The smoothing Fletcher-Reeves conjugate gradient method for solving finite minimax problems

Deyan Pang, Shouqiang Du*, Jingjie Ju
College of Mathematics, Qingdao University, Qingdao 266071, China

*Corresponding author, e-mail: sqdu@qdu.edu.cn

Received 14 Oct 2014
Accepted 7 Feb 2016

ABSTRACT: In this paper, we give a smoothing Fletcher-Reeves conjugate gradient method for finite minimax problems. The functions of the finite minimax problem are all continuous differentiable functions. Under general conditions, we present the global convergence of the method. The final discussion and preliminary numerical experiments indicate that the method works quite well in practice.

KEYWORDS: nonsmooth unconstrained optimization problem, smoothing function, global convergence

INTRODUCTION
In this paper, we consider the following finite minimax problem

\[ \min \max_{1 \leq i \leq m} f_i(x), \]  

where \( f_i : \mathbb{R}^n \to \mathbb{R} \) are continuously differentiable functions for \( i = 1, \ldots, m \). Problem (1) and its related optimization problems have broad applications in engineering technology, economic management, and many other fields\(^1\)–\(^3\). How to design the numerical algorithms for solving (1) is one of the most fundamental themes in optimization research\(^4\)–\(^6\).

Let

\[ f(x) = \max_{1 \leq i \leq m} f_i(x). \]

Then (1) can be transformed into the following nonsmooth (nondifferentiable) unconstrained optimization problem

\[ \min_{x \in \mathbb{R}^n} f(x). \]  

The nonsmooth unconstrained optimization problem is a kind of optimization problem with a broad application background (see, e.g., Ref. 7). The methods for solving the nonsmooth unconstrained optimization problem are more complex than the methods for solving the smooth optimization problem. During the last thirty years, many numerical methods have been proposed for the solution of the finite minimax problem, such as in Refs. 2, 3, but there have only been a few methods for solving large-scale problems. Furthermore, \( f_i, i = 1, \ldots, m \) in (1) are often twice continuously differentiable functions. To overcome the drawback, in this paper, we consider using the smoothing Fletcher-Reeves conjugate gradient method to solve (1) and the functions in (1) are only continuously differentiable, not necessarily twice continuously differentiable. The main advantage of our method is that the conjugate gradient method only uses the first order information\(^8\),\(^9\) and so is suitable for solving large-scale optimization problems\(^10\)–\(^12\).

The first nonlinear conjugate gradient method was introduced by Fletcher and Reeves\(^13\), it is one of the earliest known techniques for solving nonlinear optimization problems. Zoutendijk\(^14\) established a convergence result for the Fletcher-Reeves conjugate gradient method under the exact line search. Al-Baali\(^15\), Liu et al\(^16\) and Dai and Yuan\(^17\) dealt with the convergence of the Fletcher-Reeves conjugate gradient method with the strong Wolfe line search. A comprehensive study on the convergence of the Fletcher-Reeves conjugate gradient method has been made in Ref. 18. Notwithstanding the strong convergence properties, numerical performance of the Fletcher-Reeves conjugate gradient method is essentially affected by jamming\(^18\),\(^19\), i.e., the method can take many short steps without making significant progress to the minimum. To achieve good computational performance and to maintain the attractive feature of strong global convergence, we focus on hybridizing the Fletcher-
Reeves conjugate gradient method with smoothing techniques. Also, the simple use of smoothing techniques can help prevent ill-conditioning.

Here, we consider using the following iterative formula for solving (1):

\[ x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, \ldots, \]

where \( \alpha_k > 0 \) is the step size and \( d_k \) is a search direction.

This paper is organized as follows. In the second section, we present the smoothing Fletcher-Reeves conjugate gradient method for solving (1) and give its convergence analysis. In the third section, we discuss the convergence of the method with other inexact line search and exact line search conditions. In the fourth section, we give some numerical results of the method. In the last section, we give the applications in solving constrained optimization problems and conclusions.

Throughout the paper, \( \| \cdot \| \) denotes the \( l_2 \) norm; \( R^n_+ = \{ t \mid t \geq 0, t \in \mathbb{R} \} \); \( \nabla \tilde{f}(x, \mu) = \nabla f(x, \mu); \tilde{g}_k = \nabla \tilde{f}(x_k, \mu_k) \).

METHOD AND ITS CONVERGENCE ANALYSIS

We start by giving some definitions. If \( H(x) \) is locally Lipschitz continuous but not necessarily differentiable, then the generalized gradient of \( H \) at \( x \) is defined by

\[ \partial H(x) = \text{conv} \left\{ \lim_{x_i \to x, x_i \in D_{H}} \nabla H(x_i) \right\}, \]

where conv denotes the convex hull of a set and \( D_{H} \) is the set of points at which \( H \) is differentiable. Because \( f \) in (2) is a locally Lipschitz continuous and not differentiable function, we need to smoothen \( f \) first. As in Ref. 7, we introduce the definition of the smoothing function and the gradient consistency property.

**Definition 1** Let \( f : \mathbb{R}^n \to \mathbb{R} \) be a continuous function. We call \( \tilde{f} : \mathbb{R}^n \times R_+ \to \mathbb{R} \) a smooth function of \( f \), if \( \tilde{f}(\cdot, \mu) \) is continuously differentiable in \( \mathbb{R}^n \) for any fixed \( \mu > 0 \) and

\[ \lim_{\mu \to 0^+} \tilde{f}(x, \mu) = f(x), \]

for any \( x \in \mathbb{R}^n \). Furthermore, if for any \( x \in \mathbb{R}^n \),

\[ \lim_{\mu \to 0^+} \nabla \tilde{f}(x, \mu) = \partial f(x^*), \]

then we say \( f \) satisfies the gradient consistency property.

Now we give the smoothing Fletcher-Reeves conjugate gradient method for solving (2).

**Algorithm 1**

Step 1: Choose \( \sigma_1 \in (0, 0.5), \gamma > 0, \gamma_1 \in (0, 1), \mu_0 > 0 \) and give an initial point \( x_0 \in \mathbb{R}^n \). Let \( k := 0 \), compute \( \tilde{g}_0 = \nabla \tilde{f}(x_0, \mu_0) \). Let \( d_0 = -\tilde{g}_0 \).

Step 2: Compute \( \alpha_k \) by the Armijo line search, where \( \alpha_k = \max \{ \rho^0, \rho^1, \ldots \} \) and \( \rho^i \) satisfies

\[ \tilde{f}(x_k + \alpha_k d_k, \mu_k) - \tilde{f}(x_k, \mu_k) \leq \sigma_1 \rho^i \| \tilde{g}_k \| d_k. \] (3)

Set \( x_{k+1} = x_k + \alpha_k d_k \).

Step 3: If \( \| \nabla \tilde{f}(x_{k+1}, \mu_k) \| \geq \gamma \mu_k \), then set \( \mu_{k+1} = \mu_k \); otherwise, let \( \mu_{k+1} = \gamma_1 \mu_k \).

Step 4: Compute \( d_{k+1} \) by the following the Fletcher-Reeves conjugate gradient formula

\[ d_{k+1} = -\tilde{g}_{k+1} + \rho_k^{FR} \cdot d_k, \] (4)

where

\[ \rho_k^{FR} = \frac{\tilde{g}_k^T \tilde{g}_{k+1}}{\tilde{g}_k^T \tilde{g}_k}. \]

Step 5: Set \( k := k + 1 \); go to Step 2.

We now give the convergence result for Algorithm 1.

**Theorem 1** Suppose \( \tilde{f}(x, \mu) : \mathbb{R}^n \to \mathbb{R} \) is continuously differentiable and \( \tilde{g}_k^T d_{k-1} < 0 \). Then

\[ \tilde{g}_k^T d_k \leq -\| \tilde{g}_k \|^2. \]

**Proof:** According to the definition,

\[ \tilde{g}_k^T d_k = -\| \tilde{g}_k \|^2 + \rho_k^{FR} \tilde{g}_k^T d_{k-1}, \]

and

\[ \tilde{g}_k^T d_{k-1} < 0. \]

We obtain

\[ \tilde{g}_k^T d_k = -\| \tilde{g}_k \|^2 + \rho_k^{FR} \tilde{g}_k^T d_{k-1} \leq -\| \tilde{g}_k \|^2. \] (5)

From (5),

\[ \tilde{g}_k^T d_k \leq -\| \tilde{g}_k \|^2. \]

**Theorem 2** Suppose that \( \tilde{f}(\cdot, \mu) \) is a smoothing function of \( f \). If for any constant \( \mu > 0 \), \( \tilde{f}(\cdot, \mu) \) is bounded on the level set \( L(x_0) = \{ x \in \mathbb{R}^n \mid f(x) \leq f(x_0) \} \), then \( \{ x^k \} \) generated by Algorithm 1 satisfies

\[ \lim_{k \to \infty} \mu_k = 0 \]

and

\[ \liminf_{k \to \infty} \left\| \nabla \tilde{f}(x_k, \mu_{k-1}) \right\| = 0. \]
Proof: Define $K = \{ k \mid \mu_{k+1} = \gamma_1 \mu_k \}$. If $K$ is a finite set, then there exists an integer $k$ such that for all $k > \hat{k}$

$$\| \nabla \tilde{f}(x_k, \mu_k) \| \geq \gamma \mu_{k-1}. \quad (6)$$

And $\mu_k = \mu = \bar{\mu}$ in Step 3 of Algorithm 1. Since $\tilde{f}(\cdot, \bar{\mu})$ is bounded on the level set $L(x_0)$, from Theorem 1 and the conjugate gradient method, we deduce that

$$\lim_{k \to \infty} \inf_{\mu_k} \| \nabla \tilde{f}(x_k, \bar{\mu}) \| = 0,$$

which contradicts (6). This shows that $K$ must be infinite and

$$\lim_{k \to \infty} \mu_k = 0.$$

Since $K$ is infinite, we can assume that $K = \{ k_0, k_1, \ldots \}$, where $k_0 < k_1 < \ldots$. Then we have

$$\lim_{i \to \infty} \| \nabla \tilde{f}(x_{k+1}, \mu_k) \| \leq \gamma \lim_{i \to \infty} \mu_k = 0. \quad \square$$

According to Theorem 2 and the definition of gradient consistency, it is easy to obtain the following result.

Theorem 3 Any accumulation point $x^*$ of $\{ x_k \}$ generated by Algorithm 1 is a Clark stationary point, i.e., $0 \in \partial f(x^*)$.

COMPARISON OF THE GLOBAL CONVERGENCE OF THE METHOD WITH OTHER LINE SEARCHES

The line search in Algorithm 1 can also be changed to a strong Wolfe line search, i.e., $\alpha_k$ satisfying (3) and

$$\| \nabla \tilde{f}(x_k + \alpha_k d_k, \mu_k) \|^2 \leq \sigma_2 \| \tilde{g}_k d_k \|^2, \quad (7)$$

where $0 < \sigma_1 < \sigma_2 < \frac{1}{2}$. In order to obtain the global convergence of Algorithm 1 under the strong Wolfe line search, we first consider the descent property of the search direction.

Lemma 1 Suppose $\{ x_k \}$ is generated by Algorithm 1 with a strong Wolfe line search, which is under the strong Wolfe line search ($0 < \sigma_1 < \sigma_2 < \frac{1}{2}$). Then we have

$$\tilde{g}_k^T d_k < 0.$$ 

Proof: We use induction to prove the following inequality:

$$- \sum_{i=0}^{k} \sigma_i^2 \leq \frac{\tilde{g}_k^T d_k}{\| \tilde{g}_k \|^2} \leq -2 + \sum_{i=0}^{k} \sigma_i^2, \quad (8)$$

for $\forall k = 0, 1, \ldots$. Suppose (8) is established. By

$$\sum_{i=0}^{k} \sigma_i^2 \leq \sum_{i=0}^{\infty} \sigma_i = \frac{1}{1-\sigma_2} < 2,$$

we deduce that the right-hand side of (8) is less than 0. It follows that

$$\tilde{g}_k^T d_k < 0.$$ 

Obviously, when $k = 0$, $d_0 = -\tilde{g}_0$. Suppose (8) holds for some $k \geq 0$. From (4), we can know that

$$\frac{\tilde{g}_k^T d_{k+1}}{\| \tilde{g}_{k+1} \|^2} = -1 + \frac{\tilde{g}_k^T d_k}{\| \tilde{g}_k \|^2}.$$ 

By (7) and $\tilde{g}_k^T d_k < 0$, we have

$$-1 + \sigma_2 \frac{\tilde{g}_k^T d_{k+1}}{\| \tilde{g}_{k+1} \|^2} \leq \frac{\tilde{g}_k^T d_k}{\| \tilde{g}_k \|^2} \leq -1 - \sigma_2 \frac{\tilde{g}_k^T d_k}{\| \tilde{g}_k \|^2}.$$ 

Using the left side of the induction hypothesis,

$$- \sum_{i=0}^{k+1} \sigma_i^2 \leq -1 - \sigma_2 \sum_{i=0}^{k} \sigma_i^2 \leq -1 - \sigma_2 \frac{\tilde{g}_k^T d_{k+1}}{\| \tilde{g}_{k+1} \|^2} \leq -1 + \sigma_2 \sum_{i=0}^{k+1} \sigma_i^2 - 2 + \sum_{i=0}^{k+1} \sigma_i^2.$$ 

Hence (8) holds. Then from the right-hand side of (8) we obtain

$$\tilde{g}_k^T d_k < 0. \quad \square$$

Theorem 4 Suppose $\tilde{f}(x, \mu) : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable, the level set $L(x_0)$ is bounded, and the gradient function $\tilde{g}(x)$ is Lipschitz continuous on this level set. Then $\{ x_k \}$ generated by Algorithm 1 satisfies

$$\lim_{k \to \infty} \inf_{\tilde{g}_k} \| \tilde{g}_k \| = 0.$$ 

Proof: From (3), (7), and (8), we have

$$\| \tilde{g}_k^T d_{k-1} \| \leq -\sigma_2 \tilde{g}_k^T d_{k-1} \leq \frac{\sigma_2}{1-\sigma_2} \| \tilde{g}_{k-1} \|^2. \quad (9)$$

From (4), we know that

$$\| d_k \|^2 = \| \tilde{g}_k \|^2 - 2 \beta_{k-1} \tilde{g}_k^T d_{k-1} + \beta_{k-1}^2 \| d_{k-1} \|^2 = \| \tilde{g}_k \|^2 - 2 \frac{\| \tilde{g}_k \|^2}{\tilde{g}_k^T \tilde{g}_{k-1}} \| \tilde{g}_{k-1} \| d_{k-1} + \beta_{k-1}^2 \| d_{k-1} \|^2.$$ 

www.scienceasia.org
By (9), we know that
\[ \|d_k\|^2 \leq \|g_k\|^2 + \frac{2\sigma_2}{1 - \sigma_2} \|g_k\|^2 + \beta_{k-1} \|d_{k-1}\|^2 \]
\[ = \frac{1 + \sigma_2}{1 - \sigma_2} \|g_k\|^2 + \beta_{k-1} \|d_{k-1}\|^2. \]
Using the recursion relations, we obtain
\[ \|d_k\|^2 \leq \frac{1 + \sigma_2}{1 - \sigma_2} \|g_k\|^2 + \beta_{k-1} \|d_{k-1}\|^2. \tag{10} \]
If the conclusion of the theorem is not true, then there exists a constant \( \varepsilon_0 > 0 \) such that for any \( k \)
\[ \|g_k\| \geq \varepsilon_0. \tag{11} \]
Because the gradient function \( g(x) \) is Lipschitz continuous on the level set, the norm of \( g(x) \) is bounded on the level set. From (10), there exists a constant \( M_1 > 0 \) such that for any \( k \)
\[ \|d_k\|^2 \leq M_1(1 + k). \tag{12} \]
Assume the angle between \( d_k \) and \( (-g_k) \) is \( \theta_k \). From (8), we have
\[ \cos \theta_k = -\frac{\tilde{g}_k^T d_k}{\|\tilde{g}_k\| \|d_k\|} \geq \frac{1 - 2\sigma_2}{1 - \sigma_2} \frac{\|g_k\|}{\|d_k\|}. \tag{13} \]
Since \( \sigma_2 < \frac{1}{2} \)
\[ \frac{1 - 2\sigma_2}{1 - \sigma_2} > 0. \]
By (11), (12), and (13), there exists a constant \( M_2 > 0 \) such that
\[ \sum_{k=0}^{\infty} \cos^2 \theta_k \geq \left( \frac{1 - 2\sigma_2}{1 - \sigma_2} \right) \sum_{k=0}^{\infty} \frac{\|g_k\|^2}{\|d_k\|^2} \geq M_2 \sum_{k=0}^{\infty} \frac{1}{k + 1}. \]
So \( \sum_{k=0}^{\infty} \cos^2 \theta_k \) is not convergent. On the other hand, because \( g(x) \) is Lipschitz continuous on the level set (suppose the Lipschitz constant is \( L \)),
\[ \sigma_2 \tilde{g}_k^T d_k \leq \tilde{g}_{k+1}^T d_k = \tilde{g}_k^T d_k + (\tilde{g}_{k+1} - \tilde{g}_k)^T d_k \leq \tilde{g}_k^T d_k + \alpha_k L \|d_k\|^2. \]
Thus
\[ \alpha_k \geq \frac{1 - \sigma_2}{L \|d_k\|} \frac{\|g_k\|^2}{\|d_k\|^2}. \]
By (3), we obtain
\[ \bar{f}_{k+1} \leq \bar{f}_k - \frac{1}{L} \|d_k\|^2 \|g_k\|^2 \cos^2 \theta_k. \]
Because \( \{\bar{f}_k\} \) is monotonically decreasing and has a lower bound, \( \sum_{k=0}^{\infty} \|g_k\|^2 \cos^2 \theta_k \) is convergent. From \( \|g_k\| \geq \varepsilon_0 \), we know that \( \sum_{k=0}^{\infty} \cos^2 \theta_k \) is convergent. This contradicts that \( \sum_{k=0}^{\infty} \cos^2 \theta_k \) is not convergent. So the assumption does not hold. That is
\[ \lim_{k \to \infty} \inf \|g_k\| = 0. \]

**Remark 1** Theorem 2 and Theorem 3 obviously hold for Algorithm 1 with a strong Wolfe line search. If we choose the exact line search to determine the step-size factor \( \alpha_k \) in Algorithm 1, i.e.,
\[ \alpha_k = \arg \min_{\alpha \geq 0} f(x_k + \alpha d_k), \]
then we can obtain the convergence theorem of Algorithm 1 with the exact line search.

**Theorem 5** Let \( \{x_k\} \) be generated by Algorithm 1 with the exact line search. Suppose \( f(x, \mu) : \mathbb{R}^n \to \mathbb{R} \) is first order continuously differentiable, and the level set \( L(x_0) \) is bounded. Then Algorithm 1 either terminates or \( \lim_{k \to \infty} g(x_k) = 0 \).

**Proof:** Without loss of generality, suppose \( \{x_k\} \) is an infinite sequence. Because we choose the exact line search in Algorithm 1, we have
\[ \tilde{g}_k^T d_{k-1} = 0. \]
From
\[ \tilde{g}_k^T d_k = -\|\tilde{g}_k\|^2 + \beta_{k-1} \tilde{g}_k^T d_{k-1} = -\|\tilde{g}_k\|^2 < 0, \]
we know that \( d_k \) is a descent direction. Thus from the exact line search rule we know that \( \{f(x_k, \mu_k)\} \) is monotonically decreasing, and so \( \{x_k\} \subset L(x_0) \). Hence \( \{x_k\} \) is bounded. There exists an accumulation point \( x^* \), i.e., there exists \( \{x_k \mid k \in K_1\} \) convergent to \( x^* \). According to the continuity of \( \bar{f} \), we have
\[ f^* = \lim_{k \in K_1} \bar{f}(x_k, \mu_k) = \bar{f}(x^*). \]
Similarly, \( \{x_{k+1} \mid k \in K_2\} \) is a bounded subsequence, and so there exists a subsequence \( \{x_{k+1} \mid k \in K_2\} \) which converges to \( \bar{x}^* \), where \( K_2 \subset K_1 \) is an infinite subsequence. Hence we obtain
\[ f^* = \lim_{k \in K_2} \bar{f}(x_{k+1}, \mu_k) = \bar{f}(\bar{x}^*). \]

www.scienceasia.org
and the smoothing gradient method

Table 1 Numerical results for Example 1 by Algorithm 1 and the smoothing gradient method.\(^7\)

<table>
<thead>
<tr>
<th>(x_0)</th>
<th>(x^*(\text{Alg1}))</th>
<th>(x^* (\text{SGM}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,−1)</td>
<td>(1.1358, 0.9002)</td>
<td>(1.1413, 0.8961)</td>
</tr>
<tr>
<td>(1.3,−0.8)</td>
<td>(1.1391, 0.8986)</td>
<td>(1.1415, 0.8959)</td>
</tr>
<tr>
<td>(1.2,−0.69)</td>
<td>(1.1360, 0.9009)</td>
<td>(1.1410, 0.8963)</td>
</tr>
<tr>
<td>(1.3,−1.6)</td>
<td>(1.1402, 0.8976)</td>
<td>(1.1415, 0.8959)</td>
</tr>
<tr>
<td>(1.4,−0.9)</td>
<td>(1.1396, 0.8981)</td>
<td>(1.1413, 0.8960)</td>
</tr>
<tr>
<td>(1.4,−0.7)</td>
<td>(1.1370, 0.9002)</td>
<td>(1.1414, 0.8960)</td>
</tr>
</tbody>
</table>

Table 2 Numerical results for Example 2 by Algorithm 1 and the smoothing gradient method.\(^7\)

<table>
<thead>
<tr>
<th>(x_0)</th>
<th>(x^*(\text{Alg1}))</th>
<th>(x^* (\text{SGM}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1.4,−0.7)</td>
<td>(1.0033, 0.9916)</td>
<td>(1.0033, 0.9916)</td>
</tr>
<tr>
<td>(3.1,−2.7)</td>
<td>(1.0033, 0.9916)</td>
<td>(1.0033, 0.9916)</td>
</tr>
<tr>
<td>(2.9,−1.7)</td>
<td>(1.001, 0.9957)</td>
<td>(1.0033, 0.9916)</td>
</tr>
<tr>
<td>(2.4,−1.9)</td>
<td>(1.0016, 0.9958)</td>
<td>(1.0033, 0.9916)</td>
</tr>
<tr>
<td>(3,−2)</td>
<td>(1.0017, 0.9956)</td>
<td>(1.0033, 0.9916)</td>
</tr>
<tr>
<td>(1,−1)</td>
<td>(1.0018, 0.9956)</td>
<td>(1.0033, 0.9916)</td>
</tr>
</tbody>
</table>

It follows that
\[
\tilde{f}(x^*) = \tilde{f}(x^+) = f^*.
\] (14)

Next, we will use reduction to absurdity to prove \(\tilde{g}(x^*) = 0\). If \(\tilde{g}(x^*) \neq 0\), then for sufficiently small \(\alpha > 0\), we have

\[
\tilde{f}(x_{k+1}, \mu_k) = \tilde{f}(x_k + \alpha_d, \mu_k) \leq \tilde{f}(x_k + \alpha_d, \mu_k).
\]

For \(k \in K_2 \subset K_1\), let \(k \to \infty\). We obtain

\[
\tilde{f}(x^*) \leq \tilde{f}(x^* + \alpha_d^*) < \tilde{f}(x^*).
\]

This contradicts (14). Hence

\[
\tilde{g}(x^*) = 0.
\]

Remark 2 Theorem 2 and Theorem 3 clearly hold for Algorithm 1 with the exact line search.

**NUMERICAL EXPERIMENTS**

We used the examples in Ref. 3 to test Algorithm 1. All codes were run on MATLAB 8.0. Throughout our computational experiments, the parameters used in Algorithm 1 were \(\delta_1 = 0.25\), \(\gamma = \gamma_1 = 0.5\). We used the following smoothing function\(^2\)

\[
\tilde{f}(x, \mu) = \mu \ln \sum_{i=1}^m \exp \left( \frac{f_i(x)}{\mu} \right).
\]

We use \(\|\Delta x\| \leq 10^{-5}\) as the stopping rule. \(x_0\) is the initial point, \(x^*\) is the optimal value point. In the following, we will give several initial points and the results in Table 1, Table 2, Table 3, and Table 4. We also compare Algorithm 1 with the smoothing gradient method.\(^7\)

**Example 1** Charalambous-Conn 1\(^3\).

\[
f(x) = \max\{x_1^2 + x_2^4, (2 - x_1)^2 + (2 - x_2)^2, 2 \exp(-x_1 + x_2)\}.
\]

**Example 2** Charalambous-Conn 2\(^3\).

\[
f(x) = \max\{x_1^4 + x_2^2, (2 - x_1)^2 + (2 - x_2)^2, 2 \exp(-x_1 + x_2)\}.
\]

**Example 3** Crescent\(^3\).

\[
f(x) = \max\{x_1^2 + (x_2 - 1)^2 + x_2 - 1, -x_1^2 - (x_2 - 1)^2 + x_2 + 1\}.
\]

**Example 4** Demyanov-Malozemov\(^3\).

\[
f(x) = \max\{5x_1 + x_2, -5x_1 + x_2, x_1^2 + x_2^2 + 4x_2\}.
\]

**DISCUSSION**

Bandler and Charalambous gave a special transformation method in Ref. 5, where the nonlinear
Table 5 Numerical results for Example 5 by Algorithm 1 and the smoothing gradient method.

<table>
<thead>
<tr>
<th>x₀</th>
<th>x*</th>
</tr>
</thead>
<tbody>
<tr>
<td>By Algorithm 1.</td>
<td></td>
</tr>
<tr>
<td>(0.3,1,4,1,−0.4)</td>
<td>(0.0021, 1.0035, 1.9896, −0.9962)</td>
</tr>
<tr>
<td>(0.2,1,2,1,−0.3)</td>
<td>(0.0035, 0.9997, 2.0001, −0.9948)</td>
</tr>
<tr>
<td>(0.2,1,2,2,−0.1)</td>
<td>(0.0057, 0.9980, 1.9998, −0.9905)</td>
</tr>
<tr>
<td>(0.2,1,9,1,−0.17)</td>
<td>(−0.0018, 0.9971, 1.9973, −0.9937)</td>
</tr>
<tr>
<td>(0.18,1,4,1,89,−0.25)</td>
<td>(0.0000, 0.9971, 1.9960, −0.9953)</td>
</tr>
<tr>
<td>(0.28,1,6,1,79,−0.23)</td>
<td>(−0.0011, 0.9992, 1.9967, −0.9936)</td>
</tr>
</tbody>
</table>

By the SGM.
| (0.3,1,4,1,−0.4) | (0.0483, 0.8403, 1.9961, −1.0171) |
| (0.2,1,2,1,9,−0.3) | (0.0037, 0.9667, 1.9995, −0.9880) |
| (0.2,1,1,2,2,−0.1) | (−0.1158, 0.9993, 2.0759, −0.8289) |
| (0.2,1,9,1,3,−0.17) | (−0.1157, 0.9940, 2.0791, −0.8238) |
| (0.18,1,4,1,89,−0.25) | (0.0005, 0.9667, 2.0010, −0.9868) |
| (0.28,1,6,1,79,−0.23) | (−0.1531, 0.9676, 2.1100, −0.7765) |

constrained optimization problem

\[ \min g_i(x) \]

s.t. \[ g_i(x) \geq 0 \quad i = 2, \ldots, m, \quad x \in \mathbb{R}^n, \]

where \( g_i : \mathbb{R}^n \to \mathbb{R} \) are continuously differentiable functions for \( i = 1, \ldots, m \) was transformed into (1). From Ref. 5, for sufficiently large \( \alpha_i, i = 1, \ldots, m \), we know that the above nonlinear constrained optimization problem coincides with the unconstrained minimax problem

\[ \min \max_{1 \leq i \leq m} f_i(x), \]

where \( f_i(x) = g_i(x), f_i(x) = g_i(x) - \alpha_i g_i(x), 2 \leq i \leq m \). We can see that (15) is a special case of (1). Hence we can solve the nonlinear constrained optimization problem by Algorithm 1.

Example 5 Rosen-Suzuki problem

\[ \min g_i(x) \]

s.t. \[ g_i(x) \geq 0 \quad i = 2, 3, 4, \quad x \in \mathbb{R}^n, \]

where \( g_i(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4, g_2(x) = -x_1^2 - x_2^2 - x_3^2 - x_4^2 + 21x_3 + 7x_4, g_3(x) = -x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 + 10, g_4(x) = -x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 + 5. \]

Using Algorithm 1, we obtain the numerical results for Example 5 in Table 5 where the numerical results for Example 5 by the smoothing gradient method are also given.

Acknowledgements: This work is supported by National Natural Science Foundation of China (11101231, 11401331), Natural Natural Science Foundation of Shandong (No. ZR2015AQ013) and Key Issues of Statistical Research in Shandong Province (KT15173). The authors are grateful to the referees for their insightful comments.

REFERENCES