Journal of Computational Science &Engineering 58 (2022) 1-8

Available at http://www.asocse.org ISSN 1710-4068

A smoothing conjugate gradient method for a kind of generalized tensor complementarity problem

Jingjing Sun, Yuanyuan Chen*

[1]School of Mathematics and Statistics, Qingdao University, Qingdao 266071, China

Abstract: **The complementarity problem is widely used in engineering, game and traffic equilibrium. In this paper, a kind of generalized tensor complementarity problem is studied. The generalized tensor complementarity problem is transformed into a system of nonsmooth equations by using complementary function, and then it is equivalently transformed into an unconstrained optimization problem by smoothing the nonsmooth equations. Finally, a smoothing conjugate gradient method is presented. The global convergence of the proposed method is proved under general assumptions, and the effectiveness of the proposed method is also verified by numerical examples.**

Keywords: Generalized tensor complementarity problem; Complementarity function; Smoothing conjugate gradient method; Global convergence

Received 03 September 2022, Revised 10 September 2022, Accepted 17. September 2022

* Corresponding Author: Yuanyuan Chen, yychen@qdu.edu.cn

1. Introduction

Complementary problem has an important application background in engineering and technical problems, and is also an important mathematical model in traffic balance problems [1-8]. When the function involved is a linear function, it is called linear complementarity problem (LCP) [3]. When the function involved in complementarity problem is a nonlinear function, it is called nonlinear complementarity problem (NCP) [1,2,4]. Matrices with special structures and various special types of functions also play an important role in the study of LCP and NCP.

In recent years, tensors with various special structures have been extensively studied, such as [9-11]. In 2015, as an application of structural tensors, a class of complementarity problem called tensor complementarity problem (TCP) was proposed for the first time in [12], and studied initially by Song and Qi [13] and Che, Qi and Wei [14]. Since then, TCP has attracted extensive attention [15-18]. It is of great significance to study TCP, including existence and compactness of solution sets, global uniqueness, error bound theory and stability of solutions. With the continuous application of tensors in signal processing, data analysis, image processing and related fields, the problem of TCP has obtained great research results from theory to numerical methods and applications [19-23].

As a kind of tensor complementarity problems, generalized tensor complementarity problem is a meaningful problem to study. On the one hand, the generalized tensor complementarity problem covers LCP and TCP, and the solution method of complementarity problem largely depends on the linearity of the function involved and the special structure of the matrix involved. Due to the functions involved in the generalized tensor complementarity problems are nonlinear and the structure of the tensor involved is far more complex than that of the matrix, the design and analysis of method about the generalized tensor complementarity problems are different from those of the general complementarity problems. On the other hand, while the generalized tensor complementarity problem is a subclass of NCP, due to the generalized tensor complementarity problem involved in the function is defined by a tensor of a special kind of polynomial, thus it is meaningful to design the efficient method for the generalized tensor complementarity problem by using tensor structure involved.

In this paper, we first give a kind of generalized tensor complementarity problem, which is closely connected with absolute value equation problems [24-26] and equivalent to the tensor absolute value equation problem under certain conditions [25]. In particular, when the generalized tensor complementarity problem involved in the tensor for the two-order tensor, the generalized tensor complementarity problem is equivalent to the absolute value equation problem [24].

The structure of this paper is as follows: in Section 2, a kind of generalized tensor complementarity problem is introduced and transformed into an unconstrained optimization problem by a smoothing function. In Section 3, a smoothing conjugate gradient method is proposed to solve the proposed generalized tensor complementarity problem. Finally, in Section 4, numerical results of the generalized tensor complementarity problem are given.

2022@ The American Computational Science Society. All rights reserved.

2. The generalized tensor complementarity problem

With the continuous development of science and technology, the scale of data is getting larger and larger, and the structural relationship between data is becoming more and more complex. The matrix can no longer well meet the needs of data representation in practical problems. Vectors are one-dimensional arrays, matrices are two-dimensional arrays, and tensors can be regarded as the generalization of vectors and moment matrices in high-latitude structures [9-11]. There are two concepts of "order" and "dimension" of tensors. The order determines the basic space structure of tensors, and the dimension determines the scale of tensors. An m-order n-dimensional tensors can be expressed as follows

$$\mathcal{A} = (a_{i_1 i_2 \cdots i_m}) \in R^{[m,n]},$$

where $R^{[m,n]}$ is the set of all mth-order n-dimensional real tensors.

$\mathcal{A}x^{m-1}$ represents an n-dimensional column vector, and each of its components is denoted as

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

Similarly, $\mathcal{A}x^{m-2}$ represents an n-order matrix, and each element of it is denoted as

$$(\mathcal{A}x^{m-2})_{ij} = \sum_{i_3,\cdots,i_m=1}^{n} a_{iji_3\cdots i_m} x_{i_2} \cdots x_{i_m}, \quad \forall i, j \in \{1,2,\cdots,n\}.$$

For any given $\mathcal{A} = (a_{i_1\cdots i_m}) \in R^{[m,n]}$ and $x \in R^n$, there exists a unique symmetric tensor $\hat{\mathcal{A}}$ satisfying $\hat{\mathcal{A}}x^{m-1} = \mathcal{A}x^{m-1}$, and the Jacobian of $\mathcal{A}x^{m-1}$ is $(m-1)\hat{\mathcal{A}}x^{m-2}$.

Generalized tensor complementarity problems are a class NCPs, in which the functions are defined by tensors. In this paper, we mainly consider a kind of generalized tensor complementarity problem, which can be expressed as follows

$$(\mathcal{A}+I)x^{m-1} - q \geq 0, (\mathcal{A}-I)x^{m-1} - q \geq 0, \quad ((\mathcal{A}+I)x^{m-1} - q)^T((\mathcal{A}-I)x^{m-1} - q) = 0. \qquad (2.1)$$

where $\mathcal{A}$ is an m-order n-dimension tensor, I is an m-order n-dimension identity tensor and $q \in R^n$. We denote it as $GTCP(\mathcal{A}, q)$.

Many solution methods for this kind of problems are based on reformulating them as a systems of equations using NCP-funtions [1-7,27-29].

Definition 2.1[1] A function $\varphi : R^2 \to R$ is called an NCP-function if

$$\varphi(a,b) = 0 \Leftrightarrow a \geq 0, b \geq 0, ab = 0.$$

The Fischer-Burmeister (FB) function, which is a special case of NCP-function, is defined as follows

$$\varphi_{FB}(a,b) = \sqrt{a^2 + b^2} - a - b, \quad a,b \in R.$$

We know that FB function is convex and differentiable almost everywhere. And $\varphi^2(a,b)$ is continuous and differentiable everywhere. So we use FB function to solve (2.1).

Denote $\phi(x) = \begin{pmatrix} \varphi_{FB}(F_1(x), G_1(x)) \\ \vdots \\ \varphi_{FB}(F_n(x), G_n(x)) \end{pmatrix}$,

2022@ The American Computational Science Society. All rights reserved.

where $F_i(x) = ((\mathcal{A}+I)x^{m-1} - q)_i$, $G_i(x) = ((\mathcal{A}-I)x^{m-1} - q)_i$, $i = 1,2,\cdots,n$. In addition, $x \in R^n$ is the solution of (2.1) if and only if $\phi(x) = 0$. However, FB function is not differentiable at the origin [1-7, 28]. Here, we use the smoothing FB function $\varphi_\mu(a,b) = \sqrt{a^2 + b^2 + \mu} - a - b,\ a,b \in R$,

where $\mu > 0$. Then solving (2.1) is equivalent to solving the following problem

$$\phi_\mu(x) = \begin{bmatrix} \varphi_\mu(F_1(x), G_1(x)) \\ \vdots \\ \varphi_\mu(F_n(x), G_n(x)) \end{bmatrix} = 0, \quad (2.2)$$

Where $\varphi_\mu(F_i(x), G_i(x)) = \sqrt{((\mathcal{A}+I)x^{m-1} - q)_i^2 + ((\mathcal{A}-I)x^{m-1} - q)_i^2 + \mu} - ((\mathcal{A}+I)x^{m-1} - q)_i - ((\mathcal{A}-I)x^{m-1} - q)_i$, $i = 1,2,\cdots n$.

Define the value function of (2.2) as $\Phi_\mu(x) = \frac{1}{2}\left\|\phi_\mu(x)\right\|^2$, then (2.1) is finally transformed into the following unconstrained problem $\min\ \Phi_\mu(x)$, (2.3) where $x \in R^n$, $\mu > 0$.

3. A smoothing conjugate gradient method

The gradient of $\Phi_\mu(x)$ is

$$\nabla\Phi_\mu(x) = J\phi_\mu(x)^T \phi_\mu(x),$$

where $J\phi_\mu(x)$ is the Jacobian matrix of $\phi_\mu(x)$, which is calculated by the following

$$J\phi_\mu(x) = (m-1)D_a(x)(\mathcal{A}+I)x^{m-1} + (m-1)D_b(x)(\mathcal{A}-I)x^{m-1},$$

where $D_a(x)$ and $D_b(x)$ are diagonal matrix, whose ith diagonal element is

$$a_i(x) = \frac{((\mathcal{A}+I)x^{m-1} - q)_i}{\sqrt{((\mathcal{A}+I)x^{m-1} - q)_i^2 + ((\mathcal{A}-I)x^{m-1} - q)_i^2 + \mu}} - 1$$

and

$$b_i(x) = \frac{((\mathcal{A}-I)x^{m-1} - q)_i}{\sqrt{((\mathcal{A}+I)x^{m-1} - q)_i^2 + ((\mathcal{A}-I)x^{m-1} - q)_i^2 + \mu}} - 1,$$

respectively.

Let x_0 be the initial point for solving (2.3). The general form of conjugate gradient method is

$$x_{k+1} = x_k + \alpha_k d_k,\ k = 0,1,\cdots,$$

where step size α_k is found by linear search. Based on [30], we propose a smoothing conjugate gradient method whose search direction is

$$d_k = \begin{cases} -\Phi_\mu(x_k), & k = 0, \\ -\Phi_\mu(x_k) + \beta_k d_{k-1}, & k \geq 1, \end{cases} \quad (3.1)$$

where β_k is defined as

2022@ The American Computational Science Society. All rights reserved.

$$\beta_k = \frac{\Phi_\mu(x_k)^T y_{k-1}}{(d_{k-1})^T z_{k-1}} \quad \xi \frac{\|y_{k-1}\|^2}{((d_{k-1})^T z_{k-1})^2} \Phi_\mu(x_k)^T d_{k-1}, \quad (3.2)$$

and

$$z_{k-1} = y_{k-1} + t_{k-1} s_{k-1}, \; t_{k-1} = \varepsilon_0 + \max\left\{-\frac{s_{k-1}^T y_{k-1}}{s_{k-1}^T s_{k-1}}, 0\right\}, \quad (3.3)$$

where $\varepsilon_0 \in [0, c], \xi > \frac{1}{4}, s_{k-1} = x_k - x_{k-1}, y_{k-1} = \nabla\Phi_\mu(x_k) - \nabla\Phi_\mu(x_{k-1})$. Similar to the methods in [30]. We give the smoothing conjugate gradient method for (2.1).

Algorithm 3.1 Smoothing conjugate gradient method

Step 1 Given the initial point $x_0 \in R^n$, $\mu_0 > 0$, parameters $0 < \rho < \sigma < 1$, $\gamma \in (0,1)$, $\varepsilon > 0$, $\kappa > 0$. Set $k = 0$;

Step 2 If $\left\|\Phi_{\mu_k}(x_k)\right\| \le \varepsilon$, stop. Otherwise, go to Step 3;

Step 3 According to (3.1), we get d_k.

Step 4 Determine α_k satisfying line search

$$\Phi_{\mu_k}(x_k) - \Phi_{\mu_k}(x_k + \alpha_k d_k) \ge \rho\alpha_k^2 \|d_k\|^2,$$

$$\nabla\Phi_{\mu_k}(x_k + \alpha_k d_k)^T d_k \ge -2\sigma\alpha_k \|d_k\|^2.$$

Step 5 Set $x_{k+1} = x_k + \alpha_k d_k$. If $\left\|\Phi_{\mu_k}(x_k)\right\| \geqslant \kappa\mu_k$, then set $\mu_{k+1} = \mu_k$; Otherwise choose $\mu_{k+1} = \gamma\mu_k$.

Step 6 Set $k = k+1$, go to Step 1.

Now, we give the convergence analysis of Algorithm 3.1. Firstly, we give a basic assumption of convergence analysis of gradient methods.

Assumption 3.1 Let the level set $\Omega = \{x \in R^n \mid \Phi_\mu(x) \le \Phi_\mu(x_0), \mu > 0\}$ is bounded.

Lemma 3.1[30] Let

$$\begin{cases} B(s,z) = (m-1)[D_a(s)(\hat{A} + \hat{I})x^{m-2}\phi_\mu(s) \\ \qquad - D_a(z)(\hat{A} + \hat{I})x^{m-2}\phi_\mu(z) \\ C(s,z) = (m-1)[D_a(s)(\hat{A} - \hat{I})x^{m-2}\phi_\mu(s) \\ \qquad - D_a(z)(\hat{A} - \hat{I})x^{m-2}\phi_\mu(z) \end{cases}.$$

If there are $\lambda_1 > 0, \lambda_2 > 0$, such that

a. $\|B(s,z)\| \le \lambda_1 \|s - z\|$,

b. $\|C(s,z)\| \le \lambda_2 \|s - z\|$.

Then for any $s, z \in U$, we have

$$\|\nabla\Phi_\mu(s) - \nabla\Phi_\mu(z)\| \le L\|s - z\|, \quad (3.4)$$

where $L = \lambda_1 + \lambda_2$, and U is the neighbourhood of Ω.

Lemma 3.2[30] Let $\{x_k\}$ and $\{d_k\}$ be generated by Algorithm 3.1. If $(d_k)^T z_k \ne 0$, then

$$\nabla\Phi_\mu(x_k)^T d_k \le -(1 - \frac{1}{4\xi})\|\nabla\Phi_\mu(x_k)\|^2,$$

where $\xi > \frac{1}{4}$.

2022@ The American Computational Science Society. All rights reserved.

Lemma 3.3[30] Under Assumption 3.1, let $\{x_k\}$ and $\{d_k\}$ be generated by Algorithm 3.1, then

$$\sum_{k\geq 0}\frac{\left\|\nabla\Phi_{\mu}(x_k)\right\|^4}{\left\|d_k\right\|^2}<+\infty. \quad (3.5)$$

Theorem 3.1 Let $\{x_k\}$ be generated by Algorithm 3.1. If $\varepsilon_0>0$ in (3.3), then

$$\lim_{k\to\infty}\left\|\nabla\Phi_{\mu_k}(x_k)\right\|=0.$$

Proof Define $K=\{k\mid \mu_{k+1}=\gamma\mu_k\}$. If K is a finite set, then there exists an integer $\hat{k}$, for all $k>\hat{k}$, such that

$$\left\|\nabla\Phi_{\mu_k}(x_k)\right\|\geq\kappa\mu_k. \quad (3.6)$$

And for all $k\geq\hat{k}$, let $\mu_k=\overline{\mu}$. Since $\left\|\nabla\Phi_{\overline{\mu}}(x_k)\right\|$ is a smoothing function, (2.3) can be transformed into $\min\Phi_{\overline{\mu}}(x)$. By (3.3), we get

$$\begin{aligned}
&(d_{k-1})^T z_{k-1}\\
&=(d_{k-1})^T y_{k-1}+\varepsilon_0(d_{k-1})^T s_{k-1}+\max\left\{0,-\frac{s_{k-1}^T y_{k-1}}{s_{k-1}^T s_{k-1}}\right\}(d_{k-1})^T s_{k-1}\\
&\geq(d_{k-1})^T y_{k-1}+\varepsilon_0(d_{k-1})^T s_{k-1}-\frac{s_{k-1}^T y_{k-1}}{s_{k-1}^T s_{k-1}}(d_{k-1})^T s_{k-1}\\
&=\varepsilon_0(d_{k-1})^T s_{k-1}.
\end{aligned}$$

Combined with (3.2) and (3.4), we get

$$\begin{aligned}
|\beta_k|&=\left|\frac{\nabla\Phi_{\overline{\mu}}(x_k)^T y_{k-1}}{(d_{k-1})^T z_{k-1}}-\xi\frac{\left\|y_{k-1}\right\|^2}{\left((d_{k-1})^T z_{k-1}\right)^2}\nabla\Phi_{\overline{\mu}}(x_k)^T d_{k-1}\right|\\
&\leq\frac{L\alpha_{k-1}\left\|d_{k-1}\right\|\left\|\nabla\Phi_{\overline{\mu}}(x_k)\right\|}{\varepsilon_0\alpha_{k-1}\left\|d_{k-1}\right\|^2}+\xi\frac{L^2\alpha_{k-1}^2\left\|d_{k-1}\right\|^2}{\varepsilon_0^2\alpha_{k-1}^2\left\|d_{k-1}\right\|^4}\left\|\nabla\Phi_{\overline{\mu}}(x_k)\right\|\left\|d_{k-1}\right\|\\
&\leq\left(\frac{L}{\varepsilon_0}+\xi\frac{L^2}{\varepsilon_0^2}\right)\frac{\left\|\nabla\Phi_{\overline{\mu}}(x_k)\right\|}{\left\|d_{k-1}\right\|}.
\end{aligned}$$

Thus by (3.1), we have

$$\begin{aligned}
\left\|d_k\right\|&\leq\left\|\nabla\Phi_{\overline{\mu}}(x_k)\right\|+|\beta_k|\left\|d_{k-1}\right\|\\
&\leq\left\|\nabla\Phi_{\overline{\mu}}(x_k)\right\|+\left(\frac{L}{\varepsilon_0}+\xi\frac{L^2}{\varepsilon_0^2}\right)\left\|\nabla\Phi_{\overline{\mu}}(x_k)\right\|\\
&=\left(1+\frac{L}{\varepsilon_0}+\xi\frac{L^2}{\varepsilon_0^2}\right)\left\|\nabla\Phi_{\overline{\mu}}(x_k)\right\|.
\end{aligned}$$

Set $\sqrt{A}=\left(1+\frac{L}{\varepsilon_0}+\xi\frac{L^2}{\varepsilon_0^2}\right)\left\|\nabla\Phi_{\overline{\mu}}(x_k)\right\|$, then $\left\|d_k\right\|^2\leq A\left\|\nabla\Phi_{\overline{\mu}}(x_k)\right\|^2$. Substitute it into (3.5), we get

$$\sum_{k=0}^{\infty}\left\|\nabla\Phi_{\overline{\mu}}(x_k)\right\|^2\leq A\sum_{k=0}^{\infty}\frac{\left\|\nabla\Phi_{\overline{\mu}}(x_k)\right\|^4}{\left\|d_k\right\|^2}<\infty,$$

which contradicts with (3.6). So K is an infinite set, that is $\lim_{k\to\infty}\mu_k=0$. By Algorithm 3.1, we know that $\left\|\nabla\Phi_{\mu_k}(x_k)\right\|\leq\kappa\mu_k$, that is

$$\lim_{k\to\infty}\left\|\nabla\Phi_{\mu_k}(x_k)\right\|\leq\kappa\lim_{k\to\infty}\mu_k=0.$$

The proof is completed.

4. Numerical examples

In this section, we use Algorithms 3.1 to solve $GTCP(\mathcal{A},q)$. The parameter used in Algorithm 3.1 are chosen as

2022@ The American Computational Science Society. All rights reserved.

$\xi = 2$, $\varepsilon = 10^{-6}$, $\varepsilon_0 = 0.1$, $\rho = 0.001$, $\sigma = 0.01$, $\mu_0 = 0.3$, $\gamma = 0.3$, $\kappa = 0.5$, $c = 1$. We implement all numerical test in MATLAB R2019b and run the codes on a PC with 1.80GHz CPU and 16.00GB of RAM. The tensors used in the following examples have the similar structure with that of tensors used in [31,32].

Example 4.1 Consider the problem (2.1) with tensor $\mathcal{A} \in R^{[4,2]}$ and $q = (1,1)^T$, where $\mathcal{A}$ with entries $a_{1111} = 2$, $a_{1112} = -2$, $a_{1122} = 1$, $a_{2222} = 2$ and other $a_{i_1i_2i_3i_4} = 0$.

We randomly generate the initial value $x_0 \in R^2$. The numerical results are given in Tab.4.1, where "sol", "k" and "val" denote the solution, iterations and the optimal value of the objective function respectively.

Tab.4.1 Nuemrical results of Example 4.1

x_0	sol	k	val
$(0.4709,0.2305)^T$	$(1.7549,1.0000)^T$	28	1.7942e-14
$(0.1270,0.9134)^T$	$(1.7549,1.0000)^T$	18	2.4009e-14
$(0.8443,0.1948)^T$	$(1.7549,1.0000)^T$	30	5.2199e-14
$(0.6225,0.5870)^T$	$(1.7549,1.0001)^T$	18	5.7676e-15
$(0.8759,0.5502)^T$	$(1.7549,1.0000)^T$	22	3.9455e-14

Tab.4.2 Nuemrical results of Example 4.2(n=10)

x_0	sol	k	val
$(0.4389,0.1111,\ldots,0.1174)^T$	$(1.0000,1.0000,\ldots,1.0000)^T$	35	4.2446e-15
$(0.6791,0.3955,\ldots,0.2619)^T$	$(1.0000,1.0000,\ldots,1.0000)^T$	37	1.6200e-14
$(0.3354,0.6797,\ldots,0.9037)^T$	$(1.0000,1.0000,\ldots,1.0000)^T$	36	8.0358e-15
$(0.6177,0.8594,\ldots,0.1679)^T$	$(1.0000,1.0000,\ldots,1.0000)^T$	34	1.4115e-14
$(0.8181,0.8175,\ldots,0.4538)^T$	$(1.0000,1.0000,\ldots,1.0000)^T$	39	1.5982e-14

Tab.4.3 Nuemrical results of Example 4.2(n=100)

x_0	sol	k	val
$(0.4573,0.6181,\ldots,0.3554)^T$	$(1.0000,1.0000,\ldots,1.0000)^T$	40	1.0183e-14
$(0.1048,0.8584,\ldots,0.8045)^T$	$(1.0000,1.0000,\ldots,1.0000)^T$	41	2.5813e-14
$(0.4229,0.0942,\ldots,0.6135)^T$	$(1.0000,1.0000,\ldots,1.0000)$T	45	3.2476e-14
$(0.6622,1.2943,\ldots,1.2948)^T$	$(1.0000,1.0000,\ldots,1.0000)$T	47	1.5260e-14
$(1.0822,1.0407,\ldots,0.5550)^T$	$(1.0000,1.0000,\ldots,1.0000)$T	42	3.3234e-14

Example 4.2 Consider the problem (2.1) with tensor $\mathcal{A} \in R^{[4,2]}$ and $q = (1,\cdots,1)^T$, where $\mathcal{A}$ with entries

2022@ The American Computational Science Society. All rights reserved.

$$\begin{cases} a_{iiii} = a_{i+1,i+1,i+1,i+1} = 2, & i = 1,3,\cdots,n-1, \\ a_{i,i+1,i+1,i+1} = a_{i+1,i,i,i} = -1, & i = 1,3,\cdots,n-1, \\ a_{i,i,i+1,i+1} = a_{i,i+1,i,i+1} = a_{i,i+1,i+1,i} = \dfrac{1}{3}, & i = 1,3,\cdots,n-1, \\ a_{i+1,i+1,i,i} = a_{i+1,i,i+1,i} = a_{i+1,i,i,i+1} = \dfrac{1}{3}, & i = 1,3,\cdots,n-1, \end{cases}$$

and other $a_{i_1i_2i_3i_4} = 0$.

We randomly generate the initial value $x_0 \in R^n$. The numerical results for $n = 10$ and $n = 100$ are given in Tab.4.2 and Tab.4.3, where "sol" , "k" and "val" denote the solution, iterations and the optimal value of the objective function respectively.

From the above numerical results, we can see the proposed method can effectively solve the given generalized tensor complementarity problems.

References

[1] Harker P, Pang J. Finite-dimensional variational inequality and nonlinear complementarity problems: a survey of theory, algorithms and applications [J]. Mathematical Programming, 1990, 48: 161 – 220.

[2] Facchinei F, Pang J. Finite-dimensional variational inequalities and complementarity problems[M]. New York: Springer, 2003.

[3] Cottle R, Pang J, Stone R. The linear complementarity problem [M]. Boston: Academic Press, 1992.

[4] Han J, Xiu N, Qi H. Nonlinear complementarity theory and algorithms [M]. Shanghai: Shanghai Science and Technology Press, 2006 (in Chinese).

[5] Pang J, Sun D, Sun J. Semismooth homeomorphisms and strong stability of semidefinite and Lorentz complementarity problems [J]. Mathematics of Operations Research, 2003, 28(1): 39-63.

[6] Lu N, Huang Z, Han J. Properties of a class of nonlinear transformations over Euclidean Jordan algebras with applications to complementarity problems [J]. Numerical Functional Analysis and Optimization, 2009, 30(7-8): 799-821.

[7] Kong L, Tuncel L, Xiu N. Monotonicity of Lowner operators and its applications to symmetric cone complementarity problems [J]. Mathematical Programming, 2012, 133(1-2): 327-336.

[8] Lv S, Du S. A new smoothing spectral conjugate gradient method for solving tensor complementarity problems [J]. Journal of Industrial and Management Optimization. 2021, doi:10.3934/ jimo.2021150.

[9] Yang Y, Yang Q. Optimization of eigenvalues of higher order tensors and related polynomials [M]. Beijing: Science Press, 2015 (in Chinese).

[10] Wei Y, Ding W. Theory and computation of tensors: multi-dimensional arrays [M]. London: Academic Press, 2016.

[11] Qi L, Luo Z. Tensor analysis: spectral theory and special tensors [J]. SIAM, Philadelphia, 2017.

[12] Song Y, Qi L. Properties of some classes of structured tensors [J]. Journal of Optimization Theory and Applications, 2015, 165: 854 – 873.

[13] Song Y, Qi L. Properties of tensor complementarity problem and some classes of structured tensors [J]. Annals of Applied Mathematics, 2017, 33: 308 – 323.

[14] Che M, Qi L, Wei Y. Positive-definite tensors to nonlinear complementarity problems [J]. Journal of Optimization Theory and Applications, 2016, 168(2): 475 – 487.

[15] Zheng Y, Wu W. On a class of semi-positive tensor in tensor complementarity problem [J]. Journal of Optimization Theory and Applications, 2018, 177(1): 127-136.

[16] Ding W, Luo Z, Qi L. P-tensors, P0-tensors, and tensor complementarity problem [J]. Linear Algebra and its Applications, 2018, 555: 336-354.

[17] Balaji R, Palpandi K. Positive definite and Gram tensor complementarity problems [J]. Optimization Letters, 2018, 12(3): 639-648.

[18] Du S, Ding W, Wei Y. Acceptable solutions and backward errors for tensor complementarity problems [J]. Journal of Optimization Theory and Applications, 2021, 188(1): 260-276.

[19] Huang Z, Qi L. Formulating an n-person noncooperative game as a tensor complementarity problem [J]. Computational Optimization and Applications, 2017, 66: 557-576.

[20] Huang Z, Qi L. Tensor complementarity problems-part I: basic theory [J]. Journal of Optimization Theory and Applications, 2019, 183(1): 1-23.

2022@ The American Computational Science Society. All rights reserved.

[21] Qi L, Huang Z. Tensor complementarity problems-part II: solution methods [J]. Journal of Optimization Theory and Applications, 2019, 183(2): 365-385.
[22] Huang Z, Qi L. Tensor complementarity problems-part III: applications [J]. Journal of Optimization Theory and Applications, 2019, 183(3): 771-791.
[23] Chen C, Zhang L. Finding Nash equilibrium for a class of multi-person noncooperative games via solving tensor complementarity problem [J]. Applied Numerical Mathematics, 2019, 145: 458-468.
[24] Mangasarian O, Mayer R. Absolute value equation [J]. Linear Algebra and its Applications, 2006, 419: 359-367.
[25] Du S, Zhang L, Chen C, Qi L. Tensor absolute value equations [J]. Science China-Mathematics, 2018, 61: 1695-1710.
[26] Du S, Sun J, Niu S, Zhang L. Stochastic absolute value equations [J]. Journal of Applied Mathematics and Computing, 2022, https://doi.org/10.1007/s12190-022-01777-0.
[27] Fischer A. A special Newton-type optimization method [J]. Optimization, 1992, 24: 269-284.
[28] Kanzow C. Some noninterior continuation methods for linear complementarityproblems[J]. SIAM Journal on Matrix Analysis and Applications, 1996, 17(4): 851-868.
[29] Xiu N, Gao Z. New development of complementarity problem algorithms [J]. Advances in Mathematics (China), 1999, 28: 193-210 (in Chinese).
[30] Lv S. Smoothing gradient algorithms for tensor complementarity problems [D]. Qingdao University, 2022 (in Chinese).
[31] Bai X, Huang Z, Wang Y. Global uniqueness and solvability for tensor complementarity problems [J]. Journal of Optimization Theory and Applications, 2016, 170, 72 – 84.
[32] Song S, Li L. A class of smooth modulus-based matrix iteration methods for solving tensor complementarity problem. Mathematica Numerica Sinica, 2022, 44(2): 178-186 (in Chinese).

2022@ The American Computational Science Society. All rights reserved.