Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 852074, 7 pages
http://dx.doi.org/10.1155/2014/852074

*Research Article*

# A Spline Smoothing Newton Method for Semi-Infinite Minimax Problems

**Li Dong,[1] Bo Yu,[2,3] and Yu Xiao[4]**

[1] *College of Science, Dalian Nationalities University, Dalian 116600, China*
[2] *School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China*
[3] *School of Sciences, Dalian University of Technology, Dalian at Panjin 124221, China*
[4] *School of Basic Science, East China Jiaotong University, Nanchang 330013, China*

Correspondence should be addressed to Li Dong; dongli@dlnu.edu.cn

Received 20 May 2014; Accepted 30 June 2014; Published 17 July 2014

Academic Editor: Mariano Torrisi

Based on discretization methods for solving semi-infinite programming problems, this paper presents a spline smoothing Newton method for semi-infinite minimax problems. The spline smoothing technique uses a smooth cubic spline instead of max function and only few components in the max function are computed; that is, it introduces an active set technique, so it is more efficient for solving large-scale minimax problems arising from the discretization of semi-infinite minimax problems. Numerical tests show that the new method is very efficient.

## 1. Introduction

In this paper, we consider the following semi-infinite minimax problems:

$$(P) \quad \min_{x\in R^n} \max_{y\in Y} \psi(x, y), \tag{1}$$

where $\psi : R^n \times R^m \to R$. We assume (as in Assumption 3.4.1 in [1]) that both $\psi(\cdot,\cdot)$ and $\nabla_x \psi(\cdot,\cdot)$ are Lipschtiz continuous on bounded sets. Such a semi-infinite minimax problem $P$ is an exciting part of mathematical programming. It has very widespread application backgrounds in optimal electronic circuit design, linear Chebyshev approximation, minimization of floor area, optimal control, computer-aided design, numerous engineering problems, and so forth (see [1–8]). Over the past decade, many researchers had done a lot of works on it and proposed some algorithms (see [9–14]). However, efficient algorithms for solving the problem $P$ are few, because it is difficult to design an algorithm to deal with the nondifferentiability of objective function and the infinite set $Y$. A common approach for solving $P$ is the discretization method. Generally, discretization of multidimensional domains gives rise to minimax problems with thousands of component functions. Computation cost is increased; efficiency of the discretization method is affected. To overcome these problems, Polak et al. proposed algorithms with smoothing techniques for solving finite and semi-infinite minimax problems (see [15, 16]). In [16], an active set strategy which can be used in conjunction with exponential (entropic) smoothing for solving large-scale minimax problems arising from the discretization of semi-infinite minimax problems had been proposed. But the active set grew monotonically. In this paper, using the feedback precision-adjustment smoothing parameter rule which was proposed by Polak et al. in [15], we propose a new discretization smoothing algorithm for solving $P$. The smoothing function is cubic spline function not exponential function in our algorithm. The spline smoothing technique uses a smooth spline function instead of thousands of component functions and acts also as an active set technique, so only few components in the max function are computed at each iteration. And the active set does not grow monotonically; hence the number of gradients and Hessian calculations is dramatically reduced, and the computation cost is greatly reduced. Numerical tests show that the new method is very efficient for semi-infinite minimax problems with complicated component functions.

We assume that the problem $P$ can be approximated by a sequence of finite minimax problems of the form

$$\left(P_{Y_N}\right) \quad \min_{x \in R^n} \max_{y \in Y_N} \psi(x, y), \tag{2}$$

where the sets $Y_N$, $N = 1, 2, 3, \ldots$ have finite cardinality. In practice, we can expect that $Y_N \subset Y_{N+1}$, that is, that they grow monotonically, and that the closure of $\cup_{N=1}^{\infty} Y_N$ is equal to $Y$. However, it suffices to assume, as in Assumption 3.4.2 in [1].

*Assumption 1.* There exist a strictly positive valued, strictly monotone decreasing function $\Theta : N \to R$ and constants $K > 0$ and $N_0 > 0$, such that for every $N > N_0$ and $y \in Y$, there exists a $y' \in Y_N$ such that

$$\left\| y - y' \right\| \leq K\Theta(N). \tag{3}$$

It is easy to satisify with Assumption 1. For example, let $Y = [a, b], Y_N = [a, a + (b - a)/2^N, a + 2((b - a)/2^N), \ldots, a + (2^N - 1)((b - a)/2^N), b], \Theta(N) = 1/(2^N - 1), K = 1/2\|Y\| = (1/2)(b - a)$.

We assume that $Y_N = \{y_{N_1}, \ldots, y_{N_{q_N}}\}$.

Next, we let

$$f^{N_j}(x) \triangleq \psi\left(x, y_{N_j}\right), \quad j \in Q_N = \{1, 2, \ldots, q_N\}, \tag{4}$$

and define

$$\begin{aligned} \phi(x) &= \max_{y \in Y} \psi(x, y), \\ \phi_N(x) &= \max_{j \in Q_N} f^{N_j}(x), \end{aligned} \tag{5}$$

and with this notation, the problems $P_{Y_N}$ assume the form

$$(P_N) \quad \min_{x \in R^n} \phi_N(x). \tag{6}$$

The optimality functions $\rho_N$ for the problems $P_N$ are defined by

$$\begin{aligned} \rho_N(x) \triangleq -\min_{\lambda \in \Sigma} \Bigg\{ & \sum_{j \in Q_N} \lambda^j \left[\phi_N(x) - f^{N_j}(x)\right] \\ & + \frac{1}{2} \left\| \sum_{j \in Q_N} \lambda^j \nabla f^{N_j}(x) \right\|^2 \Bigg\}, \end{aligned} \tag{7}$$

where $\Sigma \triangleq \{\lambda \in R^{Q_N} \mid \lambda^j \geq 0, \sum_{j \in Q_N} \lambda^j = 1\}$.

In [1], we find the following result.

**Theorem 2.** *Suppose that $\hat{x}$ is an optimal solution to problems $P_N$; then, $\rho_N(\hat{x}) = 0$.*

The corresponding optimality function for $P$ is defined (see Theorem 3.1.6 in [1]) by

$$\rho(x) = -\min_{\eta \in \overline{F}\phi(x)} \left\{ \eta^0 + \frac{1}{2}\|\eta\|^2 \right\}, \tag{8}$$

where

$$\overline{F}\phi(x) = \operatorname{conv}_{y \in Y} \left\{ \begin{pmatrix} \phi(x) - \psi(x, y) \\ \nabla_x \psi(x, y) \end{pmatrix} \right\}. \tag{9}$$

Referring to Lemma 3.4.3 in [1], we see that, under Assumption 1, the following result must hold.

**Theorem 3.** *Suppose that $\{x_N\}_{N=0}^{\infty}$ is a sequence in $R^n$ converging to a point $\hat{x}$. Then, $\phi_N(x_N) \to \phi(\hat{x})$ and $\rho_N(x_N) \to \rho(\hat{x})$ as $N \to \infty$.*

In this paper, we consider to approximate uniformly $\phi_N(x)$ by the smooth spline introduced in [17].

Let us first recall the formulation of multivariate spline. Let $D$ be a polyhedral domain of $R^{q_N}$ which is partitioned with irreducible algebraic surfaces into cells $\Delta = \{\Delta_i \mid i = 1, \ldots, N\}$. A function $s(z)$ defined on $D$ is called a $k$-spline function with $r$th order smoothness, expressed for short as $s(z) \in S_k^r(D, \Delta)$, if $s(z) \in C^r(D)$ and $s(z)|_{\Delta_i} = p_i \in P_k$, where $P_k$ is the set of all polynomials of degree $k$ or less in $q_N$ variables. Similar to the smooth spline which approximates uniformly $\min\{z_1, z_2, \ldots, z_{q_N}\}$ given in [17], we can construct a spline function $s_3^2(z; \varepsilon) \in S_3^2(R^{q_N}, \Delta_{\mathrm{MS}}^2)$ to approximate uniformly $\max\{z_1, z_2, \ldots, z_{q_N}\}$ (as $\varepsilon \to +0$), where $\Delta_{\mathrm{MS}}^2$ is the homogenous Morgan-Scott partition of type two in [17], as follows:

$$\begin{aligned} & s_3^2\left(z_1, z_2, \ldots, z_{q_N}; \varepsilon\right) = z_{i_1} \\ & \quad + \sum_{l=1}^{k-1} c_l \left( l z_{i_{l+1}} - \sum_{j=1}^{l} z_{i_j} + \varepsilon \right)^3, \\ & \quad \text{for } z \in \Delta_{i_1, \ldots, i_k}(\varepsilon), \end{aligned} \tag{10}$$

where $c_1 = 1/(6\varepsilon^2), c_k/c_{k+1} = (k + 2)/k, 1 \leq k \leq q_N$, and the cell $\Delta_{i_1, \ldots, i_k}(\varepsilon)$ is the region defined by the following inequalities:

$$\begin{aligned} & z_{i_l} - z_{i_{l+1}} \geq 0, \quad \text{when } 1 \leq l < k, \\ & (k - 1) z_{i_k} - \sum_{j=1}^{k-1} z_{i_j} + \varepsilon \geq 0, \\ & k z_{i_l} - \sum_{j=1}^{k} z_{i_j} + \varepsilon \leq 0, \quad \text{when } k + 1 \leq l \leq q_N. \end{aligned} \tag{11}$$

The composite function $\gamma_{q_N, t_N}(x)$ approximates uniformly $\phi_N(x)$ as $t_N \to +0$, where

$$\begin{aligned} & \gamma_{q_N, t_N}(x) = s_3^2\left(f^1(x), f^2(x), \ldots, f^{q_N}(x); t_N\right), \\ & \text{for } x \text{ such that} \\ & \left(f^1(x), f^2(x), \ldots, f^{q_N}(x)\right) \in \Delta_{i_1, \ldots, i_k}(t_N). \end{aligned} \tag{12}$$

**Proposition 4.** *Suppose that $t_N > 0$. For any $x \in R^n$, we define*

$$\widehat{\Omega}(x) \triangleq \left\{ j \in Q_N \mid f^{N_j}(x) = \phi_N(x) \right\}. \tag{13}$$

*If the function $f^j(\cdot)$ is continuous, then $\gamma_{q_N,t_N}(\cdot)$ is continuous and is increasing with respect to $t_N$. Furthermore,*

$$\begin{aligned}\phi_N(x) + \frac{t_N}{3}\left(1 - \frac{1}{\left|\widehat{\Omega}(x)\right|}\right) &\le \gamma_{q_N,t_N}(x) \\ &\le \phi_N(x) + \frac{t_N}{3}\left(1 - \frac{1}{k}\right),\end{aligned} \tag{14}$$

*where $|\widehat{\Omega}(x)|$ denotes the cardinality of $\widehat{\Omega}(x)$.*

*Proof.* $s_3^2(z_1, z_2, \ldots, z_{q_N}; \varepsilon)$ is twice continuously differentiable and if the functions $f^j(\cdot)$, $j \in Q_N$ are continuous, it is easy to know that $\gamma_{q_N,t_N}(\cdot)$ is continuous. From Lemma 1.1 in [18], we know $\gamma_{q_N,t_N}(\cdot)$ is increasing with respect to $t_N$.

According to (12), we have

$$\begin{aligned}&\gamma_{q_N,t_N}(x) \\ &= f^{N_{i_1}}(x) + c_1\left(f^{N_{i_1}}(x) - f^{N_{i_2}}(x) + t_N\right)^3 \\ &\quad + \cdots + c_{|\widehat{\Omega}(x)|}\left(\left|\widehat{\Omega}(x)\right| f^{N_{i_{|\widehat{\Omega}(x)|+1}}}(x)\right. \\ &\qquad\qquad \left. - \left(f^{N_{i_1}}(x) + \cdots + f^{N_{i_{|\widehat{\Omega}(x)|}}}(x)\right) + t_N\right)^3 \\ &\quad + \cdots + c_{k-1}\left((k-1) f^{N_{i_k}}(x)\right. \\ &\qquad\qquad \left. - \left(f^{N_{i_1}}(x) + \cdots + f^{N_{i_{k-1}}}(x)\right) + t_N\right)^3.\end{aligned} \tag{15}$$

From the definition of $\widehat{\Omega}(x)$, we know $(l-1)f^{N_{i_l}}(x) - (f^{N_{i_1}}(x) + \cdots + f^{N_{i_{l-1}}}(x)) + t_N = t_N$, $1 \le l \le |\widehat{\Omega}(x)|$. From the definition of $\Delta_{i_1,\ldots,i_k}(t_N)$, we know $t_N \ge (l-1)f^{N_{i_l}}(x) - (f^{N_{i_1}}(x) + \cdots + f^{N_{i_{l-1}}}(x)) + t_N \ge 0$, $1 \le l < k$. Then, we have

$$f^{N_{i_1}}(x) + \sum_{l=1}^{|\widehat{\Omega}(x)|-1} c_l t_N^3 \le \gamma_{q_N,t_N}(x) \le f^{N_{i_1}}(x) + \sum_{l=1}^{k-1} c_l t_N^3. \tag{16}$$

By $c_l/c_{l+1} = (l+2)/l$, we can obtain $c_l = (2/(l(l+1)))c_1 = 1/(3l(l+1)t_N^2)$. Thus $\sum_{l=1}^{|\widehat{\Omega}(x)|-1} c_l t_N^3 = (t_N/3)(1 - (1/|\widehat{\Omega}(x)|))$ and $\sum_{l=1}^{k-1} c_l t_N^3 = (t_N/3)(1 - (1/k))$ and $f^{N_{i_1}}(x) = \phi_N(x)$. Hence the desired result follows. □

The following proposition is proved in [19].

**Proposition 5.** *(1) If all the functions $f^{N_j}(\cdot)$, $j \in Q_N$ are continuously differentiable, then $\gamma_{q_N,t_N}(\cdot)$ is continuously differentiable and*

$$\begin{aligned}\nabla_x \gamma_{q_N,t_N}(x) &= \sum_{i=1}^{q_N} \lambda_{q_N,t_N}^{i}(x)\nabla_x f^{N_i}(x) \\ &= \sum_{j=1}^{k} \lambda_{q_N,t_N}^{i_j}(x)\nabla_x f^{N_{i_j}}(x),\end{aligned} \tag{17}$$

*where*

$$\begin{aligned}&\lambda_{q_N,t_N}^{i_j}(x) \\ &= \begin{cases} 1 - 3\sum_{l=1}^{k-1} c_l\left(h_l(x,t_N)\right)^2, & \text{for } j = 1, \\ 3(j-1)c_{j-1}\left(h_{j-1}(x,t_N)\right)^2 \\ \quad -3\sum_{l=j}^{k-1} c_l\left(h_l(x,t_N)\right)^2, & \text{for } 2 \le j < k, \\ 3(k-1)c_{k-1}\left(h_{k-1}(x,t_N)\right)^2, & \text{for } j = k, \\ 0, & \text{for } k < j \le q_N, \end{cases}\end{aligned} \tag{18}$$

*and $h_l(x,t_N) = l f^{N_{i_{l+1}}}(x) - \sum_{r=1}^{l} f^{N_{i_r}}(x) + t_N$.*

*(2) For any $x \in R^n$ and $t_N > 0$, $\lambda_{q_N,t_N}^{i_j}(x) \in [0,1)$, $i_j \in Q_N$ and $\sum_{i_j \in Q_N} \lambda_{q_N,t_N}^{i_j}(x) = 1$.*

*(3) If all the functions, $f^{N_j}(\cdot)$, $j \in Q_N$, are twice continuously differentiable, then $\gamma_{q_N,t_N}(\cdot)$ is twice continuously differentiable and*

$$\begin{aligned}\nabla_{xx}^2 \gamma_{q_N,t_N}(x) &= \sum_{j=1}^{k} \lambda_{q_N,t_N}^{i_j}(x)\nabla_{xx}^2 f^{N_{i_j}}(x) \\ &\quad + \sum_{j=1}^{k}\left(\sum_{\breve{j}=1}^{k} \lambda_{q_N,t_N}^{i_{j,\breve{j}}}(x)\nabla_x f^{N_{i_{\breve{j}}}}(x)\right) \\ &\quad \times \left(\nabla_x f^{N_{i_j}}(x)\right)^T,\end{aligned} \tag{19}$$

*where*

$$\begin{aligned}&\lambda_{q_N,t_N}^{i_{1,\breve{j}}}(x) \\ &= \begin{cases} 6\sum_{l=1}^{k-1} c_l\left(h_l(x,t_N)\right) & \text{when } \breve{j} = 1, \\ -6(\breve{j}-1)c_{\breve{j}-1}\left(h_{\breve{j}-1}(x,t_N)\right) \\ +6\sum_{l=\breve{j}}^{k-1} c_l\left(h_l(x,t_N)\right) & \text{when } 2 \le \breve{j} < k, \\ -6(k-1)c_{k-1}\left(h_{k-1}(x,t_N)\right) & \text{when } \breve{j} = k, \end{cases} \\ &\lambda_{q_N,t_N}^{i_{j,\breve{j}}}(x) \\ &= \begin{cases} -6(j-1)c_{j-1}\left(h_{j-1}(x,t_N)\right) \\ +6\sum_{l=j}^{k-1} c_l\left(h_l(x,t_N)\right) & \text{when } 1 \le \breve{j} < j, \\ -6(\breve{j}-1)c_{\breve{j}-1}\left(h_{\breve{j}-1}(x,t_N)\right) \\ +6\sum_{l=\breve{j}}^{k-1} c_l\left(h_l(x,t_N)\right) & \text{when } j < \breve{j} < k, \\ 6(\breve{j}-1)^2 c_{\breve{j}-1}\left(h_{\breve{j}-1}(x,t_N)\right) \\ +6\sum_{l=\breve{j}}^{k-1} c_l\left(h_l(x,t_N)\right) & \text{when } \breve{j} = j, \\ -6(k-1)c_{k-1}\left(h_{k-1}(x,t_N)\right) \\ \qquad \text{when } \breve{j} = k, \text{ for } 2 \le j < k, \end{cases}\end{aligned}$$

$$\lambda^{i_{k,\breve{j}}}_{q_N,t_N}(x) = \begin{cases} -6(k-1)\,c_{k-1}\left(h_{k-1}(x,t_N)\right) & when\ 1 \le \breve{j} < k, \\ 6(k-1)^2 c_{k-1}\left(h_{k-1}(x,t_N)\right) & when\ \breve{j} = k. \end{cases} \tag{20}$$

From Proposition 5, we obtain the following results.

**Corollary 6.** *For any $x \in R^n$ and $i_j \in Q_N$,*

$$\lim_{t_N\to 0}\lambda^{i_j}_{q_N,t_N}(x) = \begin{cases} \dfrac{1}{\left|\widehat{\Omega}(x)\right|}, & if\ i_j \in \widehat{\Omega}(x), \\ 0, & if\ i_j \notin \widehat{\Omega}(x), \end{cases} \tag{21}$$

$$\lim_{t_N\to 0}\nabla_x \gamma_{q_N,t_N}(x) = \sum_{j\in\widehat{\Omega}(x)} \frac{1}{\left|\widehat{\Omega}(x)\right|}\nabla_x f^j(x).$$

*Proof.* According to the definition of $\Delta_{i_1,\dots,i_k}(t_N)$, we know that $k \to |\widehat{\Omega}(x)|$ as $t_N \to 0$. From Proposition 5 (1), we know

$$\begin{aligned} \lim_{t_N\to 0}\lambda^{i_1}_{q_N,t_N}(x) &= 1 - 3\sum_{l=1}^{|\widehat{\Omega}(x)|-1} c_l t_N^2 \\ &= 1 - 3\sum_{l=1}^{|\widehat{\Omega}(x)|-1} \frac{1}{3l(l+1)t_N^2}t_N^2 = \frac{1}{\left|\widehat{\Omega}(x)\right|}. \end{aligned} \tag{22}$$

If $i_j \in \widehat{\Omega}(x)$, $(1 < j < k)$, then

$$\begin{aligned} \lim_{t_N\to 0}\lambda^{i_j}_{q_N,t_N}(x) &= 3(j-1)\,c_{j-1}t_N^2 - 3\sum_{l=j}^{|\widehat{\Omega}(x)|-1} c_l t_N^2 \\ &= 3(j-1)\frac{1}{3(j-1)jt_N^2}t_N^2 \\ &\quad - 3\sum_{l=j}^{|\widehat{\Omega}(x)|-1}\frac{1}{3t_N^2}\left(\frac{1}{l}-\frac{1}{l+1}\right)t_N^2 = \frac{1}{\left|\widehat{\Omega}(x)\right|}. \end{aligned} \tag{23}$$

If $i_k \in \widehat{\Omega}(x)$, then $k = |\widehat{\Omega}(x)|$. That is

$$\lim_{t_N\to 0}\lambda^{i_k}_{q_N,t_N}(x) = 3\left(\left|\widehat{\Omega}(x)\right|-1\right)c_{|\widehat{\Omega}(x)|-1}t_N^2 = \frac{1}{\left|\widehat{\Omega}(x)\right|}. \tag{24}$$

Then, $\lim_{t_N\to 0}\lambda^{i_j}_{q_N,t_N}(x) = 1/|\widehat{\Omega}(x)|$, $i_j \in \widehat{\Omega}(x)$.

If $i_j \notin \widehat{\Omega}(x)$, $(1 < j < k)$, then $\lim_{t_N\to 0}\lambda^{i_j}_{q_N,t_N}(x) = 0$.

It now follows from (21) and (17) that (21) holds. The proof is completed. □

Next, let $\{t_N\}_{N=1}^{\infty}$ be an infinite sequence such that $t_N \to 0$, as $N \to \infty$, and consider the sequence of approximating problems

$$\left(P_{q_N,t_N}\right)\min_{x\in R^n}\gamma_{q_N,t_N}(x), \quad N = 1,2,3,\dots, \tag{25}$$

with $\gamma_{q_N,t_N}(x)$ defined as in (12).

**Theorem 7.** *The problems $(P_{q_N,t_N})$ epiconverge to the problem $P$.*

*Proof.* Referring to Theorem 3.3.2 in [1], we see that to prove the theorem it is sufficient to show that if $\{x_N\}_{N=1}^{\infty}$ is a sequence in $R^n$ converging to a point $\widehat{x}$, then $\gamma_{q_N,t_N}(x) \to \phi(\widehat{x})$.

Thus, suppose that $x_N \to \widehat{x}$ as $N \to \infty$. Then,

$$\begin{aligned} &\left|\gamma_{q_N,t_N}(x_N) - \phi(\widehat{x})\right| \\ &\quad \le \left|\gamma_{q_N,t_N}(x_N) - \phi_N(x_N)\right| + \left|\phi_N(x_N) - \phi(\widehat{x})\right|. \end{aligned} \tag{26}$$

Now, by Theorem 2, $\phi_N(x_N) - \phi(\widehat{x}) \to 0$ as $N \to \infty$, and, since by assumption of $t_N \to 0$, as $N \to \infty$, it follows from (14) that $\gamma_{q_N,t_N}(x_N) - \phi_N(x_N) \to 0$ as $N \to \infty$, which completes our proof. □

## 2. Spline Smoothing Newton Method and Its Convergence

We combine Algorithm 3.1 in [18] with a discretization precision test to produce an algorithm for solving the semi-infinite minimax problems $P$. The Hessian $\nabla^2_{xx}\gamma_{q_N,t_N}(x)$ of the smoothing spline function $\gamma_{q_N,t_N}(x)$ can be modified by adding a multiple of the identity introduced in [20]; that is,

$$B_{q_N,t_N}(x) = \theta(x)\,I + \nabla^2_{xx}\gamma_{q_N,t_N}(x), \tag{27}$$

where $\theta(x) = \max\{0, \vartheta - e(x)\}$ with $e(x)$ denoting the minimum eigenvalue of $\nabla^2_{xx}\gamma_{q_N,t_N}(x)$ and $\vartheta > 0$.

*Algorithm 8.*

*Data.* Given $x^{(0)} \in R^n$, a monotone increasing sequence of sets $\{Y_N\}_{N=0}^{\infty}$, $Y_N = \{y_1, y_2, \dots, y_{q_N}\}$, of cardinality $q_N$, with $q_N \to \infty$ as $N \to \infty$, satisfying Assumption 1, and defining the functions $f^{N_j}(x) = \psi(x, y_{N_j})$, $j = 1,2,\dots,q_N$, a sequence of monotone decreasing parameters $\{\pi_N\}_{N=0}^{\infty} > 0$, such that $\pi_N \to 0$ as $N \to \infty$, $\vartheta > 0$, and $\delta > 0$. Functions $\epsilon_a(t), \epsilon_b(t), \tau(t)$: $(0,\infty) \to (0,\infty)$, satisfying $\epsilon_b(t) \ge \epsilon_a(t) > \tau(t)$, for all $t > 0$, $\widehat{t} \gg 1$, $\kappa_1 \in (0,1)$, $\kappa_2 \gg 1$, $0 < \kappa_3 \ll 1$.

*Parameter.* Set $\alpha, \beta \in (0,1)$.

*Step 0.* Set $N = 0$.

*Step 1.* Set $i = 0$, $t_0 = \pi_N$, $k = 0$, $s = 1$, $l = 9$, $x^{k,i} = x^{(0)}$.

*Step 2.* Let $\overline{I} = \{j \mid \max\{f^{N_1}(x^{k,i}),\dots,f^{N_{q_N}}(x^{k,i})\} - f^{N_j}(x^{k,i}) < t_k\}$; let $\overline{k}$ be the cardinality of $\overline{I}$, and $\overline{I} = \{i_1, i_2, \dots, i_{\overline{k}}\}$. Range

$\{f^{N_{i_j}}(x^{k,i})\}_{j=1}^{\overline{k}}$ according to $f^{N_{i_1}}(x^{k,i}) \geq f^{N_{i_2}}(x^{k,i}) \geq \cdots \geq f^{N_{i_{\overline{k}}}}(x^{k,i})$.

If $\overline{k} = 1$, the cell is $\Delta_{i_1}(t_k)$.

Else if $(\widetilde{k} - 1)f^{N_{i_{\widetilde{k}}}}(x^{(k,i)}) - \sum_{j=1}^{\widetilde{k}-1} f^{N_{i_j}}(x^{(k,i)}) + t_k \geq 0$, for every $\widetilde{k} \in \{\overline{k}, \overline{k}-1, \ldots, 2\}$, we have $\widetilde{k} \in I \subseteq \{\overline{k}, \overline{k}-1, \ldots, 2\}$. Let $\widehat{k}$ be the maximum element of $I$; then the cell is $\Delta_{i_1,\ldots,i_{\widehat{k}}}(t_k)$.

*Step 3*. Compute $\nabla_x \gamma_{q_N,t_k}(x^{k,i})$. If $\|\nabla_x \gamma_{q_N,t_k}(x^{k,i})\|^2 > \tau(t_k)$, go to Step 4. Else go to Step 9.

*Step 4*. Compute $B_{q_N,t_k}(x^{k,i})$ according to (27); then compute Cholesky factor $R$ such that $B_{q_N,t_k}(x^{k,i}) = RR^T$ and the reciprocal condition number $c(R)$ of $R$.

If $c(R) \geq \kappa_1$ and $t_k \geq \kappa_3$, go to Step 5.

Else if $c(R) \geq \kappa_1$ and the largest eigenvalue $\sigma_{t_k,\max}(x^{k,i})$ of $B_{q_N,t_k}(x^{k,i})$ satisfies $\sigma_{t_k,\max}(x^{k,i}) \leq \kappa_2$, go to Step 5.

Else go to Step 6.

*Step 5*. Compute the search direction

$$h_{k,i} = -B_{q_N,t_k}\left(x^{k,i}\right)^{-1} \nabla_x \gamma_{q_N,t_k}\left(x^{k,i}\right); \tag{28}$$

go to Step 7.

*Step 6*. Compute the search direction

$$h_{k,i} = -\nabla_x \gamma_{q_N,t_k}\left(x^{k,i}\right). \tag{29}$$

*Step 7*. Compute the step length $\lambda_{k,i} = \beta^l$, where $l \geq 0$ is the smallest integer satisfying

$$\begin{aligned} &\gamma_{q_N,t_k}\left(x^{k,i} + \beta^l h_{k,i}\right) - \gamma_{q_N,t_k}\left(x^{k,i}\right) \\ &\quad \leq \alpha\beta^l \left\langle \nabla_x \gamma_{q_N,t_k}\left(x^{k,i}\right), h_{k,i} \right\rangle . \end{aligned} \tag{30}$$

*Step 8*. Set $x^{k,i+1} = x^{k,i} + \lambda_{k,i} h_{k,i}$, $i = i + 1$. Go to Step 2.

*Step 9*. If $s = 1$, compute $t^*$ such that

$$\epsilon_a\left(t_k\right) \leq \left\|\nabla_x \gamma_{q_N,t^*}\left(x^{k,i}\right)\right\|^2 \leq \epsilon_b\left(t_k\right); \tag{31}$$

go to Step 10.

Else set $t_{k+1} = 1/s(k+2)$, $k = k+1$, and $i = 0$; go to Step 2.

*Step 10*. If $t^* \geq \widehat{t}$, set $t_{k+1} = \min\{t^*, t_k/(t_k+1)\}$, $k = k+1$, and $i = 0$; go to Step 2. Else set $s = \max\{2, ((1/\widehat{t}) + 2)/(k+1)\}$, $t_{k+1} = 1/s(k+2)$, $k = k+1$, and $i = 0$; go to Step 2.

*Step 11*. If

$$\begin{aligned} &\frac{1}{2}\left\|\nabla_x \gamma_{q_N,t_k}(x^{k,i})\right\|^2 \leq \delta, \\ &\sum_{j=1}^{q_N} \lambda^j_{q_N,t_k}\left(x^{k,i}\right)\left[\phi_N\left(x^{k,i}\right) - f^{N_j}\left(x^{k,i}\right)\right] \leq \delta, \end{aligned} \tag{32}$$

where the $\lambda^j_{q_N,t_k}(x^{k,i})$ are defined by (17), set $\xi_N = x^{k,i}$, $t_N = t_k$, replace $N$ by $N+1$, $\delta$ by $\delta/2$, and go to Step 1.

Else go to Step 1.

**Theorem 9.** *Suppose that $\{\xi_N\}_{N=0}^{\infty}$ is a sequence constructed by Algorithm 8. Then, any accumulation point $\widehat{\xi}$ of this sequence satisfies $\theta(\widehat{\xi}) = 0$.*

*Proof.* First note that it follows from Corollary 6 that condition (32) will be eventually satisfied, since Algorithm 8 keeps abating $t_k$. Next, note that

$$\begin{aligned} &-\sum_{j=1}^{q_N} \lambda^j_{q_N,t_N}\left[\phi_N\left(\xi_N\right) - f^{N_j}\left(\xi_N\right)\right] \\ &\quad - \frac{1}{2}\left\|\nabla_x \gamma_{q_N,t_N}(\xi_N)\right\|^2 \leq \rho_N\left(\xi_N\right) \leq 0. \end{aligned} \tag{33}$$

Since by construction

$$\begin{aligned} &-\sum_{j=1}^{q_N} \lambda^j_{q_N,t_N}\left[\phi_N\left(\xi_N\right) - f^{N_j}\left(\xi_N\right)\right] \\ &\quad - \frac{1}{2}\left\|\nabla_x \gamma_{q_N,t_N}\left(\xi_N\right)\right\|^2 \longrightarrow 0, \quad \text{as } N \longrightarrow \infty, \end{aligned} \tag{34}$$

and $\rho_N(\xi_N) \to^{k} \rho(\widehat{\xi})$ on any infinite subsequence $\{\xi_N\}_{N\in K}$ that converges to $\widehat{\xi}$, the desired result follows. □

## 3. Numerical Experiment

We have implemented Algorithm 8 using Matlab. In order to show the efficiency of the algorithm, we also have implemented algorithm in [16] (denote PWY) using similar procedures. Algorithm PWY was proposed by Polak et al. in [16], which has been introduced in Section 1.

The test results were obtained by running Matlab R2011a on a desktop with Windows XP Professional operation system, Intel(R) Core (TM) i3-370 2.40 GHz processor, and 2.92 GB of memory.

In Algorithm 8, parameters are chosen as follows: $\alpha = 0.8$, $\beta = 0.77$, $\widehat{t} = 10^5 \ln q_N$, $\kappa_1 = 10^{-7}$, $\kappa_2 = 10^{30}$, $\kappa_3 = 1000\,\widehat{t}$, $\tau(t) = \min\{0.1, 1000t/(N+1)^2\}$, $(\epsilon_a, \epsilon_b) = (0.01, 0.2)$, $\vartheta = 0.1$, $\delta = 1$, $q_N = 100 \cdot 2^N + 1$, and $\pi_N = 1/(N^2 + (1/p_N))$. In the PWY algorithm, parameters are chosen as follows: $\varepsilon = 0.1$, $\tau(p) = \min\{0.1, 1000/((N+1)^2 p)\}$. The results are listed in Tables 1, 2, 3, 4, and 5. $x^*$ denotes the final approximate solution point and $\phi^*$ is the value of the objective function of discretized problems $P_{Y_N}$ at $x^*$. $m$ is the maximum number of discrete points. Time is the CPU time in seconds.

*Example 1* (see [21]). Let $x = (x_1, x_2, x_3, x_4, x_5, x_6) \in R^6$

$$\begin{aligned} \phi\left(x, y\right) = {} & x_1^2 \exp\left(-x_2 y\right) \cos^2\left(x_3 y + x_4\right) - \cos\left(y\right) \\ & + x_2^2 x_3^2 \exp\left(-x_1 y\right) \sin^2\left(x_2 y\right) \\ & + \exp\left(\left(1 - x_6\right)^2 y\right) + x_5^2, \\ & Y = [0, 10] . \end{aligned} \tag{35}$$

Table 1: Test results for Example 1.

| $m$ | Method | $\phi^*$ | Time | $x^*$ |
|---|---|---|---|---|
| 1638401 | Algorithm 8 | 2.000062 | 9.4203 | (1.1065, 1.0071, 1.1483, 1.0785, 0.0022, 1.0000) |
| 819201 | PWY | 2.000056 | 51.4509 | (0.9586, 1.0027, 1.1604, 1.0501, −0.0023, 0.9999) |

Table 2: Test results for Example 2.

| $m$ | Method | $\phi^*$ | Time | $x^*$ |
|---|---|---|---|---|
| 1638401 | Algorithm 8 | 0.948184 | 5.8026 | (−4.9016, 0.0002, 1.0046, −0.1800) |
| 1638401 | PWY | 0.948046 | 14.8421 | (−4.7261, −0.0000, 1.0000, −0.1472) |

Table 3: Test results for Example 3.

| $m$ | Method | $\phi^*$ | Time | $x^*$ |
|---|---|---|---|---|
| 819201 | Algorithm 8 | 4.999999 | 1.5466 | (2.0363, 1.9958, 0.0095, 1.5724) |
| 819201 | PWY | 5.000483 | 18.2670 | (1.4796, 1.3277, 0.62892, 1.5704) |

Table 4: Test results for Example 4.

| $m$ | Method | $\phi^*$ | Time | $x^*$ |
|---|---|---|---|---|
| 6553601 | Algorithm 8 | −0.999999 | 3.6721 | (0.0000, −44.0111) |
| 3276801 | PWY | −0.999986 | 86.8183 | (−0.0036, −44.0000) |

Table 5: Test results for Example 5.

| $m$ | Method | $\phi^*$ | Time | $x^*$ |
|---|---|---|---|---|
| 2560000 | Algorithm 8 | 1.828804 | 2.8550 | (0.5548, 0.6475, 0.5946, 0.7662) |
| 640000 | PWY | 1.828698 | 4.1854 | (0.5556, 0.6497, 0.5971, 0.7684) |

*Example 2* (see [21]). Let $x = (x_1, x_2, x_3, x_4) \in R^4$

$$\begin{aligned} \phi(x, y) &= \left(x_1 + \sqrt{x_1^2 + \cos(y) + 4}\right) \\ &\quad \times \ln\left(\sin y + \exp\left(\cos x_1 + x_2^2\right)\right) \\ &\quad + \left(\exp(-y) - \frac{x_3 + y x_4}{1 + x_3^2}\right)^2, \\ Y &= [0, 10]. \end{aligned} \tag{36}$$

*Example 3.* Let $x = (x_1, x_2, x_3, x_4) \in R^4$

$$\begin{aligned} \phi(x, y) &= x_2^2 x_3^2 \exp(-x_1 y) \sin^2(x_2 y) \\ &\quad - x_1 \cos(y) - x_4 \sin(y) + 5, \\ Y &= [0, 2\pi]. \end{aligned} \tag{37}$$

*Example 4.* Let $x = (x_1, x_2) \in R^2$

$$\begin{aligned} \phi(x, y) &= x_1^2 + 2x_1 y^2 + e^{x_1 + x_2} - e^y, \\ Y &= [0, 10]. \end{aligned} \tag{38}$$

*Example 5.* Let $x = (x_1, x_2, x_3, x_4) \in R^4$

$$\begin{aligned} \phi(x, y) &= \frac{(y_1 - x_1)^2}{x_3^2} + \frac{(y_2 - x_2)^2}{x_4^2 - 1}, \\ Y &= [0, 1] \times [0, 1]. \end{aligned} \tag{39}$$

## 4. Conclusion

We have developed a spline smoothing Newton method for the solution of semi-infinite minimax problems using smooth cubic spline and discretization strategy. At each iteration, only few components in the max function are computed; hence, the computation cost is greatly reduced. For semi-infinite minimax problems with complicated component functions, numerical tests show that the new method is very efficient.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (11171051, 91230103) and the Fundamental Research Funds for the Central Universities (DC13010214) and the General Project of the Education Department of Liaoning Province (0908-330006).

## References

[1] E. Polak, *Optimization: Algorithms and Consistent Approximations*, Springer, New York, NY, USA, 1997.

[2] J. W. Bandler, P. C. Liu, and H. Tromp, "A nonlinear programming approach to optimal design centering, tolerancing, and tuning," *IEEE Transactions on Circuits and Systems*, vol. 23, no. 3, pp. 155–165, 1976.

[3] T. Chen and M. K. H. Fan, "On convex formulation of the floorplan area minimization problem," in *Proceedings of the International Symposium on Physical Design (ISPD '98)*, pp. 124–128, Academic Press, Monterey, Calif, USA, April 1998.

[4] A. V. Fiacco and K. O. Kortanek, *Semi-Infinite Programming and Applications*, Lecture Notes in Economics and Mathematical Systems, Springer, Berlin, Germany, 1983.

[5] R. Hettich and K. O. Kortanek, "Semi-infinite programming: theory, methods, and applications," *SIAM Review*, vol. 35, no. 3, pp. 380–429, 1993.

[6] A. A. Kassim and B. V. K. Vijaya Kumar, "Path planning for autonomous robots using neural networks," *Journal of Intelligent Systems*, vol. 7, no. 1-2, pp. 33–55, 1997.

[7] E. Polak, "On the use of consistent approximations in the solution of semi-infinite optimization and optimal control problems," *Mathematical Programming*, vol. 62, no. 2, pp. 385–414, 1993.

[8] E. Polak and J. O. Royset, "Algorithms for finite and semi-infinite min-max-min problems using adaptive smoothing techniques," *Journal of Optimization Theory and Applications*, vol. 119, no. 3, pp. 421–457, 2003.

[9] Z. Guanglu, C. Wang, and Q. Sun, "A globally convergent method for semi-infinite minimax problems," *Organ Transplantation*, vol. 2, no. 2, p. 42, 1998.

[10] G. Zhou, C. Wang, and Y. Zhang, "An efficient method for semi-infinite programming," *Mathematics Numerical Sinica*, vol. 21, no. 1, pp. 1–8, 1999.

[11] E. Polak, D. Q. Mayne, and J. E. Higgins, "On the extension of Newton's method to semi-infinite minimax problems," *SIAM Journal on Control and Optimization*, vol. 30, no. 2, pp. 367–389, 1992.

[12] O. Yigui and Z. Qian, "A nonmonotonic trust region algorithm for a class of semi-infinite minimax programming," *Applied Mathematics and Computation*, vol. 215, no. 2, pp. 474–480, 2009.

[13] A. Auslender, M. A. Goberna, and M. A. López, "Penalty and smoothing methods for convex semi-infinite programming," *Mathematics of Operations Research*, vol. 34, no. 2, pp. 303–319, 2009.

[14] L. Zhang, S. Fang, and S. Wu, "An entropy based central cutting plane algorithm for convex min-max semi-infinite programming problems," *Science China Mathematics*, vol. 56, no. 1, pp. 201–211, 2013.

[15] E. Polak, J. O. Royset, and R. S. Womersley, "Algorithms with adaptive smoothing for finite minimax problems," *Journal of Optimization Theory and Applications*, vol. 119, no. 3, pp. 459–484, 2003.

[16] E. Polak, R. S. Womersley, and H. X. Yin, "An algorithm based on active sets and smoothing for discretized semi-infinite minimax problems," *Journal of Optimization Theory and Applications*, vol. 138, no. 2, pp. 311–328, 2008.

[17] G. Zhao, Z. Wang, and H. Mou, "Uniform approximation of min/max functions by smooth splines," *Journal of Computational and Applied Mathematics*, vol. 236, no. 5, pp. 699–703, 2011.

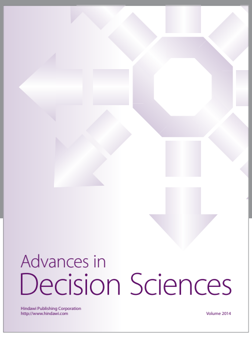
[18] L. Dong and B. Yu, "A spline smoothing Newton method for finite minimax problems," *Journal of Engineering Mathematics*. In press.

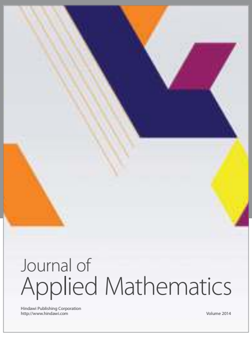
[19] L. Dong, B. Yu, and G. H. Zhao, "A smoothing spline homotopy method for nonconvex nonlinear programming," submitted.

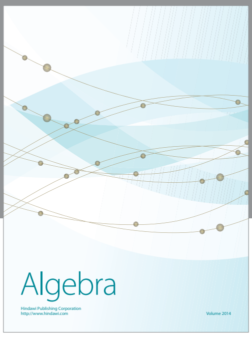
[20] J. Nocedal, S. J. Wright, P. Glynn, and S. M. Robinson, *Numerical Optimization*, Springer Series in Operations Research, Springer, New York, NY, USA, 1999.

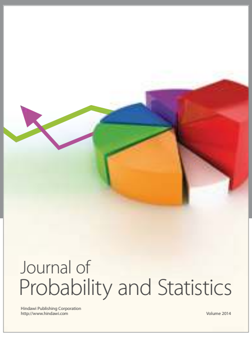
[21] X. Yu, *Truncated Aggregate Smoothing Algorithms*, School of Mathematical Sciences, Dalian University of Technology, Dalian, China, 2010.

Advances in
Operations Research

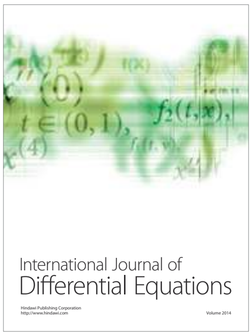

Advances in
Decision Sciences


Journal of
Applied Mathematics

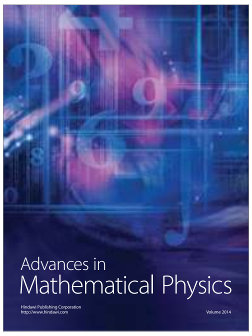

Algebra

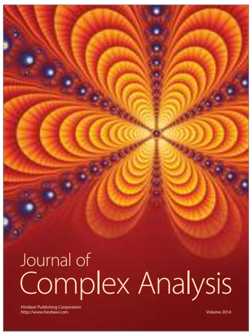

Journal of
Probability and Statistics

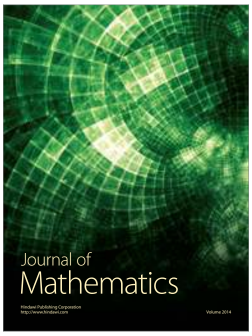
The Scientific
World Journal

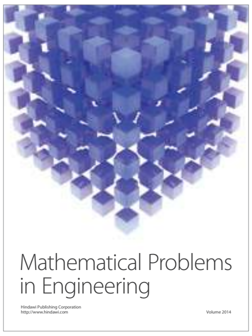

International Journal of
Differential Equations

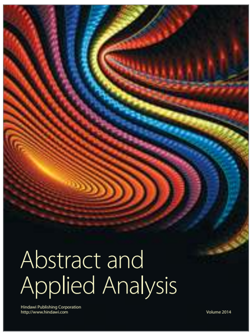
International Journal of
Combinatorics

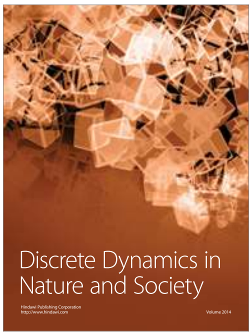

Advances in
Mathematical Physics

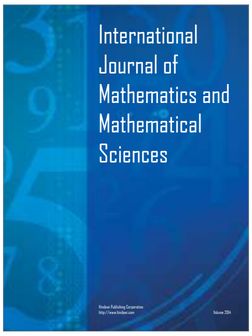

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

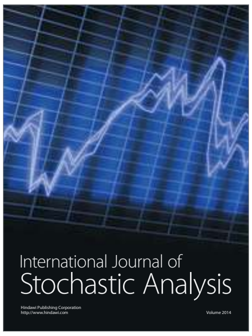

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization