Filomat 35:3 (2021), 737–758
https://doi.org/10.2298/FIL2103737H

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

A New Three-Term Conjugate Gradient Method for Solving the Finite Minimax Problems

Yue Hao[a], Shouqiang Du[a], Yuanyuan Chen[a]

[a] *School of Mathematics and Statistics, Qingdao University*

Abstract. In this paper, we consider the method for solving the finite minimax problems. By using the exponential penalty function to smooth the finite minimax problems, a new three-term nonlinear conjugate gradient method is proposed for solving the finite minimax problems, which generates sufficient descent direction at each iteration. Under standard assumptions, the global convergence of the proposed new three-term nonlinear conjugate gradient method with Armijo-type line search is established. Numerical results are given to illustrate that the proposed method can efficiently solve several kinds of optimization problems, including the finite minimax problem, the finite minimax problem with tensor structure, the constrained optimization problem and the constrained optimization problem with tensor structure.

Key Words: finite minimax problem; tensor structure; three-term nonlinear conjugate gradient method; global convergence; polynomial complementarity problem

1. Introduction

The finite minimax problem is one of the most important research topics in the field of optimization research[1–7]. This kind of problem is also an important nonsmooth optimization problem, which is widely used in engineering design, economic decision-making, game theory, nonlinear programming problems and multi-objective programming problems (such as [8–10] and the references therein).

The general form of the finite minimax problem is

$$\min_{x\in\mathbb{R}^n} \quad F(x), F(x) = \max_{i=1,\dots,m} f_i(x), \tag{1.1}$$

where $f_i(x) : \mathbb{R}^n \to \mathbb{R}$ are continuous differentiable functions.

It is important to design an efficient algorithm to solve the finite minimax problem. As far as we know, many well-established methods are used to solve the finite minimax problems [1–9, 11]. Such as in [1], the researchers showed that they could formulate the conventional nonlinear programming problem as an unconstrained minimax problem. In [2], the researcher proposed an effective method to solve minimax problems referred to the aggregate method. A similar approach has been described in [3] with constructing a penalty function. By constructing an interval extension of adjustable entropy

2010 *Mathematics Subject Classification*. Primary 90C47; Secondary 90C30, 15A69
Keywords. finite minimax problem; tensor structure; three-term nonlinear conjugate gradient method; global convergence; polynomial complementarity problem
Received: 28 January 2020; Revised: 03 December 2020; Accepted: 28 February 2021
Communicated by Predrag Stanimirović
Research supported by National Nature Science Foundation of China(Grant No.11671220)
Email addresses: hytawh1218@163.com (Yue Hao), sqdu@qdu.edu.cn (Shouqiang Du), yychen@qdu.edu.cn (Yuanyuan Chen)

function and some region deletion test rules, the researchers presented a new interval algorithm in [4].A truncated aggregate smoothing Newton method was presented for solving minimax problems in [5].In [6], the researcher concerned on the smoothing method to solve the minimax problems. The researchers gave a smoothing FR conjugate gradient method for solving minimax problem in [7].

By the above literatures, we know that penalty function plays a very important role in solving the finite minimax problems. The form of the exponential penalty function proposed in [11] is

$$g(x,t) = t\ln\sum_{i=1}^{m} exp(\frac{g_i(x)}{t}), \tag{1.2}$$

where t is the penalty parameter, $g_i(x)$ are continuous differential functions.

For the finite minimax problem (1.1), we can transform it into a smooth unconstrained optimization problem, which can be showed as

$$\min \tilde{F}(x,t). \tag{1.3}$$

In (1.3), $\tilde{F}(x,t)$ is a differentiable function defined as

$$\tilde{F}(x,t) = t\ln\sum_{i=1}^{m} exp(\frac{f_i(x)}{t}),$$

where $t > 0$ is a penalty parameter.

For three-term conjugate gradient methods [12–22] and the related methods [23–26] posse low memory requirement and simple implementation, conjugate gradient methods are often used for solving unconstrained optimization problems. Recently, many researchers have presented kinds of three-term conjugate gradient method for solving the unconstrained optimization problems.Hence, we consider using three-term conjugate gradient method to solve the finite minimax problems. This is also one of our motivations of this paper. On the other hand, tensor optimization problem is a new proposed optimization problem(one can see [27–33] and the references therein). In particular, the finite minimax problems with the tensor structure have not been considered. Therefore, we will study the finite minimax problems with the tensor structure in this paper. And this is the other motivation of this paper. Now, we give the finite minimax problems with the tensor structure.

Throughout this paper, we denote $\mathcal{A}$ as an m-th order n-dimensional tensor. $\mathcal{A} = (a_{i_1,i_2\ldots i_m})$, where $a_{i_1,i_2\ldots i_m} \in \mathbb{R}$ for $1 \le i_1, i_2 \ldots i_m \le n$. We denote $T_{m,n}$ as the set of all real m-th order n-dimensional square tensors. For a tensor $\mathcal{A} \in T_{m,n}$, $x \in \mathbb{R}^n$, $\mathcal{A}x^{m-1}$ is a vector, whose i-th component is

$$(\mathcal{A}x^{m-1})_i = \sum_{i_2,\ldots,i_m=1}^{n} a_{ii_2\ldots i_m} x_{i_2}\cdots x_{i_m}, \quad i = 1,2,\ldots,n.$$

For symmetric tensor $\mathcal{A}$, we know that $(\mathcal{A}x^{m-1})' = (m-1)\mathcal{A}x^{m-2}$, where $\mathcal{A}x^{m-2}$ denotes an $n \times n$ matrix, whose (i, j)-th component is defined as

$$(\mathcal{A}x^{m-2})_{ij} = \sum_{i_3,\ldots,i_m=1}^{n} a_{iji_3\ldots i_m} x_{i_3}\cdots x_{i_m}, \quad i,j = 1,2,\ldots,n.$$

Denote

$$h^i(x) = \mathcal{A}^i x^{m-1} - |x| - b^i, \quad i = 1,2,\ldots,l, \tag{1.4}$$

where $\mathcal{A}^i$ is an m-th order n-dimensional tensor, b^i is an n-th dimensional vector. The j-th of (1.4) is

$$h^i_j(x) = (\mathcal{A}^i x^{m-1})_j - \left|x_j\right| - b^i_j, \quad j = 1,2,\ldots,n.$$

It is obviously that $\sqrt{x^2+t} \longrightarrow |x|$ as $t \longrightarrow 0$. $h^i(x)$ can be converted to

$$h^i(x,t) = \mathcal{A}^i x^{m-1} - \sqrt{x^2+t} - b^i, \quad i = 1, 2, \ldots, l.$$

For

$$\max\{h_1^i(x,t), h_2^i(x,t), \ldots, h_m^i(x,t)\},$$

by (1.2), it can be defined as

$$\tilde{h}^i(x,t) = t \ln \sum_{j=1}^{n} exp\left(\frac{(\mathcal{A}^i x^{m-1})_j - \sqrt{x_j^2+t} - b_j^i}{t}\right),$$

where $t > 0$ is a penalty parameter.

For

$$\max\{\tilde{h}^1(x,t), \tilde{h}^2(x,t), \ldots, \tilde{h}^l(x,t)\},$$

similar to (1.3), we can transform the finite minimax problem with tensor structure into a smooth unconstrained optimization problems (1.3), $\tilde{F}(x,t)$ in (1.3) can be defined as

$$\tilde{F}(x,t) = t \ln \sum_{i=1}^{l} exp\left(\frac{\tilde{h}^i(x,t)}{t}\right). \tag{1.5}$$

The structure of the following paper is organized as follows. In Section 2, we propose the new three-term conjugate gradient method with Armijo-type line search, and use it to solve the finite minimax problems and the finite minimax problem with tensor structure.Under some wild assumptions, we prove the global convergence of the proposed method.In Section 3,we report some numerical results for some well-known finite minimax problems, the minimax problem with tensor structure, the constrained optimization problem and the constrained optimization problem with tensor structure to show the effectiveness of the proposed method. In Section 4, we complete this paper by drawing some conclusions.

2. New three-term conjugate gradient method

In this section, the new three-term conjugate gradient method with Armijo-type line search is proposed. We also give the convergence analysis of the proposed method.

2.1. New three-term conjugate gradient method for smooth case

The unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x),$$

where $f : \mathbb{R}^n \to \mathbb{R}$, and the gradient of f is available. Denote g is the gradient of f, g_k is the function value of g at x_k.

The iterative process of the nonlinear conjugate gradient method is given by $x_{k+1} = x_k + \alpha_k d_k$, $k = 0, 1, \ldots,$ where the step length α_k can be obtained by line search, and the search direction is defined by

$$d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1}, & \text{if } k > 0. \end{cases}$$

There are many well-known conjugate gradient methods, such as [12–26, 34–42]. In [12], the iterative formula of three-term conjugate gradient method is

$$x_{k+1} = x_k + \alpha_k d_k,$$

where

$$d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -\beta_k^1 o_k + \beta_k^2 p_k + \beta_k^3 q_k, & \text{if } k > 0, \end{cases}$$

o_k, p_k and q_k are directions and β_k^1, β_k^2 and β_k^3 are parameters.

Based on the above methods,we give a new three-term conjugate gradient method with Armijo-type line search.

The direction is generated by

$$d_k = \begin{cases} -\beta_k^1 o_k + \beta_k^2 p_k + \beta_k^3 q_k, & k > 0 \ \& \left|g_k^T d_k^T\right| \geq \Delta\|d_k\|\|g_k\|, \\ -g_k, & \text{otherwise}, \end{cases} \tag{2.1}$$

where $\Delta \in (0, 1)$, the parameter β_k^i is defined by

$$\beta_k^1 = 1 + \gamma_k \frac{g_k^\top d_{k-1}}{y_{k-1}^\top d_{k-1}}, \qquad \beta_k^2 = \gamma_k \beta_k^{DY} + \beta_k^{PR}, \qquad \beta_k^3 = -\frac{g_k^\top d_{k-1}}{\|g_{k-1}\|^2} \tag{2.2}$$

and

$$o_k := g_k, \quad p_k := d_{k-1}, \quad q_k := y_{k-1}, \tag{2.3}$$

where

$$\beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1} y_{k-1}}, \qquad \beta_k^{PRP} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2},$$

and $y_{k-1} := g_k - g_{k-1}$, $\{\gamma_k\}_{k\geq 0}$ is a decreasing sequence satisfying $\lim_{k\to\infty} \gamma_k = 0$. The step length α_k satisfies the following conditions

$$\alpha_k = \max_j \{\sigma^j \frac{\tau \left|g_k^\top d_k\right|}{\|d_k\|^2}, \quad j = 0, 1, 2, \cdots\}, \tag{2.4}$$

$$f(x_k + \alpha_k d_k) - f(x_k) \leq -\delta \alpha_k^2 \|d_k\|^2, \tag{2.5}$$

where $\delta > 0$, $\sigma \in (0, 1)$, $\tau > 0$.

Now, we give the new three-term conjugate gradient method.

Algorithm 2.1

Step0. Choose constants $\delta > 0$, $\sigma \in (0, 1)$, $\tau > 0$, $\Delta \in (0, 1)$, $\varepsilon > 0$, an initial point $x_0 \in \mathbb{R}^n$, compute g_0, let $d_0 = -g_0$.

Step1. If $\|d_0\| \leq \varepsilon$, stop; otherwise, go to *Step* 2;

Step2. Compute $d_k = \beta_k^1 o_k + \beta_k^2 p_k + \beta_k^3 q_k$ from (2.1), if $\left|g_k^T d_k\right| \geq \Delta\|d_k\|\|g_k\|$, go to *Step* 3; otherwise, let $d_k = -g_k$, go to *Step* 3;

Step3. Determine α_k satisfying (2.4) and (2.5);

Step4. Get the next iterate $x_{k+1} = x_k + \alpha_k d_k$, and compute f_{k+1} and g_{k+1};

Step5. If $\|d_k\| \leq \varepsilon$, stop; otherwise, go to *Step* 6;

Step6. Let $k = k + 1$, go to *Step* 2.

Next, we will establish the global convergence of Algorithm 2.1, which need the following standard assumptions:

(A1) The level set $L(x_0) = \{x \in \mathbb{R}^n | f(x) \leq f(x_0)\}$ is bounded.

(A2) The gradient of objective function f is Lipschitz continuous on the open bounded convex set Ω of

$L(x_0)$, *i.e.*,

$$\|g(x) - g(y)\| \le L_1\|x - y\|, \qquad \forall x, y \in \Omega,$$

where L_1 is the Lipschitz constant.

From Assumption (A1) and (A2), it can be obtained that there exists a constant $\Gamma > 0$, subject to

$$\|g(x)\| \le \Gamma. \tag{2.6}$$

From Assumption(A2), we also have

$$\|y_k\| \le L_1\alpha_k\|d_k\|. \tag{2.7}$$

Similar to [12],we get the following lemma.

Lemma 2.1 Let d_k be generated by Algorithm 2.1. Then

$$g_k^\top d_k = -\|g_k\|^2. \tag{2.8}$$

Proof. From (2.1)-(2.3), we have

$$\begin{aligned} g_k^T d_k &= g_k^T(-\beta_k^1 o_k + \beta_k^2 p_k + \beta_k^3 q_k) \\ &= -(1 + \gamma_k \frac{g_k^T d_{k-1}}{y_{k-1}^T d_{k-1}})\|g_k\|^2 + \gamma_k \frac{g_k^T g_k}{y_{k-1}^T d_{k-1}} g_k^T d_{k-1} \\ &+ \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2} g_k^T d_{k-1} - \frac{g_k^T d_{k-1}}{\|g_{k-1}\|^2} g_k^T y_{k-1} \\ &= -\|g_k\|^2 < 0. \end{aligned}$$

□

Lemma 2.2 Suppose that Assumption (A1) and (A2) hold, we have

$$\sum_{k\ge 0} \frac{(g_k^\top d_k)^2}{\|d_k\|^2} = \sum_{k\ge 0} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty.$$

Proof. Firstly, we prove that there exists a constant $c_1 > 0$ such that

$$\alpha_k \ge c_1 \frac{\left|g_k^\top d_k\right|}{\|d_k\|^2}, \quad \forall k \ge 0. \tag{2.9}$$

Now,there are two cases.

Case(i) When $\alpha_k = \tau \frac{\left|g_k^\top d_k\right|}{\|d_k\|^2}$, we have $\alpha_k \ge \tau \frac{\left|g_k^\top d_k\right|}{\|d_k\|^2}, \quad \forall k \ge 0$, let $c_1 = \tau$, then (2.9) holds.

Case(ii) When $\alpha_k < \tau \frac{\left|g_k^\top d_k\right|}{\|d_k\|^2}$, it follows that $\sigma^{-1}\alpha_k$ is contradict with (2.5). So, we have

$$f(x_k + \sigma^{-1}\alpha_k d_k) - f(x_k) > -\delta(\sigma^{-1}\alpha_k)^2\|d_k\|^2. \tag{2.10}$$

By Assumption(A2), we know that there exists $\xi \in (0, 1)$ such that

$$
\begin{aligned}
f(x_k+\sigma^{-1}\alpha_k d_k)-f(x_k) &= \sigma^{-1}\alpha_k g(x_k+\xi\sigma^{-1}\alpha_k d_k)^T d_k \\
&= \sigma^{-1}\alpha_k g_k^T d_k + \sigma^{-1}\alpha_k[g(x_k+\xi\sigma^{-1}\alpha_k d_k)-g(x_k)]^T d_k \\
&\le \sigma^{-1}\alpha_k g_k^T d_k + \sigma^{-1}\alpha_k d_k L_1 \xi \sigma^{-1}\alpha_k d_k \\
&= \sigma^{-1}\alpha_k g_k^T d_k + L_1\sigma^{-2}\alpha_k^2\|d_k\|^2.
\end{aligned}
$$

Combined with the above inequation and (2.10), we obtain that

$$\alpha_k > \frac{\sigma}{\delta+L_1}\frac{\left|g_k^\top d_k\right|}{\|d_k\|^2}.$$

Denote $c=\min\{\tau, \dfrac{\sigma}{\delta+L_1}\}$, then we get (2.9).

From (2.5), Assumption (A1) and (A2), we know that

$$\delta\sum_{k\ge 0}\alpha_k^2\|d_k\|^2<\infty. \tag{2.11}$$

By (2.8) and (2.11), we get

$$\sum_{k\ge 0}\frac{(g_k^\top d_k)^2}{\|d_k\|^2}=\sum_{k\ge 0}\frac{\|g_k\|^4}{\|d_k\|^2}<\infty.$$

We finish the proof of this lemma. □

Lemma 2.3 Let $\{x_k\}$ is the sequence generated by Algorithm 2.1, then

$$\lim_{k\to\infty}\alpha_k\|d_k\|=0. \tag{2.12}$$

Proof. By (2.5), we have

$$\sum_{k=0}^{\infty}\alpha_k^2\|d_k\|^2<\infty.$$

Then

$$\lim_{k\to\infty}\alpha_k^2\|d_k\|^2=0,$$

i.e.,

$$\lim_{k\to\infty}\alpha_k\|d_k\|=0.$$

We get this lemma. □

Lemma 2.4 Let $\{x_k\}$ is generated by Algorithm 1. If

$$y_k^\top d_k\ge\Delta\|g_k\|\|d_k\|$$

and $\varepsilon>0$, for $\forall k$,we have

$$\|g_k\|\ge\varepsilon.$$

Then, there exists a constant $M>0$, satisfies

$$\|d_k\|\le M.$$

Proof. For convenience, we denote $I=\{k|k>0\&\left|g_k^T d_k^T\right|\ge\Delta\|d_k\|\|g_k\|\}$. From (2.1), (2.2), (2.3), (2.6) and (2.7), the following inequation holds for any $k\in I$:

$$
\begin{aligned}
\|d_k\| &\le \left|1+\gamma_k\frac{g_k^T d_{k-1}}{y_{k-1}^T d_{k-1}}\right|\|g_k\| + \gamma_k\left|\beta^{DY}\right|\|d_{k-1}\| + \frac{\left|g_k^T y_{k-1}\right|}{\|g_{k-1}\|^2}\|d_{k-1}\| + \frac{\left|g_k^T d_{k-1}\right|}{\|g_{k-1}\|^2}\|y_{k-1}\| \\
&\le \|g_k\| + \gamma_k\frac{\left|g_k^T d_{k-1}\right|}{\left|y_{k-1}^T d_{k-1}\right|}\|g_k\| + \gamma_k\frac{\|g_k\|^2}{\left|d_{k-1}^T y_{k-1}\right|}\|d_{k-1}\| + \frac{\|g_k\|\|y_{k-1}\|}{\|g_{k-1}\|^2}\|d_{k-1}\| \\
&\quad + \frac{\|g_k\|\|d_{k-1}\|}{\|g_{k-1}\|^2}\|y_{k-1}\| \\
&\le \|g_k\| + \gamma_k\frac{\|g_k\|^2\|d_{k-1}\|}{\Delta\|g_{k-1}\|\|d_{k-1}\|} + \gamma_k\frac{\|g_k\|^2\|d_{k-1}\|}{\Delta\|g_{k-1}\|\|d_{k-1}\|} + \frac{2L_1\|g_k\|\alpha_{k-1}\|d_{k-1}\|^2}{\|g_{k-1}\|^2} \\
&\le \Gamma + 2\gamma_k\frac{\Gamma^2}{\Delta\varepsilon} + 2L_1\alpha_{k-1}\frac{\Gamma}{\varepsilon^2}\|d_{k-1}\|^2 \\
&= \Gamma + \hat{\gamma_{k1}} + \hat{\gamma_{k2}}\|d_{k-1}\|,
\end{aligned}
$$

where $\hat{\gamma_{k1}} = 2\gamma_k\frac{\Gamma^2}{\Delta\varepsilon}$, $\hat{\gamma_{k2}} = 2L_1\alpha_{k-1}\frac{\Gamma}{\varepsilon^2}\|d_{k-1}\|$.
From (2.12), we have

$$\lim_{k\to\infty}\alpha_k\|d_k\| = 0$$

and

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

Hence, there are $\Lambda_1 \in (0,1)$, $\Lambda_2 \in (0,1)$, and integer k_1, k_2 such that for $k \ge k_1$, there exists $\hat{\gamma_{k1}} < \Lambda_1$, for $k \ge k_2$, there exists $\hat{\gamma_{k2}} < \Lambda_2$, respectively.

Taking $k_0 = \max\{k_1, k_2\}$, $\Lambda = \max\{\Lambda_1, \Lambda_2\}$, then, for any $k \ge k_0$, we have $\hat{\gamma_{k1}} < \Lambda$ and $\hat{\gamma_{k2}} < \Lambda$. Therefore, we get

$$
\begin{aligned}
\|d_k\| &\le \Gamma + \hat{\gamma_{k1}} + \hat{\gamma_{k2}}\|d_{k-1}\| \\
&\le \Gamma + \Lambda + \Lambda\|d_{k-1}\| \\
&\le (\Lambda + \Lambda^2 + \cdots + \Lambda^{k-k0}) + \Gamma(1 + \Lambda + \Lambda^2 + \cdots + \Lambda^{k-k0-1}) + \Lambda^{k-k0}\|d_{k0}\| \\
&\le \frac{1}{1-\Lambda} + \Gamma\frac{1}{1-\Lambda} + \|d_{k0}\|.
\end{aligned}
$$

Denote

$$M = \max\{\|d_1\|, \|d_2\|, \cdots, \|d_{k0}\|, \frac{1}{1-\Lambda} + \Gamma\frac{1}{1-\Lambda} + \|d_{k0}\|\}.$$

We have

$$\|d_k\| \le M.$$

For any $k \notin I$, we also get

$$\|d_k\| = \|g_k\| \le \Gamma =: M.$$

□

Theorem 2.1 Suppose that Assumptions (A1) and (A2) hold, we have

$$\lim_{k\to\infty}\|g_k\| = 0.$$

Proof. From Lemma 2.4, we have

$$\sum_{k=0}^{\infty} M^{-2}\|g_k\|^4 \le \sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} = \sum_{k=0}^{\infty} \frac{(g_k^\top d_k)^2}{\|d_k\|^2} < \infty.$$

Then, we have $\lim_{k\to\infty}\|g_k\| = 0$. The proof is completed. □

2.2. New three-term conjugate gradient method for solving the finite minimax problems

In the following of this subsection, we use the proposed method to solve the finite minimax problems.

Denote $\tilde{g}(x_k, t_k)$ as the gradient of $\tilde{F}(x, t)$ at x_k, and e_j^T is the j-th row of the unit matrix I. For any $t > 0$, from (1.5), we get

$$\nabla_x \tilde{F}(x,t) = \left(\frac{\partial \tilde{F}(x,t)}{\partial x_1}, \frac{\partial \tilde{F}(x,t)}{\partial x_2}, \dots, \frac{\partial \tilde{F}(x,t)}{\partial x_n}\right),$$

and

$$\nabla_x \tilde{F}(x,t) = \frac{I}{\sum_{i=1}^{l} exp\left[\ln \sum_{j=1}^{n} exp\left(\frac{h_j^i(x,t)}{t}\right) \cdot \sum_{j=1}^{n} exp\left(\frac{h_j^i(x,t)}{t}\right)\right]},$$

where

$$I = \sum_{i=1}^{l}\left\{exp\left[\ln \sum_{j=1}^{n} exp\left(\frac{h_j^i(x,t)}{t}\right)\right] \cdot \sum_{j=1}^{n}\left[exp\left(\frac{h_j^i(x,t)}{t}\right)\left((\mathcal{A}^i x^{m-1})_j' - \frac{x_j}{\sqrt{x_j^2+t}} e_j^T\right)\right]\right\}.$$

Now, we give the new smoothing three-term conjugate gradient method for solving the finite minimax problems.

Algorithm 2.2

Step0. Choose constants $\tau > 0$, $\gamma_1 > 0$, $\sigma \in (0,1)$, $\Delta \in (0,1)$, $\delta > 0$, $\sigma_1 \in (0,1)$, $\varepsilon > 0$. Choose an initial point (x_0, t_0), compute g_0, let $d_0 = -g_0$;

Step1. If $\|d_0\| \le \varepsilon$, stop; otherwise, go to *Step* 2;

Step2. Compute $d_k = \beta_k^1 o_k + \beta_k^2 p_k + \beta_k^3 q_k$ from (2.1), if $\left|\tilde{g}_k^T d_k\right| \ge \Delta\|d_k\|\|g_k\|$, go to *Step* 3; otherwise, let $d_k = -g_k$, go to *Step* 3;

Step3. Compute α_k satisfies

$$\alpha_k = \max_j\{\sigma^j \frac{\tau\left|\tilde{g}_k^\top d_k\right|}{\|d_k\|^2}, \quad j = 0,1,2,\cdots\},$$

$$\tilde{F}(x_k + \alpha_k d_k, t_k) - \tilde{F}(x_k, t_k) \le -\delta\alpha_k^2\|d_k\|^2;$$

Step4. Get the next iterate $x_{k+1} = x_k + \alpha_k d_k$, and compute ${}_{k+1}(x,t)$ and $\tilde{g}_{k+1}$;

Step5. If $\|d_k\| \le \varepsilon$, stop; otherwise, go to *Step* 6;

Step6. If $\|\nabla_x \tilde{F}(x_{k+1}, t_k)\| \ge \gamma_1 t_k$, let $t_{k+1} = t_k$; otherwise, let $t_{k+1} = \sigma_1 t_k$;

Step7. Let $k = k+1$, go to *Step* 2.

The following theorem give the convergence of the new smoothing three-term conjugate gradient method.

Theorem 2.2 Suppose that $\tilde{F}(\cdot, t)$ is a smoothing function of $F(\cdot, t)$. If for every fixed $t > 0$, $\tilde{F}(\cdot, t)$ satisfies

Assumption (A1) and (A2), and $\{x_k\}$ is generated by Algorithm 2.2, then

$$\liminf_{k\to\infty} \|\nabla_x \tilde{F}(x_{k+1}, t_k)\| = 0.$$

Proof. Denote $K = \{k | t_{k+1} = \sigma_1 t_k\}$. Assume K is finite, there exists an integer $\hat{k}$ such that

$$\|\nabla_x \tilde{F}(x_k, t_{k-1})\| \geq \sigma_1 t_k, \quad for \quad all \quad k > \hat{k}. \tag{2.13}$$

Let $t_k = t_{\hat{k}} = \bar{t}$, (1.3) can be converted to solve

$$\min_{x\in\mathcal{R}^n} \tilde{F}(x, \bar{t}).$$

From Theorem 2.1, we have

$$\liminf_{k\to\infty} \|\nabla_x \tilde{F}(x_k, \bar{t})\| = 0,$$

which is contradicted with (2.13). Hence, we get K is an infinite set. So, $\lim_{k\to\infty} t_k = 0$. Taking $K = \{k_0, k_1, \cdots\}$, where $k_0 < k_1 < \cdots$. Then, we have

$$\liminf_{i\to\infty} \|\nabla_x \tilde{F}(x_{k_i+1}, t_{k_i})\| \leq \sigma_1 \lim_{i\to\infty} t_{k_i} = 0.$$

□

3. Numerical results

In order to show the effectiveness of the proposed method, four types of the finite minimax problems are considered in this section. We further report some numerical results, which are the numerical comparisons of our method with the smoothing Fletcher-Reeves conjugate gradient method (SFR)[7] and fminunc in the MATLAB tool box. We list some tables and figures to show the numerical results.

3.1. *The finite minimax problem*

In this subsection, we use the test problems taken from [3]. We define

$$\gamma_k = \frac{\delta_1}{(1+5k)^{\zeta}},$$

where k is the number of iterations. The parameters are taken as follows:
$\varepsilon = 10^{-5}$, $\Delta = 0.1$, $\delta_1 = 10^{-4}$, $\zeta = 0.25$, $\sigma = 0.3$, $\delta = 0.9$, $\tau = 0.7$, $\sigma_1 = 0.5$, $\gamma_1 = 0.5$, $t_0 = 2$.

We choose $\|\nabla_x \tilde{F}(x_k, t_k)\| \leq \varepsilon$ as the terminate condition. And we give the numerical results in Table 3.1, where n, m and *iter* is denoted the number of variables, the number of functions and the iterations of the algorithm, respectively. Let $\Delta(h) = |h(x_k) - h(x^*)|$. We also give the numerical results in Figures 3.1-3.5. In addition to SFR, Algorithm 2.2 is also compared with fminunc, we can find that Algorithm 2.2 has much smaller error than SFR, and the fminunc can only solve Examples 3.1-3.2. From Table 3.1, Figures 3.1-3.5, we can see that our method is effectively to solve the finite minimax problems.

Remark 3.1. *In addition to SFR and fminunc, we also compared with HSM[23], which is also efficient for solving unconstrained optimization problems proposed recently.We find out that HSM is very sensitive to its parameters when solving unconstrained optimization problems after smoothing with t. We will do more research in the future.*

Example 3.1 Crescent [3].
$h(x) = \max\{x_1^2 + (x_2-1)^2 + x_2 - 1, -x_1^2 - (x_2-1)^2 + x_2 + 1\}$,
$n = 2$, $h(x^*) = 0$, $x_0 = (-1.5, 2)$.

Example 3.2 Mifflin 1 [3].
$h(x) = -x_1 + \max\{x_1^2 + x_2^2 - 1, 0\}$,
$n = 2$, $h(x^*) = -1$, $x_0 = (0.8, 0.6)$.

Example 3.3 Mifflin 2 [3].
$h(x) = -x_1 + 2(x_1^2 + x_2^2 - 1) + 1.75 \max\{\pm(x_1^2 + x_2^2 - 1\}$,
$n = 2$, $h(x^*) = -1$, $x_0 = (-1, -1)$.

Example 3.4 Hald-Madsen 1 [3].
$h(x) = \max\{\pm 10(x_2 - x_1^2), \pm(1 - x_1)\}$,
$n = 2$, $h(x^*) = 0$, $x_0 = (1.2, 1)$.

Example 3.5 Maxq [3].
$h(x) = \max\{x_u^2\}$,
$n = 20$, $h(x^*) = 0$, $x_0 = (1, \ldots, 10, -11, \ldots, -20)$.

Table 3.1: The numerical results for Examples 3.1-3.5

problems	n	m	iter	$\Delta(h)$
By Algorithm 2.2				
Crescent	2	2	282	$4.2903e-06$
Mifflin 1	2	2	79	$1.0577e-05$
Mifflin 2	2	2	223	$6.8712e-06$
Hald-Madsen 1	2	4	161	$1.0577e-05$
Maxq	20	20	193	$4.5712e-05$
By SFR				
Crescent	2	2	9	0.3906
Mifflin 1	2	2	29	0.0014
Mifflin 2	2	2	18	0.0606
Hald-Madsen 1	2	4	78	0.0873
Maxq	20	20	115	$1.1428e-05$
By fminunc				
Crescent	2	2	\	0.3571
Mifflin 1	2	2	\	0.2
Mifflin 2	2	2	\	\
Hald-Madsen 1	2	4	\	\
Maxq	20	20	\	\

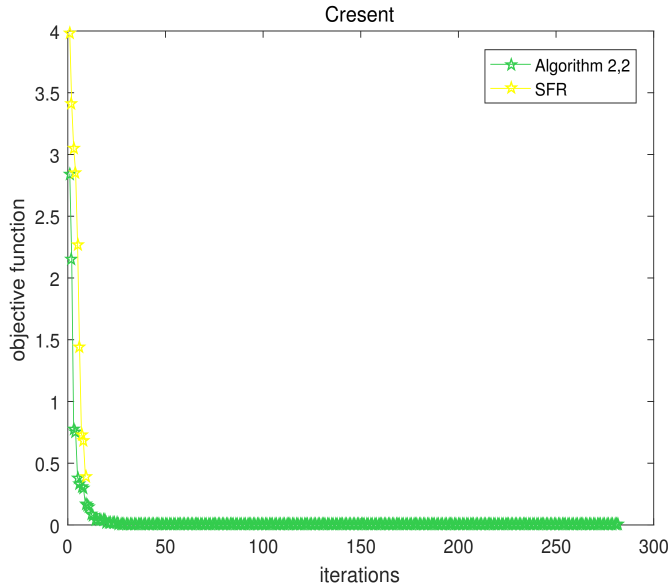

Figure 3.1: The numerical results for Example 3.1

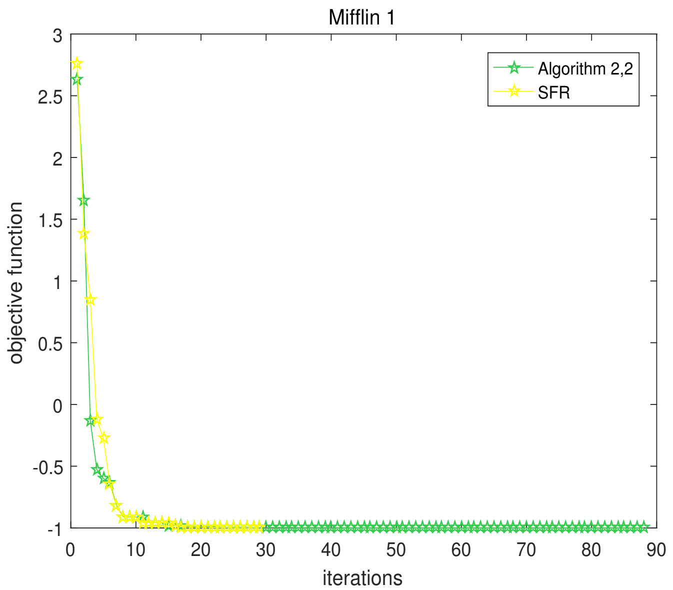

Figure 3.2: The numerical results for Example 3.2

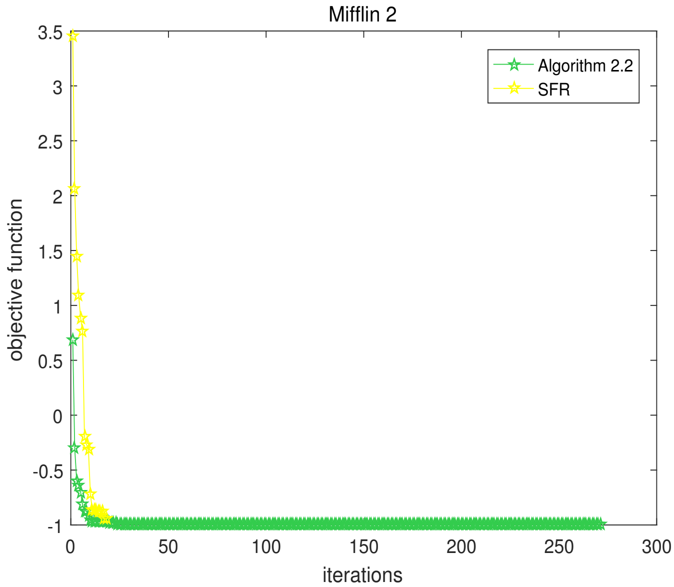

Figure 3.3: The numerical results for Example 3.3

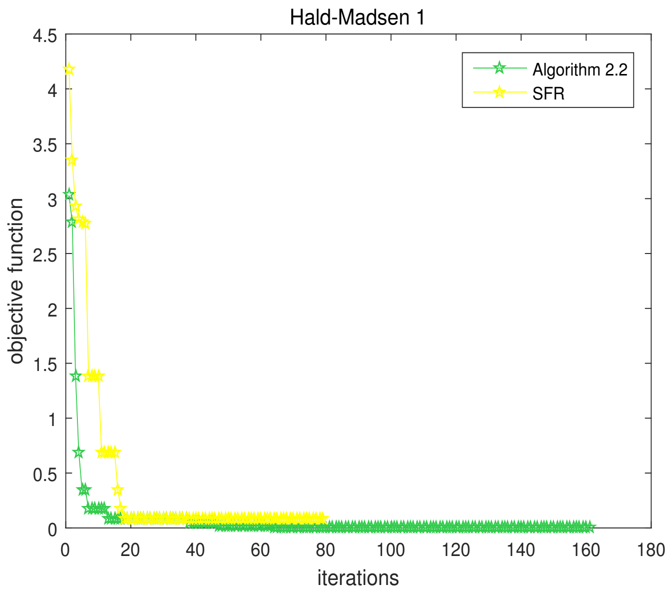

Figure 3.4: The numerical results for Example 3.4

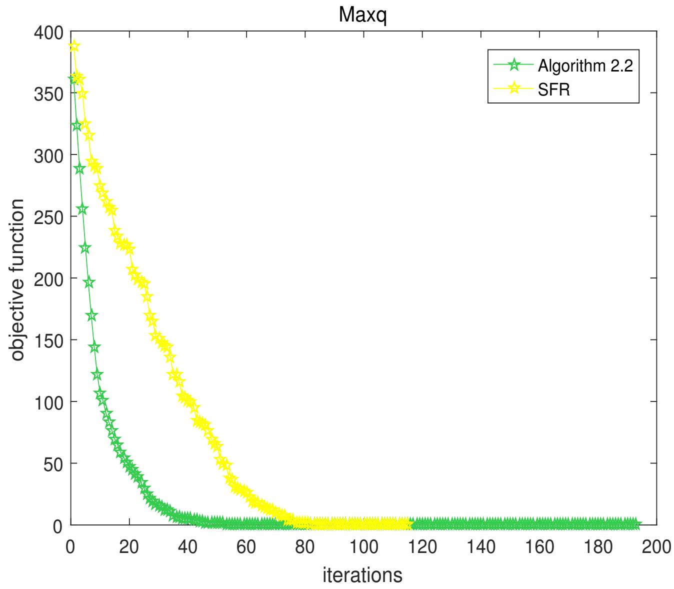

Figure 3.5: The numerical results for Example 3.5

3.2. *The finite minimax problem with tensor structure*

In this subsection, we give the numerical experiments of the finite minimax problem with tensor structure. From [43],we know that for any tensor $\mathcal{A} \in T_{m,n}$, there exists a unique semi-symmetric tensor $\hat{\mathcal{A}} \in T_{m,n}$ such that $\hat{\mathcal{A}}x^{m-1} = \mathcal{A}x^{m-1}$ for any $x \in \mathbb{R}^n$. Hence, we assume that $\mathcal{A}x^{m-1}$ is semi-symmetric tensor, and the Jacobian of $\mathcal{A}x^{m-1}$ at x is given by $(m-1)\mathcal{A}x^{x-2}$. We choose $\varepsilon = 10^{-3}$, $\Delta = 0.7$, $\delta_1 = 10^{-4}$, $\zeta = 0.4$, $\sigma = 0.3$, $\delta = 0.3$, $\tau = 0.4$, $\sigma_1 = 0.5$, $\gamma_1 = 0.3$. We use $\|x_{k+1} - x_k\| \leq \varepsilon$ as the terminate condition.

The numerical results are given in Table 3.2, in which the m, n and iter denote the order of the tensor, the dimensional of the tensor and the number of iterations of the algorithm, respectively. We also give the numerical results in Figures 3.6-3.8. From Table 3.2 and Figures 3.6-3.8, we know that Algorithm 2.2 and SFR have similar numerical results and both of them are effectively for solving the finite minimax problems with tensor structure.

Example 3.6 Consider the finite minimax problem (1.1) with tensor structure (1.4), where the two 3-th order 2-dimensional tensors $\mathcal{A}^1 = (a^1_{i_1i_2i_3})$ and $\mathcal{A}^2 = (a^2_{i_1i_2i_3})$. $\mathcal{A}^1$ is defined by $a^1_{111} = a^1_{222} = 10$, and $a^1_{ijk} = 3$ for all other i, j, k. $\mathcal{A}^2$ is defined by $a^2_{111} = a^2_{222} = 3$, and $a^2_{ijk} = 6$ for all other i, j, k. $b^1 = (8, 8)^T$, $b^2 = (6, 0)^T$. The initial point is $x_0 = (2, 6)^T$, $t_0 = 2$.

Example 3.7 Consider the finite minimax problem (1.1) with tensor structure (1.4), where the two 3-th order 2-dimensional tensors $\mathcal{A}^1 = (a^1_{i_1i_2i_3})$ and $\mathcal{A}^2 = (a^2_{i_1i_2i_3})$. $\mathcal{A}^1$ is defined by $a^1_{111} = a^1_{222} = 1$, and $a^1_{ijk} = 3$ for all other i, j, k. $\mathcal{A}^2$ is defined by $a^2_{111} = a^2_{222} = 10$, and $a^2_{ijk} = 6$ for all other i, j, k. $b^1 = (8, 2)^T$, $b^2 = (6, 8)^T$. The initial point is $x_0 = (5, 2)^T$, $t_0 = 2$.

Example 3.8 Consider the finite minimax problem (1.1) with tensor structure (1.4), where the two 3-th order 3-dimensional tensors $\mathcal{A}^1 = (a^1_{i_1i_2i_3})$ and $\mathcal{A}^2 = (a^2_{i_1i_2i_3})$. $\mathcal{A}^1$ is defined by $a^1_{111} = a^1_{222} = a^1_{333} = 0$, and $a^1_{ijk} = 3$ for all other i, j, k. $\mathcal{A}^2$ is defined by $a^2_{111} = a^2_{222} = a^2_{333} = 6$, and $a^2_{ijk} = 2$ for all other i, j, k. $b^1 = (0, 3, 8)^T$, $b^2 = (6, 4, 6)^T$. The initial point is $x_0 = (4, 0, 2)^T$, $t_0 = 2$.

Table 3.2: The numerical results for Examples 3.6-3.8

Example	m	n	iter	optimal value
By Algorithm 2.2				
3.6	3	2	11	-2.9339
3.7	3	2	7	-4.0784
3.8	3	3	20	-3.4054
By SFR				
3.6	3	2	9	-2.3490
3.7	3	2	16	-4.0753
3.8	3	3	25	-3.2134

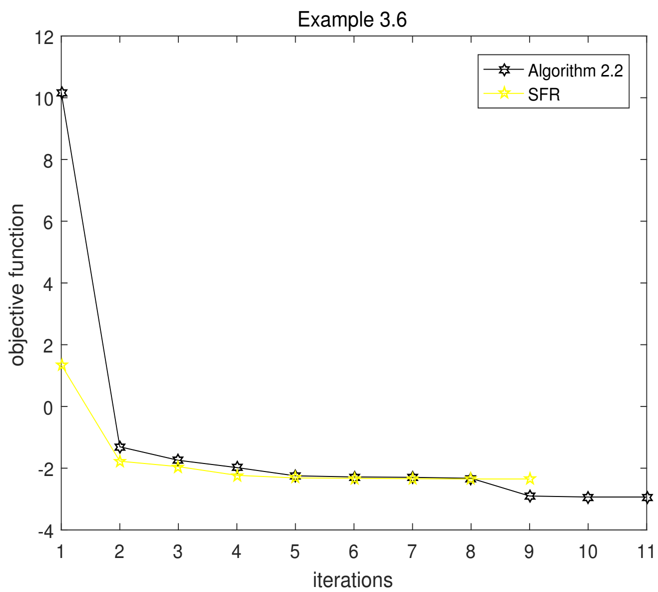

Figure 3.6: The numerical results for Example 3.6

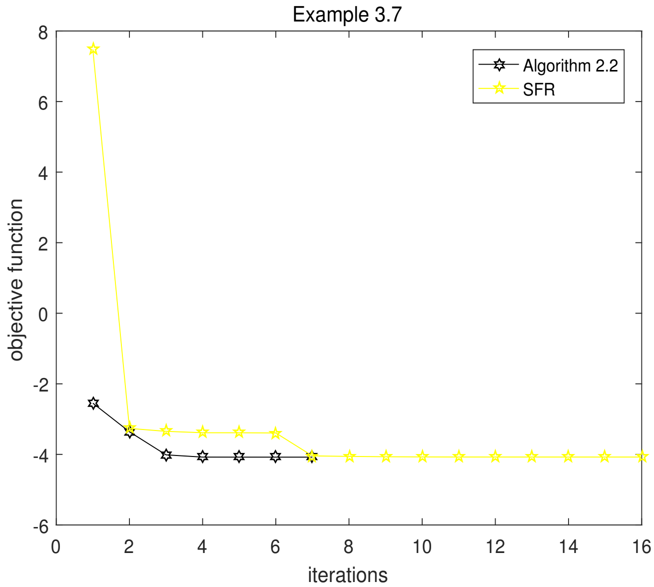

Figure 3.7: The numerical results for Example 3.7

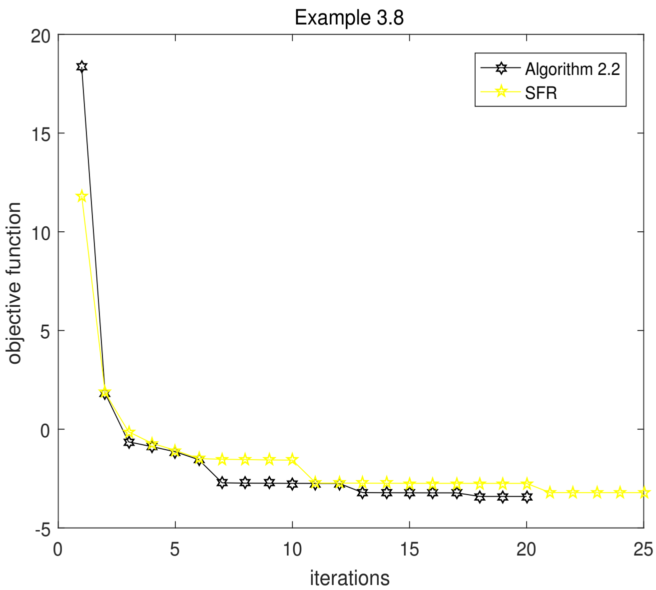

Figure 3.8: The numerical results for Example 3.8

3.3. Constrained optimization problem

In this subsection, we concentrate on solving the constrained optimization problem by the finite minimax problems. The constrained optimization problem

$$\begin{aligned} \min \quad & U(x) \\ \text{s.t.} \quad & g_i(x) \le 0, \quad i = 1, 2, \ldots, m\,. \end{aligned} \tag{3.1}$$

where $g_i(x)$ are continuous differential functions.

From [1], we know that (3.1) is equivalent to the problem of minimizing the unconstrained optimization problem, which can be rewritten as

$$\min V(x,\alpha), \tag{3.2}$$

where

$$V(x,\alpha) = \max\{U(x), U(x) - \alpha_i g_i(x)\}, \qquad i = 1,2,\ldots,m,$$

$\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_m]^T$, $\alpha_i > 0$, $U(x)$ and $g_i(x)$ are all continuous differential functions.

We can transform (3.2) into a smooth unconstrained optimization problem by (1.2), the unconstrained optimization problem is transformed as

$$\min \tilde{V}(x,\alpha,t), \tilde{V}(x,\alpha,t) = t\ln[exp(\frac{U(x)}{t}) + \textstyle\sum_{i=1}^{m} exp(\frac{U(x) - \alpha_i g_i(x)}{t})]. \tag{3.3}$$

Denote the derivative of $\tilde{V}(x,\alpha,t)$ as $g(x_k, t_k, \alpha)$. Then, the unconstrained optimization problem (3.3) can be solved by Algorithm 2.2, the parameter values are taken as $\varepsilon = 10^{-5}$, $\Delta = 0.1$, $\delta_1 = 10^{-4}$, $\zeta = 0.25$, $\sigma = 0.3$, $\delta = 0.9$, $\tau = 0.7$, $\sigma_1 = 0.5$, $\gamma_1 = 0.5$, and $\alpha = \alpha_i$, $i = 1,2,\ldots,m$. When computing Example 3.9 and Example 3.10 by Algorithm 2.2, we choose the values of t as 2 and 1, respectively. We use $\|x_{k+1} - x_k\| \leq \varepsilon$ as the terminate condition.

We give the numerical results in Table 3.3 and Table 3.4, where the x_0, iter and x^* denote the initial point, the number of iterations of the algorithms, and the solution point, respectively. Moreover, Figure 3.9 and Figure 3.10 also give the numerical results of Algorithm 2.2 and SFR[7]. We can see that Algorithm 2.2 has fewer iteration steps than SFR and both of the two algorithms are effectively to solve this kind of optimization problems.

Example 3.9 Rosen-Suzuki.[44]

$$U(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4$$

subject to

$$\begin{aligned} g_1(x) &= -x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4 + 8 \geq 0, \\ g_2(x) &= -x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 + 10 \geq 0, \\ g_3(x) &= -2x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 + 5 \geq 0. \end{aligned}$$

The solution is

$$U = -44, \qquad x = (0,1,2,-1)^T.$$

Example 3.10 Beale.[44]

$$U(x) = 9 - 8x_1 - 6x_2 - 4x_3 + 2x_1^2 + 2x_2^2 + x_3^2 + 2x_1x_2 + 2x_1x_3$$

subject to

$$g_1(x) = x_1 \geq 0, \quad g_2(x) = x_2 \geq 0, \quad g_3(x) = x_3 \geq 0,$$

$$g_4(x) = 3 - x_1 - x_2 - 2x_3 \geq 0.$$

The solution is

$$U = \frac{1}{9}, \qquad x = (\frac{4}{3}, \frac{7}{9}, \frac{4}{9})^T.$$

3.4. Constrained minimax problem with tensor structure

In this subsection, we consider the constrained optimization problem with tensor structure. For any $(\mathcal{A}^1, \mathcal{A}^2, \ldots, \mathcal{A}^m) \in \mathbb{R}^{[2,n]} \times \mathbb{R}^{[3,n]} \times \cdots \times \mathbb{R}^{[m,n]}$ and $q \in \mathbb{R}^n$, we consider the polynomial complementarity

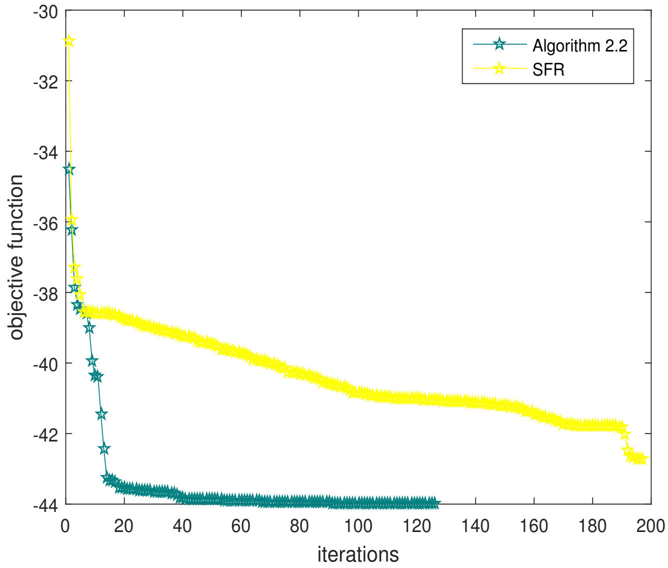

Figure 3.9: The numerical results of Example 3.9 (the initial point is $x_0 = (0.3, 1.4, 1, -0.4)^T$)

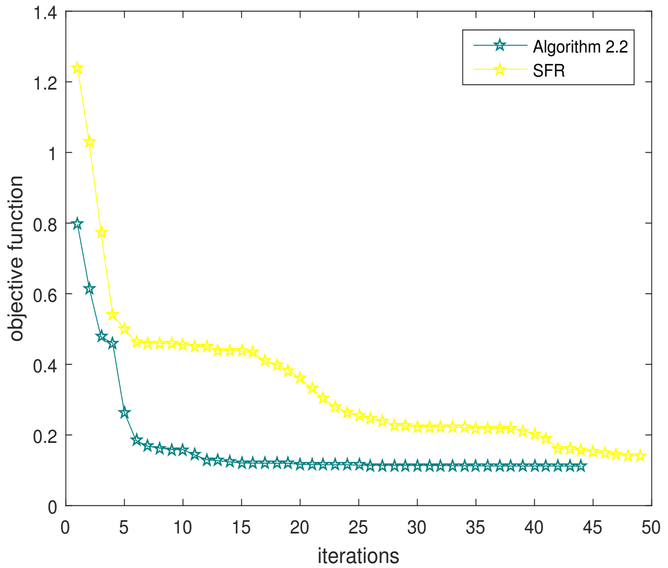

Figure 3.10: The numerical results of Example 3.10 (the initial point is $x_0 = (0.5, 0.5, 0.5)^T$)

Table 3.3: The numerical results for Example 3.9 by Algorithm 2.2 and SFR

x_0	α	iter	optimal value	x^*
By Algorithm 2.2				
$(0,0,0,0)^T$	100	54	-43.9806	$(0.0039, 0.9979, 1.9964, -1.0034)^T$
$(0.3, 1.4, 1, -0.4)^T$	100	126	-43.9605	$(-0.0076, 1.0123, 1.9993, -0.9957)^T$
$(0.28, 1.6, 1.79, -0.23)^T$	200	91	-43.9776	$(-0.0070, 0.9959, 2.0047, -0.9936)^T$
$(0.18, 1.4, 1.89. - 0.25)^T$	200	86	-43.9772	$(-0.0091, 0.9954, 2.0063, -0.9918)^T$
By SFR				
$(0,0,0,0)^T$	100	1093	-43.6508	$(0.0150, 0.9733, 1.9640, -1.0378)^T$
$(0.3, 1.4, 1, -0.4)^T$	100	197	-42.7042	$(0.1007, 0.8701, 2.0412, -0.6514)^T$
$(0.28, 1.6, 1.79, -0.23)^T$	200	1091	-43.7840	$(-0.0410, 0.9743, 2.0137, -0.9840)^T$
$(0.18, 1.4, 1.89. - 0.25)^T$	200	122	-43.8772	$(-0.0428, 1.0743, 1.9992, -0.9803)^T$

Table 3.4: The numerical results for Example 3.10 by Algorithm 2.2 and SFR

x_0	α	iter	optimal value	x^*
By Algorithm 2.2				
$(0.5, 0.5, 0.5)^T$	100	44	0.1118	$(1.3325, 0.7790, 0.4429)^T$
$(0,0,0)^T$	100	35	0.1121	$(1.3410, 0.7753, 0.4398)^T$
$(0.1, 0.7, -0.3)^T$	150	45	0.1118	$(1.3265, 0.7807, 0.4450)^T$
$(1, 0.5, 0.5)^T$	150	42	0.1118	$(1.3278, 0.7797, 0.4449)^T$
By SFR				
$(0.5, 0.5, 0.5)^T$	100	49	0.1415	$(1.4748, 0.6838, 0.4040)^T$
$(0,0,0)^T$	100	42	0.1131	$(1.3491, 0.7816, 0.4313)^T$
$(0.1, 0.7, -0.3)^T$	150	271	0.1176	$(1.3121, 0.8154, 0.4293)^T$
$(1, 0.5, 0.5)^T$	150	172	0.1202	$(1.2885, 0.7542, 0.4721)^T$

problem[30]

$$x \geq 0, \qquad \sum_{k=1}^{m} \mathcal{A}_k x^{k-1} + q \geq 0, \qquad x^T(\sum_{k=1}^{m} \mathcal{A}_k x^{k-1} + q) = 0. \tag{3.4}$$

(3.4) is equivalent to the following constrained optimization problem with tensor structure

$$\begin{aligned} \min \quad & x^T F(x) \\ \text{s.t.} \quad & F_i(x) = (\sum_{k=1}^{m} \mathcal{A}_k x^{k-1} + q)_i \geq 0\,, \\ & x_i \geq 0\,, \end{aligned} \tag{3.5}$$

where $F_i(x)$ are continuous differential functions, $i = 1, 2, \ldots, n$. Then, we can rewrite (3.5) to the minimax problem

$$\min U(x, \alpha, \beta)$$

$$U(x, \alpha, \beta) = \max\{x^T F(x), x^T F(x) - \alpha_i F_i(x), x^T F(x) - \beta_i x_i\},$$

where $\alpha_i > 0$ and $\beta_i > 0$. Then

$$\min \tilde{U}(x, \alpha, \beta, t),$$

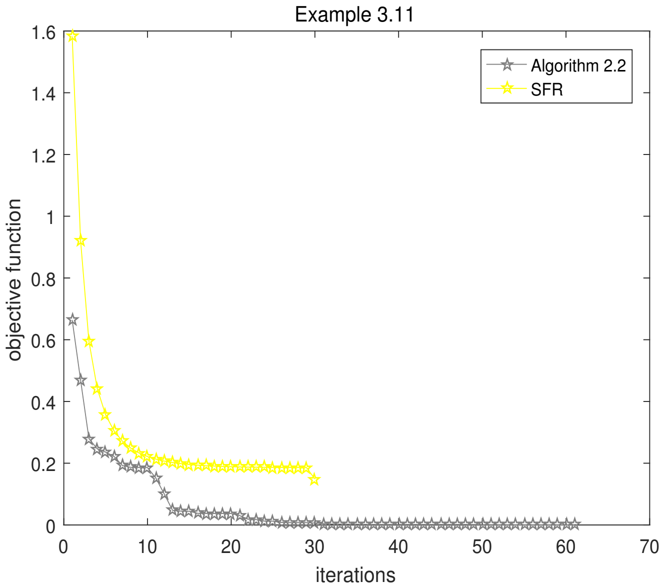

Figure 3.11: The numerical results for Example 3.11

where

$$\tilde{U}(x,\alpha,\beta,t) = t\ln[exp\frac{x^TF(x)}{t} + \sum_{i=1}^{n} exp\frac{x^TF(x)-\alpha_iF_i(x)}{t} + \sum_{i=1}^{n} exp\frac{x^TF(x)-\beta_ix_i}{t}].$$

Denote the derivative of $\tilde{U}(x,\alpha,t)$ as $g(x_k,t_k,\alpha)$. We can solve (24) by Algorithm 2, the parameters are taken as $\varepsilon = 10^{-8}$, $\Delta = 0.1$, $\delta_1 = 10^{-4}$, $\zeta = 0.99$, $\sigma = 0.3$, $\delta = 0.9$, $\tau = 0.7$, $\sigma_1 = 0.2$, $\gamma_1 = 0.5$, $t_0 = 2$, and $\alpha_i = \beta_i$. The initial points are taken randomly. We use $\|x_{k+1} - x_k\| \leq \varepsilon$ as the terminate condition.

Two polynomial complementarity problems are considered, which are taken from [45]. We give the numerical results in Table 3.5, where the iter, $\Delta(x)$ denote the number of iterations of the algorithm, the norm of the vector between the solution point and the optimal point,respectly. We also give the numerical results in Figure 3.11 and Figure 3.12. From Table 3.5, Figure 3.11 and Figure 3.12, we can see that Algorithm 2.2 is valid to solve the polynomial complementarity problems.

Example 3.11 Consider (3.4) with $\mathcal{A} \in \mathbb{R}^{[3,2]}$ and $q \in \mathbb{R}^2$, where $\mathcal{A}$ is defined by $a_{111} = a_{121} = a_{222} = a_{221} = 1$, $a_{122} = a_{211} = -1$ and zero otherwise. Let $q = (5,3)^T$, t_0=2, $x^* = (0,0)^T$.

Example 3.12 Consider (3.4) with $\mathcal{A} \in \mathbb{R}^{[3,2]}$ and $q \in \mathbb{R}^2$ where $\mathcal{A}$ is defined by $a_{111} = a_{122} = a_{222} = 1$, $a_{221} = a_{211} = -1$, $a_{121} = 2$ and zero otherwise. Let $q = (5,3)^T$, t_0=2, $x^* = (0,0)^T$.

Table 3.5: The numerical results for Examples 3.11 and 3.12

Example	α	iter	optimal value	$\Delta(x)$
By Algorithm 2.2				
3.11	500	67	$2.9638e-04$	$4.3912e-05$
3.12	600	63	$2.5482e-04$	$3.7985e-05$
By SFR				
3.11	500	30	0.1487	0.0265
3.12	600	38	0.1293	0.0230

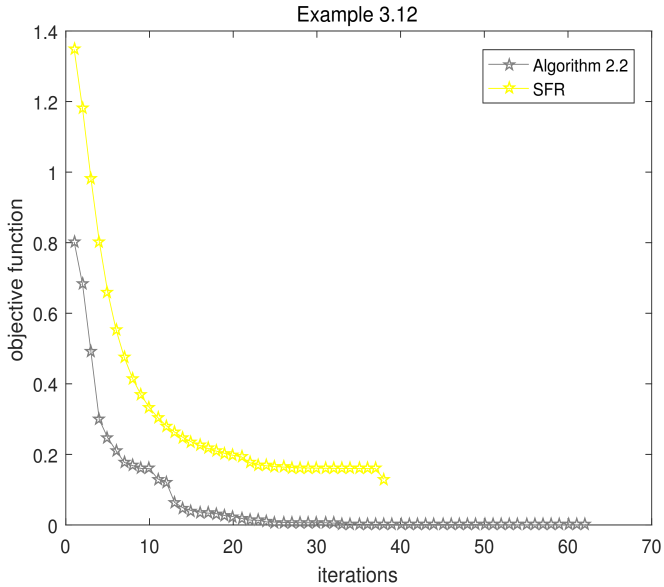

Figure 3.12: The numerical results for Example 3.12

4. Concluding remarks

We proposed a new three-term conjugate gradient method for solving the finite minimax problems. Under the mild assumptions, we prove the convergence of the method. We also consider four kinds of finite minimax problems, which can be solved by the proposed method. We give the numerical tests of our method, and compared our method with other related methods. Obvious advantages of our method for solving minimax problems are shown by the given numerical results. Moreover, the finite minimax problems with tensor structure is proposed for the first time. And the polynomial complementarity problem is also solved by the finite minimax formulation. In the future, new conjugate gradient methods can be considered to solve the minimax problems proposed in this paper.

References

[1] J.W. Bandler, C. Charalambous, Nonlinear programming using minimax techniques, Journal of Optimization Theory and Applications 13(1974) 607-619.

[2] X.S. Li, An entropy-based aggregate method for minimax optimization, Engineering Optimization 18(1992) 277-285.

[3] G. Pillo, L. Grippo, S. Lucidi, A smooth method for the finite minimax problem, Mathematical Programming 60(1993) 187-214.

[4] S.B. Li, D.X. Cao, H.J. Wang and K.Z. Deng, Interval adjustable entropy algorithm for a class of unconstrained discrete minimax problems, Applied Mathematics 19(2004) 37-43.

[5] Y. Xiao, B. Yu, A truncated aggregate smoothing Newton method for minimax problems, Applied Mathematics and Computation 216(2010) 1868-1879.

[6] X.J. Chen, Smoothing methods for nonsmooth, nonconvex minimization, Mathematical Programming 134(2012) 71-99.

[7] D.Y. Pang, S.Q. Du, J.J. Ju, The smoothing Fletcher-Reeves conjugate gradient method for solving finite minimax problems, ScienceAsia 42(2016) 40-45.

[8] E. Polak, On the mathematical foundations of nondifferentiable optimization in engineering design, SIAM Review 29(1987) 21-91.

[9] E. Polak, J.E. Higgins, D.Q. Maynes, A barrier function method for minimax problems, Mathematical Programming 54(1992) 155-176.

[10] R.T. Rockafellar, Computational schemes for large-scale problems in extended linear quadratic programming, Mathematical Programming 48(1990) 447-474.

[11] X.S. Li, An aggregate function method for nonlinear programming, Science in China, Series A 12(1991) 1467-1473.

[12] F. Rahpeymaii, K. Amini, T. Allahviranloo, M.R. Malkhalifeh, A new class of conjugate gradient methods for unconstrained smooth optimization and absolute value equations, Calcolo 56(2019) 1-28.

[13] M.S. Koorapetse, P. Kaelo, Globally convergent three-term conjugate gradient projection methods for solving nonlinear monotone equations, Arabian Journal of Mathematics 7(2018) 289-301.

[14] N. Andrei, A simple three-term conjugate gradient algorithm for unconstrained optimization, Journal of Computational and Applied Mathematics 241(2013) 19-29.

[15] X.L. Dong, H.W. Liu, Y.B. He, New version of the three-term conjugate gradient method based on spectral scaling conjugacy condition that generates descent search direction, Applied Mathematics and Computation 269(2015) 606-617.

[16] J. Sabi¡¯u, A. Shah, M.Y. Waziri, M.K. Dauda, A new hybrid approach for solving large-scale monotone nonlinear equations, Journal of Mathematical and Fundamental Sciences 52(2020) 7-26.

[17] J. Sabi¡¯u, A. Shah, An efficient three-term conjugate gradient-type algorithm for monotone nonlinear equations, RAIRO-Operations Research. doi: 10.1051/ro/2020061, (2020).

[18] M. Y. Waziri, L. Muhammad, J. Sabi¡¯u, A simple three-term conjugate gradient algorithm for solving symmetric systems of nonlinear equations, International Journal of Advances in Applied Sciences 5(2016) 128-136.

[19] J. Sabi¡¯u, A. Shah, M.Y. Waziri, Two optimal Hager-Zhang conjugate gradient methods for solving monotone nonlinear equations, Applied Numerical Mathematics 153(2020) 217-233.

[20] M.Y. Waziri, K. Ahmeda, J. Sabi¡¯u, A family of Hager-Zhang conjugate gradient methods for system of monotone nonlinear equations, Applied Mathematics and Computation 361(2019) 645-660.

[21] J. Sabi¡¯u, M.Y. Waziri, Effective modified hybrid conjugate gradient method for large-scale symmetric nonlinear equations, Application and Applied Mathematics 12(2017) 1036-1056.

[22] M.Y. Waziri, K.A. Hungu, J. Sabi¡¯u, Descent Perry conjugate gradient methods for systems of monotone nonlinear equations, Numerical Algorithms 85(2020) 763-785.

[23] J. Petrovic, V. Rakocevic, N. Kontrec, S. Panic, D. Ilic, Hybridization of accelerated gradient descent method, Numerical Algorithms 79(2018) 769-786.

[24] J. Petrovic, V. Rakocevic, D. Valjarevic, D. Ilic, A note on hybridization process applied on transformed double step size model, Numerical Algorithms 85(2020) 449-465.

[25] J. Petrovic, P.S. Stanimirovic, N. Kontrec, J. Mladenovic, Hybrid modification of accelerated double direction method, Mathematical Problems in Engineering 79(2018) 769-786.

[26] J. Petrovic, Hybridization rule applied on accelerated double step size optimization scheme, Filomat 33(2019) 655-665.

[27] Y.S. Song, L.Q. Qi, Properties of some classes of structured tensors, Journal of Optimization Theory and Applications 165(2015) 854-873.

[28] Z.H. Huang, L.Q. Qi, Formulating an n -person noncooperative game as a tensor complementarity problem, Computational Optimization and Applications 66(2017) 557-576.
[29] Z.Y. Luo, L.Q. Qi, N.H, Xiu The sparsest solutions to Z -tensor complementarity problems, Optimization Letters 11(2017) 471-482.
[30] Z.H. Huang, L.Q. Qi, Tensor complementarity problems¡ªpart I: basic theory, Journal of Optimization Theory and Applications 183(2019) 365-385.
[31] M.L. Che, C.J. Bu, L.Q. Qi, Y.M. Wei, Nonnegative tensors revisited: plane stochastic tensors, Linear and Multilinear Algebra 67(2019) 1364-1391.
[32] L.P. Zhang, L.Q. Qi, Z.Y. Luo, Y. Xu, The dominant eigenvalue of an essentially nonnegative tensor, Numerical Linear Algebra with Applications 20(2013) 929-941.
[33] S.Q. Du, L.P. Zhang, C.Y. Chen, L.Q. Qi, Tensor absolute value equations, Science China(Mathematics) 61(2018) 157-172.
[34] M.R. Hestenes, E.L. Stiefel, Methods of conjugate gradients for solving linear systems, Journal of Research of the National Bureau of Standards 49(1952) 409-436.
[35] R. Fletcher, C. Reeves, Function minimization by conjugate gradients, Computer Journal 7(1964) 149-154.
[36] E. Polak, G. Ribière, Note sur la convergence de méthodes de directions conjugées, Revue Française d¡¯Informatique et Recherche Opératinelle, Série Rouge Tome 316(1969) 35-43.
[37] Y.H. Dai, Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, SIAM Journal on Optimization 10(1999) 177-182.
[38] N. Andrei, Open problems in nonlinear conjugate gradient algorithms for unconstrained optimization, Bulletin of the Malaysian Mathematical Sciences Society 34(2011) 319-330.
[39] C.Y. Wu, A modified PRP conjugate gradient algorithm for unconstrained optimization problems, Mathematica Applicata 1(2011) 25-29.
[40] G.L. Yuan, Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems, Optimization Letters 3(2009) 11-21.
[41] S.Y. Lan,C.Y.Wang ,Convergence properties of the conjugte descent method with Armijo-type line searches,Chinese Journal of Engineering Mathematics 1(2005) 133-138.
[42] G.M. Zhou,Convergence properties of a conjugate gradient method with Armijo-type line searches,Chinese Journal of Engineering Mathematics 3(2008) 405-410.
[43] Q. Ni, L.Q. Qi, A quadratically convergent algorithm for finding the largest eigenvalue of a nonnegative homogeneous polynomial map, Journal of Global Optimization 61(2016) 627-641.
[44] J. Kowalik, M.R. Osborne, D.M. Ryan, A new method for constrained optimization problems, Operations Research 17(1969) 973-983.
[45] L.X. Han, A continuation method for tensor complementarity problems, Journal of Optimization Theory and Appllications 180(2019) 949-963.