A NONMONOTONE SPECTRAL PROJECTED GRADIENT METHOD FOR TENSOR EIGENVALUE COMPLEMENTARITY PROBLEMS

WANBIN TONG, HONGJIN HE, CHEN LING* AND LIQUN QI
Department of Mathematics, School of Science
Hangzhou Dianzi University, Hangzhou, 310018, China.

Abstract. The tensor eigenvalue complementarity problem (TEiCP) is a higher order extension model of the classical matrix eigenvalue complementarity problem (EiCP), which has been studied extensively in the literature from theoretical perspective to algorithmic design. Due to the high nonlinearity resulted by tensors, the corresponding TEiCPs are often not easy to be solved directly by the algorithms tailored for EiCPs. In this paper, we introduce a nonmonotone spectral projected gradient (NSPG) method equipped with a positive Barzilai-Borwein step size to find a solution of TEiCPs. A series of numerical experiments show that the proposed NSPG method can greatly improve the efficiency of solving TEiCPs in terms of taking much less computing time for higher dimensional cases. Moreover, computational results show that our NSPG method is less sensitive to choices of starting points than some state-of-the-art algorithms.

1. Introduction. A real mth order n-dimensional square tensor A can be represented as A = (ai1i2⋯im) with aii1i2⋯im ∈ ℝ for all ij ∈ [n] := \{1, 2, ⋯, n\} and \( i \in [m] := \{1, 2, ⋯, m\} \). Throughout this paper, we denote by \( \mathbb{T}^{[m,n]} \) the set of all real mth order n-dimensional tensors. A is called symmetric if the value of \( a_{i_1i_2⋯i_m} \) is invariant under any permutation of its indices \( \{i_1, i_2, ⋯, i_m\} \). In what follows, we denote by \( \mathbb{S}^{[m,n]} \) the set of all real symmetric mth order n-dimensional tensors. Given \( A \in \mathbb{T}^{[m,n]} \) and \( x = (x_1, ⋯, x_n)^\top \in \mathbb{R}^n \), \( Ax^{m-1} \) is defined as an n-dimensional vector, whose \( i \)-th component is given by

\[
(Ax^{m-1})_i = \sum_{i_2, ⋯, i_m = 1}^{n} a_{i_1i_2⋯i_m}x_{i_2}⋯x_{i_m}, \quad \forall i \in [n],
\]

and \( Ax^n = x^\top (Ax^{m-1}) \) is a homogeneous polynomial function with respect to \( x \), where \( u^\top v = (u, v) \) for \( u, v \in \mathbb{R}^n \) represents the standard inner product of vectors. Then, as defined in [17], we call that tensor \( A \in \mathbb{T}^{[m,n]} \) is copositive if \( Ax^n \geq 0 \) for all \( x \in \mathbb{R}_n^+ \) and strictly copositive if \( Ax^n > 0 \) for all \( x \in \mathbb{R}_n^+ \) and \( x \neq 0 \).

2020 Mathematics Subject Classification. Primary: 15A68, 90C30; Secondary: 90C33.

Keywords and phrases. Tensor eigenvalue complementarity problem, Projected gradient method, Barzilai-Borwein step size, Polynomial optimization.

H. He and C. Ling were supported in part by National Natural Science Foundation of China (Nos. 11771113 and 11971138) and Natural Science Foundation of Zhejiang Province (Nos. LY20A010018, LY19A010019, and LD19A010002).

* Corresponding author: Chen Ling.
With the importance of tensors in massive data analysis, Qi [16] and Lim [13] independently introduced the concepts of eigenvalues and eigenvectors for higher order tensors. Later, Chang et al. in [5] gave a unified definition on tensor eigenvalues. Specifically, for $\mathcal{A}, \mathcal{B} \in \mathbb{T}^{[m,n]}$, we call $(\lambda, x) \in \mathbb{C} \times \mathbb{C}^n \setminus \{0\}$ a generalized eigenvalue-eigenvector pair of $(\mathcal{A}, \mathcal{B})$ if

$$\mathcal{A}x^{m-1} = \lambda \mathcal{B}x^{m-1}. \tag{2}$$

In particular, when tensor $\mathcal{B} = (b_{i_1i_2\ldots i_m})$ reduces to the unit tensor $\mathcal{I} \in \mathbb{T}^{[m,n]}$, i.e.,

$$b_{i_1i_2\ldots i_m} = (\mathcal{I})_{i_1i_2\ldots i_m} = \begin{cases} 1, & \text{if } i_1 = i_2 = \cdots = i_m, \\ 0, & \text{otherwise,} \end{cases}$$

the real scalar $\lambda$ in (2) is called an H-eigenvalue and $x$ is the associated H-eigenvector of tensor $\mathcal{A}$ (see [13, 16]). In the past decade, it has been well documented that tensor eigenvalues have many applications in exponential data fitting, diffusion kurtosis tensor imaging, quantum physics, spectral hypergraph theory, etc. Here, we only refer the reader to the most recent monograph [18] for theory and applications of tensor eigenvalues.

In this paper, we consider the tensor eigenvalue complementarity problem (TEiCP), which refers to the task of finding a scalar $\lambda \in \mathbb{R}$ and $x \in \mathbb{R}^n \setminus \{0\}$ such that

$$x \geq 0, \quad \lambda \mathcal{B}x^{m-1} - \mathcal{A}x^{m-1} \geq 0, \quad \langle x, \lambda \mathcal{B}x^{m-1} - \mathcal{A}x^{m-1} \rangle = 0, \tag{3}$$

where $\mathcal{A}, \mathcal{B} \in \mathbb{T}^{[m,n]}$. In what follows, we call the scalar $\lambda$ and the nonzero $x$ a Pareto-eigenvalue and the associated Pareto-eigenvector of $(\mathcal{A}, \mathcal{B})$, respectively. The TEiCP, which covers the well-known eigenvalue complementarity problems for matrices (i.e., by setting $m = 2$ in (3)) as special cases, is originally proposed by Ling et al. [14] and is closely related to a class of differential inclusions with nonconvex processes (see [14]) and a kind of nonlinear differential dynamical system (see [7]). As stated in [14], TEiCP (3) has at least one solution if the condition $\mathcal{B}x^m \neq 0$ holds for all $x \in \mathbb{R}^n \setminus \{0\}$. When a TEiCP has solutions, ones may be immediately concerned with how to find a solution efficiently for such a problem. Accordingly, Ling et al. [14] first introduced a scaling-and-projection algorithm (SPA), which has simple iterative schemes, however, with a slow convergence behavior. To improve the efficiency of solving TEiCPs, Chen and Qi [8] reformulated TEiCP (3) as a system of nonlinear equations and proposed a semismooth Newton method for the resulted nonlinear equations. Due to the high nonlinearity caused by tensor-vector multiplication (see (1)), reformulating TEiCP (3) as a nonlinear optimization problem usually yields a nonconvex model. Therefore, Chen et al. [7] introduced a shifted projection power (SPP) method for TEiCP, which employs an adaptive shift to force the objective function to be (locally) convex. Actually, the nonlinear program reformulation of TEiCP opens a window for solving TEiCP efficiently. In this direction, Fan et al. [9] exploited the polynomial nature of TEiCP and reformulated such a problem as a constrained polynomial optimization, which can be solved by Lasserre’s hierarchy of semidefinite relaxation. Moreover, they proved that their method has finite convergence for generic tensors. Most recently, Yu et al. [19] introduced some spectral projected gradient (SPG) methods for generalized TEiCPs. A series of computational results in [19] show that their SPG methods work well and perform better than the existing first-order algorithms (e.g., SPA and SPP) for TEiCPs.
Even though several methods have been developed for TEiCPs, to our experiment observations, it is necessary to design algorithms for higher-order and -dimension TEiCPs. In recent years, first-order methods received much considerable attention for large-scale structured optimization problems. As shown in [14, 19], finding a solution to a symmetric TEiCP amounts to maximizing a fractional function over a simplex set (see (4)), which is also a structured optimization model to be solved by first-order methods. Actually, it is not difficult to observe that the success of the SPG methods in [19] comes from twofold: (i) SPG methods are equipped with the well-known BB step size originally proposed by Barzilai and Borwein [2] in 1988. In the literature, we have witnessed that such a step size has become one of the most popular choices for scientific computing. To guarantee the convergence of BB-type methods, ones usually restricts BB step size in an interval $[\beta_{\text{min}}, \beta_{\text{max}}]$ with $\beta_{\text{max}} > \beta_{\text{min}} > 0$. However, as we mentioned, the high nonlinearity of TEiCPs often yields a nonconcave (or nonconvex) objective function. In this situation, the BB step size is negative, which immediately makes SPG methods take a very small $\beta_{\text{min}}$ as its updating step size for TEiCPs. A direct consequence is that SPG methods require more iterations and computing time to get an approximate solution. Therefore, a natural question is that how can we keep the nature of BB step size as much as possible. (ii) SPG methods employ a monotone line search to find an updating direction such that the objective function always increases monotonically, which is a promising property of SPG methods. However, the monotone line search procedure is often computational expensive since it is a relative stronger requirement with more inner iterations for evaluating tensor-vector multiplications. As we know, the tensor-vector multiplication defined by (1) is computational expensive for general cases (see numerical reports in [11]). Hence, another question is that how could we alleviate the computational burden to find an updating direction.

In this paper, our objective is to answer the aforementioned two questions. Specifically, we will modify the SPG method by incorporating a positive BB step size (i.e., the absolute value of the BB step size) and a nonmonotone line search technique proposed in [4]. The positive BB step size keep its nature as much as possible so that a larger step size can be absorbed in the SPG method for acceleration. On the other hand, the nonmonotone line search procedure can efficiently reduce inner iterations, thereby reducing evaluations of tensor-vector multiplications. A series of numerical results support that the proposed techniques can greatly improve the efficiency of the SPG method for TEiCPs.

The rest of this paper is organized as follows. In Section 2, we will give the details of the nonmonotone SPG method. Theoretically, we also show that the proposed algorithm is globally convergent to a solution of TEiCPs or a stationary point of its fractional program formulation. In Section 3, we will report extensive numerical results to show that the proposed method outperforms the existing first-order methods. Finally, we complete this paper with drawing a conclusion.

2. The nonmonotone SPG method. In this section, we first recall a fractional programming reformulation to TEiCP (3). Then, we present the nonmonotone spectral projected gradient algorithm for the resulted fractional programming model. Moreover, some convergence results will be given for the proposed algorithm. As shown in [14, Corollary 2.1], when $\mathcal{B}$ is strictly copositive, then TEiCP (3) has at least one solution. Hence, we assume throughout this paper that $\mathcal{B}$ is strictly copositive for ensuring that solution set of the problem under consideration is
nonempty. Below, we recall the following proposition proved in [14, Proposition 3.1], which is fundamental for the fractional programming reformulation.

**Proposition 1.** Assume that both $\mathcal{A}$ and $\mathcal{B}$ are symmetric tensors and $\mathcal{B}$ is strictly copositive. Let $\hat{x}$ be a stationary point of the following problem:

$$
\max \Phi(x) = \frac{A x^m}{B x^m}
$$

s.t. $x \in \Delta^n = \left\{ x \in \mathbb{R}^n : x \geq 0, \sum_{i=1}^n x_i = 1 \right\}$, \hspace{1cm} (4)

Then $(\lambda(\hat{x}) = \frac{A \hat{x}^m}{B \hat{x}^m}, \hat{x})$ is a solution (or eigenpair) of TEiCP (3).

According to Proposition 1, finding a solution to the symmetric TEiCP (3) amounts to finding a stationary point of the optimization problem (4). Obviously, $\Delta^n$ is a closed convex set in $\mathbb{R}^n$ and the gradient function

$$
\varphi(x) = \nabla \Phi(x) = \frac{m}{B x^m} (A x^{m-1} - \frac{A x^m}{B x^m} B x^{m-1}).
$$

Throughout, we define $P_{\Delta^n} : \mathbb{R}^n \to \Delta^n$ as the orthogonal projection operator onto $\Delta^n$, i.e.,

$$
P_{\Delta^n}(x) = \arg \min \left\{ ||x - y|| : y \in \Delta^n \right\}.
$$

By using the property of projection operator, we have the following lemma which has been given in [4, Lemma 2.1] (see also [19, Lemma 1]), and the proof is skipped here.

**Lemma 2.1.** Let $\varphi(x)$ be the gradient of $\Phi(x)$ given in (4). Then, it holds, for all $x \in \Delta^n$ and $\beta \in (0, \beta_{\text{max}}]$, that

$$
\varphi(x)^\top d_{\beta}(x) \geq \frac{1}{\beta} ||d_{\beta}(x)||^2 \geq \frac{1}{\beta_{\text{max}}} ||d_{\beta}(x)||^2 \geq 0,
$$

where $d_{\beta}(x) = P_{\Delta^n}(x + \beta \varphi(x)) - x$.

Apparently, the assertion of Lemma 2.1 means that $d_{\beta}(x)$ can be an ascent direction at $x$ when $\beta > 0$. Accordingly, the authors of [19] employed the novel BB step size [2] with a positivity preserving strategy to update iterates, i.e., setting

$$
\beta_{k+1} = \begin{cases} 
\beta_{\text{max}}, & \text{if } \langle s_k, y_k \rangle \leq 0, \\
\max \left\{ \beta_{\text{min}}, \min \left\{ \beta_{\text{max}}, \frac{\langle s_k, s_k \rangle}{\langle s_k, y_k \rangle} \right\} \right\}, & \text{if } \langle s_k, y_k \rangle > 0,
\end{cases}
$$

where $s_k = x_k - x_{k-1}$, $y_k = \varphi(x_k) - \varphi(x_{k-1})$, and $\frac{\langle s_k, s_k \rangle}{\langle s_k, y_k \rangle}$ is the so-called BB step size. It can be easily seen that the $\beta_{k+1}$ defined by (6) is always a positive number.

In this situation, the algorithm in [19] can always update its iterates along an ascent direction until it gets a solution to (4). However, the optimization reformulation (4) is often a nonconvex problem due to the high nonlinearity of $\Phi(x)$ induced by tensor-vector multiplication. Therefore, it can be observed from (6) that $\beta_{k+1}$ is usually specified as $\beta_{\text{max}}$ in practice since $\langle s_k, y_k \rangle \leq 0$ caused by the nonconvexity of $\Phi(x)$. It is well-known that a constant $\beta_{k+1}$ is not an ideal choice for the algorithm. In other words, the updating strategy (6) did not make the best use of BB step size when solving the problem under consideration, which also encourages us to find a better strategy to update $\beta_{k+1}$. Actually, if we take the absolute value of the BB step size, we can also keep the positivity of $\beta_{k+1}$ and do not care about the sign of $\langle s_k, y_k \rangle$. 


It is noteworthy that the original SPG method proposed in [19] for TEiCPs employs a monotone line search strategy. However, it has been shown numerically that the nonmonotone line search technique is often more efficient than monotone ones in terms of taking less inner iterations to satisfy some descent (or ascent) inequality (we refer the reader to [4, 10, 20] for numerical reports). To our numerical experience, we observed that the tensor-vector multiplication (see (1)) is comparatively time-consuming. So, it is reasonable to employ a nonmonotone line search strategy to find a new iterate. Specifically, we can show details of the proposed method in Algorithm 1.

**Algorithm 1** The nonmonotone SPG method for TEiCPs

1. Let $\varepsilon > 0$ be a tolerance of termination, $\sigma, \rho \in (0, 1)$. $M \geq 1$ be an integer, $0 < \beta_{\min} < \beta_{\max}$, and $x_0 \geq 0$ be an initial point. Set $\beta_0 = 1/\|\varphi(x_0)\|$ and $k = 0$.
2. Compute $z_k = P_{\Delta^n}(x_k + \beta_k \varphi(x_k))$, and the direction $d_{\beta_k}(x_k) = z_k - x_k$.
3. If $\|d_{\beta_k}(x_k)\| \leq \varepsilon$ then stop, the output $\lambda = A x_m B x_m$ is a Pareto-eigenvalue and $x_k$ is the corresponding Pareto-eigenvector of TEiCP.
4. Set $\alpha_k = \rho^i$, where $i \geq 0$ is the smallest integer, such that $\Phi(x_k + \alpha_k d_{\beta_k}(x_k)) \geq \min_{0 \leq j \leq \min\{k, M-1\}} \Phi(x_{k-j}) + \sigma \alpha_k \varphi(x_k)^T d_{\beta_k}(x_k)$, and let $x_{k+1} = x_k + \alpha_k d_{\beta_k}(x_k)$.
5. Let $s_k = x_{k+1} - x_k$ and $y_k = \varphi(x_{k+1}) - \varphi(x_k)$. Update $\beta_{k+1} = \max \left\{ \beta_{\min}, \min \left\{ \beta_{\max}, \frac{\langle s_k, s_k \rangle}{\langle s_k, y_k \rangle} \right\} \right\}$.

Go to Step 2.

**Remark 1.** We notice that Step 2 in Algorithm 1 contains a projection onto a convex set. Roughly speaking, computing a projection onto a general set is not an easy task. However, there is no computational worry about this projection since such a convex set $\Delta^n$ is a simplex set and many efficient solvers are available, e.g., see [3, 6].

**Remark 2.** It can be easily seen from (7) that Algorithm 1 reduces to a monotone SPG method with an absolute value BB step size when setting $M = 1$. Empirically, larger $M$’s usually perform better than smaller ones, which will be highlighted in our computational results (see Table 1). Moreover, we can see that $\beta_{k+1}$ defined in Algorithm 1 always is a positive number in $[\beta_{\min}, \beta_{\max}]$, which, together with Lemma 2.1, ensures that $d_{\beta_k}(x_k)$ given in Step 2 is an ascent direction. Consequently, we can successfully find an appropriate $\alpha_k$ satisfying (7) in finite trials. Therefore, our algorithm is well defined.

Below, we show that the sequence generated by Algorithm 1 converges to a solution of (3). Moreover, we can follow the way of [4] to establish its proof and show it here for the completeness of this paper.

**Theorem 2.2.** Let $\{x_k\}$ be generated by Algorithm 1. If there is a vector $x_k$ such that $d_{\beta_k}(x_k) = 0$, then $\Phi(x_k, x_k)$ is a solution to the symmetric TEiCP (3). Otherwise, any accumulation point of the sequence $\{x_k\}$ is a constrained stationary
point, i.e., the sequence \( \{\Phi(x_k)\} \) converges to a Pareto eigenvalue of the symmetric TEiCP (3).

**Proof.** It is clear from (5) that the search direction \( d_{\beta_k}(x_k) \) is always an ascent direction. Hence, as stated in Remark 2, Algorithm 1 is well defined.

If there is a vector \( x_k \) satisfying \( d_{\beta_k}(x_k) = 0 \), then it follows from [3, Theorem 9.10] and Proposition 1 that \( (\Phi(x_k), x_k) \) is a solution to TEiCP (3). Otherwise, let \( x^* \) is an accumulation point of \( \{x_k\} \) and relabel \( \{x_k\} \) a subsequence converging to \( x^* \). Below, we consider two cases to show \( x^* \) being a constrained stationary point of (4).

**Case 1.** Assume that \( \alpha_k \rightarrow 0 \). Suppose, by contradiction, that \( x^* \) is not a stationary point, which immediately means that \( d_{\beta}(x^*) \neq 0 \). By continuity and compactness, there exists a constant \( \delta > 0 \) such that

\[
\frac{\varphi(x^*)^\top d_{\beta}(x^*)}{\|d_{\beta}(x^*)\|} > \delta, \quad \text{for all} \quad \beta \in [\beta_{\min}, \beta_{\max}].
\]

Consequently, for large enough \( k \) on the subsequence converging to \( x^* \), we have

\[
\frac{\varphi(x_k)^\top d_{\beta}(x_k)}{\|d_{\beta}(x_k)\|} > \frac{\delta}{2}, \quad \text{for all} \quad \beta \in [\beta_{\min}, \beta_{\max}].
\]  

(8)

Since \( \inf \alpha_k = 0 \), there exists a subsequence \( \{x_k\}_K \) such that \( \lim_{k \in K} \alpha_k = 0 \), where \( K \) is an index set. In that case, from the way \( \alpha_k \) chosen in (7), there exists an index \( \bar{k} \) sufficiently large such that for all \( k \geq \bar{k} \), \( k \in K \), there exists \( p_k = \rho^i \in (0, 1) \) for which \( p_k \) fails to satisfy the line search condition (7), i.e.,

\[
\Phi(x_k + p_k d_{\beta_k}(x_k)) < \min_{0 \leq j \leq M-1} \Phi(x_{k-j}) + p_k \sigma \varphi(x_k)^\top d_{\beta_k}(x_k)
\]

\[
\leq \Phi(x_k) + p_k \sigma \varphi(x_k)^\top d_{\beta_k}(x_k),
\]

As a consequence,

\[
\frac{\Phi(x_k + p_k d_{\beta_k}(x_k)) - \Phi(x_k)}{p_k} < \sigma \varphi(x_k)^\top d_{\beta_k}(x_k).
\]

An application of the mean value theorem to the above relation immediately yields

\[
\varphi(x_k + t_k d_{\beta_k}(x_k))^\top d_{\beta_k}(x_k) < \sigma \varphi(x_k)^\top d_{\beta_k}(x_k) \quad \text{for all} \quad k \in K, \ k \geq \bar{k}, \quad (9)
\]

where \( t_k \in [0, p_k] \) that goes to zero as \( k \in K \) goes to infinity.

Taking a convenient subsequence so that \( d_{\beta_k}(x_k)/\|d_{\beta_k}(x_k)\| \) converges to \( d^* \). Notice that the sequence \( \{d_{\beta_k}(x_k)\}_K \) is bounded and then \( t_k \|d_{\beta_k}(x_k)\| \rightarrow 0 \) since \( t_k \rightarrow 0 \). Therefore, by taking limits in (9), we arrive at \( (1 - \sigma) \varphi(x^*)^\top d^* \leq 0 \). Since \( 1 - \sigma > 0 \) with \( \sigma \in (0, 1) \) and \( \varphi(x_k)^\top d_{\beta_k}(x_k) > 0 \) for all \( k \), then \( \varphi(x^*)^\top d^* = 0 \). By continuity and the definition of \( d_{\beta}(x) \) (see Lemma 2.1), we have, for all enough large \( k \) on that subsequence,

\[
\frac{\varphi(x_k)^\top d_{\beta_k}(x_k)}{\|d_{\beta_k}(x_k)\|} < \delta/2,
\]

which contradicts to (8).

**Case 2.** Assume that \( \inf \alpha_k \geq \mu > 0 \). Similarly, we show \( x^* \) being a constrained stationary point by a contradiction. Suppose \( x^* \) is not a constrained stationary point, by continuity and compactness, there exists a constant \( \delta > 0 \) such that
\[\|d_\beta(x^*)\| \geq \delta > 0 \text{ for all } \beta \in [\mu, \beta_{\text{max}}].\]
Let \(l(k)\) be an integer such that \(k - \min\{k, M - 1\} \leq l(k) \leq k\) and
\[
\Phi(x_{l(k)}) = \min_{0 \leq j \leq \min\{k, M - 1\}} \Phi(x_{k-j})
\]
is a monotonically nondecreasing sequence. It then follows from (7) that, for \(k > M - 1\) (see [10] for more details),
\[
\Phi(x_{l(k)}) \geq \min_{0 \leq j \leq M-1} \Phi(x_{l(k) - j}) + \sigma \alpha_{l(k) - 1} (x_{l(k) - 1}) \Phi(x_{l(k) - 1})
\]
\[
= \Phi(x_{l(l(k) - 1)}) + \sigma \alpha_{l(k) - 1} (x_{l(k) - 1}) \Phi(x_{l(k) - 1})
\]
By the continuity of \(d_\beta(x)\), it holds for sufficiently large \(k \geq k\) that \(\|d_\beta_k(x_k)\| \geq \delta/2\).
Hence, it follows from Lemma 2.1 that
\[
\Phi(x_{l(k)}) - \Phi(x_{l(l(k) - 1)}) \geq \frac{\sigma \mu}{\beta_{\text{max}}} \|d_\beta_{l(k) - 1} (x_{l(k) - 1})\|^2 \geq \frac{\sigma \mu \delta^2}{4 \beta_{\text{max}}}.
\]
When \(k \to \infty\), it is clear that \(\Phi(x_{l(k)}) \to \infty\), which is a contradiction to the fact \(\Phi(x)\) being finite on the simplex set \(\Delta^n\). Therefore, the accumulation point \(x^*\) is a constrained stationary point. Moreover, the continuity of \(\Phi(\cdot)\) implies that \(\{x_k\}\) converges to \(\Phi(x^*)\), which is Pareto eigenvalue of the symmetric TEiCP (3).

3. Numerical experiments.
In this section, we will show the efficiency of Algorithm 1 (NSPG for short) through extensive numerical results on structured tensors, in addition to comparing Algorithm 1 with some existing efficient solvers such as the SPP method (see [7]) and the first SPG method (see [19, Algorithm 1]) which is fastest one by the results reported in [19] and will be denoted by “SPG1” for short. All algorithms were coded by MATLAB R2017a, and all numerical experiments were conducted on a laptop with Windows 10 system and Intel(R) Core(TM) i5-2410U CPU processor with a 6 GB memory.

Notice that there are some parameters in the algorithms. First, we set \(\beta_{\text{min}} = 10^{-10}, \beta_{\text{max}} = 10^{10}, \sigma = 0.05, \text{ and } \rho = 0.6\) for the proposed NSPG method. For SPG1, we follow the settings as used in [19], where \(\rho = 0.05, \beta_{\text{max}} = \|\varphi(x_k)\|, \text{ and } \beta_{\text{min}} = 1/\|\varphi(x_k)\|\); For the SPP method, we take \(\tau = 10^{-6}\) as specified in [7]. Additionally, since both NSPG and SPG1 share the similar idea, we follow the stopping criterion used in [19] to terminate the algorithms for computational fairness, i.e., min\(\{\|x_{k+1} - x_k\|, \|\varphi(x_{k+1}) - \varphi(x_k)\|\} \leq \varepsilon\). Unlike, we take \(\|x_{k+1} - x_k\| \leq \varepsilon\) as the stopping criterion for SPP method. Throughout, we set \(\varepsilon = 10^{-6}\) and employ the method proposed in [6] to compute the projection onto a simplex set. Moreover, in our experiments, we start iteration from the point \(x_0 = (1, 1, \cdots, 1)^T\) and set the maximum iteration as 1000 for all algorithms.

In the following experiments, we will consider three types of eigenpairs to TE-iCPs, which cover all cases tested in the literature. Specifically, the first one is \(Z\)-eigenpair, which refers to \(B\) being an identity tensor such that \(B x^m = \|x\|^{m-2} x\) for all \(x \in \mathbb{R}^n\) (see [5, 13]). Another type is \(B\)-eigenpair which corresponds to the case where \(B\) is a general tensor (however, we only consider some structured tensors for \(B\)). The last one is the so-named \(H\)-eigenpair mentioned in introduction (see also [13, 16]).

Below, we consider ten examples and some of them come from [12, 15]. First, we deal with the case of computing Pareto \(Z\)-eigenpairs in Examples 1-3.
Example 1. This example comes from [12], where $A$ is a 4th order 3-dimensional symmetric tensor whose components are $a_{1111} = 0.2883$, $a_{1112} = -0.0031$, $a_{1113} = 0.1973$, $a_{1122} = -0.2485$, $a_{1223} = 0.1862$, $a_{1133} = 0.3847$, $a_{1223} = 0.2972$, $a_{1123} = -0.2939$, $a_{1233} = 0.0919$, $a_{1333} = -0.3619$, $a_{2222} = 0.1241$, $a_{2223} = -0.3420$, $a_{2233} = 0.2127$, $a_{2333} = 0.2727$, $a_{3333} = -0.3054$, where $[i_1 i_2 \cdots i_n]$ represents all permutations of $\{i_1, i_2, \cdots, i_n\}$.

Example 2. Let $A \in S^{[4,n]}$ be a diagonal tensor whose diagonals are given by $a_{iiii} = (i-1)/i$ for $i = 1, 2, \cdots, n$.

Example 3. Let $A$ be a 4th order 3-dimensional symmetric tensor, whose non-zero elements are given by $a_{[1111]} = 1.00397$, $a_{[2222]} = 0.99397$, $a_{[3333]} = 1.00207$, $a_{[1222]} = 0.00788$, $a_{[2111]} = 0.00788$, $a_{[3111]} = 0.00001$, $a_{[3222]} = 0.00005$, $a_{[1333]} = 0.99603$, $a_{[2333]} = 1.00400$.

The following three problems (i.e., Examples 4-6) correspond to finding Pareto $H$-eigenpairs of TEICP (3) with 4th order tensors. Such three examples come from [15]. Since they are structured tensors, we consider some higher dimensional scenarios on $n$, i.e., $n = \{50, 75, 100\}$ for the three problems.

Example 4. Let $A$ be a symmetric tensor whose elements are given by $a_{ijkl} = \sin(i+j+k+l)$, for $1 \leq i,j,k,l \leq n$.

Example 5. Let $A$ be a symmetric tensor defined by $a_{ijkl} = \tan(i) + \tan(j) + \tan(k) + \tan(l)$, $(1 \leq i,j,k,l \leq n)$.

Example 6. Let $A$ be a symmetric tensor given by $a_{ijkl} = \frac{(-1)^i}{i} + \frac{(-1)^j}{j} + \frac{(-1)^k}{k} + \frac{(-1)^l}{l}$, $(1 \leq i,j,k,l \leq n)$.

Hereafter, we consider two examples (i.e., Examples 7-8) on computing $B$-eigenpairs of TEICP (3) with 4th order tensors $A$ and $B$, and three scenarios on $n$, i.e., $n = \{50, 75, 100\}$ will be tested in our numerical simulation.

Example 7. Let $A$ be a symmetric tensor defined by $a_{ijkl} = \sin(i+j+k+l)$ for $1 \leq i,j,k,l \leq n$, and $B$ be a diagonal tensor with $b_{iiii} = 1/\sin^2(i)$ for $1 \leq i \leq n$.

Example 8. Let $A$ be a symmetric tensor defined by $a_{ijkl} = \cos(i+j+k+l)$, and $B$ be the Hilbert tensor given by $b_{ijkl} = 1/(i+j+k+l-3)$ for $1 \leq i,j,k,l \leq n$.

More generally, we below consider two random cases on tensor $A$. Specifically, we consider computing $Z$-eigenpairs and $H$-eigenpairs in Examples 9 and 10, respectively. Moreover, we test a 4th order tensor $A$ with three scenarios on $n$, i.e., $n = \{50, 75, 100\}$, and conduct an $m$th order tensor $A$ in Example 10 with three scenarios on $m$ and $n$, i.e., $m = \{3, 4, 5\}$ and $n = \{40, 50, 60\}$ respectively.

Example 9. Let $A$ be a 4th order symmetric tensor whose elements satisfy a normal distribution with mean 0 and standard deviation 1, which can be generated by the Matlab script $A$ = randn$(n,n,n)$ and symmetrized by the tensor toolbox [1].

Example 10. Let $A$ be an $m$th order $n$-dimensional symmetric tensor, which can be generated by the way used in Example 9.

As we know, different $M$’s will result in different performance of Algorithm 1. Therefore, we consider three choices on $M$, i.e., $M = \{1, 5, 15\}$ and report their
numerical results for Examples 1-10. Throughout, we report number of iteration \( \text{(its)} \) and number of inner iterations \( \text{(iits)} \) which is presented for the purpose of highlighting the necessity of nonmonotone strategy in solving TEiCPs. Also, we list computing time in seconds \( \text{(time)} \) and the final error \( \text{(err)} \) stopped at an approximate solution. All computational results are summarized in Table 1.

It can be easily seen from Table 1 that the proposed NSPG equipped with an absolute value BB step sizes performs better than both SPG1 and SPP in many cases. The behind reason is that NSPG possesses a better step size than SPG1. Moreover, we can see from Table 1 that NSPG equipped with \( M = 5 \) and 15 perform better than the two monotone ones (i.e., NSPG with \( M = 1 \) and SPG1) due to their less inner iterations, which also efficiently support the promising property of nonmonotone technique. Another remarkable conclusion of NSPG is that larger \( M \)'s are better than smaller ones in most cases. Therefore, the results in Table 1 efficiently support the motivation of this paper.

In the above experiments, all algorithms started from a fixed starting point. Hence, to further support the idea of Algorithm 1, we below investigate the sensitivity of starting points to algorithms. Here, we randomly generate ten initial points which are uniformly distributed in \((0, 1)\) and then plot standard error bars with respect to number of iterations and computing time in Figs. 1-3, where the height of a bar represents the mean value of ten trials and the length of a line segment means the standard deviation of ten trials. Here, we only consider NSPG with two scenarios on \( M \), i.e., \( M = 1 \) and 5. Moreover, since Examples 1-3 are much easier than others, we just summarize the plots for Examples 4-10, where a lower dimensional case \( n = 25 \) for each problem is investigated.

Clearly, we can observe from Figs. 1-3 that NSPG equipped with \( M = 1 \) and \( M = 5 \) perform much more stable than both SPG1 and SPP. In other words, the proposed NSPG is more insensitive to starting points than SPG1 and SPP, which further highlight the necessity of the absolute value BB step size and nonmonotone strategy for computing TEiCPs.

4. Conclusion. In this paper, we proposed a nonmonotone spectral projected gradient method by absorbing an absolute value BB step size for finding solutions to TEiCPs. The remarkable advantage is that the proposed method can efficiently exploit the benefit of BB step size to reduce the probability of taking a constant step size as appeared in the original spectral projected gradient method. Moreover, the nonmonotone line search technique can reduce the number of inner iterations to save computing time. All mentioned promising properties of our method are demonstrated through extensive numerical results on synthetic problems with structured tensors.

Acknowledgments. We would like to thank the reviewers for their close reading and comments, which helped us improve the presentation of this paper.
### Table 1: Computational results for Examples 1-10.

| Example | Computational results | Description |
|---------|-----------------------|-------------|
| 1       |                       |             |
| 2       |                       |             |
| 3       |                       |             |
| 4       |                       |             |
| 5       |                       |             |
| 6       |                       |             |
| 7       |                       |             |
| 8       |                       |             |
| 9       |                       |             |
| 10      |                       |             |

The symbol *err* denotes that the number of iterations exceeds 1000.
Figure 1. Mean and standard deviation of iterations and computing time of ten trials with random starting points for the case of computing Pareto $H$-eigenpairs. From top to bottom corresponding to Example 4, 5, and 6, respectively.
Figure 2. Mean and standard deviation of iterations and computing time of ten trials with random starting points for the case of computing Pareto $B$-eigenpairs. The first and second rows correspond to Example 7 and 8, respectively.

Figure 3. Mean and standard deviation of iterations and computing time of ten trials with random starting points for the case of computing Pareto $Z$- and $H$-eigenpairs, respectively. The first and second rows correspond to Example 9 and 10, respectively.
REFERENCES

[1] B. W. Bader, T. G. Kolda et al., MATLAB Tensor Toolbox Version 2.6, Available online, 2015, URL http://www.sandia.gov/~tgkolda/TensorToolbox/.
[2] J. Barzilai and J. M. Borwein, Two-point step size gradient methods, *IMA J. Numer. Anal.*, 8 (1988), 141–148.
[3] A. Beck, *Introduction to Nonlinear Optimization Theory, Algorithms, and Applications with MATLAB*, SIAM, Philadelphia, 2014.
[4] E. Birgin, J. Martinez and M. Raydan, Nonmonotone spectral projected gradient methods on convex sets, *SIAM J. Optim.*, 10 (2000), 1196–1211.
[5] K. Chang, K. Pearson and T. Zhang, On eigenvalue problems of real symmetric tensors, *J. Math. Anal. Appl.*, 350 (2009), 416–422.
[6] Y. Chen and X. Ye, Projection onto a simplex. arXiv:1101.6081.
[7] Z. Chen, Q. Yang and L. Ye, Generalized eigenvalue complementarity problem for tensors, *Pac. J. Optim.*, 13 (2017), 527–545.
[8] Z. Chen and L. Qi, A semismooth Newton method for tensor eigenvalue complementarity problem, *Comput. Optim. Appl.*, 65 (2016), 109–126.
[9] J. Fan, J. Nie and A. Zhou, Tensor eigenvalue complementarity problems, *Math. Program. Ser. A*, 170 (2018), 507–539.
[10] L. Grippo, F. Lampariello and S. Lucidi, A nonmonotone line search technique for Newton’s method, *SIAM J. Numer. Anal.*, 23 (1986), 707–716.
[11] H. He, C. Ling, L. Qi and G. Zhou, A globally and quadratically convergent algorithm for solving multilinear systems with M-tensors, *J. Sci. Comput.*, 76 (2018), 1718–1741.
[12] E. Koldis and P. Regalia, On the best rank-1 approximation of higher-order supersymmetric tensors, *SIAM J. Matrix Anal. Appl.*, 23 (2002), 863–884.
[13] L. Lim, Singular values and eigenvalues of tensors: A variational approach, in *IEEE CAMSAP 2005: First International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, CINVESTAV; IEEE Signal Proc Soc, 2005, 129–132, (1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, Puerto Vallarta, MEXICO, DEC 13-15, 2005).
[14] C. Ling, H. He and L. Qi, On the cone eigenvalue complementarity problem for higher-order tensors, *Comput. Optim. Appl.*, 63 (2016), 143–168.
[15] J. Nie and L. Wang, Semidefinite relaxations for best rank-1 tensor approximations, *SIAM J. Matrix Anal. Appl.*, 35 (2014), 1155–1179.
[16] L. Qi, Eigenvalues of a real supersymmetric tensor, *J. Symbolic Comput.*, 40 (2005), 1302–1324.
[17] L. Qi, Symmetric nonnegative tensors and copositive tensors, *Linear Algebra Appl.*, 439 (2013), 228–238.
[18] L. Qi, H. Chen and Y. Chen, *Tensor Eigenvalues and Their Applications*, Springer, Singapore, 2018.
[19] G. Yu, Y. Song, Y. Xu and Z. Yu, Spectral projected gradient methods for generalized tensor eigenvalue complementarity problem, *Numer. Algor.*, 80 (2019), 1181–1201.
[20] H. Zhang and W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, *SIAM J. Optim.*, 14 (2004), 1043–1056.

Received March 2020; 1st revision September 2020; Final revision September 2020.

E–mail address: danyi_tong@qq.com
E–mail address: hehjmath@hdu.edu.cn
E–mail address: macling@hdu.edu.cn
E–mail address: maqilq@polyu.edu.hk