Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 9108150, 25 pages
https://doi.org/10.1155/2020/9108150

Research Article

An Active Set Smoothing Method for Solving Unconstrained Minimax Problems

Zhengyong Zhou and Qi Yang

School of Mathematics and Computer Sciences, Shanxi Normal University, Linfen, Shanxi 041004, China

Correspondence should be addressed to Zhengyong Zhou; zzy198300@163.com

Received 14 February 2020; Accepted 5 June 2020; Published 24 June 2020

Academic Editor: Mohammad D. Aliyu

In this paper, an active set smoothing function based on the plus function is constructed for the maximum function. The active set strategy used in the smoothing function reduces the number of gradients and Hessians evaluations of the component functions in the optimization. Combing the active set smoothing function, a simple adjustment rule for the smoothing parameters, and an unconstrained minimization method, an active set smoothing method is proposed for solving unconstrained minimax problems. The active set smoothing function is continuously differentiable, and its gradient is locally Lipschitz continuous and strongly semismooth. Under the boundedness assumption on the level set of the objective function, the convergence of the proposed method is established. Numerical experiments show that the proposed method is feasible and efficient, particularly for the minimax problems with very many component functions.

1. Introduction

In this paper, we consider the following unconstrained minimax problem:

$$\min_{x\in R^n}\left\{F(x)=\max_{j\in Q} f_j(x)\right\}, \tag{1}$$

where the component functions $f_j\colon R^n \longrightarrow R$, $j\in Q=\{1,\ldots,q\}$, are twice continuously differentiable. Minimax problem (1) is a typical nonsmooth optimization problem and arises in many fields, such as engineering design ([1]), vehicle routing ([2, 3]), structural optimization ([4]), electronic circuit design ([5]), and game theory ([6, 7]).

Many methods have been proposed for solving minimax problem (1), such as subgradient methods ([8]), bundle type methods ([9, 10]), cutting plane methods ([11]), sequential quadratic programming methods ([12–14]), interior point methods ([15–17]), conjugate gradient methods ([18]), and smoothing methods ([19–26]). The main advantage of smoothing methods is that the minimax problem is transformed into a sequence of simple, smooth, and unconstrained optimization problems, which can be solved by standard unconstrained minimization solvers.

In [27], the following aggregate function (also called the exponential penalty function) induced from Jaynes' maximum entropy principle was introduced:

$$F_t(x)=t\ln\left(\sum_{j\in Q}\exp\left(\frac{f_j(x)}{t}\right)\right), \tag{2}$$

where $t>0$ is the smoothing parameter. It approaches $F(x)$ uniformly with respect to $x\in R^n$ as the smoothing parameter goes to 0, and has been wildly used in the smoothing methods for solving the minimax problems. Its gradient can be written as follows:

$$\nabla F_t(x)=\sum_{j\in Q}\mu_t^j(x)\nabla f_j(x), \tag{3}$$

with

$$\begin{aligned}\mu_t^j(x)&=\frac{\exp\left(f_j(x)/t\right)}{\sum_{j\in Q}\exp\left(f_j(x)/t\right)}\in(0,1],\\ \sum_{j\in Q}\mu_t^j(x)&=1,\end{aligned} \tag{4}$$

which is a convex combination of the gradients of all the component functions, and its Hessian

$$\nabla^2 F_t(x) = \sum_{j\in Q} \mu_t^j(x)\nabla^2 f_j(x) + \frac{1}{t}\sum_{j\in Q} \mu_t^j(x)\nabla f_j(x)\nabla f_j(x)^T - \frac{1}{t}\sum_{j\in Q} \mu_t^j(x)\nabla f_j(x)\sum_{j\in Q} \mu_t^j(x)\nabla f_j(x)^T, \tag{5}$$

is a complicated combination of the gradients and Hessians of all component functions. Therefore, for the maximum function with very many nonlinear component functions, the evaluations for the gradient and Hessian of the aggregate function always consume a large amount of computation.

For the minimax problems with very many component functions, several active set strategies have been developed for the smoothing methods to reduce the number of gradients or Hessians evaluations of the component functions at each iteration. In [18], the following active set smoothing function for $F(x,\alpha) = \alpha + \sum_{j\in Q}\max(f_j(x) - \alpha, 0)$ was presented:

$$F_t(x,\alpha) = \alpha + \sum_{j\in Q} \varphi\left(f_j(x) - \alpha; t\right), \tag{6}$$

where $t > 0$ is the smoothing parameter,

$$\varphi(z;t) = \begin{cases} 0, & z < -t, \\ \dfrac{(z+t)^3}{6t^2}, & -t \le z \le 0, \\ z + \dfrac{(z-t)^3}{6t^2}, & 0 < z \le t, \\ z, & z > t. \end{cases} \tag{7}$$

The active set used in $F_t(x,\alpha)$ at $(x,\alpha) \in R^n \times R$ can be written as follows:

$$Q_t(x,\alpha) = \left\{ j \in Q \mid f_j(x) - \alpha + t > 0 \right\}. \tag{8}$$

In [28], a cubic spline smoothing function for $F(x)$ was presented. For any smoothing parameter $t > 0$, the active set used in the cubic spline smoothing function at $x \in R^n$ can be represented as follows:

$$Q_t(x) = \left\{ j \in Q \mid F(x) - f_j(x) \le t \right\}. \tag{9}$$

In [25], an active set strategy for the aggregate function was introduced. For a given $\epsilon > 0$, let

$$Q_\epsilon(x) = \left\{ j \in Q \mid F(x) - f_j(x) \le \epsilon \right\}. \tag{10}$$

Then, the active set Q^k used for the aggregate function at $x^k \in R^n$ is updated as

$$\begin{aligned} Q^0 &= Q_\epsilon\left(x^0\right), \\ Q^k &= Q^{k-1} \cup Q_\epsilon\left(x^k\right), \quad k \ge 1. \end{aligned} \tag{11}$$

In [26], another active set strategy for the aggregate function was presented. For any smoothing parameter $t > 0$, the active set used for the aggregate function at $x \in R^n$ is defined as follows:

$$Q_t(x,q) = \left\{ j \in Q \mid F(x) - f_j(x) \le \epsilon(t,q) \right\}, \tag{12}$$

where $\epsilon(t,q) > 0$ is a complicated combination of several parameters.

In this paper, based on the plus function, an active set smoothing function for the maximum function is proposed, and the smoothing function only relates to a part of component functions, whose function values are close to $F(x)$. It is continuously differentiable, and its gradient is locally Lipschitz continuous and strongly semismooth. Combing the active set smoothing function, a geometric reduction rule for the smoothing parameters, the Armijo line search strategy, the steepest decent direction, and the Newton direction, an active set smoothing method is proposed for solving unconstrained minimax problems. Under the boundedness assumption on the level set of $F(x)$, the convergence of the active set smoothing method is established. Numerical experiments show that the resulting method is stable and efficient, especially for the minimax problems with very many component functions.

The following assumptions and results will be used in this paper:

Assumption 1: the component functions $f_j\colon R^n \longrightarrow R$, $j \in Q$, are twice continuously differentiable, and $\nabla f_j\colon R^n \longrightarrow R^n$, $j \in Q$, is strongly semismooth

Assumption 2: for any $M > 0$, the level set $\Omega_M = \{x \in R^n \mid F(x) \le M\}$ is bounded

Definition 1 (see [29]). Suppose that $\Phi\colon R^n \longrightarrow R^m$ is locally Lipschitz continuous, if for any $V \in \partial\Phi(x+h)$, $h \longrightarrow 0$,

$$Vh - \Phi'(x;h) = O\left(\|h\|^2\right), \tag{13}$$

where $\partial\Phi(x)$ is the generalized Jacobian of Φ at x, $\Phi_j'(x;h)$ is the directional derivative of Φ_j at x in the direction h for $j = 1,\ldots,m$, and then Φ is said to be strongly semismooth at x.

Lemma 1 (see [29]). *Suppose that* $\phi\colon R \longrightarrow R$ *and* $\psi\colon R \longrightarrow R$ *are strongly semismooth, then*

(i) For any a, $b \in R$, $a\phi + b\psi\colon R \longrightarrow R$ *is strongly semismooth*

(ii) $\phi \cdot \psi\colon R \longrightarrow R$ *is strongly semismooth*

(iii) If $|\psi(x)| \ge c$ *for the constant* $c > 0$, $\phi/\psi\colon R \longrightarrow R$ *is strongly semismooth*

(iv) $\phi^\circ\psi\colon R \longrightarrow R$ *is strongly semismooth*

Lemma 2 (see [29]). *Suppose that* $\Phi\colon R^n \longrightarrow R^m$ *is locally Lipschitz continuous, if all* $\Phi_i\colon R^n \longrightarrow R$, $i = 1,\ldots,m$, *are strongly semismooth, then* Φ *is strongly semismooth.*

Lemma 3 (see [30]). *For the function* $\Phi\colon R^n \longrightarrow R^m$, *if* Φ' *is locally Lipschitz continuous, then* Φ *is strongly semismooth.*

Theorem 1 (see [24]). *Suppose that the component functions f_j: $R^n \longrightarrow R$, $j \in Q$, are continuously differentiable. If x^* is a local minimizer of problem (1), then*

$$0 \in \partial F(x^*) := \mathrm{conv}_{j\in I(x^*)}\{\nabla f_j(x^*)\}, \tag{14}$$

where

$$I(x^*) := \{j \in Q \mid f_j(x^*) = F(x^*)\}, \tag{15}$$

and conv $\{A\}$ denotes the convex hull of A.

2. An Active Set Smoothing Function for the Maximum Function

In this section, based on the plus function z_+: $R \longrightarrow R_+$:

$$z_+ = \begin{cases} z, & z > 0, \\ 0, & z \le 0, \end{cases} \tag{16}$$

we construct the following function F_t^c: $R^n \times R \longrightarrow R$:

$$F_t^c(x,\alpha) = c\alpha + \frac{t}{2}\sum_{j\in Q}\left(1 + \frac{f_j(x) - c\alpha}{t}\right)_+^2, \tag{17}$$

where $t > 0$ is the smoothing parameter and $c > 0$ is the scaling parameter. By the definition of the plus function, Assumption 1, and $t > 0$, we have the following result.

Lemma 4. *For any $t > 0$ and $c > 0$, F_t^c: $R^n \times R \longrightarrow R$ is continuously differentiable.*

For any $t > 0$, $c > 0$, and $(x,\alpha) \in R^n \times R$, let

$$Q_t^c(x,\alpha) = \{j \in Q \mid f_j(x) - c\alpha + t > 0\}, \tag{18}$$

then we know that

$$F_t^c(x,\alpha) = c\alpha + \frac{t}{2}\sum_{j\in Q_t^c(x,\alpha)}\left(1 + \frac{f_j(x) - c\alpha}{t}\right)^2, \tag{19}$$

only relates to the component functions f_j for $j \in Q_t^c(x,\alpha)$, whose function values are close to $F(x)$. Therefore, $F_t^c(x,\alpha)$ is called an active set smoothing function for the maximum function in this paper. By direct calculation, we can obtain the gradient of $F_t^c(x,\alpha)$:

$$\nabla F_t^c(x,\alpha) = \begin{pmatrix} \displaystyle\sum_{j\in Q}\left(1 + \frac{f_j(x) - c\alpha}{t}\right)_+ \nabla f_j(x) \\ \displaystyle c - c\sum_{j\in Q}\left(1 + \frac{f_j(x) - c\alpha}{t}\right)_+ \end{pmatrix}, \tag{20}$$

which can be also written as follows:

$$\nabla F_t^c(x,\alpha) = \begin{pmatrix} \displaystyle\sum_{j\in Q_t^c(x,\alpha)}\left(1 + \frac{f_j(x) - c\alpha}{t}\right)\nabla f_j(x) \\ \displaystyle c - c\sum_{j\in Q_t^c(x,\alpha)}\left(1 + \frac{f_j(x) - c\alpha}{t}\right) \end{pmatrix}. \tag{21}$$

Lemma 5. *For any $t > 0$, $c > 0$, and $(x,\alpha) \in R^n \times R$,*

$$F_t^c(x,\alpha) > F(x). \tag{22}$$

Proof. If $c\alpha > F(x)$, by $t > 0$, we have

$$F_t^c(x,\alpha) = c\alpha + \frac{t}{2}\sum_{j\in Q}\left(1 + \frac{f_j(x) - c\alpha}{t}\right)_+^2 \ge c\alpha > F(x). \tag{23}$$

If $c\alpha \le F(x)$, we know $(1 + ((F(x) - c\alpha)/t))_+^2 = (1 + ((F(x) - c\alpha)/t))^2$ by $t > 0$, then we have

$$\begin{aligned} F_t^c(x,\alpha) &= c\alpha + \frac{t}{2}\sum_{j\in Q}\left(1 + \frac{f_j(x) - c\alpha}{t}\right)_+^2 \\ &\ge c\alpha + \frac{t}{2}\left(1 + \frac{F(x) - c\alpha}{t}\right)^2 \\ &= c\alpha + \frac{t}{2}\left(1 + \frac{2(F(x) - c\alpha)}{t} + \frac{(F(x) - c\alpha)^2}{t^2}\right) \\ &= \frac{t}{2} + F(x) + \frac{(F(x) - c\alpha)^2}{2t} \\ &> F(x). \end{aligned} \tag{24}$$

By (23) and (24), the conclusion holds. □

Lemma 6. *For any $t > 0$, $c > 0$, $\tau > 0$, and $(x,\alpha) \in R^n \times R$ satisfying $|\nabla_\alpha F_t^c(x,\alpha)| < \tau \le c$,*

(i) $F(x) - (\tau/c)t < c\alpha < F(x) + (1 - ((c-\tau)/c\overline{q}))t$

(ii) $F_t^c(x,\alpha) < F(x) + (1 - ((c-\tau)/c\overline{q}) + (((c+\tau)^2)/2c^2))t$, where $\overline{q} = \sharp(Q_t^c(x,\alpha))$.

Proof. By (10) and $|\nabla_\alpha F_t^c(x,\alpha)| < \tau$, we know

$$-\tau < c - c\sum_{j\in Q_t^c(x,\alpha)}\left(1 + \frac{f_j(x) - c\alpha}{t}\right) < \tau. \tag{25}$$

Then, by $c \ge \tau > 0$, we have $Q_t^c(x,\alpha) \ne \varnothing$,

$$0 \le 1 - \frac{\tau}{c} < \sum_{j\in Q_t^c(x,\alpha)}\left(1 + \frac{f_j(x) - c\alpha}{t}\right) < 1 + \frac{\tau}{c}, \tag{26}$$

and hence, $\max_{j\in Q_t^c(x,\alpha)}(1 + ((f_j(x) - c\alpha)/t)) > 0$. Therefore, by $t > 0$ and $Q_t^c(x,\alpha) \subseteq Q$, we have

$$1 + \frac{F(x) - c\alpha}{t} = \max_{j\in Q}\left(1 + \frac{f_j(x) - c\alpha}{t}\right) > 0. \tag{27}$$

(i) By (26) and (27), we have

$$\begin{aligned} \overline{q}\left(1 + \frac{F(x) - c\alpha}{t}\right) &\ge \sum_{j\in Q_t^c(x,\alpha)}\left(1 + \frac{f_j(x) - c\alpha}{t}\right) > 1 - \frac{\tau}{c}, \\ 1 + \frac{F(x) - c\alpha}{t} &\le \sum_{j\in Q_t^c(x,\alpha)}\left(1 + \frac{f_j(x) - c\alpha}{t}\right) < 1 + \frac{\tau}{c}. \end{aligned} \tag{28}$$

Then, by $t > 0$, we have

$$F(x) - \frac{\tau}{c}t < c\alpha < F(x) + \left(1 - \frac{c-\tau}{c\bar{q}}\right)t. \tag{29}$$

(ii) By (26), (29) and $t > 0$, we have

$$\begin{aligned} F_t^c(x,\alpha) &= c\alpha + \frac{t}{2}\sum_{j\in Q_t^c(x,\alpha)}\left(1 + \frac{f_j(x) - c\alpha}{t}\right)^2 \\ &\le c\alpha + \frac{t}{2}\left(\sum_{j\in Q_t^c(x,\alpha)}\left(1 + \frac{f_j(x) - c\alpha}{t}\right)\right)^2 \\ &< F(x) + \left(1 - \frac{c-\tau}{c\bar{q}}\right)t + \frac{t}{2}\left(1 + \frac{\tau}{c}\right)^2 \\ &= F(x) + \left(1 - \frac{c-\tau}{c\bar{q}} + \frac{(c+\tau)^2}{2c^2}\right)t. \end{aligned} \tag{30}$$

According to Lemmas 5 and 6 and $(c+\tau)^2 \le 4c^2$ for $c \ge \tau > 0$, we have the following approximation of $F_t^c(x,\alpha)$ for $F(x)$. □

Lemma 7. *For any $t > 0$, $c > 0$, and $(x,\alpha) \in R^n \times R$ satisfying $|\nabla_\alpha F_t^c(x,\alpha)| < c$,*

(i) $F(x) - t < c\alpha < F(x) + t$

(ii) $F(x) < F_t^c(x,\alpha) < F(x) + 3t$

For convenience of discussion, for any $t > 0$, $c > 0$, and $(x,\alpha) \in R^n \times R$, let

$$z_+^j(x,\alpha) = \left(1 + \frac{f_j(x) - c\alpha}{t}\right)_+, \quad j \in Q, \tag{31}$$

then the gradient of $F_t^c(x,\alpha)$ in (20) can be rewritten as follows:

$$\nabla F_t^c(x,\alpha) = \begin{pmatrix} \sum_{j\in Q} z_+^j(x,\alpha)\nabla f_j(x) \\ c - c\sum_{j\in Q} z_+^j(x,\alpha) \end{pmatrix}. \tag{32}$$

Lemma 8. *Suppose that Assumption 1 holds, then for any $t > 0$ and $c > 0$, $z_+^j(x,\alpha)$: $R^n \times R \longrightarrow R$, $j \in Q$, is locally Lipschitz continuous.*

Proof. By Assumption 1, we know that $1 + ((f_j(x) - c\alpha)/t)$: $R^n \times R \longrightarrow R$, $j \in Q$, is locally Lipschitz continuous with respect to the variables (x,α) for any $t > 0$ and $c > 0$. By the definition of the plus function, for any $z^1, z^2 \in R$, we know that

(i) If $z^1 \le 0$, $z^2 \le 0$, then we have $z_+^1 = z_+^2 = 0$, and hence, $|z_+^1 - z_+^2| = 0$

(ii) If $z^1 \le 0$, $z^2 > 0$, then we have $z_+^1 = 0$, $z_+^2 = z^2$, and hence, $|z_+^1 - z_+^2| = |0 - z^2| \le |z^1 - z^2|$

(iii) If $z^1 > 0$, $z^2 > 0$, then we have $z_+^1 = z^1$, $z_+^2 = z^2$, and hence, $|z_+^1 - z_+^2| = |z^1 - z^2|$

Therefore, the plus function is Lipschitz continuous. Hence, $z_+^j(x,\alpha)$: $R^n \times R \longrightarrow R$, $j \in Q$, is locally Lipschitz continuous. □

Lemma 9. *Suppose that Assumption 1 holds, then for any $t > 0$ and $c > 0$, $\nabla F_t^c(x,\alpha)$: $R^n \times R \longrightarrow R^{n+1}$ is locally Lipschitz continuous.*

Proof. By Assumption 1, $\nabla f_j(x)$: $R^n \longrightarrow R^n$, $j \in Q$, is locally Lipschitz continuous. Then, by Lemma 8, $\sum_{j\in Q} z_+^j(x,\alpha)\nabla f_j(x)$: $R^n \times R \longrightarrow R^n$ and $c - c\sum_{j\in Q} z_+^j(x,\alpha)$: $R^n \times R \longrightarrow R$, are locally Lipschitz continuous, which implies that $\nabla F_t^c(x,\alpha)$: $R^n \times R \longrightarrow R^{n+1}$ is locally Lipschitz continuous. □

Lemma 10. *Suppose that Assumption 1 holds, then for any $t > 0$ and $c > 0$, $z_+^j(x,\alpha)$: $R^n \times R \longrightarrow R$, $j \in Q$, is strongly semismooth.*

Proof. By the proof of Lemma 8, the plus function is Lipschitz continuous. For any $h > 0$ and $\xi > 0$, we know $\xi h > 0$ and $h + \xi h > 0$, and hence, $h_+ = h$, $(\xi h)_+ = \xi h$, $(h + \xi h)_+ = h + \xi h$; then, we have

$$\begin{aligned} \partial(0+h)_+ &= \{1\}, \\ z_+'(0;h) &= \lim_{\xi\downarrow 0}\frac{(\xi h)_+ - 0}{\xi} = h. \end{aligned} \tag{33}$$

For any $h < 0$ and $\xi > 0$, we know $\xi h < 0$ and $h + \xi h < 0$, and hence, $h_+ = (\xi h)_+ = (h + \xi h)_+ = 0$; then, we have

$$\begin{aligned} \partial(0+h)_+ &= \{0\}, \\ z_+'(0;h) &= \lim_{\xi\downarrow 0}\frac{(\xi h)_+ - 0}{\xi} = 0. \end{aligned} \tag{34}$$

Therefore, for any $V \in \partial(0+h)_+$ and $h \longrightarrow 0$, we have

$$Vh - z_+'(x;h) = O\left(\|h\|^2\right), \tag{35}$$

which implies that the plus function is strongly semismooth at $z = 0$ by Definition 1. Since the plus function is sufficiently smooth on $R\backslash\{0\}$, we know that the plus function is strongly semismooth on R.

By Assumption 1, $\nabla f_j(x)$: $R^n \longrightarrow R^n$, $j \in Q$, is locally Lipschitz continuous. Then, by Lemma 3, the component functions $f_j(x)$: $R^n \longrightarrow R$, $j \in Q$, are strongly semismooth, and hence, $1 + ((f_j(x) - c\alpha)/t)$: $R^n \times R \longrightarrow R$, $j \in Q$, is strongly semismooth with respect to (x,α) for any $t > 0$ and $c > 0$. Therefore, by (iv) of Lemma 1, $z_+^j(x,\alpha)$: $R^n \times R \longrightarrow R$, $j \in Q$, is strongly semismooth. □

Lemma 11. *Suppose that Assumption 1 holds, then for any $t > 0$ and $c > 0$, $\nabla F_t^c(x,\alpha)$: $R^n \times R \longrightarrow R^{n+1}$ is strongly semismooth.*

Proof. By Lemma 9, $\nabla F_t^c(x,\alpha)$: $R^n \times R \longrightarrow R^{n+1}$ is locally Lipschitz continuous. By Assumption 1, $\nabla f_j(x)$: $R^n \longrightarrow R^n$, $j \in Q$, is strongly semismooth. By Lemma 10, $z_+^j(x,\alpha)$: $R^n \times R \longrightarrow R$, $j \in Q$, is strongly semismooth. Then, by (i) and (ii) of Lemma 1, $\sum_{j\in Q} z_+^j(x,\alpha)\nabla f_j(x)$: $R^n \times R \longrightarrow R^n$ and $c - c\sum_{j\in Q} z_+^j(x,\alpha)$: $R^n \times R \longrightarrow R$, are strongly semismooth. Therefore, $\nabla F_t^c(x,\alpha)$: $R^n \times R \longrightarrow R^{n+1}$ is strongly semismooth by Lemma 2.

For any $(x,\alpha) \in R^n \times R$, let $\overline{Q}_t^c(x,\alpha) = \{j \in Q \mid f_j(x) - c\alpha + t = 0\}$, $\overline{q} = \sharp(Q_t^c(x,\alpha))$. By (32) and the definition of the plus function, the Clarke generalized Jacobian of $\nabla F_t^c(x,\alpha)$ at (x,α) can be represented as follows:

$$\partial(\nabla F_t^c(x,\alpha)) = \left\{ \begin{pmatrix} H_1(x,\alpha) & H_2(x,\alpha) \\ H_2(x,\alpha)^T & H_3(x,\alpha) \end{pmatrix} \middle| \zeta_j \in [0,1], j \in \overline{Q}_t^c(x,\alpha) \right\}, \tag{36}$$

where

$$\begin{aligned} H_1(x,\alpha) &= \sum_{j\in Q_t^c(x,\alpha)} z_+^j(x,\alpha)\nabla^2 f_j(x) + \frac{1}{t}\left(\sum_{j\in \overline{Q}_t^c(x,\alpha)} \zeta_j \nabla f_j(x)\nabla f_j(x)^T + \sum_{j\in Q_t^c(x,\alpha)} \nabla f_j(x)\nabla f_j(x)^T \right), \\ H_2(x,\alpha) &= -\frac{c}{t}\left(\sum_{j\in \overline{Q}_t^c(x,\alpha)} \zeta_j \nabla f_j(x) + \sum_{j\in Q_t^c(x,\alpha)} \nabla f_j(x) \right), \\ H_3(x,\alpha) &= \frac{c^2}{t}\left(\sum_{j\in \overline{Q}_t^c(x,\alpha)} \zeta_j + \overline{q} \right). \end{aligned} \tag{37}$$

For efficient numerical evaluation of $\partial(\nabla F_t^c(x,\alpha))$, we can set $\zeta_j = 0$ for $j \in \overline{Q}_t^c(x,\alpha)$; then, we know

$$V_t^c(x,\alpha) = \begin{pmatrix} H_1(x,\alpha) & H_2(x,\alpha) \\ H_2(x,\alpha)^T & H_3(x,\alpha) \end{pmatrix} \in \partial(\nabla F_t^c(x,\alpha)), \tag{38}$$

where

$$\begin{aligned} H_1(x,\alpha) &= \sum_{j\in Q_t^c(x,\alpha)} \left(z_+^j(x,\alpha)\nabla^2 f_j(x) + \frac{1}{t}\nabla f_j(x)\nabla f_j(x)^T \right), \\ H_2(x,\alpha) &= -\frac{c}{t}\sum_{j\in Q_t^c(x,\alpha)} \nabla f_j(x), \\ H_3(x,\alpha) &= \frac{c^2 \overline{q}}{t}. \end{aligned} \tag{39}$$

□

3. An Active Set Smoothing Method and Its Convergence

In this section, based on the active set smoothing function $F_t^c(x,\alpha)$ for $F(x)$ and the smoothing methods introduced in [24], an active set smoothing method is proposed to solve problem (1). For a starting point $(x^0,\alpha^0) \in R^n \times R$ and an initial smoothing parameter $t^0 > 0$, the initial scaling parameter $c^0 > 0$ is chosen from a bounded region in Subroutine 1, which reduces the ill-conditioning of $V_{t^0}^{c^0}(x^0,\alpha^0/c^0)$ caused by the scaling problem of the variable $\alpha \in R$; then, $\alpha^{0,0}$ is set to be α^0/c^0. The Armijo line search strategy, the steepest decent direction, and the Newton direction, in which the selection of the search direction depends on the condition number of $V_{t^0}^{c^0}(x^{k,0},\alpha^{k,0})$ and two convergence conditions for $k \geq 0$, are used to compute an approximate solution $(x^{k^0,0},\alpha^{k^0,0})$ of the smoothing problem $P_{t^0}^{c^0}$:

$$\min_{(x,\alpha)\in R^n\times R} F_{t^0}^{c^0}(x,\alpha). \tag{40}$$

Then, the smoothing parameter t^0 geometrically reduces to t^1, the scaling parameter c^1 is chosen from a bounded region in Subroutine 1, $\alpha^{k^0,0}$ is updated to $\alpha^{k^0+1,1}$ in two ways to balance the efficiency and convergence of the resulting algorithm in Subroutine 1, and the smoothing problem $P_{t^1}^{c^1}$:

$$\min_{(x,\alpha)\in R^n\times R} F_{t^1}^{c^1}(x,\alpha), \tag{41}$$

is solved with the starting point $(x^{k^0,0},\alpha^{k^0+1,1})$. By repeating this process, a sequence of smooth, unconstrained optimization problems is solved. As the smoothing parameters t^i go to 0, a solution of problem (1) can be obtained by the solutions of the smoothing problems $P_{t^i}^{c^i}$.

Algorithm 1. An active set smoothing algorithm.

Data. Input $\beta,\gamma \in (0,1)$; $C_l \in (0,1)$, $C_u > 1$, $\hat{t} \in (0,1)$; $\kappa \gg 1$, $\kappa_1^P, \kappa_2^P, \kappa_1, \kappa_2, \kappa_3 > 0$; $\omega_t, \omega_c \in (0,1)$; $\tau(t)$: $(0,\infty) \longrightarrow (0,\infty)$ satisfying $\lim_{t\longrightarrow 0}\tau(t) = 0$; $x^0 \in R^n$, $\alpha^0 \in R$, $t^0 > 0$.

Step 0: set $i = 0$, $k = 0$, $x^{0,0} = x^0$, $\alpha^{0,0} = \alpha^0$, and $c^0 = 1$, and go to Subroutine 1.

Step 1: (compute the search direction) compute the condition number $C^{k,i}$ of $V_{t^i}^{c^i}(x^{k,i}, \alpha^{k,i})$. If $C^{k,i} < \kappa$, compute the Newton direction $d_{t^i}^{c^i}(x^{k,i}, \alpha^{k,i})$ by solving

$$V_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right) d_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right) + \nabla F_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right) = 0. \quad (42)$$

If $V_{t^i}^{c^i}(x^{k,i}, \alpha^{k,i})$ is positive and the Newton direction $d_{t^i}^{c^i}(x^{k,i}, \alpha^{k,i})$ satisfies

$$\left\| d_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right) \right\| \le \kappa_1^P \left\| \nabla F_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right) \right\|, \quad (43)$$

$$\left\langle -\nabla F_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right), d_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right) \right\rangle \ge \kappa_2^P \left\| \nabla F_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right) \right\|^2, \quad (44)$$

go to Step 2.

If $V_{t^i}^{c^i}(x^{k,i}, \alpha^{k,i})$ is not positive and the Newton direction $d_{t^i}^{c^i}(x^{k,i}, \alpha^{k,i})$ satisfies

$$\left\| d_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right) \right\| \le \kappa_1 \left\| \nabla F_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right) \right\|, \quad (45)$$

$$\begin{aligned} &\left\langle -\nabla F_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right), d_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right) \right\rangle \ge \kappa_2 \left\| \nabla F_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right) \right\|^2, \\ &\left\langle -\nabla F_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right), d_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right) \right\rangle \ge \kappa_3 \left\| \nabla F_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right) \right\| \\ &\qquad \cdot \left\| d_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right) \right\|, \end{aligned} \quad (46)$$

go to Step 2.

Else, compute the steepest decent direction

$$d_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right) = -\nabla F_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right), \quad (47)$$

go to Step 2.

Step 2: (compute the stepsize) let $\lambda_{t^i}^{c^i}(x^{k,i}, \alpha^{k,i}) = \beta^l$, where l is the smallest nonnegative integer satisfying

$$\begin{aligned} &F_{t^i}^{c^i}\left(\left(x^{k,i}, \alpha^{k,i}\right) + \beta^l d_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right)\right) - F_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right) \\ &\quad \le \gamma \beta^l \left\langle \nabla F_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right), d_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right) \right\rangle, \end{aligned} \quad (48)$$

go to Step 3.

Step 3: set $(x^{k+1,i}, \alpha^{k+1,i}) = (x^{k,i}, \alpha^{k,i}) + \lambda_{t^i}^{c^i}(x^{k,i}, \alpha^{k,i}) d_{t^i}^{c^i}(x^{k,i}, \alpha^{k,i})$, replace k by $k+1$, and go to Step 4.

Step 4: if $\|\nabla F_{t^i}^{c^i}(x^{k,i}, \alpha^{k,i})\| < \tau(t^i)$, go to Step 5; else, go to Step 1.

Step 5: (adjustment of the smoothing parameter) set $k^i = k$, $x^{k+1,i+1} = x^{k,i}$, $\alpha^{k+1,i+1} = \alpha^{k,i}$, $t^{i+1} = \omega_t t^i$, and $c^{i+1} = c^i$, replace i by $i+1$ and k by $k+1$, and go to Subroutine 1.

Subroutine 1: adjustment of the scaling parameter.

Substep 0: set $x = x^{k,i}$, $t = t^i$, $c^l = \omega_c c^i$, and $c^u = c^i/\omega_c$. If $i = 0$ or $t \le \hat{t}$, set $\overline{\alpha}^i = c^i \alpha^{k,i}$:

$$\alpha_l^{k,i} = \frac{\overline{\alpha}^i}{c^l}, \qquad \alpha_u^{k,i} = \frac{\overline{\alpha}^i}{c^u}; \quad (49)$$

else, set $\overline{\alpha}^i = c^i \alpha^{k,i} - (1/\omega_t - 1)t$:

$$\alpha_l^{k,i} = \frac{\overline{\alpha}^i}{c^l}, \qquad \alpha^{k,i} = \frac{\overline{\alpha}^i}{c^i}, \qquad \alpha_u^{k,i} = \frac{\overline{\alpha}^i}{c^u}. \quad (50)$$

Compute the condition numbers $C^{k,i}$ of $V_t^{c^i}(x, \alpha^{k,i})$, $C_l^{k,i}$ of $V_t^{c^l}(x, \alpha_l^{k,i})$, and $C_u^{k,i}$ of $V_t^{c^u}(x, \alpha_u^{k,i})$. If $c^l \ge C_l$ and $C_l^{k,i} < C^{k,i}$, go to Substep 1; else, if $c^u \le C_u$ and $C_u^{k,i} < C^{k,i}$, go to Substep 2; else, go to Step 4 of Algorithm 1 with c^i, $C^{k,i}$, $V_t^{c^i}(x, \alpha^{k,i})$, and $\alpha^{k,i}$.

Substep 1: set $c^i = c^l$, $C^{k,i} = C_l^{k,i}$, $V_t^{c^i}(x, \alpha^{k,i}) = V_t^{c^l}(x, \alpha_l^{k,i})$, and $\alpha^{k,i} = \alpha_l^{k,i}$, and go to Substep 3.

Substep 2: set $c^i = c^u$, $C^{k,i} = C_u^{k,i}$, $V_t^{c^i}(x, \alpha^{k,i}) = V_t^{c^u}(x, \alpha_u^{k,i})$, and $\alpha^{k,i} = \alpha_u^{k,i}$, and go to Substep 4.

Substep 3: set $c^l = \omega_c c^l$ and $\alpha_l^{k,i} = \overline{\alpha}^i / c^l$ and compute the condition number $C_l^{k,i}$ of $V_t^{c^l}(x, \alpha_l^{k,i})$. If $C_l^{k,i} < C^{k,i}$ and $c^l \ge C_l$, go to Substep 1; else, go to Step 4 of Algorithm 1 with c^i, $C^{k,i}$, $V_t^{c^i}(x, \alpha^{k,i})$, and $\alpha^{k,i}$.

Substep 4: set $c^u = c^u/\omega_c$ and $\alpha_u^{k,i} = \overline{\alpha}^i / c^u$ and compute the condition number $C_u^{k,i}$ of $V_t^{c^u}(x, \alpha_u^{k,i})$. If $C_u^{k,i} < C^{k,i}$ and $c^u \le C_u$, go to Substep 2; else, go to Step 4 of Algorithm 1 with c^i, $C^{k,i}$, $V_t^{c^i}(x, \alpha^{k,i})$, and $\alpha^{k,i}$.

Remark 1. In Subroutine 1 for adjusting the scaling parameter c^i, if $t^i > \hat{t}$, $\alpha^{k,i}$ is updated to satisfy

$$c^{i+1} \alpha^{k^i+1, i+1} = c^i \alpha^{k^i, i} - (1 - \omega_t) t^i. \quad (51)$$

Then, by $x^{k^i+1,i+1} = x^{k^i,i}$ and $t^{i+1} = \omega_t t^i$, we have

$$\begin{aligned} &f_j\left(x^{k^i+1,i+1}\right) - c^{i+1}\alpha^{k^i+1,i+1} + t^{i+1} = f_j\left(x^{k^i,i}\right) - c^i \alpha^{k^i,i} + t^i, \quad j \in Q, \\ &Q_{t^{i+1}}^{c^{i+1}}\left(x^{k^i+1,i+1}, \alpha^{k^i+1,i+1}\right) = Q_{t^i}^{c^i}\left(x^{k^i,i}, \alpha^{k^i,i}\right), \\ &z_+^j\left(x^{k^i+1,i+1}, \alpha^{k^i+1,i+1}\right) = \frac{z_+^j\left(x^{k^i,i}, \alpha^{k^i,i}\right)}{\omega_t}, \quad j \in Q, \\ &\nabla_x F_{t^{i+1}}^{c^{i+1}}\left(x^{k^i+1,i+1}, \alpha^{k^i+1,i+1}\right) = \frac{\nabla_x F_{t^i}^{c^i}\left(x^{k^i,i}, \alpha^{k^i,i}\right)}{\omega_t}, \\ &\nabla_\alpha F_{t^{i+1}}^{c^{i+1}}\left(x^{k^i+1,i+1}, \alpha^{k^i+1,i+1}\right) = c^{i+1}\left(1 - \frac{1 - \nabla_\alpha F_{t^i}^{c^i}\left(x^{k^i,i}, \alpha^{k^i,i}\right)/c^i}{\omega_t}\right), \\ &V_{t^{i+1}}^{c^{i+1}}\left(x^{k^i+1,i+1}, \alpha^{k^i+1,i+1}\right) = \frac{V_{t^i}^{c^i}\left(x^{k^i,i}, \alpha^{k^i,i}\right)}{\omega_t}. \end{aligned} \quad (52)$$

If $t^i \leq \hat{t}$, $\alpha^{k,i}$ is updated to satisfy

$$c^{i+1}\alpha^{k^i+1,i+1} = c^i\alpha^{k^i,i}, \tag{53}$$

which keeps the monotonicity of $\left\{F_{t^i}^{c^i}(x^{k,i}, \alpha^{k,i})\right\}$ with respect to k in Lemma 13.

Theorem 2 (local convergence [31]). *For any $t > 0$ and $c > 0$, suppose that (x^*, α^*) is a stationary point of the problem P_t^c. If all $V \in \partial(\nabla F_t^c(x^*, \alpha^*))$ are nonsingular, then there exist a neighborhood $N(x^*, \alpha^*)$ of (x^*, α^*) and a constant M such that for any $(x, \alpha) \in N(x^*, \alpha^*)$ and any $V \in \partial(\nabla F_t^c(x, \alpha))$, V is nonsingular and*

$$\left\|V^{-1}\right\| \leq M. \tag{54}$$

The sequence $\{(x^k, \alpha^k)\}$ produced by any initial point $(x^0, \alpha^0) \in N(x^*, \alpha^*)$ and the semismooth Newton method $(x^{k+1}, \alpha^{k+1}) = (x^k, \alpha^k) - (V^k)^{-1}\nabla F_t^c(x^k, \alpha^k)(V^k \in \partial (\nabla F_t^c(x^k, \alpha^k)))$ quadratically converges to (x^*, α^*).

Lemma 12. *Suppose that Assumption 1 holds, then for any bounded set $S \subset R^n \times R$ and parameters β, $\gamma \in (0,1)$, $t > 0$, and $c > 0$, there exists a $\lambda_S < \infty$ such that for any $(x, \alpha) \in S$, $\lambda_t^c(x, \alpha) \geq \lambda_S$ and*

$$F_t^c((x, \alpha) + \lambda_t^c(x, \alpha)d_t^c(x, \alpha)) - F_t^c(x, \alpha) \leq -\lambda_S\left\|\nabla F_t^c(x, \alpha)\right\|^2, \tag{55}$$

where $\lambda_t^c(x, \alpha)$ is the stepsize computed in Step 2 of Algorithm 1.

Proof. Let $\kappa_1^* = \max\{\kappa_1^P, \kappa_1, 1\}$ and $\kappa_2^* = \min\{\kappa_2^P, \kappa_2, 1\}$. By (43)–(47), the search direction $d_t^c(x, \alpha)$ satisfies

$$\left\|d_t^c(x, \alpha)\right\| \leq \kappa_1^*\left\|\nabla F_t^c(x, \alpha)\right\|, \tag{56}$$

$$\langle -\nabla F_t^c(x, \alpha), d_t^c(x, \alpha)\rangle \geq \kappa_2^*\left\|\nabla F_t^c(x, \alpha)\right\|^2. \tag{57}$$

By Lemma 9, $\nabla F_t^c(x, \alpha)$: $R^n \times R \longrightarrow R^{n+1}$ is locally Lipschitz continuous, and then, there exists a Lipschitz constant $L_S > 0$ such that for any (x^1, α^1), $(x^2, \alpha^2) \in S$,

$$\left\|\nabla F_t^c\left(x^1, \alpha^1\right) - \nabla F_t^c\left(x^2, \alpha^2\right)\right\| \leq L_S\left\|\left(x^1, \alpha^1\right) - \left(x^2, \alpha^2\right)\right\|. \tag{58}$$

For any $(x, \alpha) \in S$ and $\lambda \in (0,1]$, by the mean value theorem, there exists a $\xi \in (0,1)$ such that

$$\begin{aligned}
&F_t^c((x, \alpha) + \lambda d_t^c(x, \alpha)) - F_t^c(x, \alpha) - \gamma\lambda\langle\nabla F_t^c(x, \alpha), d_t^c(x, \alpha)\rangle \\
&\quad = \langle\nabla F_t^c((x, \alpha) + \xi\lambda d_t^c(x, \alpha)), \lambda d_t^c(x, \alpha)\rangle - \gamma\lambda\langle\nabla F_t^c(x, \alpha), d_t^c(x, \alpha)\rangle \\
&\quad = \lambda(1-\gamma)\langle\nabla F_t^c(x, \alpha), d_t^c(x, \alpha)\rangle + \langle\nabla F_t^c((x, \alpha) + \xi\lambda d_t^c(x, \alpha)) - \nabla F_t^c(x, \alpha), \lambda d_t^c(x, \alpha)\rangle \\
&\quad \leq \lambda(1-\gamma)\langle\nabla F_t^c(x, \alpha), d_t^c(x, \alpha)\rangle + \lambda\left\|\nabla F_t^c((x, \alpha) + \xi\lambda d_t^c(x, \alpha)) - \nabla F_t^c(x, \alpha)\right\|\left\|d_t^c(x, \alpha)\right\| \\
&\quad \leq \lambda(1-\gamma)\langle\nabla F_t^c(x, \alpha), d_t^c(x, \alpha)\rangle + \lambda^2\xi L_S\left\|d_t^c(x, \alpha)\right\|^2 \\
&\quad \leq -\lambda\left(\kappa_2^*(1-\gamma) - \kappa_1^{*2}\lambda L_S\right)\langle\nabla F_t^c(x, \alpha)\rangle^2,
\end{aligned} \tag{59}$$

where the second inequality comes from (58) and the last inequality comes from (56), (57), and $\xi \in (0,1)$. Let

$$\lambda^* = \min\left\{1, \frac{\kappa_2^*(1-\gamma)}{\kappa_1^{*2}L_S}\right\}, \tag{60}$$

then it follows from (59) that for any $\lambda \in (0, \lambda^*]$,

$$F_t^c((x, \alpha) + \lambda d_t^c(x, \alpha)) - F_t^c(x, \alpha) - \gamma\lambda\langle\nabla F_t^c(x, \alpha), d_t^c(x, \alpha)\rangle \leq 0, \tag{61}$$

and hence, $\lambda_t^c(x, \alpha) \geq \beta\lambda^*$. Therefore, by (57), we have

$$\begin{aligned}
&F_t^c((x, \alpha) + \lambda_t^c(x, \alpha)d_t^c(x, \alpha)) - F_t^c(x, \alpha) \\
&\quad \leq \gamma\lambda_t^c(x, \alpha)\langle\nabla F_t^c(x, \alpha), d_t^c(x, \alpha)\rangle \\
&\quad \leq -\beta\gamma\lambda^*\kappa_2^*\left\|\nabla F_t^c(x, \alpha)\right\|^2.
\end{aligned} \tag{62}$$

Then, by γ, $\kappa_2^* \in (0,1]$, the conclusion holds for

$$\lambda_S = \min\left\{\beta\gamma\kappa_2^*, \frac{\beta\gamma(1-\gamma)\left(\kappa_2^*\right)^2}{\left(\kappa_1^*\right)^2L_S}\right\}. \tag{63}$$

□

Lemma 13. *Suppose that Assumptions 1 and 2 hold and $\tau(t^i) \leq \omega_t c^i$, then for any $(x^0, \alpha^0) \in R^n \times R$, $t^0 > 0$ and $c^0 > 0$, the sequence $\{(x^{k,i}, \alpha^{k,i})\}$ generated by Algorithm 1 satisfies the following:*

(i) For any $i \geq 0$, $\left\{F_{t^i}^{c^i}(x^{k,i}, \alpha^{k,i})\right\}$ is monotone decreasing with respect to k

(ii) There exists an $\hat{i}$ such that the sequence $\left\{F_{t^i}^{c^i}(x^{k,i}, \alpha^{k,i})\right\}_{i \geq \hat{i}}$ is monotone decreasing with respect to k

(iii) The sequence $\{x^{k,i}\}$ is bounded

(iv) The sequence $\{\alpha^{k,i}\}$ is bounded

Proof.

(i) For any $i \geq 0$ and $(x^{k,i}, \alpha^{k,i}) \in R^n \times R$ satisfying $\|d_{t^i}^{c^i}(x^{k,i}, \alpha^{k,i})\| \geq \tau(t^i)$, by Lemma 12, there exists a $l^{k,i} \geq 0$ such that

$$\begin{aligned}
&F_{t^i}^{c^i}\left(x^{k+1,i}, \alpha^{k+1,i}\right) - F_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right) \\
&\quad \leq \gamma\beta^{l^{k,i}}\langle\nabla F_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right), d_{t^i}^{c^i}\left(x^{k,i}, \alpha^{k,i}\right)\rangle < 0,
\end{aligned} \tag{64}$$

then we know

$$F_{t^i}^{c^i}\left(x^{k+1,i},\alpha^{k+1,i}\right) < F_{t^i}^{c^i}\left(x^{k,i},\alpha^{k,i}\right), \quad (65)$$

which implies that the conclusion holds.

(ii) By $t^{i+1} = \omega_t t^i$ and $\omega_t \in (0,1)$, for any $t^0 > 0$ and $\hat{t} \in (0,1)$, there exists an $\hat{i}$ such that for any $i \geq \hat{i}$,

$$t^i \leq \hat{t}. \quad (66)$$

For any $i \geq \hat{i}$, if $\|\nabla F_{t^i}^{c^i}(x^{k,i},\alpha^{k,i})\| < \tau(t^i)$, by the definition of $\nabla F_t^c(x,\alpha)$ in (20), we have

$$-\tau\left(t^i\right) < c^i - c^i \sum_{j\in Q}\left(1 + \frac{f_j\left(x^{k,i}\right) - c^i\alpha^{k,i}}{t^i}\right)_+ < \tau\left(t^i\right). \quad (67)$$

Then, by $\tau(t^i) \leq \omega_t c^i < c^i$ and $t^i > 0$, we have

$$1 + \frac{F\left(x^{k,i}\right) - c^i\alpha^{k,i}}{t^i} = \max_{j\in Q}\left(1 + \frac{f_j\left(x^{k,i}\right) - c^i\alpha^{k,i}}{t^i}\right)_+ > 0. \quad (68)$$

Let t be the variable of the function $F_t^c(x,\alpha)$ for given $(x,\alpha) \in R^n \times R$ and $c > 0$, which is redefined as $F_{x,\alpha}^c(t)$, then by the mean value theorem, there exists a $\bar{t} \in (\omega_t t^i, t^i)$ such that

$$F_{x^{k,i},\alpha^{k,i}}^{c^i}\left(t^i\right) - F_{x^{k,i},\alpha^{k,i}}^{c^i}\left(\omega_t t^i\right) = \left(F_{x^{k,i},\alpha^{k,i}}^{c^i}\right)'(\bar{t})\left(t^i - \omega_t t^i\right), \quad (69)$$

with

$$\left(F_{x,\alpha}^c\right)'(t) = \frac{1}{2}\sum_{j\in Q}\left(1 + \frac{f_j(x) - c\alpha}{t}\right)_+\left(1 - \frac{f_j(x) - c\alpha}{t}\right). \quad (70)$$

If $F(x^{k,i}) \leq c^i\alpha^{k,i}$, by $\bar{t} > 0$, we have

$$1 - \frac{f_j\left(x^{k,i}\right) - c^i\alpha^{k,i}}{\bar{t}} > 0, \quad j \in Q, \quad (71)$$

and hence $(F_{x^{k,i},\alpha^{k,i}}^{c^i})'(\bar{t}) > 0$ by (68). If $F(x^{k,i}) > c^i\alpha^{k,i}$, by (67) and (68) and $\tau(t^i) \leq \omega_t c^i$, we have

$$\begin{aligned} c^i - c^i&\left(1 + \frac{F\left(x^{k,i}\right) - c^i\alpha^{k,i}}{t^i}\right) \\ &\geq c^i - c^i\sum_{j\in Q}\left(1 + \frac{f_j\left(x^{k,i}\right) - c^i\alpha^{k,i}}{t^i}\right)_+ \\ &> -\tau\left(t^i\right) \geq -\omega_t c^i, \end{aligned} \quad (72)$$

which implies

$$\frac{F\left(x^{k,i}\right) - c^i\alpha^{k,i}}{t^i} < \omega_t. \quad (73)$$

By $\bar{t} > \omega_t t^i > 0$, we have

$$\begin{aligned} &\frac{F\left(x^{k,i}\right) - c^i\alpha^{k,i}}{\bar{t}} < \frac{F\left(x^{k,i}\right) - c^i\alpha^{k,i}}{\omega_t t^i} < 1, \\ &\frac{f_j\left(x^{k,i}\right) - c^i\alpha^{k,i}}{\bar{t}} \leq \frac{F\left(x^{k,i}\right) - c^i\alpha^{k,i}}{\bar{t}} < 1, \quad j \in Q. \end{aligned} \quad (74)$$

Then, we know that (71) also holds and hence $(F_{x^{k,i},\alpha^{k,i}}^{c^i})'(\bar{t}) > 0$ by (68). Therefore, for the case of $\|\nabla F_{t^i}^{c^i}(x^{k,i},\alpha^{k,i})\| < \tau(t^i)$, by (69), we have

$$F_{x^{k,i},\alpha^{k,i}}^{c^i}\left(t^i\right) - F_{x^{k,i},\alpha^{k,i}}^{c^i}\left(\omega_t t^i\right) > 0, \quad (75)$$

which means

$$F_{t^i}^{c^i}\left(x^{k,i},\alpha^{k,i}\right) - F_{\omega_t t^i}^{c^i}\left(x^{k,i},\alpha^{k,i}\right) > 0. \quad (76)$$

By $x^{k+1,i+1} = x^{k,i}$ and $c^{i+1}\alpha^{k+1,i+1} = c^i\alpha^{k,i}$ according to (53) and (66), we have

$$F_{t^{i+1}}^{c^{i+1}}\left(x^{k+1,i+1},\alpha^{k+1,i+1}\right) = F_{t^{i+1}}^{c^i}\left(x^{k,i},\alpha^{k,i}\right). \quad (77)$$

Then, by (76) and $t^{i+1} = \omega_t t^i$, we know

$$F_{t^{i+1}}^{c^{i+1}}\left(x^{k+1,i+1},\alpha^{k+1,i+1}\right) < F_{t^i}^{c^i}\left(x^{k,i},\alpha^{k,i}\right). \quad (78)$$

Therefore, the sequence $\{F_{t^i}^{c^i}(x^{k,i},\alpha^{k,i})\}_{i\geq\hat{i}}$ is monotone decreasing with respect to k by (65) and (78).

(iii) By (i), for any $0 \leq k \leq k^0$, we have

$$F_{t^0}^{c^0}\left(x^{k,0},\alpha^{k,0}\right) \leq F_{t^0}^{c^0}\left(x^{0,0},\alpha^{0,0}\right), \quad (79)$$

and for any $1 \leq i < \hat{i}$, $k^{i-1} + 1 \leq k \leq k^i$, we have

$$F_{t^i}^{c^i}\left(x^{k,i},\alpha^{k,i}\right) \leq F_{t^i}^{c^i}\left(x^{k^{i-1}+1,i},\alpha^{k^{i-1}+1,i}\right). \quad (80)$$

By (ii), for any $i \geq \hat{i}$ and $k^{i-1} + 1 \leq k \leq k^i$,

$$F_{t^i}^{c^i}\left(x^{k,i},\alpha^{k,i}\right) \leq F_{t^{\hat{i}}}^{c^{\hat{i}}}\left(x^{k^{\hat{i}-1}+1,\hat{i}},\alpha^{k^{\hat{i}-1}+1,\hat{i}}\right). \quad (81)$$

By the finiteness of i satisfying $i < \hat{i}$, (79), (80), and (81), there exists a constant $F^0 \in R$ such that for any $(x^{k,i},\alpha^{k,i})$,

$$\left(x^{k,i},\alpha^{k,i}\right) \in \Omega^{F^0} = \left\{(x,\alpha) \in R^n \times R \mid F_t^c(x,\alpha) \leq F^0\right\}. \quad (82)$$

Suppose that for any $M > 0$, there exists a point $(x^{\bar{k},\bar{i}},\alpha^{\bar{k},\bar{i}}) \in \Omega^{F^0}$ satisfying $F(x^{\bar{k},\bar{i}}) > M$. If $c^{\bar{i}}\alpha^{\bar{k},\bar{i}} \geq M - t^{\bar{i}}$, by the definition of the plus function and $0 < t^{\bar{i}} \leq t^0$, we have

$$F_{t^{\bar{i}}}^{c^{\bar{i}}}\left(x^{\bar{k},\bar{i}},\alpha^{\bar{k},\bar{i}}\right) \geq c^{\bar{i}}\alpha^{\bar{k},\bar{i}} \geq M - t^{\bar{i}} \geq M - t^0. \quad (83)$$

If $c^{\bar{i}}\alpha^{\bar{k},\bar{i}} < M - t^{\bar{i}}$, by $F(x^{\bar{k},\bar{i}}) > M$ and $t^{\bar{i}} > 0$, we have

$$1 + \frac{F\left(x^{\bar{k},\bar{i}}\right) - c^{\bar{i}}\alpha^{\bar{k},\bar{i}}}{t^{\bar{i}}} > 1 + \frac{M - c^{\bar{i}}\alpha^{\bar{k},\bar{i}}}{t^{\bar{i}}} > 2, \quad (84)$$

and hence $(1+((F(x^{\bar{k},\bar{i}})-c^{\bar{i}}\alpha^{\bar{k},\bar{i}})/t^{\bar{i}}))_+ = 1+((F(x^{\bar{k},\bar{i}})-c^{\bar{i}}\alpha^{\bar{k},\bar{i}})/t^{\bar{i}})$ and $(1+((M-c^{\bar{i}}\alpha^{\bar{k},\bar{i}})/t^{\bar{i}}))_+ = 1+((M-c^{\bar{i}}\alpha^{\bar{k},\bar{i}})/t^{\bar{i}})$. Then, we know

$$\begin{aligned}F_{t^{\bar{i}}}^{c^{\bar{i}}}\left(x^{\bar{k},\bar{i}},\alpha^{\bar{k},\bar{i}}\right) &= c^{\bar{i}}\alpha^{\bar{k},\bar{i}} + \frac{t^{\bar{i}}}{2}\sum_{j\in Q}\left(1+\frac{f_j\left(x^{\bar{k},\bar{i}}\right)-c^{\bar{i}}\alpha^{\bar{k},\bar{i}}}{t^{\bar{i}}}\right)_+^2 \\ &\geq c^{\bar{i}}\alpha^{\bar{k},\bar{i}} + \frac{t^{\bar{i}}}{2}\left(1+\frac{F\left(x^{\bar{k},\bar{i}}\right)-c^{\bar{i}}\alpha^{\bar{k},\bar{i}}}{t^{\bar{i}}}\right)^2 \\ &> c^{\bar{i}}\alpha^{\bar{k},\bar{i}} + \frac{t^{\bar{i}}}{2}\left(1+\frac{M-c^{\bar{i}}\alpha^{\bar{k},\bar{i}}}{t^{\bar{i}}}\right)^2 \\ &> c^{\bar{i}}\alpha^{\bar{k},\bar{i}} + t^{\bar{i}}\left(1+\frac{M-c^{\bar{i}}\alpha^{\bar{k},\bar{i}}}{t^{\bar{i}}}\right) \\ &> M.\end{aligned} \tag{85}$$

However, the arbitrariness of M, (83), and (85) contradict $x^{\bar{k},\bar{i}} \in \Omega^{F^0}$; then, there must exist a $M^* > 0$ such that $F(x^{k,i}) \leq M^*$ for any $F(x^{k,i}) \leq M^*$ and $i \geq 0$. Therefore, we have that $\{x^{k,i}\} \subseteq \Omega_{M^*}$, and hence, the sequence $\{x^{k,i}\}$ is bounded by Assumption 2.

(iv) For any $(x^{k,i},\alpha^{k,i})$ with $k \geq 0$ and $i \geq 0$, if $Q_{t^i}^{c^i}(x^{k,i},\alpha^{k,i}) = \emptyset$, by the definition of $Q_t^c(x,\alpha)$, we know $(1+((f_j(x^{k,i})-c^i\alpha^{k,i})/t^i))_+ = 0$ for any $j \in Q$ by $t^i > 0$. Then, by $(x^{k,i},\alpha^{k,i}) \in \Omega^{F^0}$, we have

$$F_{t^i}^{c^i}\left(x^{k,i},\alpha^{k,i}\right) = c^i\alpha^{k,i} \leq F^0. \tag{86}$$

Hence, by $c^i \geq C_l > 0$, we know

$$\alpha^{k,i} \leq \frac{F^0}{c^i} \leq \frac{F^0}{C_l}. \tag{87}$$

If $Q_{t^i}^{c^i}(x^{k,i},\alpha^{k,i}) \neq \emptyset$, by $F(x^{k,i}) \leq M^*$ from (iii), we have

$$t^i + M^* - c^i\alpha^{k,i} \geq t^i + F\left(x^{k,i}\right) - c^i\alpha^{k,i} > 0. \tag{88}$$

Then, by $0 < t^i \leq t^0$ and $c^i \geq C_l > 0$, we know

$$\alpha^{k,i} < \frac{t^i + M^*}{c^i} \leq \frac{t^0 + M^*}{c^i} \leq \frac{t^0 + M^*}{C_l}. \tag{89}$$

Therefore, the sequence $\{\alpha^{k,i}\}$ is bounded by (87) and (89). □

Lemma 14. *Suppose that Assumptions 1 and 2 hold, then for any $(x^0,\alpha^0) \in R^n \times R$ and $t^0 > 0$, the sequences $\{(x^{k,i},\alpha^{k,i})\}$ and $\{t^i\}$ generated by Algorithm 1 satisfy the following:*

(i) For any $i \geq 0$, there exists a $k^i \in N$ such that $\|\nabla F_{t^i}^{c^i}(x^{k^i,i},\alpha^{k^i,i})\| < \tau(t^i)$

(ii) The sequence $\{t^i\}$ is infinite and strictly monotone decreasing, $t^i \longrightarrow 0$ as $k \longrightarrow \infty$

Proof.

(i) If there exists an $\bar{i} \geq 0$ such that the sequence $\{(x^{k,\bar{i}},\alpha^{k,\bar{i}})\}$ is infinite, then we have $\|\nabla F_{t^{\bar{i}}}^{c^{\bar{i}}}(x^{k,\bar{i}},\alpha^{k,\bar{i}})\| \geq \tau(t^{\bar{i}})$ and $k \longrightarrow \infty$. By Lemma 12 and the boundedness of $\{(x^{k,\bar{i}},\alpha^{k,\bar{i}})\}$ from Lemma 13, there exists a constant $C^{\bar{i}} > 0$ such that

$$\begin{aligned}&F_{t^{\bar{i}}}^{c^{\bar{i}}}\left(x^{k+1,\bar{i}},\alpha^{k+1,\bar{i}}\right) - F_{t^{\bar{i}}}^{c^{\bar{i}}}\left(x^{k,\bar{i}},\alpha^{k,\bar{i}}\right) \\ &\quad \leq -C^{\bar{i}}\left\|\nabla F_{t^{\bar{i}}}^{c^{\bar{i}}}\left(x^{k,\bar{i}},\alpha^{k,\bar{i}}\right)\right\|^2 \leq -C^{\bar{i}}\tau\left(t^{\bar{i}}\right)^2 < 0.\end{aligned} \tag{90}$$

Then, by $k \longrightarrow \infty$, we have

$$\lim_{k\longrightarrow\infty} F_{t^{\bar{i}}}^{c^{\bar{i}}}\left(x^{k,\bar{i}},\alpha^{k,\bar{i}}\right) = -\infty, \tag{91}$$

which contradicts that $\{(x^{k,\bar{i}},\alpha^{k,\bar{i}})\}$ is bounded and $F_{t^{\bar{i}}}^{c^{\bar{i}}}(x,\alpha)$ is continuous on $R^n \times R$. Therefore, for any $i \geq 0$, there exists a $k^i \in N$ such that $\|\nabla F_{t^i}^{c^i}(x^{k^i,i},\alpha^{k^i,i})\| < \tau(t^i)$.

(ii) By (i), for any $i \geq 0$, there exists a $k^i \in N$ such that $\|\nabla F_{t^i}^{c^i}(x^{k^i,i},\alpha^{k^i,i})\| < \tau(t^i)$, then t^i is updated to $t^{i+1} = \omega_t t^i$. Therefore, by $\omega_t \in (0,1)$, we know that $i \longrightarrow \infty$ as $k \longrightarrow \infty$, and hence, the sequence $\{t^i\}$ is infinite and strictly monotone decreasing, $t^i \longrightarrow 0$ as $k \longrightarrow \infty$. □

Theorem 3. *Suppose that Assumptions 1 and 2 hold, $\{(x^{k,i},\alpha^{k,i})\}$ is the sequence generated by Algorithm 1, then for any accumulation point $(x^*,\alpha^*) \in R^n \times R$ of $\{(x^{k^i,i},\alpha^{k^i,i})\}$, $0 \in \partial F(x^*)$, i.e., x^* is a stationary point of problem (1).*

Proof. By Lemmas 13 and 14, the sequence $\{(x^{k^i,i},\alpha^{k^i,i})\} \subseteq \{(x^{k,i},\alpha^{k,i})\}$ is infinite and bounded; then, there exists at least one accumulation point of $\{(x^{k^i,i},\alpha^{k^i,i})\}$. For any accumulation point (x^*,α^*) of $\{(x^{k^i,i},\alpha^{k^i,i})\}$, there exists a subsequence of $\{(x^{k^i,i},\alpha^{k^i,i})\}$ (denoted also by $\{(x^{k^i,i},\alpha^{k^i,i})\}$ for convenience) converging to (x^*,α^*). By Lemma 14, $\|\nabla F_{t^i}^{c^i}(x^{k^i,i},\alpha^{k^i,i})\| < \tau(t^i)$ and $\lim_{t\longrightarrow 0}\tau(t) = 0$, we have

$$\lim_{i\longrightarrow\infty}\left\|\nabla F_{t^i}^{c^i}\left(x^{k^i,i},\alpha^{k^i,i}\right)\right\| = 0. \tag{92}$$

It follows from (32) and (92) that

$$\lim_{i\longrightarrow\infty}\sum_{j\in Q} z_+^j\left(x^{k^i,i},\alpha^{k^i,i}\right)\nabla f_j\left(x^{k^i,i}\right) = 0, \tag{93}$$

$$\lim_{i\longrightarrow\infty}\sum_{j\in Q} z_+^j\left(x^{k^i,i},\alpha^{k^i,i}\right) = 1. \tag{94}$$

By Lemma 7, $(x^{k^i,i},\alpha^{k^i,i}) \longrightarrow (x^*,\alpha^*)$, $i \longrightarrow \infty$, and $t^i \longrightarrow 0$, we have

$$\lim_{i\longrightarrow\infty} F\left(x^{k^i,i}\right) = \lim_{i\longrightarrow\infty} c^i\alpha^{k^i,i} = F(x^*). \tag{95}$$

Then, by Assumption 1, we have

$$\lim_{i\longrightarrow\infty}\left(f_j\left(x^{k_i,i}\right)-c^i\alpha^{k_i,i}\right)=f_j\left(x^*\right)-F\left(x^*\right)<0,\quad j\notin I\left(x^*\right). \tag{96}$$

Hence, by the finiteness of indexes in $Q\setminus I(x^*)$, there exists an $i_0>0$ such that for any $i>i_0$,

$$f_j\left(x^{k_i,i}\right)-c^i\alpha^{k_i,i}<-t^i,\quad j\notin I\left(x^*\right), \tag{97}$$

which implies

$$\lim_{i\longrightarrow\infty} z_+^j\left(x^{k_i,i},\alpha^{k_i,i}\right)=0,\quad j\notin I\left(x^*\right). \tag{98}$$

By (94), $z_+^j(x^{k_i,i},\alpha^{k_i,i})\ge 0$ for $j\in I(x^*)$, and by the finiteness of indexes in $I(x^*)$, there exists a subsequence of $\{(x^{k_i,i},\alpha^{k_i,i})\}$ (denoted also by $\{(x^{k_i,i},\alpha^{k_i,i})\}$ for convenience) and $z_j^*\in[0,1]$ for $j\in I(x^*)$, such that

$$\begin{gathered}\sum_{j\in I(x^*)} z_j^*=1,\\ \lim_{i\longrightarrow\infty} z_+^j\left(x^{k_i,i},\alpha^{k_i,i}\right)=z_j^*\in[0,1],\\ j\in I\left(x^*\right).\end{gathered} \tag{99}$$

Therefore, by $x^{k_i,i}\longrightarrow x^*$, (93), (98), (99), and Assumption 1, we have

$$\begin{gathered}\sum_{j\in I(x^*)} z_j^*\nabla f_j\left(x^*\right)=0,\\ \sum_{j\in I(x^*)} z_j^*=1, z_j^*\ge 0, j\in I\left(x^*\right),\end{gathered} \tag{100}$$

which implies that x^* is a stationary point of problem (1). □

4. Numerical Experiments

In this section, we present the numerical results of Algorithm 1 and several related algorithms for solving unconstrained minimax problems. Algorithm 1 is recorded as ASSF. Fminimax is the MATLAB algorithm "fminimax". Fmincon is the MATLAB algorithm "fmincon" applied to

$$\min_{(x,\alpha)\in R^{n+1}}\left\{\alpha\mid f_j(x)-\alpha\le 0, j\in Q\right\}, \tag{101}$$

which is equivalent to problem (1). To show the efficiency of the proposed active set smoothing function, we replace it by some other smoothing techniques in Algorithm 1 to obtain several smoothing methods. AF, SSF, and SPF are constructed by Algorithm 1 with aggregate function (2), cubic spline smoothing function introduced in [28], and exact penalty function technique introduced in [18], respectively. TAF and ASAF are constructed by Algorithm 1 and the aggregate function with the active set strategies introduced in [25, 26], respectively.

The parameters in Algorithm 1 are set as follows:

$$\begin{gathered}\beta=0.8,\ \gamma=0.5;\ C_l=10^{-2},\ C_u=10^2,\ \hat{t}=10^{-3};\\ \kappa=10^{10},\ \kappa_1^P=\kappa_1=10^5,\ \kappa_2^P=\kappa_2=10^{-10},\ \kappa_3=0.25;\\ \omega_t=0.1,\ \omega_c=0.1,\ \tau\left(t^i\right)=10^{-3}.\end{gathered} \tag{102}$$

For the moderately sized test problems, t^0 and α^0 are set as follows:

$$\begin{gathered}t^0=\sum_{j\in Q} f_j\left(x^0\right)-q\min_{j\in Q} f_j\left(x^0\right)+1,\\ \alpha^0=\frac{\sum_{j\in Q} f_j\left(x^0\right)}{c^0 q}+\frac{t^0}{c^0}-\frac{t^0}{c^0 q},\end{gathered} \tag{103}$$

and then, we have

$$\begin{gathered}Q_{t^0}^{c^0}\left(x^0,\alpha^0\right)=Q,\\ \sum_{j\in Q}\left(1+\frac{f_j\left(x^0\right)-c^0\alpha^0}{t^0}\right)_+=1,\end{gathered} \tag{104}$$

which implies that $\nabla_x F_{t^0}^{c^0}(x^0,\alpha^0)$ is a convex combination of the gradients of all the component functions. For the test problems with very many component functions, t^0 is set as

$$t^0=\max\left\{1,\frac{q\left(F\left(x^0\right)-\min_{j\in Q} f_j\left(x^0\right)\right)}{10}\right\}. \tag{105}$$

$\alpha^0\in(F(x^0)-t^0,F(x^0)+t^0)$ is computed by the bisection method according to

$$\left|1-\sum_{j\in Q}\left(1+\frac{f_j\left(x^0\right)-c^0\alpha^0}{t^0}\right)_+\right|<0.1. \tag{106}$$

For the algorithm ASAF, the parameter ϵ in (10) is set as

$$\epsilon=0.8\left(F(x)-\min_{j\in Q} f_j(x)\right), \tag{107}$$

for the moderately sized test problems, and

$$\epsilon=0.4\left(F(x)-\min_{j\in Q} f_j(x)\right), \tag{108}$$

for the test examples with very many component functions. For the algorithm TAF, the parameter $\epsilon(t,q)$ in (12) is set as

$$\epsilon(t,q)=t\ln\left(\max\left\{1,\frac{(2\epsilon_1-\epsilon_2)(q-1)}{\epsilon_2},\left(\frac{2\epsilon_3+6\epsilon_1^2}{t-\epsilon_4}\right)\frac{(q-1)}{\epsilon_4}\right\}\right), \tag{109}$$

with $\epsilon_1=0.1$, $\epsilon_2=0.01$, $\epsilon_3=0.01$, and $\epsilon_4=0.1$. For the algorithms AF, ASAF, TAF, SSF, and SPF, the initial smoothing parameters are set to $t^0=1$. The termination criterion for the algorithm AF is set as

$$\begin{gathered}t<10^{-3},\\ \left\|\nabla F_t(x)\right\|\le 10^{-3}.\end{gathered} \tag{110}$$

Table 1: The CPU time for Examples 1–10.

Ex.	q	ASSF	SSF	SPF	TAF	ASAF	AF	Fminimax	Fmincon
1	10^5	**0.1035**	0.7598	1.2153	0.3283	0.2872	0.2924	34.7621	226.4573
	10^6	**0.8800**	7.5375	7.3612	3.2775	2.8821	3.0391	2204.3730	$fail^3$
	10^7	**10.7318**	75.3454	75.8656	32.6120	28.2999	31.4882	$fail^3$	$fail^3$
2	10^5	**0.0862**	0.0965	0.1222	0.0954	0.2447	0.0909	7.1100	116.8411
	10^6	**0.5647**	0.9765	2.8985	0.9576	3.1395	0.9704	116.5674	1491.4805
	10^7	**9.1149**	10.7622	26.5794	9.7821	32.9736	9.4956	1487.9868	$fail^3$
3	10^5	**1.4902**	4.4586	27.7835	4.4605	4.0761	7.7392	122.9883	$fail^2$
	10^6	**19.8937**	59.5392	839.3762	68.2482	55.0242	103.4272	3044.0625	$fail^2$
	10^7	**203.0458**	514.4458	3569.5139	676.4006	501.4630	926.5762	$fail^3$	$fail^2$
4	10^5	**0.3202**	0.5836	4.5430	0.6396	2.1124	2.1355	69.3814	$fail^2$
	10^6	**3.4366**	7.9853	44.6778	8.8451	27.3115	27.6770	1545.7716	$fail^2$
	10^7	**36.0206**	71.4938	645.1772	78.9203	230.0224	237.1365	$fail^3$	$fail^2$
5	10^5	**0.9396**	2.6478	9.6803	3.7163	3.2187	3.7483	230.2623	$fail^1$
	10^6	**16.4395**	31.1775	92.1263	45.2874	43.4843	45.5968	$fail^3$	$fail^1$
	10^7	**158.8988**	307.4791	1002.5981	432.1560	401.5995	436.5213	$fail^3$	$fail^1$
6	10^5	**0.6966**	3.7246	3.1238	4.4810	3.4686	4.5208	42.9713	$fail^3$
	10^6	**7.2618**	49.6754	46.5345	59.6989	46.1585	60.0291	2912.3607	$fail^3$
	10^7	**74.8332**	478.8652	455.7022	561.2579	425.4030	566.2980	$fail^3$	$fail^3$
7	10^5	**1.1056**	7.0048	13.4588	3.2790	9.3249	3.2450	169.5213	$fail^2$
	10^6	**26.3445**	74.6292	177.1446	35.2551	99.2012	34.5205	$fail^3$	$fail^2$
	10^7	**123.4221**	749.7656	1834.9277	343.0618	984.0483	341.2569	$fail^3$	$fail^2$
8	10^5	**0.9361**	1.8597	8.1073	1.3515	2.6522	1.5489	157.1865	$fail^2$
	10^6	**12.6151**	25.1838	86.3288	19.5087	35.3458	20.1976	$fail^3$	$fail^2$
	10^7	**108.6392**	236.4630	949.2546	184.0629	328.1475	189.5273	$fail^3$	$fail^2$
9	10^5	**0.4863**	1.5641	10.0722	1.4539	0.7162	1.4945	21.4071	$fail^3$
	10^6	**8.0179**	18.8813	174.8562	18.4460	9.4945	18.7385	283.0764	$fail^3$
	10^7	**58.8060**	177.5352	2275.1924	162.9317	101.1354	165.5642	3543.2556	$fail^3$
10	10^3	**0.9273**	1.9002	1.1572	14.5336	28.9550	30.5158	1.3285	$fail^2$
	10^4	**0.6413**	19.4758	10.9090	97.0372	177.8060	186.0173	14.5788	$fail^2$
	10^5	**1.6718**	183.3566	129.7202	471.6195	792.5688	809.8969	131.9020	$fail^2$

The termination criteria for the algorithms ASSF, SSF, SPF, ASAF, and TAF are set as

$$t < 10^{-3}, \quad \left\|\nabla \mathscr{F}_t(x)\right\| \le 10^{-3}, \tag{111}$$

or

$$F(x) \le F(x^*_{AF}), \tag{112}$$

where $\nabla \mathscr{F}_t(x)$ represents the gradient of the smoothing function with respect to the variable x and x^*_{AF} is the approximation solution computed by the algorithm AF. The numerical results were obtained by running MATLAB R2014a on a laptop with Inter(R) Core(TM) i5-7300HQ CPU 2.50GHZ and 4.00 GB memory.

We carry out a comparison on three categories of test problems described in the Appendix. The first category of problems, Examples 1–10, emanates from the discretized semi-infinite minimax problems, and the number of the component functions is at least 1000. The second category of problems, Example 11, possesses many variables and many component functions. The third category of problems, Examples 12–45, is composed by various moderately sized test problems. Tables 1–3 list the CPU time; Tables 4–6 list the number of function evaluations and iterations; Tables 7 and 8 list the average proportion of the component functions used in the active set strategy; the word $fail^1$ means that the stepsize cannot be computed in the region $[10^{-10}, 1]$; the word $fail^2$ means that the number of iterations in Fminimax or Fmincon reaches the upper limit; the word $fail^3$ means that the CPU time exceeds 3600 seconds. In order to make the advantages of Algorithm 1 clearer and more explicit, the corresponding Dolan–Morée performance profiles proposed in [32] are shown in Figures 1–3 for three categories of examples above.

For all the test problems with very many component functions, we see that Algorithm 1 is predominantly faster than other algorithms from Tables 1 and 2 and Figures 1 and 2, the proposed active set strategy results in more significant reduction of gradient evaluations than the active set strategies in [18, 25, 26, 28] from Tables 7 and 8, and Fminimax and Fmincon have poor stability and low efficiency. For most moderately sized test problems, we see from Tables 3 and 6 and Figure 3 that Algorithm 1 requires fewer iterations and function evaluations and takes less CPU time than the other algorithms considered.

Table 2: The CPU time for Example 11.

n	q	ASSF	SSF	SPF	TAF	ASAF	AF	Fminimax	Fmincon
10^2	10^2	**0.5275**	0.7747	12.2677	0.7556	0.7715	0.7928	464.3152	$fail^3$
10^2	10^3	**2.7369**	15.8849	76.2165	14.4669	15.1963	15.2076	$fail^3$	$fail^3$
10^2	10^4	**25.7550**	587.5630	$fail^3$	533.8570	615.3645	665.1596	$fail^3$	$fail^3$
10	10^3	**0.2127**	2.4103	2.8646	2.3061	2.3695	2.4319	2.8159	20.2106
10^2	10^3	**2.7141**	12.0482	74.3301	10.7229	10.0013	11.2506	$fail^3$	$fail^3$
10^3	10^3	**806.9206**	2286.4695	$fail^3$	2186.7536	2098.2569	2304.5143	$fail^3$	$fail^3$

Table 3: The CPU time for Examples 12–45.

Ex.	ASSF	SSF	SPF	TAF	ASAF	AF	Fminimax	Fmincon
12	0.0018	0.0026	0.1070	**0.0013**	0.0023	0.0017	0.0636	0.0569
13	**0.0011**	0.0012	0.0167	0.0021	0.0037	0.0045	0.0931	0.0578
14	0.2468	0.4617	0.6172	0.2469	0.4213	**0.0013**	0.0959	$fail^2$
15	0.0190	0.0132	0.1243	**0.0087**	0.0117	0.0125	1.4777	$fail^2$
16	0.0437	0.0316	2.3675	0.0298	0.0353	**0.0297**	5.8776	$fail^2$
17	**0.0099**	0.0193	0.3882	0.0129	0.0137	0.0193	0.8454	0.1955
18	**0.0376**	0.0417	0.8257	0.0839	0.0849	0.0702	1.8279	$fail^2$
19	**0.1895**	0.7442	0.5214	0.8614	0.9749	0.8750	0.5122	0.6373
20	**1.8828**	3.6656	3.0255	3.5755	3.8257	3.5837	3.0208	3.0462
21	**0.0054**	0.4821	$fail^1$	0.0178	0.0229	0.0198	0.0845	0.1491
22	**0.0074**	0.1384	3.0235	0.0169	0.0312	0.0228	0.0321	0.0607
23	**0.0113**	0.8615	0.3981	0.0432	0.0588	0.0481	0.0897	0.2281
24	**0.0026**	0.0881	0.0248	0.0073	0.0092	0.0077	0.0211	0.0360
25	**1.7941**	16.1289	6.0794	6.2755	11.2465	12.2501	6.1201	6.2704
26	**0.0006**	0.0057	0.0096	0.0190	0.0088	0.0051	0.1165	0.0926
27	0.0990	0.1668	1.7366	0.8320	1.1020	0.7201	0.4449	**0.0894**
28	**0.0018**	0.0055	0.0147	0.0038	0.0044	0.0032	0.0126	0.0179
29	0.0010	0.0022	0.0126	0.0011	0.0027	**0.0007**	0.0180	0.0247
30	**0.0015**	0.0036	0.0093	0.0016	0.0040	0.0025	0.0101	0.0170
31	**0.0005**	0.0010	0.0014	0.0009	0.0007	0.0006	0.0177	0.0185
32	**0.0018**	0.0055	0.1108	0.0041	0.0055	0.0038	0.0101	0.0193
33	**0.0017**	0.0089	0.0165	0.0038	0.0060	0.0040	0.0101	0.0174
34	**0.0004**	0.0017	0.0031	0.0016	0.0019	0.0015	0.0106	0.0184
35	**0.0018**	0.0061	0.0206	0.0051	0.0063	0.0047	0.0087	0.0184
36	**0.0008**	0.0026	0.0093	0.0018	0.0028	0.0018	0.0082	0.0148
37	**0.0022**	0.0060	0.0210	0.0076	0.0097	0.0073	0.0160	0.0310
38	0.0030	0.0053	0.5716	0.0027	0.0037	**0.0018**	0.0543	0.1928
39	**0.0038**	0.0090	0.0263	0.0106	0.0089	0.0064	0.0327	0.0910
40	0.0228	0.0291	0.1652	0.0611	0.1229	0.1114	**0.0113**	0.9211
41	**0.0001**	0.0053	0.0061	0.0047	0.0064	0.0051	0.0145	0.0203
42	**0.0018**	0.0093	0.0127	0.0064	0.0176	0.0094	0.0068	0.8627
43	**0.0022**	0.0046	0.0590	0.0050	0.0063	0.0047	0.0157	0.0216
44	**0.0045**	0.0107	0.0505	0.0117	0.0157	0.0122	0.0168	0.0572
45	**0.0331**	$fail^1$	2.6303	0.5594	0.1648	21.6874	0.0681	$fail^1$

Table 4: The number of function evaluations and iterations for Examples 1–10.

Ex.	q	ASSF	SSF	SPF	TAF	ASAF	AF
1	10^5	(**34**, 23)	(67, **14**)	(188, 77)	(93, 15)	(79, **14**)	(84, 15)
	10^6	(**27**, 20)	(246, 22)	(88, 34)	(85, 15)	(79, **14**)	(84, 15)
	10^7	(**33**, 23)	(343, 21)	(97, 40)	(83, **14**)	(79, **14**)	(84, 15)
2	10^5	(23, 18)	(30, 13)	(**19**, 19)	(32, **10**)	(55, 13)	(32, **10**)
	10^6	(**23**, 18)	(34, 15)	(60, 35)	(32, **10**)	(55, 13)	(32, **10**)
	10^7	(32, 22)	(34, 13)	(**16**, 16)	(32, **10**)	(55, 13)	(32, **10**)
3	10^5	(**242**, **101**)	(564, 118)	(1381, 928)	(737, 219)	(580, 109)	(812, 262)
	10^6	(**238**, **103**)	(761, 137)	(4852, 4518)	(798, 245)	(583, 108)	(1039, 266)
	10^7	(**239**, **103**)	(765, 141)	(4852, 4518)	(804, 249)	(583, 108)	(819, 262)

Table 4: Continued.

Ex.	q	ASSF	SSF	SPF	TAF	ASAF	AF
	10^5	(**60**, 50)	(242, 53)	(808, 385)	(503, **33**)	(466, 35)	(610, 47)
4	10^6	(**66**, 58)	(270, 59)	(1008, 258)	(657, 41)	(550, **38**)	(614, 48)
	10^7	(**71**, 63)	(296, 68)	(1408, 271)	(632, 46)	(634, **40**)	(599, 47)
	10^5	(84, 58)	(**76**, **45**)	(407, 233)	(281, 65)	(277, 63)	(284, 67)
5	10^6	(93, 61)	(**87**, **50**)	(407, 233)	(280, 65)	(276, 63)	(284, 68)
	10^7	(**89**, 59)	(98, **55**)	(407, 233)	(280, 65)	(276, 63)	(284, 68)
	10^5	(**53**, **39**)	(293, 216)	(274, 95)	(296, 229)	(231, 231)	(299, 231)
6	10^6	(**53**, **39**)	(293, 216)	(396, 124)	(296, 229)	(231, 231)	(299, 231)
	10^7	(**53**, **39**)	(293, 216)	(426, 123)	(296, 229)	(231, 231)	(299, 231)
	10^5	(**19**, **16**)	(130, 28)	(379, 111)	(55, 17)	(342, 68)	(55, 17)
7	10^6	(67, 39)	(128, 33)	(498, 138)	(**55**, **17**)	(362, 72)	(**55**, **17**)
	10^7	(**44**, 19)	(125, 31)	(507, 137)	(55, **17**)	(364, 74)	(55, **17**)
	10^5	(**66**, **39**)	(145, 57)	(606, 180)	(214, **39**)	(550, 52)	(215, 40)
8	10^6	(**60**, **38**)	(131, 61)	(567, 180)	(216, 40)	(572, 54)	(216, 40)
	10^7	(**54**, **33**)	(137, 56)	(801, 227)	(215, 40)	(593, 56)	(215, 40)
	10^5	(79, **14**)	(47, 21)	(434, 152)	(48, 18)	(**39**, 15)	(48, 18)
9	10^6	(79, 17)	(59, 21)	(677, 200)	(48, 18)	(**39**, **15**)	(48, 18)
	10^7	(76, **11**)	(49, 20)	(935, 270)	(48, 18)	(**39**, 15)	(48, 18)
	10^3	(745, **63**)	(1071, 103)	(**243**, 107)	(107139, 4270)	(111873, 4454)	(135709, 5096)
10	10^4	(**74**, **53**)	(1094, 105)	(278, 120)	(134725, 5136)	(137748, 5240)	(136931, 5213)
	10^5	(**69**, **49**)	(2477, 495)	(423, 140)	(122067, 4649)	(112489, 4968)	(122803, 4677)

Table 5: The number of function evaluations and iterations for Example 11.

n	q	ASSF	SSF	SPF	TAF	ASAF	AF
10^2	10^2	(212, 41)	(196, 38)	(1110, 919)	(180, **32**)	(174, 31)	(**175**, **32**)
10^2	10^3	(**245**, **55**)	(561, 65)	(1010, 378)	(553, 62)	(557, 64)	(557, 64)
10^2	10^4	(**318**, **85**)	(1952, 143)	$fail^3$	(1707, 114)	(1888, 139)	(2857, 166)
10	10^3	(**208**, **45**)	(392, 49)	(5658, 249)	(391, 49)	(392, 49)	(395, 48)
10^2	10^3	(**228**, **47**)	(464, 59)	(983, 381)	(459, 56)	(561, 62)	(459, 56)
10^3	10^3	(**255**, **59**)	(562, 63)	$fail^3$	(563, 65)	(559, 60)	(559, 60)

Table 6: The number of function evaluations and iterations for Examples 12–45.

Ex.	ASSF	SSF	SPF	TAF	ASAF	AF
12	(26, 19)	(29, 13)	(56, 41)	(**24**, **11**)	(27, 12)	(36, 16)
13	(9, **5**)	(**5**, **5**)	(93, 48)	(7, **5**)	(6, **5**)	(7, **5**)
14	(5054, 2352)	(5024, 2342)	(6058, 6052)	(5195, 2341)	(5027, 2335)	(**33**, **9**)
15	(12, 6)	(**5**, **5**)	(135, 76)	(7, **5**)	(6, **5**)	(7, **5**)
16	(14, 6)	(**6**, 6)	(494, 264)	(7, **5**)	(7, 6)	(7, **5**)
17	(13, 6)	(**4**, **4**)	(316, 143)	(10, 5)	(10, 5)	(11, 6)
18	(14, **6**)	(**6**, **6**)	(180, 92)	(22, 9)	(22, 7)	(16, **6**)
19	(5466, **213**)	(11827, 582)	(**664**, 586)	(23561, 649)	(23700, 649)	(24435, 671)
20	(41561, **2428**)	(46979, 2844)	(**3103**, 3056)	(88377, 2981)	(87642, 2846)	(89606, 3016)
21	(**69**, **36**)	(3296, 3186)	$fail^1$	(618, 40)	(655, 42)	(649, 47)
22	(**121**, **40**)	(2508, 353)	(22442, 21961)	(695, 55)	(844, 61)	(740, 58)
23	(**68**, **41**)	(12151, 694)	(4623, 1197)	(1277, 55)	(1339, 81)	(1181, 82)
24	(**21**, **19**)	(638, 279)	(203, 116)	(121, 27)	(111, 26)	(119, 30)
25	(26994, **1525**)	(101048, 5186)	(**5215**, 5126)	(135661, 5598)	(220821, 7685)	(261684, 8977)
26	(**9**, **5**)	(22, 11)	(65, 44)	(151, 119)	(145, 13)	(145, 13)
27	(**237**, **237**)	(356, 239)	(3691, 2763)	(16060, 878)	(15714, 852)	(10800, 622)
28	(**30**, **18**)	(148, 21)	(164, 135)	(104, 19)	(92, 19)	(95, 19)
29	(13, 11)	(26, 10)	(105, 104)	(11, **8**)	(12, 10)	(**8**, **8**)
30	(**31**, 15)	(104, 14)	(84, 71)	(45, **10**)	(71, 16)	(71, 16)
31	(8, 8)	(8, 6)	(7, 9)	(7, 6)	(**6**, **4**)	(10, 6)

Table 6: Continued.

Ex.	ASSF	SSF	SPF	TAF	ASAF	AF
32	(**28**, **18**)	(178, **18**)	(1145, 1053)	(119, 22)	(115, 20)	(119, 22)
33	(**27**, **18**)	(270, 22)	(256, 117)	(112, 19)	(118, 25)	(118, 25)
34	(**4**, **4**)	(19, 10)	(33, 22)	(30, 11)	(44, 7)	(39, 13)
35	(**30**, **17**)	(209, 19)	(259, 175)	(171, 20)	(150, 21)	(153, 24)
36	(**12**, 8)	(42, 10)	(39, 21)	(56, **7**)	(56, 10)	(53, 10)
37	(**25**, 21)	(158, **18**)	(225, 163)	(228, 26)	(214, 25)	(218, 29)
38	(**21**, 13)	(64, 12)	(2090, 2073)	(38, 8)	(33, 10)	(20, **7**)
39	(**31**, 22)	(142, 22)	(174, 114)	(262, **17**)	(103, 22)	(105, 24)
40	(966, **45**)	(**592**, 90)	(4612, 232)	(2341, 192)	(4278, 214)	(4549, 213)
41	(**18**, **18**)	(165, 39)	(53, 38)	(97, 37)	(107, 26)	(119, 38)
42	(**26**, **15**)	(236, 36)	(180, 78)	(158, 44)	(469, 52)	(306, 41)
43	(**23**, **21**)	(165, 46)	(601, 502)	(175, 49)	(175, 28)	(167, 42)
44	(**36**, **28**)	(228, 52)	(360, 228)	(307, 53)	(339, 32)	(304, 46)
45	(**195**, **26**)	$fail^1$	(6172, 3109)	(1145, 35)	(4754, 174)	(225933, 4256)

Table 7: The proportion of the component functions used in the active set strategy for Examples 1–10.

Ex.	q	ASSF	SSF	SPF	TAF	ASAF
	10^5	**0.4992**	0.7292	0.6553	0.7582	0.9666
1	10^6	**0.4241**	0.7280	0.6497	0.7666	0.9666
	10^7	**0.4987**	0.7001	0.6178	0.8056	0.9666
	10^5	**0.4264**	0.7737	0.8901	0.7370	0.8956
2	10^6	**0.4264**	0.7528	0.8811	0.7497	0.8956
	10^7	**0.5306**	0.7793	0.8820	0.7616	0.8956
	10^5	0.8052	0.9965	**0.7298**	0.9848	0.9948
3	10^6	0.8069	0.9204	**0.5178**	0.9870	0.9947
	10^7	0.8069	0.9129	**0.5178**	0.9880	0.9946
	10^5	**0.1952**	0.3062	0.7900	0.6262	0.9549
4	10^6	**0.1672**	0.3020	0.7049	0.6334	0.9584
	10^7	**0.1540**	0.3009	0.7037	0.6301	0.9543
	10^5	**0.4633**	0.8963	0.9309	0.8392	0.9843
5	10^6	**0.4878**	0.8325	0.9309	0.8429	0.9843
	10^7	**0.4711**	0.8022	0.9308	0.8468	0.9843
	10^5	**0.6521**	0.8693	0.9028	0.9917	0.9942
6	10^6	**0.6521**	0.8356	0.9028	0.9919	0.9942
	10^7	**0.6520**	0.8260	0.8627	0.9920	0.9942
	10^5	**0.6635**	0.8454	0.8604	0.9935	0.9693
7	10^6	**0.6440**	0.8830	0.8073	0.9932	0.9710
	10^7	**0.6440**	0.8323	0.8376	0.9914	0.9691
	10^5	**0.7163**	0.9796	0.9090	0.8775	0.9975
8	10^6	**0.7088**	0.8478	0.9092	0.8760	0.9977
	10^7	**0.3990**	0.8084	0.8579	0.8850	0.9979
	10^5	**0.4992**	0.8745	0.8678	0.9997	0.5148
9	10^6	**0.4012**	0.8803	0.8509	0.9995	0.5148
	10^7	**0.3903**	0.8353	0.8345	0.9995	0.5148
	10^3	0.0382	0.5973	0.6395	**0.0289**	0.0335
10	10^4	0.5612	0.6108	0.6191	**0.0282**	0.0337
	10^5	0.4125	0.6088	0.6091	**0.0290**	0.0329

Table 8: The proportion of the component functions used in the active set strategy for Example 11.

n	q	ASSF	SSF	SPF	TAF	ASAF
10^2	10^2	**0.2741**	0.7344	0.5394	0.8765	0.9228
10^2	10^3	**0.2212**	0.7226	0.5087	0.8088	0.9227
10^2	10^4	**0.3030**	0.7951	$fail^3$	0.8364	0.9227
10	10^3	**0.1969**	0.8133	0.5097	0.8009	0.8998
10^2	10^3	**0.2570**	0.7975	0.5145	0.8371	0.8984
10^3	10^3	**0.2477**	0.7622	$fail^3$	0.8584	0.8977

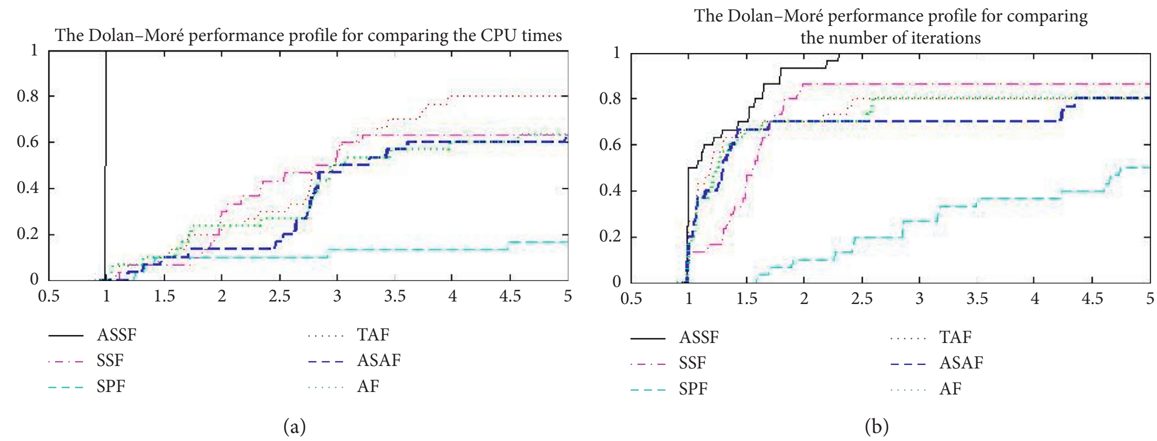

Figure 1: The Dolan–Morée performance profile of Examples 1–10.

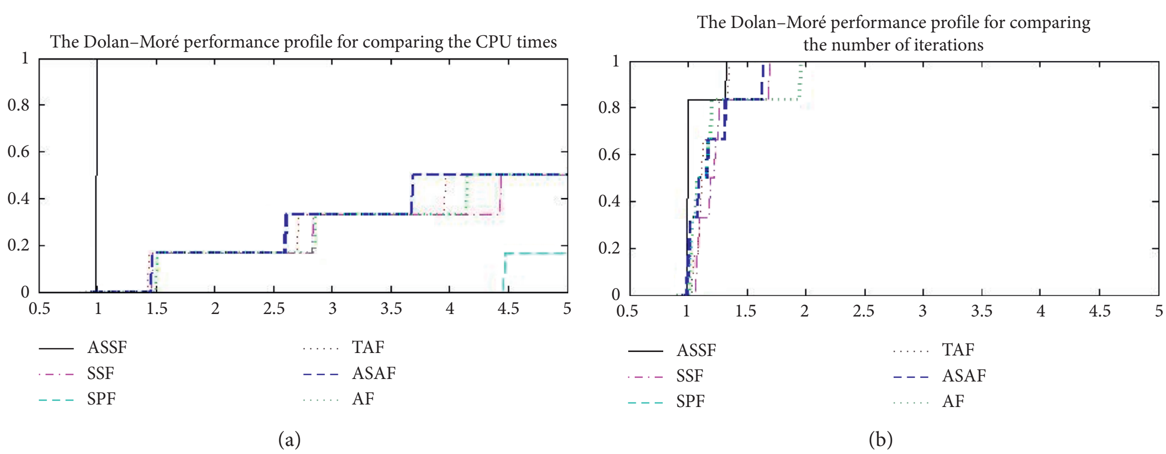

Figure 2: The Dolan–Morée performance profile of Examples 11.

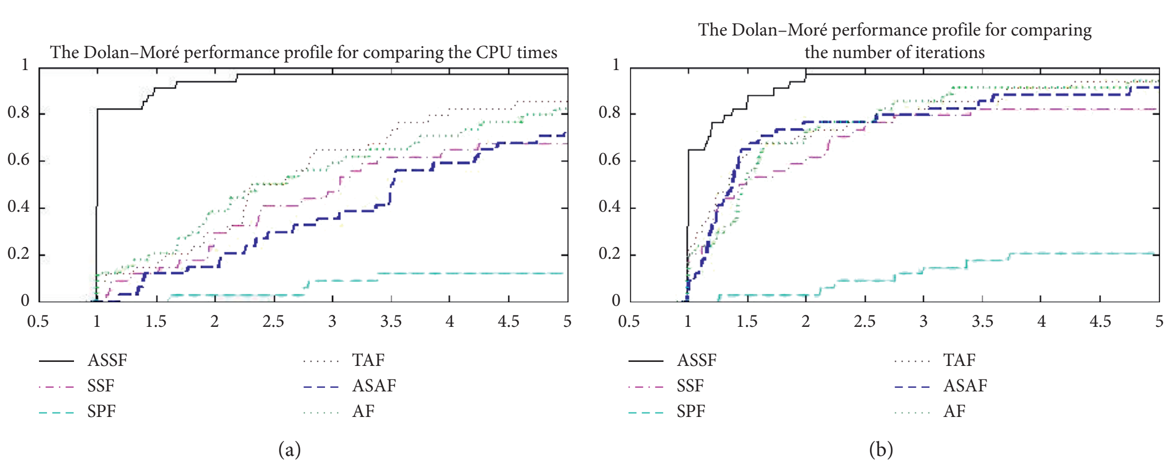

Figure 3: The Dolan–e Moré performance profile of Examples 12–25.

5. Conclusion

We have proposed an active set smoothing function for the maximum function by using the plus function and an active set smoothing method with convergence analysis for solving unconstrained minimax problems. The active smoothing function can be simply implemented in the smoothing methods. Compared with the similar smoothing algorithms based on other smoothing techniques, and the algorithms in the MATLAB environment, the proposed algorithm is competitive for wide moderately sized problems and dramatically efficient for the problems with very many component functions.

Appendix

Example 1 (see [24])

$$\begin{aligned} F(x) &= \max_{1\le j\le q} f_j(x), \\ f_j(x) &= \sin y_j - \left(x_3 y_j^2 + x_2 y_j + x_1\right), \quad j = 1,\ldots,\frac{q}{2}, \\ f_j(x) &= -f_{j-q/2}(x), \quad j = \frac{q}{2+1},\ldots,q, \\ y_j &= \frac{(j-1)}{(q/2-1)}, \quad j = 1,\ldots,\frac{q}{2}, \end{aligned} \tag{A.1}$$

$n = 3, x^0 = (1,1,1)$.

Example 2 (see [24])

$$\begin{aligned} F(x) &= \max_{1\le j\le q} f_j(x), \\ f_j(x) &= \left(2y_j^2 - 1\right)x + y_j\left(1 - y_j\right)(1-x), \quad j = 1,\ldots,\frac{q}{2}, \\ f_j(x) &= -f_{j-q/2}(x), \quad j = \frac{q}{2+1},\ldots,q, \\ y_j &= \frac{(j-1)}{(q/2-1)}, \quad j = 1,\ldots,\frac{q}{2}, \end{aligned} \tag{A.2}$$

$n = 1, x^0 = 5$.

Example 3 (see [24])

$$\begin{aligned} F(x) &= \max_{1\le j\le q} f_j(x), \\ f_j(x) &= \sqrt{y_j} - \left(x_4 - \left(x_1 y_j^2 + x_2 y_j + x_3\right)^2\right), \quad j = 1,\ldots,\frac{q}{2}, \\ f_j(x) &= -f_{j-q/2}(x), \quad j = \frac{q}{2+1},\ldots,q, \\ y_j &= 0.25 + 0.75\frac{(j-1)}{(q/2-1)}, \quad j = 1,\ldots,\frac{q}{2}, \end{aligned} \tag{A.3}$$

$n = 4, x^0 = (1,1,1,1)$.

Example 4 (see [33])

$$\begin{aligned} F(x) &= \max_{1\le j\le q} f_j(x), \\ f_j(x) &= \left(x_1 + x_2 y_j - \exp\left(y_j\right)\right)^2 + \left(x_3 + x_4 \sin y_j - \cos y_j\right)^2, \\ &\qquad j = 1,\ldots,\frac{q}{2}, \\ f_j(x) &= -f_{j-q/2}(x), \quad j = \frac{q}{2+1},\ldots,q, \\ y_j &= 4\frac{(j-1)}{(q/2-1)}, \quad j = 1,\ldots,\frac{q}{2}, \end{aligned} \tag{A.4}$$

$n = 4, x^0 = (25, 5, -5, -1)$.

Example 5 (see [34])

$$\begin{aligned} F(x) &= \max_{1\le j\le q} f_j(x), \\ f_j(x) &= \frac{x_1 + x_2 y_j + x_3 y_j^2}{1 + x_4 y_j + x_5 y_j^2} - \frac{\sqrt{\left(8y_j - 1\right)^2 + 1}\arctan\left(8y_j\right)}{8y_j}, \\ &\qquad j = 1,\ldots,\frac{q}{2}, \\ f_j(x) &= -f_{j-q/2}(x), \quad j = \frac{q}{2+1},\ldots,q, \\ y_j &= -1 + 2\frac{(j-1)}{(q/2-1)}, \quad j = 1,\ldots,\frac{q}{2}, \end{aligned} \tag{A.5}$$

$n = 5, x^0 = (0, -1, 10, 1, 10)$.

Example 6 (see [34])

$$\begin{aligned} F(x) &= \max_{1\le j\le q} f_j(x), \\ f_j(x) &= \frac{x_3}{x_2}\exp\left(-y_j x_1\right)\sin\left(y_j x_2\right) - \left(\frac{3}{20}\exp\left(-y_j\right) + \frac{1}{52}\exp\left(-5y_j\right)\right. \\ &\quad \left. - \frac{1}{65}\exp\left(-2y_j\right)\left(3\sin\left(2y_j\right) + 11\cos\left(2y_j\right)\right)\right), \quad j = 1,\ldots,\frac{q}{2}, \\ f_j(x) &= -f_{j-q/2}(x), \quad j = \frac{q}{2+1},\ldots,q, \\ y_j &= \frac{10(j-1)}{(q/2-1)}, \quad j = 1,\ldots,\frac{q}{2}, \end{aligned} \tag{A.6}$$

$n = 3, x^0 = (1,1,1)$.

Example 7 (see [35])

$$
\begin{aligned}
F(x) &= \max_{1\le j\le q} f_j(x),\\
f_j(x) &= \frac{x_1 + x_2 y_j}{1 + x_3 y_j + x_4 y_j^2 + x_5 y_j^3} - \exp(y_j), \quad j = 1,\ldots,\frac{q}{2},\\
f_j(x) &= -f_{j-q/2}(x), \quad j = \frac{q}{2+1},\ldots,q,\\
y_j &= -1 + 2\frac{(j-1)}{(q/2-1)}, \quad j = 1,\ldots,\frac{q}{2},
\end{aligned}
\tag{A.7}
$$

$n = 5$, $x^0 = (0.5, 0, 0, 0, 0)$.

Example 8 (see [13])

$$
\begin{aligned}
F(x) &= \max_{1\le j\le q} f_j(x),\\
f_j(x) &= x_1 \exp(x_3 y_j) + x_2 \exp(x_4 y_j) - \frac{1}{1+y_j}, \quad j = 1,\ldots,\frac{q}{2},\\
f_j(x) &= -f_{j-q/2}(x), \quad j = \frac{q}{2+1},\ldots,q,\\
y_j &= -0.5 + \frac{(j-1)}{(q/2-1)}, \quad j = 1,\ldots,\frac{q}{2},
\end{aligned}
\tag{A.8}
$$

$n = 4$, $x^0 = (1, 1, -3, -1)$.

Example 9 (see [36])

$$
\begin{aligned}
F(x) &= \max_{1\le j\le q} f_j(x),\\
f_j(x) &= x_1^2 \exp(-x_2 y_j)\cos^2(x_3 y_j + x_4) - \cos(y_j)\\
&\quad + x_2^2 x_3^2 \exp(-x_1 y_j)\sin^2(x_2 y_j)\\
&\quad + \exp\left((1-x_6)^2 y_j + x_5^2\right), \quad j = 1,\ldots,q,\\
y_j &= \frac{10(j-1)}{(q-1)}, \quad j = 1,\ldots,q,
\end{aligned}
\tag{A.9}
$$

$n = 6$, $x^0 = (1, 1, 1, 1, 1, 1)$.

Example 10 (see [36])

$$
\begin{aligned}
F(x) &= \max_{1\le j\le q} f_j(x),\\
f_j(x) &= \frac{x_3}{x_2}\exp(-y_j x_1)\sin(y_j x_2) + x_1 \exp(-x_2 y_j)\cos(x_3 y_j + x_4) + x_5 \exp(-x_6 y_j)\\
&\quad -\left(\frac{3}{20}\exp(-y_j) + \frac{1}{52}\exp(-5y_j) - \frac{1}{65}\exp(-2y_j)\left(3\sin(2y_j) + 11\cos(2y_j)\right) + \frac{1}{2}\exp(-y_j)\right.\\
&\quad \left. -\exp(-2y_j) + \frac{1}{2}\exp(-3y_j) + \frac{3}{2}\exp\left(-\frac{3}{2}y_j\right)\sin(7y_j) + \exp\left(-\frac{5}{2}y_j\right)\sin(5y_j)\right), \quad j = 1,\ldots,\frac{q}{2},\\
f_j(x) &= -f_{j-q/2}(x), \quad j = \frac{q}{2+1},\ldots,q,\\
y_j &= \frac{10(j-1)}{(q/2-1)}, \quad j = 1,\ldots,\frac{q}{2},
\end{aligned}
\tag{A.10}
$$

$n = 6$, $x^0 = (2, 2, 7, 0, -2, 1)$.

Example 11 (see [18])

$$
\begin{aligned}
F(x) &= \max_{1\le j\le q} f_j(x),\\
f_j(x) &= \frac{1}{2}x^T A_j x + B_j^T x + C_j,\\
&\qquad j = 1,\ldots,q,
\end{aligned}
\tag{A.11}
$$

where $A_j \in R^{n\times n}$, $j = 1,\ldots,q$, is symmetric positive definite, and $B_j \in R^n$, $j = 1,\ldots,q$, and $C_j \in R$, $j = 1,\ldots,q$, are all generated randomly as normal distribution. $x^0 = ((1/n), (1/n), \ldots, (1/n))$.

Example 12 (see [24])

$$
\begin{aligned}
F(x) &= \max_{1\le j\le 3} f_j(x),\\
f_1(x) &= (x_1)^2 + (x_2)^4,\\
f_2(x) &= (2-x_1)^2 + (2-x_2)^2,\\
f_3(x) &= 2\exp(-x_1 + x_2),
\end{aligned}
\tag{A.12}
$$

$n = 2$, $x^0 = (0, 0)$.

Example 13 (see [24]).

$$F(x) = \max_{1\le j\le 20} f_j(x),$$
$$f_j(x) = x_j^2, \tag{A.13}$$
$$j = 1, \dots, 20.$$

$n = 20$, $x^0 = (0.1, 0.2, \dots, 1, -1.1, -1.2, \dots, -2)$.

Example 14 (see [24]).

$$F(x) = \max_{1\le j\le 2} f_j(x),$$
$$f_1(x) = \left(x_1 - \sqrt{x_1^2 + x_2^2}\cos\left(x_1^2 + x_2^2\right)\right)^2 + 0.005\left(x_1^2 + x_2^2\right),$$
$$f_2(x) = \left(x_2 - \sqrt{x_1^2 + x_2^2}\sin\left(x_1^2 + x_2^2\right)\right)^2 + 0.005\left(x_1^2 + x_2^2\right), \tag{A.14}$$

$n = 2$, $x^0 = (1.41831, -4.79462)$.

Example 15 (see [24]).

$$F(x) = \max_{1\le j\le 100} f_j(x),$$
$$f_j(x) = x_j^2, \tag{A.15}$$
$$j = 1, \dots, 100,$$

$n = 100$, $x^0 = (0.1, 0.2, \dots, 1, -1.1, -1.2, \dots, -2)$.

Example 16 (see [24]).

$$F(x) = \max_{1\le j\le 200} f_j(x),$$
$$f_j(x) = x_j^2, \tag{A.16}$$
$$j = 1, \dots, 200,$$

$n = 200$, $x^0 = (0.1, 0.2, \dots, 1, -1.1, -1.2, \dots, -2)$.

Example 17 (see [24]).

$$F(x) = \max_{1\le j\le 50} f_j(x),$$
$$f_j(x) = x_{2(j-1)+1}^2 + x_{2j}^2, \tag{A.17}$$
$$j = 1, \dots, 50,$$

$n = 100$, $x^0 = (0.02, 0.04, \dots, 1, -1.02, -1.04, \dots, -2)$.

Example 18 (see [24]).

$$F(x) = \max_{1\le j\le 50} f_j(x),$$
$$f_j(x) = x_{4(j-1)+1}^2 + x_{4(j-1)+2}^2 + x_{4(j-1)+3}^2 + x_{4j}^2, \tag{A.18}$$
$$j = 1, \dots, 50,$$

$n = 200$, $x^0 = (0.01, 0.02, \dots, 1, -1.01, -1.02, \dots, -2)$.

Example 19. (see [33]).

$$F(x) = \max_{0\le j\le 30} f_j(x),$$
$$f_j(x) = x_1 + \frac{j}{(16-j)x_2 + \min(j, 16-j)x_3} - y_j, \quad j = 1, \dots, 15,$$
$$f_j(x) = -f_{j-15}(x), \quad j = 16, \dots, 30, \tag{A.19}$$
$$y = (0.14, 0.18, 0.22, 0.25, 0.29, 0.32, 0.35, 0.39, 0.37, 0.58, 0.73, 0.96, 1.34, 2.10, 4.39),$$

$n = 3$, $x^0 = (1, 1, 1)$.

Example 20 (see [33]).

$$F(x) = \max_{0\le j\le 22} f_j(x),$$
$$f_j(x) = \frac{x_1\left(u_j^2 + x_2 u_j\right)}{u_j^2 + x_3 u_j + x_4} - y_j, \quad j = 1, \dots, 11,$$
$$f_j(x) = -f_{j-11}(x), \quad j = 12, \dots, 22, \tag{A.20}$$
$$y = (0.1951, 0.1947, 0.1735, 0.1600, 0.0844, 0.0627, 0.0456, 0.0342, 0.0323, 0.0235, 0.0246),$$
$$u = (4.0000, 2.0000, 1.0000, 0.5000, 0.2500, 0.1670, 0.1250, 0.1000, 0.0833, 0.0714, 0.0625),$$

$n = 4$, $x^0 = (0.250, 0.390, 0.415, 0.390)$.

Example 21 (see [37]).

$$F(x) = \max_{1\le j\le 5} f_j(x),$$
$$f_1(x) = (x_1-10)^2 + 5(x_2-12)^2 + x_3^4 + 3(x_4-11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7,$$
$$f_2(x) = f_1(x) + 10\left(2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 - 127\right),$$
$$f_3(x) = f_1(x) + 10\left(7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 - 282\right),$$
$$f_4(x) = f_1(x) + 10\left(23x_1 + x_2^2 + 6x_6^2 - 8x_7 - 196\right),$$
$$f_5(x) = f_1(x) + 10\left(4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7\right). \tag{A.21}$$

$n = 7, x^0 = (1, 2, 0, 4, 0, 1, 1)$.

Example 22 (see [37]).

$$F(x) = \max_{1\le j\le 9} f_j(x),$$
$$f_1(x) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3-10)^2 + 4(x_4-5)^2 + (x_5-3)^2 +$$
$$+ 2(x_6-1)^2 + 5x_7^2 + 7(x_8-11)^2 + 2(x_9-10)^2 + (x_{10}-7)^2 + 45,$$
$$f_2(x) = f_1(x) + 10\left(3(x_1-2)^2 + 4(x_2-3)^2 + 2x_3^2 - 7x_4 - 120\right),$$
$$f_3(x) = f_1(x) + 10\left(5x_1^2 + 8x_2 + (x_3-6)^2 - 2x_4 - 40\right),$$
$$f_4(x) = f_1(x) + 10\left(0.5(x_1-8)^2 + 2(x_2-4)^2 + 3x_5^2 - x_6 - 30\right),$$
$$f_5(x) = f_1(x) + 10\left(x_1^2 + 2(x_2-2)^2 - 2x_1x_2 + 14x_5 - 6x_6\right),$$
$$f_6(x) = f_1(x) + 10(4x_1 + 5x_2 - 3x_7 + 9x_8 - 105),$$
$$f_7(x) = f_1(x) + 10(10x_1 - 8x_2 - 17x_7 + 2x_8),$$
$$f_8(x) = f_1(x) + 10\left(-3x_1 + 6x_2 + 12(x_9-8)^2 - 7x_10\right),$$
$$f_9(x) = f_1(x) + 10(-8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12), \tag{A.22}$$

$n = 10, x^0 = (2, 3, 5, 5, 1, 2, 7, 3, 6, 10)$.

Example 23 (see [37]).

$$F(x) = \max_{1\le j\le 18} f_j(x),$$
$$f_1(x) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3-10)^2 + 4(x_4-5)^2 + (x_5-3)^2 +$$
$$+ 2(x_6-1)^2 + 5x_7^2 + 7(x_8-11)^2 + 2(x_9-10)^2 + (x_{10}-7)^2 + (x_{11}-9)^2 +$$
$$+ 10(x_{12}-1)^2 + 5(x_{13}-7)^2 + 4(x_{14}-14)^2 + 27(x_{15}-1)^2 + x_{16}^4 + (x_{17}-2)^2 +$$
$$+ 13(x_8-2)^2 + (x_{19}-3)^2 + x_{20}^2 + 95,$$
$$f_2(x) = f_1(x) + 10\left(3(x_1-2)^2 + 4(x_2-3)^2 + 2x_3^2 - 7x_4 - 120\right),$$
$$f_3(x) = f_1(x) + 10\left(5x_1^2 + 8x_2 + (x_3-6)^2 - 2x_4 - 40\right),$$
$$f_4(x) = f_1(x) + 10\left(0.5(x_1-8)^2 + 2(x_2-4)^2 + 3x_5^2 - x_6 - 30\right),$$
$$f_5(x) = f_1(x) + 10\left(x_1^2 + 2(x_2-2)^2 - 2x_1x_2 + 14x_5 - 6x_6\right),$$
$$f_6(x) = f_1(x) + 10(4x_1 + 5x_2 - 3x_7 + 9x_8 - 105),$$

$$
\begin{aligned}
f_7(x) &= f_1(x) + 10(10x_1 - 8x_2 - 17x_7 + 2x_8),\\
f_8(x) &= f_1(x) + 10\left(-3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10}\right),\\
f_9(x) &= f_1(x) + 10(-8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12),\\
f_{10}(x) &= f_1(x) + 10(x_1 + x_2 + 4x_{11} - 21x_{12}),\\
f_{11}(x) &= f_1(x) + 10\left(x_1^2 + 15x_{11} - 8x_{12} - 28\right),\\
f_{12}(x) &= f_1(x) + 10\left(4x_1 + 9x_2 + 5x_{13}^2 - 9x_{14} - 87\right),\\
f_{13}(x) &= f_1(x) + 10\left(3x_1 + 4x_2 + 3(x_{13} - 6)^2 - 14x_{14} - 10\right),\\
f_{14}(x) &= f_1(x) + 10\left(14x_1^2 + 35x_{15} - 79x_{16} - 92\right),\\
f_{15}(x) &= f_1(x) + 10\left(15x_2^2 + 11x_{15} - 61x_{16} - 54\right),\\
f_{16}(x) &= f_1(x) + 10\left(5x_1^2 + 2x_2 + 9x_{17}^4 - x_{18} - 68\right),\\
f_{17}(x) &= f_1(x) + 10\left(x_1^2 - x_2 + 19x_{19} - 20x_{20} + 19\right),\\
f_{18}(x) &= f_1(x) + 10\left(7x_1 + 5x_2^2 + x_{19}^2 - 30x_{20}\right),
\end{aligned}
\tag{A.23}
$$

$n = 20$, $x^0 = (2, 3, 5, 5, 1, 2, 7, 3, 6, 10, 2, 2, 6, 15, 1, 2, 1, 2, 1, 3)$.

Example 24 (see [37]).

$$
\begin{gathered}
F(x) = \max_{1 \le j \le 10} f_j(x),\\
f_j(x) = b_j \sum_{i=1}^{5} \left(x_i - a_{ji}\right)^2,\\
j = 1, \dots, 10,\\
A = \begin{bmatrix}
0 & 0 & 0 & 0 & 0\\
2 & 1 & 1 & 1 & 3\\
1 & 2 & 1 & 1 & 2\\
1 & 4 & 1 & 2 & 2\\
3 & 2 & 1 & 0 & 1\\
0 & 2 & 1 & 0 & 1\\
1 & 1 & 1 & 1 & 1\\
1 & 0 & 1 & 2 & 1\\
0 & 0 & 2 & 1 & 0\\
1 & 1 & 2 & 0 & 0
\end{bmatrix},\\
b = \begin{bmatrix}
1\\ 5\\ 10\\ 2\\ 4\\ 3\\ 1.7\\ 2.5\\ 6\\ 3.5
\end{bmatrix},
\end{gathered}
\tag{A.24}
$$

$n = 5$, $x^0 = (0, 0, 0, 0, 1)$.

Example 25 (see [34]).

$$\begin{aligned}
F(x) &= \max_{1\le j\le 130} f_j(x),\\
f_j(x) &= t_j - x_1\exp\left(-x_5 y_j\right) - x_2\exp\left(-x_6\left(y_j - x_9\right)^2\right) - x_3\exp\left(-x_7\left(y_j - x_{10}\right)^2\right)\\
&\quad - x_4\exp\left(-x_8\left(y_j - x_{11}\right)^2\right), \quad j = 1,\ldots,65,\\
f_j(x) &= -f_{j-65}(x), \quad j = 66,\ldots,130,\\
y_j &= 0.1(j-1), \quad j = 1,\ldots,65,\\
t &= (1.366, 1.191, 1.112, 1.013, 0.991, 0.885, 0.831, 0.847, 0.786, 0.725, 0.746, 0.679, 0.608, 0.655,\\
&\quad 0.616, 0.606, 0.602, 0.626, 0.651, 0.724, 0.649, 0.649, 0.694, 0.644, 0.624, 0.661, 0.612, 0.558,\\
&\quad 0.533, 0.495, 0.500, 0.423, 0.395, 0.375, 0.372, 0.391, 0.396, 0.405, 0.428, 0.429, 0.523, 0.562,\\
&\quad 0.607, 0.653, 0.672, 0.708, 0.633, 0.668, 0.645, 0.632, 0.591, 0.559, 0.597, 0.625, 0.739, 0.710,\\
&\quad 0.729, 0.720, 0.636, 0.581, 0.428, 0.292, 0.162, 0.098, 0.054).
\end{aligned} \tag{A.25}$$

$n = 11, x^0 = (1.3, 0.65, 0.65, 0.7, 0.6, 3, 5, 7, 2, 4.5, 5.5)$.

Example 26 (see [38]).

$$\begin{aligned}
F(x) &= \max_{1\le j\le 40} f_j(x),\\
f_j(x) &= x_j, \quad j = 1,\ldots,20,\\
f_j(x) &= -f_{j-20}(x), \quad j = 21,\ldots,40,
\end{aligned} \tag{A.26}$$

$n = 20, x^0 = (1, 2, \ldots, 10, -11, -12, \ldots, -20)$.

Example 27 (see [38]).

$$\begin{aligned}
F(x) &= \max_{1\le j\le 50} f_j(x),\\
f_j(x) &= 50x_j - \sum_{i=1}^{50} x_i,\\
&\qquad j = 1,\ldots,50,
\end{aligned} \tag{A.27}$$

$n = 50, x^0 = (1 - 25.5, \ldots, i - 25.5, \ldots, 50 - 25.5)$.

Example 28 (see [38]).

$$\begin{aligned}
F(x) &= \max_{1\le j\le 2} f_j(x),\\
f_1(x) &= (x_1)^2 + (x_2 - 1)^2 + x_2 - 1,\\
f_2(x) &= -(x_1)^2 - (x_2 - 1)^2 + x_2 + 1,
\end{aligned} \tag{A.28}$$

$n = 2, x^0 = (-1.5, 2)$.

Example 29 (see [38]).

$$\begin{aligned}
F(x) &= \max_{1\le j\le 2} f_j(x),\\
f_1(x) &= e^{x_1^2/1000 + (x_2 - 1)^2},\\
f_2(x) &= e^{x_1^2/1000 + (x_2 + 1)^2},
\end{aligned} \tag{A.29}$$

$n = 2, x^0 = (50, 0.05)$.

Example 30 (see [38]).

$$\begin{aligned}
F(x) &= \max_{1\le j\le 2} f_j(x),\\
f_1(x) &= -x_1 - x_2,\\
f_2(x) &= -x_1 - x_2 + \left(x_1^2 + x_2^2 - 1\right),
\end{aligned} \tag{A.30}$$

$n = 2, x^0 = (-0.5, -0.5)$.

Example 31 (see [38]).

$$\begin{aligned}
F(x) &= \max_{1\le j\le 2} f_j(x),\\
f_1(x) &= -x_1,\\
f_2(x) &= -x_1 + x_1^2 + x_2^2 - 1,
\end{aligned} \tag{A.31}$$

$n = 2, x^0 = (-1, -1)$.

Example 32 (see [38]).

$$\begin{aligned}
F(x) &= \max_{1\le j\le 2} f_j(x),\\
f_1(x) &= -x_1 + 2\left(x_1^2 + x_2^2 - 1\right) + 1.75\left(x_1^2 + x_2^2 - 1\right),\\
f_2(x) &= -x_1 + 2\left(x_1^2 + x_2^2 - 1\right) - 1.75\left(x_1^2 + x_2^2 - 1\right),
\end{aligned} \tag{A.32}$$

$n = 2, x^0 = (-1, -1)$.

Example 33 (see [38]).

$$\begin{aligned}
F(x) &= \max_{1\le j\le 3} f_j(x),\\
f_1(x) &= x_1^4 + x_2^2,\\
f_2(x) &= (2 - x_1)^2 + (2 - x_2)^2,\\
f_3(x) &= e^{-x_1 + x_2},
\end{aligned} \tag{A.33}$$

$n = 2, x^0 = (2, 2)$.

Example 34 (see [38]).

$$\begin{aligned} F(x) &= \max_{1\le j\le 3} f_j(x), \\ f_1(x) &= 5x_1 + x_2, \\ f_2(x) &= -5x_1 + x_2, \\ f_3(x) &= x_1^2 + x_2^2 + 4x_2, \end{aligned} \tag{A.34}$$

$n = 2, x^0 = (1, 1)$.

Example 35 (see [38]).

$$\begin{aligned} F(x) &= \max_{1\le j\le 3} f_j(x), \\ f_1(x) &= x_1^2 + x_2^2, \\ f_2(x) &= x_1^2 + x_2^2 + 10(-4x_1 - x_2 + 4), \\ f_3(x) &= x_1^2 + x_2^2 + 10(-x_1 - 2x_2 + 6), \end{aligned} \tag{A.35}$$

$n = 2, x^0 = (-1, 5)$.

Example 36 (see [38]).

$$\begin{aligned} F(x) &= \max_{1\le j\le 4} f_j(x), \\ f_1(x) &= 10\left(x_2 - x_1^2\right), \\ f_2(x) &= -f_1(x), \\ f_3(x) &= 1 - x_1, \\ f_4(x) &= -f_3(x), \end{aligned} \tag{A.36}$$

$n = 2, x^0 = (1.2, 1)$.

Example 37 (see [38]).

$$\begin{aligned} F(x) &= \max_{1\le j\le 4} f_j(x), \\ f_1(x) &= x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4, \\ f_2(x) &= f_1(x) + 10\left(x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8\right), \\ f_3(x) &= f_1(x) + 10\left(x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10\right), \\ f_4(x) &= f_1(x) + 10\left(x_1^2 + x_2^2 + x_3^2 + 2x_1 - x_2 - x_4 - 5\right), \end{aligned} \tag{A.37}$$

$n = 4, x^0 = (0, 0, 0, 0)$.

Example 38 (see [38]).

$$\begin{aligned} F(x) &= \max_{1\le j\le 2} f_j(x), \\ f_1(x) &= f(x + 2a), \\ f_2(x) &= f(x - 2a), \end{aligned} \tag{A.38}$$

where

$$\begin{aligned} f(x) &= e^{(0.0001x_1)^2 + x_2^2 + x_3^2 + 2x_4^2 + x_5^2 + x_6^2 + \cdots + x_{10}^2}, \\ a &= (0, 1, 0, 0, 0, 0, 0, 0, 0, 0), \end{aligned} \tag{A.39}$$

$n = 10, x^0 = (100, 0.1, \ldots, 0.1)$.

Example 39 (see [38]).

$$\begin{aligned} F(x) &= \max_{1\le j\le 10} f_j(x), \\ f_j(x) &= \sum_{i=1}^{11} \frac{1}{i - j + 1} e^{\left[\left(x_i - \sin(j - 1 + 2(i-1))\right)^2\right]}, \end{aligned} \tag{A.40}$$

$n = 11, x^0 = (1, \ldots, 1)$.

Example 40 (see [38]).

$$\begin{aligned} F(x) &= \max_{1\le j\le 3} f_j(x), \\ f_1(x) &= \frac{1}{2}\left(x_1 + \frac{10x_1}{(x_1 + 0.1)} + 2x_2^2\right), \\ f_2(x) &= \frac{1}{2}\left(-x_1 + \frac{10x_1}{(x_1 + 0.1)} + 2x_2^2\right), \\ f_3(x) &= \frac{1}{2}\left(x_1 - \frac{10x_1}{(x_1 + 0.1)} - 2x_2^2\right), \end{aligned} \tag{A.41}$$

$n = 2, x^0 = (3, 1)$.

Example 41 (see [38]).

$$\begin{aligned} F(x) &= \max_{1\le j\le 6} f_j(x), \\ f_1(x) &= x_1^2 + x_2^2 + x_1 x_2, \\ f_2(x) &= -f_1(x), \\ f_3(x) &= \sin(x_1), \\ f_4(x) &= -f_3(x), \\ f_5(x) &= \cos(x_2), \\ f_6(x) &= -f_5(x), \end{aligned} \tag{A.42}$$

$n = 2, x^0 = (3, 1)$.

Example 42 (see [38]).

$$F(x) = \max_{1\le j\le 2} f_j(x),$$
$$f_1(x) = 50\left(x_1 + x_2^4 - 1\right)^2 + 3x_1^2, \quad (A.43)$$
$$f_2(x) = 50\left(x_1 - x_2^4 + 1\right)^2 + 3x_1^2,$$

$n = 2, x^0 = (0.1, 0.1)$.

Example 43 (see [39]).

$$\begin{aligned} F(x) &= \max_{1\le j\le 6} f_j(x),\\ f_1(x) &= x_1^2 + x_2^2 + x_3^2 - 1,\\ f_2(x) &= x_1^2 + x_2^2 + (x_3 - 2)^2,\\ f_3(x) &= x_1 + x_2 + x_3 - 1,\\ f_4(x) &= x_1 + x_2 - x_3 + 1,\\ f_5(x) &= 2x_1^3 + 6x_2^2 + 2(5x_3 - x_1 + 1)^2,\\ f_6(x) &= x_1^2 - 9x_3, \end{aligned} \quad (A.44)$$

$n = 3, x^0 = (1, 1, 1)$.

Example 44 (see [40]).

$$\begin{aligned} F(x) &= \max_{1\le j\le 4} f_j(x),\\ f_1(x) &= \left(x_1 - (x_4+1)^4\right)^2 + \left(x_2 - \left(x_1 - (x_4+1)^4\right)^4\right)^2 + 2x_3^2 + x_4^2 -\\ &\quad - 5\left(x_1 - (x_4+1)^4\right) - 5\left(x_2 - \left(x_1 - (x_4+1)^4\right)^4\right) - 21x_3 + 7x_4,\\ f_2(x) &= f_1(x) + 10\left(\left(x_1 - (x_4+1)^4\right)^2 + \left(x_2 - \left(x_1 - (x_4+1)^4\right)^4\right)^2 + x_3^2 + x_4^2 +\right.\\ &\quad \left. + \left(x_1 - (x_4+1)^4\right) - \left(x_2 - \left(x_1 - (x_4+1)^4\right)^4\right) + x_3 - x_4 - 8\right),\\ f_3(x) &= f_1(x) + 10\left(\left(x_1 - (x_4+1)^4\right)^2 + 2\left(x_2 - \left(x_1 - (x_4+1)^4\right)^4\right)^2 + x_3^2 + 2x_4^2 -\right.\\ &\quad \left. - \left(x_1 - (x_4+1)^4\right) - x_4 - 10\right),\\ f_4(x) &= f_1(x) + 10\left(\left(x_1 - (x_4+1)^4\right)^2 + \left(x_2 - \left(x_1 - (x_4+1)^4\right)^4\right)^2 + x_3^2 +\right.\\ &\quad \left. + 2\left(x_1 - (x_4+1)^4\right) - \left(x_2 - \left(x_1 - (x_4+1)^4\right)^4\right) - x_4 - 5\right), \end{aligned} \quad (A.45)$$

$n = 4, x^0 = (0, 0, 0, 0)$.

Example 45 (see [41]).

$$F(x) = \max_{1\le j\le 82} f_j(x),$$

$$f_j(x) = \left(\frac{\left(x_1 + (1 + x_2)\cos\vartheta_j\right)^2 + \left((1 - x_2)\sin\vartheta_j\right)^2}{\left(x_3 + (1 + x_4)\cos\vartheta_j\right)^2 + \left((1 - x_4)\sin\vartheta_j\right)^2}\right)^{1/2} \cdot \left(\frac{\left(x_5 + (1 + x_6)\cos\vartheta_j\right)^2 + \left((1 - x_6)\sin\vartheta_j\right)^2}{\left(x_7 + (1 + x_8)\cos\vartheta_j\right)^2 + \left((1 - x_8)\sin\vartheta_j\right)^2}\right)^{1/2} x_9 - y_j,$$

$$\begin{aligned} y_j &= 1 - 2t_j, \quad 1 \le j \le 41, \\ y_j &= -1 + 2t_j, \quad 42 \le j \le 82, \\ \vartheta_j &= \pi t_j, \quad 1 \le j \le 41, \\ t_j &= 0.01(j - 1), \quad 1 \le j \le 6, \\ t_j &= 0.07 + 0.03(j - 7), \quad 7 \le j \le 20, \\ t_{21} &= 0.50, \\ t_j &= 0.54 + 0.03(j - 22), \quad 22 \le j \le 35, \\ t_j &= 0.95 + 0.01(j - 36), \quad 36 \le j \le 41, \\ t_j &= t_{j-41}, \quad 42 \le j \le 82, \end{aligned} \tag{A.46}$$

$n = 9, x^0 = (0, 1, 0, -0.15, 0, -0.68, 0, -0.72, 0.37)$.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (no. 11701350).

References

[1] J. F. Sturm and S. Zhang, "A dual and interior-point approach to solve convex min-max problems," *in Minimax and Applications, Nonconvex Optimization and Its Application*, vol. 4, pp. 69–78, D. Z. Du and P. M. Pardalos, Ed, Kluwer Academic, Dordrecht, Netherlands, 1995.

[2] D. Applegate, W. Cook, S. Dash, and A. Rohe, "Solution of a min-max vehicle routing problem," *INFORMS Journal on Computing*, vol. 14, no. 2, pp. 132–143, 2002.

[3] E. M. Arkin, R. Hassin, and A. Levin, "Approximations for minimum and min-max vehicle routing problems," *Journal of Algorithms*, vol. 59, no. 1, pp. 1–18, 2006.

[4] N. V. Banichuk, "Minimax approach to structural optimization problems," *Journal of Optimization Theory and Applications*, vol. 20, no. 1, pp. 111–127, 1976.

[5] S. E. Sussman-Fort, "Approximate direct-search minimax circuit optimization," *International Journal Numerial Methods in Engineering*, vol. 28, no. 2, pp. 359–368, 1989.

[6] R. T. Pockafellar, "Computational schemes for large-scale problems in extended linear-quadratic programming," *Mathematical Programming*, vol. 48, no. 1–3, pp. 447–474, 1990.

[7] R. T. Rockafellar, "Linear-quadratic programming and optimal control," *SIAM Journal on Control and Optimization*, vol. 25, no. 3, pp. 781–814, 1987.

[8] J. A. Chatelon, D. W. Hearn, and T. J. Lowe, "A subgradient algorithm for certain minimax and minisum problems," *Mathematical Programming*, vol. 15, no. 1, pp. 130–145, 1978.

[9] M. Gaudioso and M. F. Monaco, "A bundle type approach to the unconstrained minimization of convex nonsmooth functions," *Mathematical Programming*, vol. 23, no. 1, pp. 216–226, 1982.

[10] A. Frangioni, "Generalized bundle methods," *SIAM Journal on Optimization*, vol. 13, no. 1, pp. 117–156, 2002.

[11] R. Reemtsen, "A cutting plane method for solving minimax problems in the complex plane," *Numerical Algorithms*, vol. 2, no. 3, pp. 409–436, 1992.

[12] R. S. Womersley and R. Fletcher, "An algorithm for composite nonsmooth optimization problems," *Journal of Optimization Theory and Applications*, vol. 48, no. 3, pp. 493–523, 1986.

[13] J. L. Zhou and A. L. Tits, "An SQP algorithm for finely discretized continuous minimax problems and other minimax problems with many objective functions," *SIAM Journal on Optimization*, vol. 6, no. 2, pp. 461–487, 1996.

[14] W. Murray and M. L. Overton, "A projected Lagrangian algorithm for nonlinear minimax optimization," *SIAM Journal on Scientific and Statistical Computing*, vol. 1, no. 3, pp. 345–370, 1980.

[15] J. F. Sturm and S. Zhang, "A dual and interior point approach to solve convex min-max problems," *Nonconvex Optimization and Its Applications*, pp. 69–78, 1995.

[16] L. Luksan and J. Vlcek, "Primal interior-point method for large sparse minimax optimization," *Kybernetika -Praha-*, vol. 45, no. 5, pp. 841–864, 2009.

[17] E. Obasanjo, G. Tzallas-Regas, and B. Rustem, "An interior-point algorithm for nonlinear minimax problems," *Journal of Optimization Theory and Applications*, vol. 144, no. 2, pp. 291–318, 2010.

[18] Y. Feng, L. Hongwei, Z. Shuisheng, and L. Sanyang, "A smoothing trust-region Newton-CG method for minimax problem," *Applied Mathematics and Computation*, vol. 199, no. 2, pp. 581–589, 2008.

[19] I. Zang, "A smoothing-out technique for min-max optimization," *Mathematical Programming*, vol. 19, no. 1, pp. 61–77, 1980.

[20] D. Q. Mayne and E. Polak, "Nondifferential optimization via adaptive smoothing," *Journal of Optimization Theory and Applications*, vol. 43, no. 4, pp. 601–613, 1984.

[21] R. A. Polyak, "Smooth optimization methods for minimax problems," *SIAM Journal on Control and Optimization*, vol. 26, no. 6, pp. 1274–1286, 1988.

[22] L. Xingsi, "An entropy-based aggregate method for minimax optimization," *Engineering Optimization*, vol. 18, no. 4, pp. 277–285, 1992.

[23] S. Xu, "Smoothing method for minimax problems," *Computational Optimization and Applications*, vol. 20, no. 3, pp. 267–279, 2001.

[24] 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.

[25] 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.

[26] Y. Xiao and B. Yu, "A truncated aggregate smoothing Newton method for minimax problems," *Applied Mathematics and Computation*, vol. 216, no. 6, pp. 1868–1879, 2010.

[27] X. S. Li, "An aggregate function method for nonlinear programming," *Science in China Series A-Mathematics*, vol. 34, pp. 1467–1473, 1991.

[28] L. Dong and B. Yu, "A spline smoothing Newton method for finite minimax problems," *Journal of Engineering Mathematics*, vol. 93, no. 1, pp. 145–158, 2014.

[29] A. Fischer, "Solution of monotone complementarity problems with locally Lipschitzian functions," *Mathematical Programming*, vol. 76, pp. 513–532, 1997.

[30] L. Qi, D. Sun, and G. Zhou, "A new look at smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequalities," *Mathematical Programming*, vol. 87, no. 1, pp. 1–35, 2000.

[31] L. Qi, "Convergenece analysis of some algorithms for solving nonsmooth equations," *Mathematics of Operations Research*, vol. 18, no. 1, pp. 227–244, 1993.

[32] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.

[33] G. A. Watson, "The minimax solution of an overdetermined system of non-linear equations," *IMA Journal of Applied Mathematics*, vol. 23, no. 2, pp. 167–180, 1979.

[34] J. R. Popovic, J. W. Bandler, and C. Charalambous, "General programmes for least p-th and near minimax approximation," *International Journal on Systems Science*, vol. 5, no. 10, pp. 907–932, 1974.

[35] J. Hald and K. Madsen, "Combined LP and quasi-Newton methods for minimax optimization," *Mathematical Programming*, vol. 20, no. 1, pp. 42–62, 1981.

[36] Y. Xiao and B. Yu, *Truncated Aggregate Smoothing Algorithms*, Dalian University of Technology, Dalian, China, 2010.

[37] J. Asaadi, "A computational comparison of some non-linear programs," *Mathematical Programming*, vol. 4, no. 1, pp. 144–154, 1973.

[38] G. D. Pillo, L. Grippo, and S. Lucidi, "A smooth method for the finite minimax problem," *Mathematical Programming*, vol. 60, no. 1–3, pp. 187–214, 1993.

[39] R. A. El-Attar, M. Vidyasagar, and S. R. K. Dutta, "An algorithm for l_1-norm minimization with application to nonlinear l_1-approximation," *SIAM Journal on Numerical Analysis*, vol. 16, no. 1, pp. 70–86, 1979.

[40] E. Polak, D. Q. Mayne, and J. E. Higgins, "Superlinearly convergent algorithm for min-max problems," *Journal of Optimization Theory and Applications*, vol. 69, no. 3, pp. 407–439, 1991.

[41] C. Charalambous, "Acceleration of the least pth algorithm for minimax optimization with engineering applications," *Mathematical Programming*, vol. 17, no. 1, pp. 270–297, 1979.