of numerical methods for solving Problem 2 is given by the projected gradient direction (see [12]). That is,

$$\parallel \bar{d}_G(1) \parallel = 0. \tag{5.4}$$

Here,

$$\bar{d}_G(1) = \Pi_\Omega(\omega - \gamma\nabla\Psi(\omega)) - \omega, \tag{5.5}$$

where $\gamma > 0$ is a constant related to ω, $\Pi_\Omega(\cdot)$ is an orthogonal projection operator onto Ω.

Based on this stopping rule, we state the smoothing projected Newton-type algorithm as follows.

Algorithm 1.

0. *(Initialization)*

Choose constants $\eta, \rho, \xi \in (0,1), \epsilon > 0, p_1 > 0, p_2 > 0$ *and* $\xi > 0, \bar{t} > 0, \alpha < 1$ *with* $\alpha\bar{t} < 1$. *Let* $\bar{\omega} = (\bar{t}, 0, 0)$, $t^0 = \bar{t}$ *and* $\omega^0 = (t^0, x^0, \mu^0)$ *with* $\mu^0 \geq 0$. *Set* $k := 0$.

1. *(Stop Test)*

Let

$$\gamma_k = \min\{1, \frac{t^k}{|t^k + \nabla_t \bar{W}(\omega^k)\bar{W}(\omega^k)|}, \frac{\eta\|\Upsilon(\omega^k)\|}{\|\nabla\Psi(\omega^k)\|}, \frac{\eta\Psi(\omega^k)}{\|\nabla\Psi(\omega^k)\|^2}\}, \tag{5.6}$$

where $\nabla_t \bar{W}(\omega^k)$ *is the first row of* $\nabla \bar{W}(\omega^k)$. *Compute* $\bar{d}_G^k(1)$ *by (5.5). If* $\|\Upsilon(\omega^k)\| < \epsilon$, *stop. Otherwise, compute* β_k *by*

$$\beta_0 = \alpha \min\{1, \|\bar{d}_G^0(1)\|^2\},$$

$$\beta_k = \begin{cases} \beta_{k-1}, & \textit{if } \alpha \min\{1, \|\bar{d}_G^k(1)\|^2\} > \beta_{k-1} \\ \alpha \min\{1, \|\bar{d}_G^k(1)\|^2\}, & \textit{otherwise} \end{cases} . \tag{5.7}$$

2. *(Compute Search Direction)*

Compute d_G^k *by*

$$d_G^k = -\gamma_k \nabla\Psi(\omega^k) + \beta_k \bar{\omega}. \tag{5.8}$$

Compute d_N^k *by solving the following linear system:*

$$\Upsilon(\omega^k) + \Upsilon'(\omega^k) d_N^k = \beta_k \bar{\omega}. \tag{5.9}$$

If (5.9) has no solution or

$$\nabla\Psi(\omega^k)^T d_N^k < p_1 \|d_N^k\|^{p_2},$$

then let $d_N^k := d_G^k$.

3. *(Line Search)*

Let m_k *be the smallest nonnegative integer* m *satisfying*

$$\Psi(\omega^k + \bar{d}^k(\rho^m)) \leq \Psi(\omega^k) + \xi \nabla\Psi(\omega^k)^T \tilde{d}_G^k(\rho^m), \tag{5.10}$$

where for any $\lambda \in [0,1]$,

$$\bar{d}^k(\lambda)) = \tau^*(\lambda)\tilde{d}_G^k(\lambda) + (1 - \tau^*(\lambda))\tilde{d}_N^k(\lambda). \tag{5.11}$$

Here,

$$\tilde{d}_G^k(\lambda) := \Pi_\Omega(\omega^k + \lambda d_G^k) - \omega^k, \;\; \tilde{d}_N^k(\lambda) := \Pi_\Omega(\omega^k + \lambda d_N^k) - \omega^k, \tag{5.12}$$

$\tau^*(\lambda)$ *is a solution of the following minimization problem:*

$$\min_{\tau \in [0,1]} \frac{1}{2} \|\Upsilon(\omega^k) + \Upsilon'(\omega^k)[\tau \tilde{d}_G^k(\lambda) + (1-\tau)\tilde{d}_N^k(\lambda)\|^2. \tag{5.13}$$

Let $\lambda_k = \rho^{m_k}$ *and* $\omega^{k+1} = \omega^k + \bar{d}^k(\lambda_k)$.

4. *Set* $k := k + 1$ *and go to step 1.*

Remark 2. $\tau^*(\lambda)$ *is derived as*

$$\tau^*(\lambda) = \max\{0, \min\{1, \tau(\lambda)\}\}, \tag{5.14a}$$

where

$$\tau(\lambda) = \begin{cases} 0, & \textit{if } \Upsilon'(\omega^k)[\tilde{d}_G^k(\lambda) - \tilde{d}_N^k(\lambda)] = 0, \\ -\frac{[\Upsilon(\omega^k) + \Upsilon'(\omega^k)\tilde{d}_N^k(\lambda)]^T \Upsilon'(\omega^k)[\tilde{d}_G^k(\lambda) - \tilde{d}_N^k(\lambda)]}{\|\Upsilon'(\omega^k)[\tilde{d}_G^k(\lambda) - \tilde{d}_N^k(\lambda)]\|^2}, & \textit{otherwise.} \end{cases} \tag{5.14b}$$

In this algorithm, the choice of β_k, which is defined in (5.7), is the perturbed parameter to keep $t > 0$. It is shown on the following theorem, which is also given in [6, 15].

Theorem 5.1. *Let $\{\omega^k = (t^k, x^k, \mu^k)\}$ be a sequence generated by Algorithm 1. Then, for each k, ω^k satisfies*

$$t^k \geq \beta_k \bar{t}. \tag{5.15}$$

Furthermore, if ω^k is not a stationary point of (5.2), then

$$t^k > 0. \tag{5.16}$$

We call $\tilde{d}^k_G(\lambda)$ the projected perturbed-gradient direction and $\tilde{d}^k_N(\lambda)$ the projected perturbed-Newton direction. The search direction $\bar{d}^k(\lambda))$, which is given in (5.11), is the optimal combined direction of $\tilde{d}^k_G(\lambda)$ and $\tilde{d}^k_N(\lambda)$. It guarantees the global and fast local convergence. The proof showing (5.14) is an optimal solution of (5.13) is similar to that given for Lemma 3.1 in [12].

From [6, 15], we have the local descent property of this algorithm given in the following theorems.

Theorem 5.2. *Suppose that $\omega^k = (t^k, x^k, \mu^k) \in \Omega$ with $t^k > 0$ is not a stationary point of (5.2). Then, for any $\lambda \in (0, 1]$, it holds that*

$$\nabla\Psi(\omega^k)^T \tilde{d}^k_G(\lambda) \leq -\frac{\lambda}{\gamma_k}(1 - \alpha\bar{t})\|\tilde{d}^k_G(1)\|^2 < 0. \tag{5.17}$$

Theorem 5.3. *Suppose that $\omega^k = (t^k, x^k, \mu^k) \in \Omega$ with $t^k > 0$ is not a stationary point of (5.2). Then there exists a constant $\lambda' \in (0, 1]$ such that for any $\lambda \in (0, \lambda']$, $\bar{d}^k(\lambda)$ is a decent direction of $\Psi(\omega^k)$ at ω^k and*

$$\Psi(\omega^k + \bar{d}^k(\lambda)) \leq \Psi(\omega^k) + \sigma\nabla\Psi(\omega^k)^T \tilde{d}^k_G(\lambda). \tag{5.18}$$

Details on the superlinear local convergent property and the global convergent property can be found in [6, 15].

6. Numerical Results.

In this section, the proposed method is applied to the following examples. The computation was performed in Fortran 77 double precision. It was run on a PC with the Windows system, having a CPU speed of 1.6GHz and equipped with 192MB RAM. The parameters are set as:

$$\eta = 0.9, \rho = 0.5, \sigma = 0.3, \alpha = 0.6, \epsilon = 10^{-8}, p_1 = 0.5, p_2 = 4, N = 10000.$$

Example 1. *This example is chosen from [6, 16].*

$$f(x) = \frac{1}{2}x^\intercal x, \quad g(x, v) = 3 + 4.5\sin(\frac{4.7\pi(v - 1.23)}{8}) - \sum_{i=1}^{n} x_i v^{i-1},$$

The results obtained are compared with those obtained by other methods. The comparisons are given in Table 1.

Example 2.

$$f(x) = 1.21e^{x_1} + e^{x_2}, \quad g(x, v) = v - e^{x_1+x_2}, \quad V = [0, 1].$$

Example 3.

$$f(x) = (x_1 - 2x_2 + 5x_2^2 - x_2^3 - 13)^2 + (x_1 - 14x_2 + x_2^2 + x_2^3 - 29)^2,$$
$$g(x, v) = x_1^2 + 2x_2v^2 + e^{x_1+x_2} - e^v, \quad V = [0, 1].$$

Methods	V	n	$f(x)$
[6]	[1,100]	10	0.07412
	[1,100]	20	0.08319
	[1,100]	40	3.1788
	[1,100]	60	4.8860
[16]	[0,200]	20	0.0357
Proposed method	[1,100]	10	0.072781
	[1,100]	20	0.032866
	[1,100]	40	0.016433
	[1,100]	60	0.010956
	[0,200]	20	0.035485

Table 1. Results for Example 1

Example 4.

$$f(x) = x_1^2/3 + x_1/2 + x_2^2,$$
$$g(x, v) = (1 - x_1^2 v^2)^2 - x_1 v^2 - x_2^2 + x_2, \quad V = [0, 1].$$

Example 5.

$$f(x) = (x_1 - 0.1)^2 + x_2^2 + 2.5x_3^2,$$
$$g(x, v) = 2\sin(3\pi v + x_3) - x_1^2 - x_2 - 2x_3 - 1, \quad V = [0, 1].$$

The results obtained for Examples 2-5 are given in Table 2.

Examples	(x, μ)	$f(x)$
2	$(-0.0953, 0.0953, 0.1100)$	2.2000
3	$(0.7200, -1.4505, 4.9218)$	97.1589
4	$(-0.7500, -0.6180, 0.5528)$	0.1945
5	$(0.3220, 0.3447, 0.2758, 0.6895)$	0.3583

Table 2. Results for Examples 2-5.

7. **Conclusion.** In this paper, we have presented a smoothing projected Newton-type algorithm for solving the semi-infinite programming problems. By transforming the infinite inequality constraints into a nonsmooth constraint, we constructed the smoothing function to approximate the nonsmooth constraint, leading to a sequence of approximate smoothing problems. The KKT system with a smoothing parameter is formulated for each of the approximate smoothing problems. A projected Newton-type method is applied to solve the KKT system, in which the smoothing parameter is taken as a decision. From the numerical examples solved using the method proposed, we see that it is highly efficient and effective.

REFERENCES

[1] F.H. Clarke, “Optimization and Nonsmooth Analysis,” Wiley Interscience, New York, 1983.

[2] M.A. Goberna and M.A. Lóopez, “Semi-infinite Programming: Recent Advances,” edition, Kluwer Academic Publishers, Boston, 2001.

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

[4] L.S. Jennings and K.L. Teo, *A computational algorithm for functional inequality constrained optimization problems*, Automatica, **26** (1990), 371–375.

[5] D.H. Li, L.Q. Qi, J. Tam and S.Y. Wu, *A smoothing newton method for semi-infinite programming*, Journal of Global Optimization, **30** (2004), 169–194.

[6] Q. Ni, C. Ling, L.Q. Qi and K.L. Teo, *A truncated projected Newton-type algorithm for large-scale semi-infinite programming*, SIAM Journal of Optimization, **16** (2006), 1137–1154.

[7] Burcu Özçam and Hao Cheng, *A discretization based smoothing method for solving semi-infinite variational inequalities*, Journal of Industrial and Management Optimization, **1** (2005), 219–233.

[8] L. Qi and D. Sun, *Smoothing functions and smoothing Newton method for complementarity and variational inequality problems*, J. Optim. Theory and Appl., **113** (2002), 121–147.

[9] L. Qi, D. Sun and G. Zhou, *A new look at smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequalities*, Math. Prog., **87** (2000), 1–35.

[10] L. Qi, S.Y. Wu and G. Zhou, *Semismooth Newton methods for solving semi-infinite programming problems*, J.Global Optim., **27** (2003), 215–232.

[11] R. Reemtsen and S. Görner, "Numerical methods for semi-infinite programming: a survey," In semi-infinite programming, R. Reemtsen and J. Rückmann, Eds., Kluwer Academic Publishers, Boston, 1998, 195–275.

[12] D. Sun, R.S. Womersley and H. Qi, *A feasible semismooth asymptotically Newton method for mixed complementarity problems*, Math.Prog., **94** (2002), 167–187.

[13] K.L. Teo, V. Rehbock and L.S. Jennings, *A new computational algorithm for functional inequlity constrained optimization problems*, Automatica, **29** (1993), 789–792.

[14] X.J. Tong, F.F. Wu, Y.P. Zhang, Z. Yan and Y.X. Ni, *A semismooth Newton method for solving optimal power flow*, Journal of Industrial and Management Optimization, **3** (2007), 553–567.

[15] X.J. Tong and S.Z. Zhou, *A smoothing projected Newton-type method for semismooth equations with bound constraints*, Journal of Industrial and Management Optimization, **1** (2005), 235–250.

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

E-mail address: `z.feng.scholar@gmail.com`