Received: 18 November 2017 | Revised: 13 March 2019 | Accepted: 14 June 2019

DOI: 10.1002/nla.2260

RESEARCH ARTICLE

WILEY

Relaxation methods for solving the tensor equation arising from the higher-order Markov chains

Dongdong Liu[1,2] | **Wen Li**[3] | **Seak-Weng Vong**[1]

[1]Department of Mathematics, Faculty of Science and Technology, University of Macau, Macau, China

[2]School of Applied Mathematics, Guangdong University of Technology, Guangzhou, China

[3]School of Mathematical Sciences, South China Normal University, Guangzhou, China

Correspondence
Wen Li, School of Mathematical Sciences, South China Normal University, Guangzhou 510631, China.
Email: liwen@scnu.edu.cn

Funding information
Opening Project of Guangdong Province Key Laboratory of Computational Science at the Sun Yat-sen University, Grant/Award Number: 2018008; National Natural Science Foundation of China, Grant/Award Number: 11671158, U1811464, and 11771159; Major Project, Grant/Award Number: 2016KZDXM025; Innovation Team Project of Guangdong Provincial Universities, Grant/Award Number: 2015KCXTD007; University of Macau, Grant/Award Number: MYRG2017-00098-FST and MYRG2018-00047-FST; Macao Science and Technology Development Fund, Grant/Award Number: 050/2017/A

Summary

In this paper, we propose several relaxation algorithms for solving the tensor equation arising from the higher-order Markov chain and the multilinear PageRank. The semi-symmetrization technique on the original equation is also employed to modify the proposed algorithms. The convergence analysis is given for the proposed algorithms. It is shown that the new algorithms are more efficient than the existing ones by some numerical experiments when relaxation parameters are chosen suitably.

KEYWORDS

higher-order Markov chain, multilinear PageRank, relaxation algorithm, tensor equation

1 | INTRODUCTION

Markov chains are powerful tools to analyze a variety of stochastic (probabilistic) processes over time (see other works[1,2]). Recently, Li et al.[3] have proposed an approximate tensor model for higher-order Markov chains and presented the theoretical and numerical analysis for the proposed model. Later, many researchers worked on this field (e.g., see the work of Chang et al.[4,5]) because of its applications such as in multilinear PageRank.[6]

The approximate tensor model for higher-order Markov chains[3] is given as follows:

$$\mathbf{x} = \mathcal{P}\mathbf{x}^{m-1}, \mathbf{x} \geq 0, \|\mathbf{x}\|_1 = 1, \tag{1}$$

where the tensor–vector product $\mathcal{P}\mathbf{x}^{m-1}$ is defined[7]:

$$(\mathcal{P}\mathbf{x}^{m-1})_i = \sum_{i_2,\ldots,i_m=1}^{n} p_{ii_2\cdots i_m} x_{i_2}\cdots x_{i_m},\ i = 1, 2, \ldots, n, \tag{2}$$

and $\mathcal{P} = (p_{i_1 i_2 \cdots i_m})$ is an order-m tensor representing an $(m-1)$-th-order Markov chain, which is called an order-m dimension n stochastic tensor (or transition probability tensor), that is,

$$p_{i_1 i_2 \cdots i_m} \geq 0,\ \sum_{i_1=1}^{n} p_{i_1 i_2 \cdots i_m} = 1, \tag{3}$$

and $\mathbf{x} = (x_i)$ is called a stochastic vector, that is, $x_i \geq 0$ and $\sum_{i=1}^{n} x_i = 1$.

The model (1) can be applied to the multilinear PageRank, which was first studied in the work of Gleich et al.[6]

$$\mathbf{x} = \theta \hat{\mathcal{P}} \mathbf{x}^{m-1} + (1-\theta)\mathbf{v}, \tag{4}$$

where $\hat{\mathcal{P}}$ is also a stochastic tensor, $\mathbf{v}$ is a stochastic vector, and $\theta \in (0, 1)$ is a parameter. Equation (4) can be rewritten as Equation (1).

$$\mathbf{x} = \mathcal{P}\mathbf{x}^{m-1},\ \|\mathbf{x}\|_1 = 1,\ \mathcal{P} = \theta\hat{\mathcal{P}} + (1-\theta)\mathcal{V}, \tag{5}$$

where $\mathcal{V} = (v_{i_1 i_2 \cdots i_m})$ with $v_{i_1 i_2 \cdots i_m} = v_{i_1}, \forall i_2, \ldots, i_m$. It is easy to check that $\mathcal{P}$ is also a stochastic tensor.

In the work of Li et al.,[3] they proposed a fixed-point algorithm for solving Equation (1), and then Liu et al.[8] gave a tensor splitting algorithm for solving (1). For solving (4), Gleich et al.[6] proposed some algorithms (a fixed-point method, a shifted fixed-point method, a nonlinear inner–outer iteration, an inverse iteration, and a Newton iteration). In this paper, we further develop some algorithms for solving (1) and (4). The contributions of this paper are

- to propose some new algorithms. By suitable choices of parameters, we can get a faster convergence than the existing algorithms.
- to give the convergence analysis for the proposed algorithms. Some simple computable convergence conditions are also presented.
- to employ the propose algorithms for solving some application examples given in other works.[3,6]

The rest of this paper is organized as follows. In Section 2, we introduce some definitions and notations. Furthermore, we propose relaxation algorithms and give their convergence analysis. In Section 3, an accelerated technique for the algorithm in Section 2 is proposed, and some suggestions for the choice of parameters are provided. Besides, we give an equivalent equation with (1) by the semi-symmetrization technique, which is applied to modify the proposed algorithms. In Section 4, some application examples are given to illustrate the efficiency of the proposed algorithms. The final section is a concluding remark.

2 | THE PROPOSED ALGORITHMS

Firstly, we introduce some notations and definitions.

Let $\mathbb{C}$ ($\mathbb{R}$) be the complex (real) field and $\langle n \rangle = \{1, 2, \ldots, n\}$. An order-$m$ dimension n tensors $\mathcal{A}$ with n^m entries is defined as follows:

$$\mathcal{A} = (a_{i_1 \cdots i_m}),\ a_{i_1 \cdots i_m} \in \mathbb{C}(\mathbb{R}),\ i_j \in \langle n \rangle,\ j = 1, \ldots, m.$$

Let $\mathbb{C}^{[m,n]}$ ($\mathbb{R}^{[m,n]}$) denote the set of all the order-m dimension n tensor in the complex (real) fields. Specially, $\mathbb{C}^{[2,n]}$ ($\mathbb{R}^{[2,n]}$) and $\mathbb{C}^n$ ($\mathbb{R}^n$) denote the sets of all the $n \times n$ dimension complex (real) matrices and all the n dimension complex (real) vectors, respectively. Let $\|\cdot\|_1$ denote the 1-norm.

For a tensor $\mathcal{A} \in \mathbb{C}^{[m,n]}$ and a vector $\mathbf{x} \in \mathbb{C}^n$, the tensor–vector product $\mathcal{A}\mathbf{x}^{m-2}$ is defined as a matrix given by

$$(\mathcal{A}\mathbf{x}^{m-2})_{ij} = \sum_{i_3,\cdots,i_m=1}^{n} a_{iji_3\cdots i_m} x_{i_3}\cdots x_{i_m},\ i, j = 1, 2, \ldots, n. \tag{6}$$

It is easy to check that $\mathcal{A}\mathbf{x}^{m-1} = \mathcal{A}\mathbf{x}^{m-2}\mathbf{x}$. We define

$$\mathcal{P}(\mathbf{x}^{m-1} - \mathbf{y}^{m-1}) \equiv \mathcal{P}\mathbf{x}^{m-1} - \mathcal{P}\mathbf{y}^{m-1},$$

and

$$\mathcal{P}(\mathbf{x}^{m-2} - \mathbf{y}^{m-2}) \equiv \mathcal{P}\mathbf{x}^{m-2} - \mathcal{P}\mathbf{y}^{m-2}.$$

Notice that Equation (1) is equivalent to the following equation:

$$(I - \alpha\mathcal{P}\mathbf{x}^{m-2})\mathbf{x} = (1-\alpha)\mathbf{x}, \alpha > 0. \tag{7}$$

Let $\mathbf{x} \in \mathbb{R}^n$. We define $\mathbf{x}_+ = \max(\mathbf{x}, 0)$ and $\mathbf{x}_- = \mathbf{x} - \mathbf{x}_+$. Let $\text{proj}(\mathbf{x}) = \frac{\mathbf{x}_+}{\|\mathbf{x}_+\|_1}$. It is easy to see that $\text{proj}(\mathbf{x})$ is a stochastic vector. By (7), we propose the following Algorithm 1 for solving the tensor Equation (1).

Algorithm 1

1: Given a stochastic tensor $\mathcal{P}$, α, maximum $k_{\mathbf{max}}$, termination tolerance ε and an initial stochastic vector $\mathbf{x}_0$. Initialize $k = 1$.
2: while $k < k_{\mathbf{max}}$.
3: $(I - \alpha\mathcal{P}\mathbf{x}_{k-1}^{m-2})\hat{\mathbf{x}}_k = (1-\alpha)\mathbf{x}_{k-1}$, $\mathbf{x}_k = \text{proj}(\hat{\mathbf{x}}_k)$.
4: If $\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_1 < \varepsilon$, break and output $\mathbf{x}_k$.
5: $k = k + 1$, back to 2).

Remark 1. By choosing a suitable parameter α and an initial vector $\mathbf{x}_0$, the matrix $I - \alpha\mathcal{P}\mathbf{x}_k^{m-2}$ can be proved to be nonsingular, and then Algorithm 1 performs by solving a nonsingular linear system for the step k.

Next, applying the relaxation technique to the fixed-point method given in the work of Li et al.[3] leads to Algorithm 2.

Algorithm 2

1: Given a stochastic tensor $\mathcal{P}$, β, maximum $k_{\mathbf{max}}$, termination tolerance ε and an initial stochastic vector $\mathbf{x}_0$. Initialize $k = 1$.
2: while $k < k_{\mathbf{max}}$.
3: $\mathbf{y}_k = \mathcal{P}\mathbf{x}_{k-1}^{m-1}$, $\hat{\mathbf{x}}_k = \beta\mathbf{y}_k + (1-\beta)\mathbf{x}_{k-1}$, $\mathbf{x}_k = \text{proj}(\hat{\mathbf{x}}_k)$.
4: If $\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_1 < \varepsilon$, break and output $\mathbf{x}_k$.
5: $k = k + 1$, back to 2).

Remark 2. The technique of enforcing a stochastic normalization 'proj($\mathbf{x}$)' can be found in the work of Parikh et al.,[9] which is widely used to many algorithms and numerical experiments; for example, see other works.[6,10]

Let $\mathcal{P} = (p_{i_1 \cdots i_m})$ be a stochastic tensor, and let

$$\delta_m = \min_{S \subset \langle n \rangle} \left\{ \min_{i_2, i_3, \ldots, i_m \in \langle n \rangle} \sum_{i \in S} p_{ii_2 \ldots i_m} + \min_{i_2, i_3, \ldots, i_m \in \langle n \rangle} \sum_{i \in S'} p_{ii_2 \ldots i_m} \right\},$$

where S is a proper subset of $\langle n \rangle$ and S' is its complementary set in $\langle n \rangle$, that is, $S' = \langle n \rangle \setminus S$. It is easy to check that $\delta_m \leq 1$ if $\mathcal{P}$ is a stochastic tensor. Let $\kappa_m = 1 - \delta_m$. Then, $0 \leq \kappa_m \leq 1$.

In order to discuss the convergence of the proposed algorithms, we provide some lemmas. The following uniqueness condition of the solution for Equation (1) is given in the work of Li et al.[3]

Lemma 1. *Suppose that $\mathcal{P}$ is an order-m and dimension n stochastic tensor. If $\delta_m > \frac{m-2}{m-1}$, then the model $\mathcal{P}\mathbf{x}^{m-1} = \mathbf{x}$ has a unique solution.*

Lemma 2. *Let $\mathcal{P}$ be an order-m dimension n stochastic tensor, and* $\mathbf{x}, \mathbf{y}$, *and* $\mathbf{z}$ *be dimension n stochastic vectors. Then,*

$$\|\mathcal{P}\mathbf{z}^{m-2}(\mathbf{x}-\mathbf{y})\|_1 \leq \kappa_m \|\mathbf{x}-\mathbf{y}\}_1, \tag{8}$$

$$\|\mathcal{P}(\mathbf{x}^{m-1}-\mathbf{y}^{m-1})\|_1 \leq \kappa_m (m-1)\|\mathbf{x}-\mathbf{y}\|_1, \tag{9}$$

$$\|\mathcal{P}(\mathbf{x}^{m-2}-\mathbf{y}^{m-2})\|_1 \leq \kappa_m (m-2)\|\mathbf{x}-\mathbf{y}\|_1, \tag{10}$$

and

$$\|\mathcal{P}(\mathbf{x}^{m-2}-\mathbf{y}^{m-2})\mathbf{z}\|_1 \leq \kappa_m (m-2)\|\mathbf{x}-\mathbf{y}\|_1. \tag{11}$$

Proof. The proof of the inequalities (8)–(11) is similar to Theorem 3.2 in the work of Li et al.[3] Therefore, we omit it. □

Let $\eta_m = \kappa_m(m-1)$. It is easy to check that $0 \leq \eta_m < 1$ if $\delta_m > \frac{m-2}{m-1}$.

Remark 3. Let $\mathcal{P}$ be an order-m stochastic tensor with $\delta_m > \frac{m-2}{m-1}$. Then, Equation (1) has the unique solution $\mathbf{x}$. By Lemma 2, the sequence $\{\mathbf{x}_k\}$ generated by the fixed-point algorithm given in the work of Li et al.[3] converges to the solution $\mathbf{x}$. Moreover,

$$\|\mathbf{x}_k-\mathbf{x}\|_1 \leq \eta_m^k \|\mathbf{x}_0-\mathbf{x}\|_1. \tag{12}$$

It is noted that (12) was also given in the work of Li et al.[3]

2.1 | Convergence analysis for Algorithm 1

In this subsection, we will give convergence analysis for Algorithm 1.

Lemma 3. *If $A \in \mathbb{C}^{[2,n]}$ is a nonsingular matrix, $\|\cdot\|$ is a consistent matrix norm and for $C \in \mathbb{C}^{[2,n]}$,*

$$\|A^{-1}\| \leq \theta_1, \|A-C\| \leq \theta_2, \theta_1\theta_2 < 1,$$

then C is a nonsingular matrix.

Proof. The proof can be obtained by Theorem 2.5 in the work of Stewart et al.[11] □

Lemma 4. *Let $\hat{\mathbf{c}} = (\hat{c}_i) \in \mathbb{R}^n$ with $\sum_{i=1}^n \hat{c}_i = 1$ and $\mathbf{y} = (y_i)$ is a stochastic vector. If $\mathbf{c} = \mathrm{proj}(\hat{\mathbf{c}})$, then $\|\hat{\mathbf{c}}-\mathbf{y}\|_1 \geq \|\mathbf{c}-\mathbf{y}\|_1$.*

Proof. Because $\sum_{i=1}^n \hat{c}_i = 1$ and $\mathbf{y}$ is a stochastic vector, we know that $\sum_{i=1}^n(\hat{c}_i - y_i) = 0$, thus

$$\frac{1}{2}\|\hat{\mathbf{c}}-\mathbf{y}\|_1 = \|(\hat{\mathbf{c}}-\mathbf{y})_+\|_1. \tag{13}$$

Note that $\mathbf{c}$ is also a stochastic vector. We get $\sum_{i=1}^n(c_i - y_i) = 0$ and

$$\frac{1}{2}\|\mathbf{c}-\mathbf{y}\|_1 = \|(\mathbf{c}-\mathbf{y})_+\|_1. \tag{14}$$

Applying $\|\hat{\mathbf{c}}_+\|_1 = \|\hat{\mathbf{c}}_-\|_1 + 1 \geq 1$, we get

$$\|(\hat{\mathbf{c}}-\mathbf{y})_+\|_1 = \|(\hat{\mathbf{c}}_+-\mathbf{y})_+\|_1 = \|(\mathbf{c}\|\hat{\mathbf{c}}_+\|_1-\mathbf{y})_+\|_1 \geq \|(\mathbf{c}-\mathbf{y})_+\|_1, \tag{15}$$

which together with (13) and (14) gives the desired inequality. □

Let $\mathbf{x}$ be a solution of (1), and

$$\mathbb{S}_\mathbf{x} \equiv \{\alpha : I-\alpha\mathcal{P}\mathbf{x}^{m-2} \text{ is nonsingular matrix}\} \tag{16}$$

and

$$\mathbb{T}_\mathbf{x} \equiv \{\tilde{\mathbf{x}} : \|\tilde{\mathbf{x}}-\mathbf{x}\|_1 < \mu_1\}, \tag{17}$$

where $\mu_1 = \frac{1}{\alpha\kappa_m(m-2)\|(I-\alpha\mathcal{P}\mathbf{x}^{m-2})^{-1}\|_1}$.

Notice that $\mathcal{P}\mathbf{x}^{m-2}$ is a column stochastic matrix; hence, $\forall\alpha \in (0,1)$, $I-\alpha\mathcal{P}\mathbf{x}^{m-2}$ is nonsingular. Then, $(0,1) \subseteq \mathbb{S}_{\mathbf{x}}$. Next, we give convergence analysis based on the uniqueness condition of the solution of (1) in Lemma 1.

Theorem 1. *Let $\mathcal{P}$ be an order-m stochastic tensor with $\delta_m > \frac{m-2}{m-1}$ and $\mathbf{x}$ be a solution of (1). Then, for an arbitrary initial stochastic vector $\mathbf{x}_0 \in \mathbb{T}_{\mathbf{x}}$, the vector series $\{\mathbf{x}_k\}$ generated by Algorithm 1 is convergent and*

$$\|\mathbf{x}_k - \mathbf{x}\|_1 \leq \tau_\alpha^k \|\mathbf{x}_0 - \mathbf{x}\|_1 \tag{18}$$

provided $\alpha \in (0, \frac{2}{1+\eta_m}) \cap \mathbb{S}_{\mathbf{x}}$, where $\tau_\alpha = \frac{\alpha\kappa_m(m-2)+|1-\alpha|}{1-\alpha\kappa_m}$.

Proof. By Lemma 1 and the condition $\delta_m > \frac{m-2}{m-1}$, the solution $\mathbf{x}$ is unique. By Algorithm 1, $\mathbf{x}_k \geq 0$ and $\|\mathbf{x}_k\|_1 = 1$. Next, we prove that $I-\alpha\mathcal{P}\mathbf{x}_k^{m-2}$ is nonsingular.

By (10), we have

$$\left\|(I-\alpha\mathcal{P}\mathbf{x}^{m-2}) - \left(I-\alpha\mathcal{P}\mathbf{x}_0^{m-2}\right)\right\|_1 = \alpha\left\|\mathcal{P}\mathbf{x}^{m-2} - \mathcal{P}\mathbf{x}_0^{m-2}\right\|_1 \leq \alpha\kappa_m(m-2)\|\mathbf{x}-\mathbf{x}_0\|_1. \tag{19}$$

Because $\mathbf{x}_0 \in \mathbb{T}_{\mathbf{x}}$, we have

$$\alpha\kappa_m(m-2)\|(I-\alpha\mathcal{P}\mathbf{x}^{m-2})^{-1}\|_1\|\mathbf{x}-\mathbf{x}_0\|_1 < 1.$$

By Lemma 3, $I-\alpha\mathcal{P}\mathbf{x}_0^{m-2}$ is nonsingular. Suppose that $I-\alpha\mathcal{P}\mathbf{x}_{k-1}^{m-2}$ is nonsingular. Then, solving a nonsingular linear system in Step 3 of Algorithm 1 gives the unique solution $\widehat{\mathbf{x}}_k$, and then we have $\mathbf{x}_k$ and

$$\widehat{\mathbf{x}}_k = \alpha\mathcal{P}\mathbf{x}_{k-1}^{m-2}\widehat{\mathbf{x}}_k + (1-\alpha)\mathbf{x}_{k-1}. \tag{20}$$

Because $\mathbf{x}$ is the solution of Equation (1), we obtain

$$\mathbf{x} = \alpha\mathcal{P}\mathbf{x}^{m-2}\mathbf{x} + (1-\alpha)\mathbf{x}. \tag{21}$$

Let $\widehat{\mathbf{e}}_k = \widehat{\mathbf{x}}_k - \mathbf{x}$ and $\mathbf{e}_k = \mathbf{x}_k - \mathbf{x}$. By (20) and (21), we have

$$\widehat{\mathbf{e}}_k = \alpha\mathcal{P}\mathbf{x}_{k-1}^{m-2}\widehat{\mathbf{e}}_k + \alpha\mathcal{P}\left(\mathbf{x}_{k-1}^{m-2} - \mathbf{x}^{m-2}\right)\mathbf{x} + (1-\alpha)\mathbf{e}_{k-1}. \tag{22}$$

Let $\mathbf{z}_{k-1} = \mathcal{P}(\mathbf{x}_{k-1}^{m-2} - \mathbf{x}^{m-2})\mathbf{x}$. By (11), we obtain

$$\|\mathbf{z}_{k-1}\|_1 \leq \kappa_m(m-2)\|\mathbf{e}_{k-1}\|_1. \tag{23}$$

By (20), we have

$$\sum_{i=1}^{n}\left(\left(I-\alpha\mathcal{P}\mathbf{x}_{k-1}^{m-2}\right)\widehat{\mathbf{x}}_k\right)_i = (1-\alpha)\sum_{i=1}^{n} x_{k-1,i} = 1-\alpha; \tag{24}$$

hence

$$1-\alpha = \sum_{i=1}^{n}\widehat{x}_{k,i} - \alpha\sum_{i=1}^{n}\sum_{i_2,i_3,\ldots,i_m\in\langle n\rangle} p_{ii_2\cdots i_m}\widehat{x}_{k,i_2}x_{k-1,i_3}\cdots x_{k-1,i_m} \tag{25}$$

$$= \sum_{i=1}^{n}\widehat{x}_{k,i} - \alpha\sum_{i=1}^{n}\widehat{x}_{k,i}. \tag{26}$$

Therefore, we have

$$\sum_{i=1}^{n}\widehat{x}_{k,i} = 1,\ \sum_{i=1}^{n}\widehat{e}_{k,i} = 0. \tag{27}$$

Applying (8) gives

$$\left\|\mathcal{P}\mathbf{x}_{k-1}^{m-2}\widehat{\mathbf{e}}_k\right\|_1 \leq \kappa_m\|\widehat{\mathbf{e}}_k\|_1. \tag{28}$$

By combining this with (22) and (23), we get

$$\|\widehat{\mathbf{e}}_k\|_1 \leq \alpha\kappa_m\|\widehat{\mathbf{e}}_k\|_1 + \alpha\kappa_m(m-2)\|\mathbf{e}_{k-1}\|_1 + |1-\alpha|\|\mathbf{e}_{k-1}\|_1. \tag{29}$$

Then,

$$\|\widehat{\mathbf{e}}_k\|_1 \leq \tau_\alpha\|\mathbf{e}_{k-1}\|_1. \tag{30}$$

By (27) and Lemma 4, we get

$$\|\mathbf{e}_k\|_1 \leq \|\widehat{\mathbf{e}}_k\|_1, \tag{31}$$

which combining with (30) gives

$$\|\mathbf{e}_k\|_1 \leq \tau_\alpha\|\mathbf{e}_{k-1}\|_1 \leq \tau_\alpha^k\|\mathbf{e}_0\|_1. \tag{32}$$

If $\alpha \in (0, \frac{2}{1+\eta_m}) \cap \mathbb{S}_{\mathbf{x}}$, it is easy to check that $1-\alpha\kappa_m > 0$ and $0 < \tau_\alpha < 1$. It follows from Lemma 2 and (32) that

$$\begin{aligned}\left\|(I-\alpha\mathcal{P}\mathbf{x}^{m-2}) - \left(I-\alpha\mathcal{P}\mathbf{x}_k^{m-2}\right)\right\|_1 &= \alpha\left\|\mathcal{P}\mathbf{x}^{m-2} - \mathcal{P}\mathbf{x}_k^{m-2}\right\|_1 \leq \alpha\kappa_m(m-2)\|\mathbf{e}_k\|_1 \\ &\leq \tau_\alpha^k\alpha\kappa_m(m-2)\|\mathbf{e}_0\|_1.\end{aligned}$$

Because $\mathbf{x}_0 \in \mathbb{T}_{\mathbf{x}}$, we obtain

$$\tau_\alpha^k\alpha\kappa_m(m-2)\left\|\left(I-\alpha\mathcal{P}\mathbf{x}^{m-2}\right)^{-1}\right\|_1\|\mathbf{x}-\mathbf{x}_0\|_1 < 1.$$

It follows from Lemma 3 that $I-\alpha\mathcal{P}\mathbf{x}_k^{m-2}$ is nonsingular. This shows the assertion.

The error bound (18) follows from (32). By the above proof, we have $\tau_\alpha < 1$; hence, $\{\mathbf{x}_k\}$ converges to the solution of Equation (1). This completes the proof of the theorem. □

The following corollary follows immediately from Theorem 1.

Corollary 1. *Under the same assumption as Theorem 1 on $\mathcal{P}$, if $\alpha \in (0,1)$, then the $\{\mathbf{x}_k\}$ generated by Algorithm 1 converges to the solution of (1.1) globally for an arbitrary stochastic initial vector $\mathbf{x}_0 \in \mathbb{R}^n$. Moreover, the error bound (18) holds.*

Remark 4. Let $\mathbf{x}$ be a solution of Equation (1), $\widehat{\mathbf{e}}_k = \widehat{\mathbf{x}}_k - \mathbf{x}$ and $\mathbf{e}_k = \mathbf{x}_k - \mathbf{x}$. By the proof of Theorem 1, we have

$$0 \leq \|\mathbf{e}_k\|_1 \leq \|\widehat{\mathbf{e}}_k\|_1 \leq \tau_\alpha^k\|\mathbf{e}_0\|_1.$$

and $\tau_\alpha < 1$, which shows that the series $\{\mathbf{x_k}\}$ and $\{\widehat{\mathbf{x}}_\mathbf{k}\}$ generated by Algorithm 1 converge to the solution $\mathbf{x}$.

Remark 5. Let $\nu_1 = \frac{1-\eta_m}{1-\kappa_m(m-2+\eta_m)}$, and

$$\nu_2 = \min\left\{\frac{2}{1+\eta_m}, \frac{1+\eta_m}{\kappa_m(m-2)+\kappa_m^2(m-1)+1}\right\}.$$

It is easy to check that $0 \leq \nu_1 < 1$ and $\nu_2 > 1$. If $\alpha \in [\nu_1, \nu_2] \cap \mathbb{S}_{\mathbf{x}}$, we can get $\tau_\alpha \leq \eta_m$. In this case, by (12), we know that Algorithm 1 is faster than the fixed-point algorithm in the work of Li et al.[3]

Remark 6. Because $I-\alpha\mathcal{P}\mathbf{x}_k^{m-2}$ is nonsingular, a linear system for each iterative step in Algorithm 1 can be solved by direct methods such as the LU factorization.

2.2 | Convergence analysis of Algorithm 2

By Lemma 1, we analyze the convergence of Algorithm 2.

Theorem 2. *Let $\mathcal{P}$ be an order-m stochastic tensor with $\delta_m > \frac{m-2}{m-1}$ and $\mathbf{x}$ be a solution of (1). Then, the sequence $\{\mathbf{x}_k\}$ generated by Algorithm 2 converges to the solution $\mathbf{x}$ for any initial stochastic vector $\mathbf{x}_0$ provided $\beta \in (0, \frac{2}{1+\eta_m})$. Furthermore, we have the following error bound:*

$$\|\mathbf{x}_k - \mathbf{x}\|_1 \le \epsilon_\beta^k \|\mathbf{x}_0 - \mathbf{x}\|_1, \tag{33}$$

where $\epsilon_\beta = \beta\eta_m + |1-\beta|$.

Proof. By Algorithm 2, we know that $\mathbf{x}_k$ is always stochastic. By the condition $\delta_m > \frac{m-2}{m-1}$ and Lemma 1, we know that $\mathbf{x}$ is unique.

Let $\hat{\mathbf{e}}_k = \hat{\mathbf{x}}_k - \mathbf{x}$ and $\mathbf{e}_k = \mathbf{x}_k - \mathbf{x}$. By Algorithm 2, we have

$$\hat{\mathbf{e}}_k = \beta\mathcal{P}\left(\mathbf{x}_{k-1}^{m-1} - \mathbf{x}^{m-1}\right) + (1-\beta)\mathbf{e}_{k-1}. \tag{34}$$

By Lemma 2, we have

$$\left\|\mathcal{P}\left(\mathbf{x}_{k-1}^{m-1} - \mathbf{x}^{m-1}\right)\right\|_1 \le \eta_m \|\mathbf{e}_{k-1}\|_1. \tag{35}$$

Then, by (34), we have

$$\|\hat{\mathbf{e}}_k\|_1 \le (\beta\eta_m + |1-\beta|)\|\mathbf{e}_{k-1}\|_1. \tag{36}$$

It follows from Lemma 4 and (36) that the error bound (33) holds. It is easy to check that $0 < \epsilon_\beta < 1$ if $\beta \in (0, \frac{2}{1+\eta_m})$, which proves the convergence. This completes the proof of the theorem. □

Remark 7. If $\beta = 1$, then Algorithm 2 reduces to the fixed-point algorithm in the work of Li et al.[3]

3 | RELAXATION ACCELERATED TECHNIQUE FOR ALGORITHM 1

3.1 | Relaxation accelerated technique for Algorithm 1

In this subsection, we modify Algorithm 1 for the purpose of the fast convergence.

Algorithm 3

1: Given a stochastic tensor $\mathcal{P}$, α, γ, maximum $k_{\mathbf{max}}$, termination tolerance ε and an initial stochastic vector $\mathbf{x}_0$. Initialize $k = 1$.
2: while $k < k_{\mathbf{max}}$.
3: $(I - \alpha\mathcal{P}\mathbf{x}_{k-1}^{m-2})\mathbf{y}_k = (1-\alpha)\mathbf{x}_{k-1}$, $\hat{\mathbf{x}}_k = \gamma\mathbf{y}_k + (1-\gamma)\mathbf{x}_{k-1}$, $\mathbf{x}_k = \mathrm{proj}(\hat{\mathbf{x}}_\mathrm{k})$.
4: If $\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_1 < \varepsilon$, break and output $\mathbf{x}_k$.
5: $k = k+1$, back to 2).

Next, we give the convergence analysis for Algorithm 3 based on the uniqueness condition in Lemma 1.

Theorem 3. *Let $\mathcal{P}$ be an order-m stochastic tensor with $\delta_m > \frac{m-2}{m-1}$, $\mathbf{x}$ be a solution of (1). Then, for an arbitrary initial stochastic vector $\mathbf{x}_0 \in \mathbb{T}_\mathbf{x}$, the sequence $\{\mathbf{x}_k\}$ generated by Algorithm 3 converges to the solution $\mathbf{x}$ provided the parameters α and γ satisfy one of the following conditions:*

(1) $\alpha \in (0,1)$ *and* $\gamma \in \left(\frac{2\eta_m}{1+\eta_m}, \frac{2}{\alpha(\eta_m+1)}\right)$;

(2) $\alpha \in \left(1, \frac{1+\eta_m}{2\eta_m}\right) \cap \mathbb{S}_\mathbf{x}$ *and* $\gamma \in (\frac{2\eta_m}{1+\eta_m}, \frac{1}{\alpha}]$;

(3) $\alpha \in \left(\frac{2}{1+\eta_m}, \frac{1+\eta_m}{2\eta_m}\right) \cap \mathbb{S}_\mathbf{x}$ *and* $\gamma \in [\frac{1}{\alpha}, \frac{2-2\alpha\eta_m}{\alpha(1-\eta_m)})$;

(4) $\alpha \in \left(1, \frac{2}{1+\eta_m}\right) \cap \mathbb{S}_\mathbf{x}$ *and* $\gamma \in [1, \frac{2}{\alpha(\eta_m+1)})$.

Moreover, we have the following error bound:

$$\|\mathbf{x}_k - \mathbf{x}\|_1 \leq \tilde{\tau}_\alpha^k \|\mathbf{x}_0 - \mathbf{x}\|_1, \tag{37}$$

where $\tilde{\tau}_\alpha = \frac{\alpha\kappa_m((m-2)+|1-\gamma|(m-1))+|1-\alpha\gamma|}{1-\alpha\kappa_m}$.

Proof. By Lemma 1 and the condition $\delta_m > \frac{m-2}{m-1}$, we know that $\mathbf{x}$ is unique. For Algorithm 3, we know that $\mathbf{x}_k$ is always stochastic. By the same proof as in Theorem 1, we know that $I - \alpha\mathcal{P}\mathbf{x}_0^{m-2}$ is nonsingular. Suppose that $I - \alpha\mathcal{P}\mathbf{x}_{k-1}^{m-2}$ is nonsingular, then solving a nonsingular linear system in Step 3 of Algorithm 3 gives the unique solution $\mathbf{y}_k$, and then we derive $\mathbf{x}_k$.

The iteration in Algorithm 3 can be written as follows:

$$\left(I - \alpha\mathcal{P}\mathbf{x}_{k-1}^{m-2}\right)(\widehat{\mathbf{x}}_k - (1-\gamma)\mathbf{x}_{k-1}) = \gamma(1-\alpha)\mathbf{x}_{k-1},$$

which implies

$$\widehat{\mathbf{x}}_k = \alpha\mathcal{P}\mathbf{x}_{k-1}^{m-2}\widehat{\mathbf{x}}_k + (1-\alpha\gamma)\mathbf{x}_{k-1} - \alpha(1-\gamma)\mathcal{P}\mathbf{x}_{k-1}^{m-1}. \tag{38}$$

Let $\widehat{\mathbf{e}}_k = \widehat{\mathbf{x}}_k - \mathbf{x}$ and $\mathbf{e}_k = \mathbf{x}_k - \mathbf{x}$. By (38), we can get the following equation:

$$\widehat{\mathbf{e}}_k = \alpha\mathcal{P}\mathbf{x}_{k-1}^{m-2}\widehat{\mathbf{e}}_k + \alpha\mathcal{P}\left(\mathbf{x}_{k-1}^{m-2} - \mathbf{x}^{m-2}\right)\mathbf{x} + (1-\alpha\gamma)\mathbf{e}_{k-1} - \alpha(1-\gamma)\mathcal{P}\left(\mathbf{x}_{k-1}^{m-1} - \mathbf{x}^{m-1}\right). \tag{39}$$

By the analogous proof to Theorem 1, we get

$$\|\widehat{\mathbf{e}}_k\|_1 \leq \alpha\kappa_m\|\widehat{\mathbf{e}}_k\|_1 + \alpha\kappa_m(m-2)\|\mathbf{e}_{k-1}\|_1 + |1-\alpha\gamma|\|\mathbf{e}_{k-1}\|_1 + \alpha|1-\gamma|\eta_m\|\mathbf{e}_{k-1}\|_1. \tag{40}$$

Then,

$$\|\widehat{\mathbf{e}}_k\|_1 \leq \tilde{\tau}_\alpha\|\mathbf{e}_{k-1}\|_1. \tag{41}$$

By Lemma 4 and (41), we get

$$\|\mathbf{e}_k\|_1 \leq \tilde{\tau}_\alpha\|\mathbf{e}_{k-1}\|_1 \leq \tilde{\tau}_\alpha^k\|\mathbf{e}_0\|_1. \tag{42}$$

If α and γ satisfy one of the conditions **(1)–(4)**, it is easy to check that $1 - \alpha\kappa_m > 0$ and $0 < \tilde{\tau}_\alpha < 1$. Therefore,

$$\begin{aligned}\left\|\left(I - \alpha\mathcal{P}\mathbf{x}^{m-2}\right) - \left(I - \alpha\mathcal{P}\mathbf{x}_k^{m-2}\right)\right\|_1 &= \alpha\left\|\mathcal{P}\mathbf{x}^{m-2} - \mathcal{P}\mathbf{x}_k^{m-2}\right\|_1 \leq \alpha\kappa_m(m-2)\|\mathbf{e}_k\|_1 \\ &\leq \tilde{\tau}_\alpha^k\alpha\kappa_m(m-2)\|\mathbf{e}_0\|_1.\end{aligned}$$

Because $\mathbf{x}_0 \in \mathbb{T}_{\mathbf{x}}$, we obtain

$$\|(I - \alpha\mathcal{P}\mathbf{x}^{m-2})^{-1}\|_1\tilde{\tau}_\alpha^k\alpha\kappa_m(m-2)\|\mathbf{x} - \mathbf{x}_0\|_1 < 1.$$

By Lemma 3, we get that $I - \alpha\mathcal{P}\mathbf{x}_k^{m-2}$ is nonsingular, which ensures that $\mathbf{y}_k$ in Step 3 of Algorithm 3 is unique for each k. The error bound (37) follows from (42). The convergence of the algorithm is proved by (37) and $\widetilde{\tau}_\alpha < 1$. This completes the proof of the theorem. □

If $\alpha \in (0, 1)$, we have a simple convergence condition for Algorithm 3, which follows immediately from Theorem 3.

Corollary 2. *Under the same assumption on* $\mathcal{P}$ *as in Theorem 3, if* $\alpha \in (0, 1)$*, and* $\gamma \in (\frac{2\eta_m}{1+\eta_m}, \frac{1}{\alpha}]$*, then for any initial stochastic vector* $\boldsymbol{x}_0$*, the sequence* $\{\boldsymbol{x}_k\}$ *generated by Algorithm 3 converges to the solution* $\boldsymbol{x}$ *globally. Moreover, the error bound (37) holds.*

Remark 8. If $\gamma = 1$, Algorithm 3 is reduced to Algorithm 1.

Remark 9. Recently, Li et al.[12] gave a new uniqueness condition of the solution for the mutilinear PageRank (4). This condition can be employed to discuss the uniqueness of the solution for Equation (1). Based on this uniqueness condition, we can also give the convergence analysis for the proposed algorithms. The technique is similar to the proof of Theorems 1, 2, and 3. Therefore, we omit it.

Remark 10. Theoretically, the convergence analysis is based on the uniqueness condition of the solution. However, it is a pity that the existing uniqueness conditions in literatures are all still sufficient but not necessary, that is, Equations 1 and (4) have a unique solution but the conditions in the literature may be not available. Fortunately, the proposed algorithms may also perform well. This case will be verified by numerical experiments in Subsection 4.1.

Remark 11. If $\delta_m = 1$, the conditions **(1)–(4)** in Theorem 3 reduce to

(1) $\alpha \in (0, 1)$ and $\gamma \in (0, \frac{2}{\alpha})$;
(2) $\alpha \in (1, +\infty) \cap \mathbb{S}_{\mathbf{x}}$ and $\gamma \in (0, \frac{1}{\alpha}]$;
(3) $\alpha \in (2, +\infty) \cap \mathbb{S}_{\mathbf{x}}$ and $\gamma \in [\frac{1}{\alpha}, \frac{2}{\alpha})$;
(4) $\alpha \in (1, 2) \cap \mathbb{S}_{\mathbf{x}}$ and $\gamma \in [1, \frac{2}{\alpha})$.

Then, we have $0 < \alpha\gamma < 2$ and $\widetilde{\tau}_\alpha = |1 - \alpha\gamma| < 1$.

3.2 | Computable convergent conditions for Algorithms 1 and 3

Although we give some convergence conditions for Algorithms 1 and 3, it is difficult to check because the solution $\mathbf{x}$ is contained in those conditions. In this section, we will explore some computable conditions.

Firstly, we list a definition (see the work of Lim[13]) and some lemmas.

Definition 1. An $\mathcal{A} \in \mathbb{C}^{[m,n]}$ is called reducible if there exists a nonempty proper index subset $\mathbb{I} \subsetneq \{1, 2, \dots, n\}$ such that

$$a_{i_1 i_2 \cdots i_m} = 0, \forall i_1 \in \mathbb{I},\ \forall i_2, \dots, i_m \notin \mathbb{I}.$$

If $\mathcal{A}$ is not reducible, then we call $\mathcal{A}$ irreducible.

If $\mathbf{v}$ in (4) is a positive stochastic tensor, $\mathcal{Q}_\theta$ is always a positive stochastic tensor and irreducible. Therefore, we consider the convergence analysis for the proposed algorithms in the case that $\mathcal{P}$ is irreducible in Equation (1).

Lemma 5. *Let $A \in \mathbb{R}^{[2,n]}$ be a strictly diagonally dominant matrix. Then,*

$$\|A^{-1}\|_1 \leq \max_{i \in \langle n \rangle} \left\{ \frac{1}{|a_{ii}| - \Lambda_i(A)} \right\}.$$

where $\Lambda_i(A) = \sum_{j \in \langle n \rangle \setminus \{i\}} |a_{ji}|$.

Proof. The proof can be found in the work of Varga.[14] □

By Theorem 2.2 in the work of Li et al.[3] and Lemma 1, we have the following lemma.

Lemma 6. *If a stochastic tensor $\mathcal{P}$ with $\delta_m > \frac{m-2}{m-1}$ is irreducible, Equation (1) has a unique positive solution.*

For an irreducible stochastic tensor, we have

Lemma 7. *If $\mathcal{A} = (a_{i_1 i_2 \cdots i_m}) \in \mathbb{R}^{[m,n]}$ is an irreducible stochastic tensor, then $a_{ii \cdots i} < 1, \forall i \in \langle n \rangle$.*

Proof. Assume that there exists $j \in \langle n \rangle$ such that $a_{jj \cdots j} = 1$. Because $\mathcal{A}$ is stochastic, we get that $a_{ij \cdots j} = 0\ \forall i \in \langle n \rangle \setminus \{j\}$, which contradicts to the irreducibility of $\mathcal{P}$. This proves the lemma. □

Let $\mathcal{P} = (p_{i_1 \dots i_m}) \in \mathbb{R}^{[m,n]}$. For given $i \in \langle n \rangle$, we denote the sub-tensor $\mathcal{P}_i = (p_{i i_2 \dots i_m}) \in \mathbb{R}^{[m-1,n]}$. By $p_{i \dots i_{t_1} \dots i_{t_2} \dots i_{t_k} \dots i}$ we denote the entry of $\mathcal{P}_i$ whose $t_1, \dots, t_k$ indices are $i_{t_1}, \dots, i_{t_k}$, respectively, and other indices are all equal to i. For $k = 1, 2, \dots, m-1$,let

$$p_{i \cdots j_{t_1} \cdots j_{t_2} \cdots j_{t_k} \cdots i} = \max\{p_{i \cdots i_{t_1} \cdots i_{t_2} \cdots i_{t_k} \cdots i} | i_{t_s} \in \langle n \rangle \setminus \{i\}, s = 1, \dots, k\}.$$

By $\Omega_k^i = \{(i\cdots j_{t_1}\cdots j_{t_2}\cdots j_{t_k}\cdots i), t_1, \ldots, t_k \in \{2, \ldots, m\}\}$. Then, the number of elements in the set Ω_k^i is C_{m-1}^k. With the above notation we set

$$f_1(t) = -t + \sum_{j=0}^{m-1} a_j t^{m-1-j}(1-t)^j,$$

where

$$a_0 = \max_i p_{ii\cdots i},$$
$$a_k = \max_i \sum_{(i\cdots j_{t_1}\cdots j_{t_2}\cdots j_{t_k}\cdots i)\in\Omega_k^i} p_{i\cdots j_{t_1}\cdots j_{t_2}\cdots j_{t_k}\cdots i}, 1 \le k \le m-1,$$

Remark 12. By Lemma 7, we have that $a_0 < 1$. Assume that $a_{m-1} = 0$, by the definition of a_{m-1}, for a given $j \in \langle n\rangle$ and $\forall i_2, \ldots, i_m \in \langle m\rangle \setminus \{j\}$, we have $a_{ji_2\cdots i_m} = 0$, which contradicts to the irreducibility of $\mathcal{P}$. Hence, $a_{m-1} > 0$. By Lemma 7, it is easy to check that $f_1(0) = a_{m-1} > 0$ and $f_1(1) = a_0 - 1 < 0$. Thus, $f_1(t)$ has at least a root in $(0, 1)$.

Lemma 8. *Let $\mathcal{P}$ be an irreducible stochastic tensor. For any positive solution $\mathbf{x} = (x_i)$ of (1), we obtain*

$$x_i \le x_{ub} < 1, \tag{43}$$

where x_{ub} is the maximum root of the polynomial $f_1(t)$ in $(0, 1)$.

Proof. Because $\mathcal{P}$ is irreducible, by Theorem 2.2 in the work of Li et al.,[3] there is at least one positive solution $\mathbf{x}$ of Equation (1).

Taking $l = \arg\max_i x_i$, we get

$$\begin{aligned}
x_l &= \sum_{i_2,i_3,\ldots,i_m\in\langle n\rangle} p_{li_2i_3\ldots i_m} x_{i_2i_3\ldots i_m} \\
&= p_{ll\cdots l}x_l^{m-1} + \sum_{t_1\in\langle m\rangle\setminus\{1\}, i_{t_1}\in\langle n\rangle\setminus\{l\}} p_{l\cdots i_{t_1}\cdots l}x_l^{m-2}x_{i_{t_1}} + \ldots + \sum_{\substack{t_1,\ldots,t_k\in\langle m\rangle\setminus\{1\}\\ i_{t_1},\ldots,i_{t_k}\in\langle n\rangle\setminus\{l\}}} p_{l\cdots i_{t_1}\cdots i_{t_k}\cdots l}x_l^{m-1-k}x_{i_{t_1}}\cdots x_{i_{t_k}} \\
&\quad + \ldots + \sum_{j\in\langle m\rangle\setminus\{1\}, i_j\in\langle n\rangle\setminus\{l\}} p_{li_1\cdots i_m}x_{i_2}\cdots x_{i_m} \\
&\le \max_i p_{ii\cdots i}x_l^{m-1} + \max_i \sum_{(i\cdots j_{t_1}\cdots i)\in\Omega_1^i} p_{i\cdots j_{t_1}\cdots i}x_l^{m-2} \sum_{i_{t_1}\in\langle n\rangle\setminus\{l\}} x_{i_{t_1}} + \cdots \\
&\quad + \max_i \sum_{(i\cdots j_{t_1}\cdots j_{t_2}\cdots j_{t_k}\cdots i)\in\Omega_k^i} p_{i\cdots j_{t_1}\cdots j_{t_2}\cdots j_{t_k}\cdots i}x_l^{m-1-k} \sum_{i_{t_1}\in\langle n\rangle\setminus\{l\}} x_{i_{t_1}}\cdots \sum_{i_{t_k}\in\langle n\rangle\setminus\{l\}} x_{i_{t_k}} \\
&\quad + \ldots + \max_i p_{ij_1\cdots j_{m-1}} \sum_{i_2\in\langle n\rangle\setminus\{l\}} x_{i_2}\cdots \sum_{i_m\in\langle n\rangle\setminus\{l\}} x_{i_m}
\end{aligned} \tag{44}$$

$$\le \sum_{j=0}^{m-1} a_j x_l^{m-1-j}(1-x_l)^j, \tag{45}$$

which implies that $f_1(x_l) \ge 0$. By Remark 12, we have $f_1(1) < 0$ and $f_1(0) > 0$, which together with the expression of $f_1(t)$ gives $x_l \le x_{ub}$. □

It is difficult to compute the roots of $f_1(t)$ for a large m. Therefore, we give a simple bound of x_i.

Corollary 3. *Let $\mathcal{P}$ be an irreducible stochastic tensor. For any positive solution $\mathbf{x} = (x_i)$ of (1), we obtain*

$$x_i \le x'_{ub}, \tag{46}$$

where x'_{ub} is the positive root of the polynomial $f_2(t) = (a_0 + a_{m-1} - \sum_{j=1}^{m-2} a_j)t^2 + (\sum_{j=1}^{m-2} a_j - 2a_{m-1} - 1)t + a_{m-1}$.

Proof. By (45) and $0 < x_l < 1$, we have

$$x_l \leq a_0 x_l^2 + \sum_{j=1}^{m-2} a_j (1-x_l) x_l + a_{m-1}(1-x_l)^2$$
$$= \left(a_0 + a_{m-1} - \sum_{j=1}^{m-2} a_j \right) x_l^2 + \left(\sum_{j=1}^{m-2} a_j - 2a_{m-1} \right) x_l + a_{m-1}, \tag{47}$$

which implies that $f_2(x_l) \geq 0$. By Remark 12, we have $f_2(1) < 0$ and $f_2(0) > 0$, which together with the expression of $f_2(t)$ gives $x_l \leq x'_{ub}$. □

Let

$$p = \max_i \sum_{j \in \langle n \rangle \setminus \{i\}} \sum_{i_3, \ldots, i_m \in \langle n \rangle} p_{jii_3 \cdots i_m} x_{ub}^{m-2},$$

where x_{ub} is the upper bounds of x_l given in (43) or (46). If $\mathcal{P}$ is an irreducible stochastic tensor, by Lemma 6, it is easy to check $p > 0$.

In the case of $p < 1/2$, we let

$$\mathbb{S}_p = (0, 1) \cup \left(\frac{1}{1-2p}, \infty \right),$$

and

$$\mathbb{T}'_{\mathbf{x}} \equiv \left\{ \widetilde{\mathbf{x}} : \|\widetilde{\mathbf{x}} - \mathbf{x}\|_1 < \mu'_1 \right\}, \tag{48}$$

where $\mu'_1 = \frac{1}{\alpha \kappa_m (m-2) \xi}$ and $\xi = \frac{1}{\alpha(1-2p)-1}$.

Then, we give the following convergence results for Algorithms 1 and 3.

Theorem 4. *Let $\mathcal{P}$ be an irreducible stochastic tensor with $p < 1/2$, $\delta_m > \frac{m-2}{m-1}$ and $\alpha \in \left(0, \frac{2}{1+\eta_m}\right) \cap \mathbb{S}_p$. Then Algorithm 1 converges to the solution $\mathbf{x}$ of (1) for an arbitrary initial stochastic vector $\mathbf{x}_0 \in \mathbb{T}'_{\mathbf{x}}$ with the error bound (18).*

Proof. Because $\mathcal{P}$ is an irreducible stochastic tensor with $\delta_m > \frac{m-2}{m-1}$, $\mathbf{x}$ is the unique positive solution of Equation (1). For any $i \in \langle n \rangle$, because $p_{iii_3 \cdots i_m} = 1 - \sum_{j \in \langle n \rangle \setminus \{i\}} p_{jii_3 \cdots i_m} x_{i_3} \cdots x_{i_m}$ the diagonal entry of $I - \alpha \mathcal{P} \mathbf{x}^{m-2}$ is

$$1 - \alpha \sum_{i_3 \cdots i_m \in \langle n \rangle}^{n} p_{iii_3 \cdots i_m} x_{i_3} \cdots x_{i_m} = 1 - \alpha \sum_{i_3, \ldots, i_m \in \langle n \rangle} \left(1 - \sum_{j \in \langle n \rangle \setminus \{i\}} p_{jii_3 \cdots i_m}\right) x_{i_3} \cdots x_{i_m}$$
$$= 1 - \alpha + \alpha \sum_{j \in \langle n \rangle \setminus \{i\}} \sum_{i_3, \ldots, i_m \in \langle n \rangle} p_{jii_3 \cdots i_m} x_{i_3} \cdots x_{i_m}.$$

For $\alpha \in (0, 1)$, it is easy to check that $I - \alpha \mathcal{P} \mathbf{x}^{m-2}$ is a strictly diagonally dominant matrix. By Corollary 2, Algorithm 1 converges globally. Therefore, we only consider the case of $\alpha > 1$.

Let $S_i^{\mathbf{x}} = \sum_{j \in \langle n \rangle \setminus \{i\}} \sum_{i_3, \ldots, i_m \in \langle n \rangle} p_{jii_3 \cdots i_m} x_{i_3} \cdots x_{i_m}$. Because $\mathcal{P}$ is an irreducible stochastic tensor, by Lemma 6, it is easy to check $S_i^{\mathbf{x}} > 0$. For $\alpha \in (0, \frac{2}{1+\eta_m}) \cap \mathbb{S}_p$ and $p < 1/2$, we have $S_i^{\mathbf{x}} \leq p$ and $\alpha > \frac{1}{1-2p} \geq \frac{1}{1-2S_i^{\mathbf{x}}}$. Thus, it is easy to check that

$$|1 - \alpha + \alpha S_i^{\mathbf{x}}| > \alpha \sum_{j \in \langle n \rangle \setminus \{i\}} \left| \sum_{i_3, \ldots, i_m \in \langle n \rangle} p_{jii_3 \cdots i_m} x_{i_3} \cdots x_{i_m} \right| = \alpha \sum_{j \in \langle n \rangle \setminus \{i\}} \sum_{i_3, \ldots, i_m \in \langle n \rangle} p_{jii_3 \cdots i_m} x_{i_3} \cdots x_{i_m}$$
$$= \alpha S_i^{\mathbf{x}}.$$

Therefore, $I - \alpha \mathcal{P} \mathbf{x}^{m-2}$ is a strictly diagonally dominant matrix. Furthermore, by Lemma 5, we obtain

$$\|(I - \alpha \mathcal{P} \mathbf{x}^{m-2})^{-1}\|_1 < \max_{i \in \langle n \rangle} \left\{ \frac{1}{\left|1 - \alpha + \alpha S_i^{\mathbf{x}}\right| - \alpha S_i^{\mathbf{x}}} \right\} \leq \max_{i \in \langle n \rangle} \left\{ \frac{1}{-\left(1 - \alpha + \alpha S_i^{\mathbf{x}}\right) - \alpha S_i^{\mathbf{x}}} \right\} \leq \frac{1}{\alpha(1-2p)-1}.$$

Because $\mathbf{x}_0 \in \mathbb{T}'_{\mathbf{x}}$, we have

$$\alpha\kappa_m(m-2)\|(I-\alpha\mathcal{P}\mathbf{x}^{m-2})^{-1}\|_1\|\mathbf{x}-\mathbf{x}_0\|_1 < 1.$$

It follows from Lemma 3 that $I-\alpha\mathcal{P}\mathbf{x}_0^{m-2}$ is nonsingular. By the similar proof to Theorem 1, we know that $I-\alpha\mathcal{P}\mathbf{x}_k^{m-2}$ is nonsingular for every iterative step, and the error bound (18) can be obtained by an analogous technique as in Theorem 1. This proves the theorem. □

By an analogous technique as in Theorem 4, we can give the following convergence result for Algorithm 3.

Theorem 5. *Let $\mathcal{P}$ be an irreducible stochastic tensor with $p < 1/2$ and $\delta_m > \frac{m-2}{m-1}$. Then, for an arbitrary initial stochastic vector $\mathbf{x}_0 \in \mathbb{T}'_{\mathbf{x}}$, the sequence $\{\mathbf{x}_k\}$ generated by Algorithm 3 converges to the solution $\mathbf{x}$ if the parameters α and γ satisfy one of the following conditions:*

(1) $\alpha \in (0,1)$ *and* $\gamma \in \left(\frac{2\eta_m}{1+\eta_m}, \frac{2}{\alpha(\eta_m+1)}\right)$;

(2) $\alpha \in \left(1, \frac{1+\eta_m}{2\eta_m}\right) \cap \mathbb{S}_p$ *and* $\gamma \in (\frac{2\eta_m}{1+\eta_m}, \frac{1}{\alpha}]$;

(3) $\alpha \in \left(\frac{2}{1+\eta_m}, \frac{1+\eta_m}{2\eta_m}\right) \cap \mathbb{S}_p$ *and* $\gamma \in [\frac{1}{\alpha}, \frac{2-2\alpha\eta_m}{\alpha(1-\eta_m)})$;

(4) $\alpha \in \left(1, \frac{2}{1+\eta_m}\right) \cap \mathbb{S}_p$ *and* $\gamma \in [1, \frac{2}{\alpha(\eta_m+1)})$.

Furthermore, we have the error bound (37).

Note that $\|\mathbf{x}-\mathbf{y}\|_1 \le 2$ for any stochastic vectors $\mathbf{x}$ and $\mathbf{y}$. Therefore, for the corresponding α (*or* γ) in Theorems 4 and 5, we have

Corollary 4. *Let $\mathcal{P}$ be an irreducible stochastic tensor. If $p < 1/2$, $\delta_m > \max\left\{\frac{m-2}{m-1}, 1-\frac{1}{2\alpha\xi(m-2)}\right\}$, Algorithms 1 and 3 converge globally.*

Remark 13. It is noted that the condition $p < 1/2$ is computable, which guarantees that $I-\alpha\mathcal{P}\mathbf{x}^{m-2}$ is nonsingular for $\alpha \in \mathbb{S}_p$. For example, for an order-3 dimension 2 stochastic tensor $\mathcal{P}$

$$\mathcal{P}(:,:,1) = \begin{pmatrix} 0.4938 & 0.1268 \\ 0.5062 & 0.8732 \end{pmatrix},$$
$$\mathcal{P}(:,:,2) = \begin{pmatrix} 0.9114 & 0.2113 \\ 0.0886 & 0.7887 \end{pmatrix},$$

we compute $p = 0.3696 < 1/2$ and $x_{ub} = 0.6214$ by solving the root of $f_1(t)$.

3.3 | The choice of the parameters in Algorithm 3

In this subsection, we discuss how to choose the parameters α and γ given in Algorithm 3 so that Algorithm 3 performs better than Algorithm 1. By this purpose, we consider τ_α in Theorem 1 and $\tilde{\tau}_\alpha$ in Theorem 3, and hope to choose parameters α and γ so that $\tau_\alpha - \tilde{\tau}_\alpha \le 0$, that is,

$$\tilde{\tau}_\alpha - \tau_\alpha = \frac{\alpha|1-\gamma|\eta_m + |1-\alpha\gamma| - |1-\alpha|}{1-\alpha\kappa_m} \le 0. \tag{49}$$

Hence, we have

(1) $\alpha \in (0,1)$ and $\gamma \in \left[1, \frac{2-\alpha+\alpha\eta_m}{\alpha(\eta_m+1)}\right]$.

(2) $\alpha \in \left(1, \frac{1+\eta_m}{2\eta_m}\right) \cap \mathbb{S}_p$ and $\gamma \in \left[\frac{2-\alpha+\alpha\eta_m}{\alpha(1+\eta_m)}, \frac{1}{\alpha}\right]$.

(3) $\alpha \in \left(\frac{2}{1+\eta_m}, \frac{1+\eta_m}{2\eta_m}\right) \cap \mathbb{S}_p$ and $\gamma \in [\frac{1}{\alpha}, \frac{2-2\alpha\eta_m}{\alpha(1-\eta_m)})$.

That is to say that Algorithm 3 may perform better than Algorithm 1 if the parameters α and γ satisfy one of the above conditions **(1)–(3)**. Furthermore, for a given α, satisfying one of the above conditions and the corresponding γ, it is easy to check that

$$\arg\min_{\gamma}(\tilde{\tau}_\alpha - \tau_\alpha) = \frac{1}{\alpha}.$$

Therefore, for a given α, Algorithm 3 may perform best when we take $\gamma = \frac{1}{\alpha}$ comparing Algorithm 1.

Remark 14. Although we cannot prove that Algorithms 1 and 3 are convergent globally for $\alpha > 1$ theoretically, in Section 4, we may choose the initial vector $\mathbf{x}_0 \notin \mathbb{T}_{\mathbf{x}}$ in our numerical experiments.

3.4 | The modified algorithm by the semi-symmetrization technique

We denote $p_{i_1 i_2 \cdots i_m}$ as $a_{i_1\omega}$, where $\omega = i_2 \cdots i_m$. Notice that Equations 1 are equivalent to the following equations:

$$\mathbf{x} = \overline{\mathcal{P}}\mathbf{x}^{m-1}, \|\mathbf{x}\|_1 = 1, \tag{50}$$

where $\overline{\mathcal{P}} = (\bar{p}_{i_1 i_2 \cdots i_m})$ is given by

$$\bar{p}_{i i_2 \cdots i_m} = \frac{1}{(m-1)!}\sum_{\omega \in \mathbb{G}} p_{i\omega}, \tag{51}$$

and $\mathbb{G}$ denotes the set of all permutations in $(i_2 \cdots i_m)$. Then, $\overline{\mathcal{P}}$ is symmetric on the last $m-1$ indices (In the work of Ni et al.,[15] $\overline{\mathcal{P}}$ is called semi-symmetric). It is easy to check that if $\mathcal{P}$ is a stochastic tensor, so is $\overline{\mathcal{P}}$. Thus, we can modify Algorithm 3 into the following Algorithm 4, which is called the modification of Algorithm 3.

Algorithm 4 The modified algorithm 3

1: Given a stochastic tensor $\mathcal{P}$, α, γ, maximum $k_{\mathbf{max}}$, termination tolerance ε and an initial stochastic vector $\mathbf{x}_0$. Initialize $k = 1$.
2: Semi-symmetrize the tensor $\mathcal{P}$ into $\overline{\mathcal{P}}$ by (51).
3: while $k < k_{\mathbf{max}}$.
4: $(I - \alpha\overline{\mathcal{P}}\mathbf{x}_{k-1}^{m-2})\mathbf{y}_k = (1-\alpha)\mathbf{x}_{k-1}, \hat{\mathbf{x}}_k = \gamma\mathbf{y}_k + (1-\gamma)\mathbf{x}_{k-1}, \mathbf{x}_k = \text{proj}(\hat{\mathbf{x}}_k)$.
5: If $\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_1 < \varepsilon$, break and output $\mathbf{x}_k$.
6: $k = k + 1$, back to 2).

Remark 15. Here, we explain why we use the semi-symmetrization technique given in Algorithm 4. Let $F_{\overline{\mathcal{P}}}(\mathbf{x}) = \mathbf{x} - \overline{\mathcal{P}}\mathbf{x}^{m-1}$. Note that the Jacobian matrix of $F_{\overline{\mathcal{P}}}(\mathbf{x})$ is $JF_{\overline{\mathcal{P}}}(\mathbf{x}) = I - (m-1)\overline{\mathcal{P}}\mathbf{x}^{m-2}$, Projected Newton's method (e.g., see the work of Bertsekas[10]) is given as follows:

$$\left(I - (m-1)\overline{\mathcal{P}}\mathbf{x}_{k-1}^{m-2}\right)\hat{\mathbf{x}}_k = -(m-2)\overline{\mathcal{P}}\mathbf{x}_{k-1}^{m-1}, \mathbf{x}_k = \text{proj}(\hat{\mathbf{x}}_k). \tag{52}$$

Taking $\alpha = m-1$ and $\gamma = \frac{1}{m-1}$, then Algorithm 4 may reduce to Projected Newton's method (52).

We also consider directly the Newton method for solving Equation (1) without projection and semi-symmetrization. Let $F_{\mathcal{P}}(\mathbf{x}) = \mathbf{x} - \mathcal{P}\mathbf{x}^{m-1}$. The product $\mathcal{P}\mathbf{x}^{m-1}$ can be transformed into the following matrix-vector product (see other works.[6,16]):

$$\mathcal{P}\mathbf{x}^{m-1} = \mathcal{P}_{(1)}(\underbrace{\mathbf{x} \otimes \cdots \otimes \mathbf{x}}_{m-1}), \tag{53}$$

where $\mathcal{P}_{(1)}$ is the matrices obtained from $\mathcal{P}$ flattened along the first index (see other works[16,17]) and $\otimes$ is the Kronecker product. The Jacobian matrix $JF_{\mathcal{P}}(\mathbf{x}) = I - \mathcal{P}_{(1)}(\mathbf{x} \otimes I \otimes \cdots \otimes \mathbf{x} + I \otimes \mathbf{x} \otimes \cdots \otimes \mathbf{x} + \cdots + \mathbf{x} \otimes \mathbf{x} \otimes \cdots \otimes I)$. Thus, Newton's method for solving Equation (1) is given:

$$(I - JF_{\mathcal{P}}(\mathbf{x}_{k-1}))\mathbf{x}_k = -(m-2)\mathcal{P}\mathbf{x}_{k-1}^{m-1}, \tag{54}$$

if $\mathcal{P}$ is semi-symmetric, $JF_{\mathcal{P}}(\mathbf{x}) = JF_{\overline{\mathcal{P}}}(\mathbf{x})$. Therefore, for every iterative step, it is not expensive because there is only one product between the tensor $\mathcal{P}$ and vector $\mathbf{x}$ to be calculated. However, as we know that semi-symmetrizing a tensor is expensive if the order m and the dimension n are very large. In the next section, we will give numerical examples to show that Algorithm 4 performs well for some small examples from practical applications.

Remark 16. We consider solving Equations (4) and (5) by Newton's method, respectively. Assume that $\hat{\mathcal{P}}$ is a semi-symmetric stochastic tensor. Let $F_1(\mathbf{x}) = x - \theta\hat{\mathcal{P}}\mathbf{x}^{m-1} - (1-\theta)\mathbf{v}$ and $F_2(\mathbf{x}) = x - (\theta\hat{\mathcal{P}} + (1-\theta)\mathcal{V})\mathbf{x}^{m-1}$. The Jacobian matrices of $F_1(\mathbf{x})$ and $F_2(\mathbf{x})$ are $JF_1(\mathbf{x}) = I - \theta(m-1)\hat{\mathcal{P}}\mathbf{x}^{m-2}$ and $JF_2(\mathbf{x}) = I - (m-1)(\theta\hat{\mathcal{P}} + (1-\theta)\mathcal{V})\mathbf{x}^{m-2}$, respectively. Note that $JF_2(\mathbf{x}) = JF_1(\mathbf{x}) - (m-1)(1-\theta)\mathcal{V}\mathbf{x}^{m-2} = JF_1(\mathbf{x}) - (m-1)(1-\theta)\mathbf{v}\mathbf{e}^T$, where $\cdot^T$ is the transposed operations. Because $\mathbf{v}\mathbf{e}^T$ is nonzero, $JF_1(\mathbf{x}) \neq JF_2(\mathbf{x})$. Therefore, the reduced Newton's method (52) for solving Equation (5) is totally different from the one in the work of Gleich et al.[6] for solving Equation (4). In the next section, Tables 12–17 show that Algorithm 4 may perform better than the Newton's method given in the work of Gleich et al.[6]

4 | NUMERICAL EXAMPLES

In this section, we will give numerical experiments to check the proposed algorithms. All tests were done in MATLAB R2014a with a desktop computer (Dell OptiPlex 3020) having the following configuration: Intel(R) Core(TM)i7-2600 CPU 3.40 GHz and 16.00G RAM. Three aspects are given to check the efficiency of the proposed algorithms: the number of iteration steps (denoted by IT), the CPU time in seconds (denoted by CPU), the convergence rate (denote by 'r') defined by

$$r = \frac{\|\mathbf{x}_N - \mathbf{x}_{N-1}\|_1}{\|\mathbf{x}_{N-1} - \mathbf{x}_{N-2}\|_1},$$

where N is the maximum iterative step. For Algorithms 1–3, the nonsingular linear equation for every iterative step k is solved by *mldivide* function of MATLAB. Besides, the products (2) and (6) are computed by the *ttv* function of the package Tensor Toolbox 2.6 (see the work of Bader et al.[18]) for all the experiments. The notation '-' in the tables for recording experimental results refers to the case that the corresponding algorithm is still not convergent when the number of iteration steps reaches the maximum N.

4.1 | Numerical experiments for the tensor equation (1)

We consider the following practical examples (i)–(v) given in the work of Raftery[19] (also see the work of Li et al.[3]). The first three examples come from DNA sequence data in the works of Raftery et al.[20] and Berchtold et al.[21] The fourth one is on interpersonal relationship data. Example (v) comes from occupational mobility of physicists data. In the work of Li et al.,[3] we know that Equation (1) has a unique positive solution for these examples. The value of δ_3 for Examples (i)–(v) are reported in Table 1. It is clear that $\delta_3 \geq 1/2$ for Examples (i)–(iii) but $\delta_3 \leq 1/2$ for Examples (iv) and (v). The corresponding solution of these examples is unique (see the work of Li et al.[3]). We compare our algorithms with the fixed-point method proposed in the work of Li et al.[3] We take that the maximum iterative number is

TABLE 1 The value of δ_m for Examples (i)–(v)

Example	i	ii	iii	iv	v
δ_3	0.7300	0.6472	0.5742	0.4150	0.1710

1,000, the termination tolerance $\varepsilon = 10^{-10}$, and take initial value $\mathbf{x}_0 = \frac{1}{n}\mathbf{e}$, where $\mathbf{e}$ is an n dimension vector with all entries being 1.

(i)

$$\mathcal{P}(:,:,1)=\begin{pmatrix}0.6000 & 0.4083 & 0.4935\\ 0.2000 & 0.2568 & 0.2426\\ 0.2000 & 0.3349 & 0.2639\end{pmatrix},\quad \mathcal{P}(:,:,2)=\begin{pmatrix}0.5217 & 0.3300 & 0.4152\\ 0.2232 & 0.2800 & 0.2658\\ 0.2551 & 0.3900 & 0.3190\end{pmatrix},$$

$$\mathcal{P}(:,:,3)=\begin{pmatrix}0.5565 & 0.3648 & 0.4500\\ 0.2174 & 0.2742 & 0.2600\\ 0.2261 & 0.3610 & 0.2900\end{pmatrix};$$

(ii)

$$\mathcal{P}(:,:,1)=\begin{pmatrix}0.5200 & 0.2986 & 0.4462\\ 0.2700 & 0.3930 & 0.3192\\ 0.2100 & 0.3084 & 0.2346\end{pmatrix},\mathcal{P}(:,:,2)=\begin{pmatrix}0.6514 & 0.4300 & 0.5766\\ 0.1970 & 0.3200 & 0.2462\\ 0.1516 & 0.2500 & 0.1762\end{pmatrix},$$

$$\mathcal{P}(:,:,3)=\begin{pmatrix}0.5638 & 0.3424 & 0.4900\\ 0.2408 & 0.3638 & 0.2900\\ 0.1954 & 0.2938 & 0.2200\end{pmatrix};$$

(iii)

$$\mathcal{P}(:,:,1)=\begin{pmatrix}0.2091 & 0.2834 & 0.2194 & 0.1830\\ 0.3371 & 0.3997 & 0.3219 & 0.3377\\ 0.3265 & 0.0560 & 0.3119 & 0.2961\\ 0.1723 & 0.2608 & 0.1468 & 0.1832\end{pmatrix},\mathcal{P}(:,:,2)=\begin{pmatrix}0.1952 & 0.2695 & 0.2055 & 0.1690\\ 0.3336 & 0.3962 & 0.3184 & 0.3342\\ 0.2954 & 0.0249 & 0.2808 & 0.2650\\ 0.1758 & 0.3094 & 0.1953 & 0.2318\end{pmatrix},$$

$$\mathcal{P}(:,:,3)=\begin{pmatrix}0.3145 & 0.3887 & 0.3248 & 0.2883\\ 0.0603 & 0.1203 & 0.0451 & 0.0609\\ 0.3960 & 0.1255 & 0.3814 & 0.3656\\ 0.2293 & 0.3628 & 0.2487 & 0.2852\end{pmatrix},\mathcal{P}(:,:,4)=\begin{pmatrix}0.1685 & 0.2429 & 0.1789 & 0.1425\\ 0.3553 & 0.4180 & 0.3402 & 0.3559\\ 0.3189 & 0.0484 & 0.3043 & 0.2885\\ 0.1571 & 0.2907 & 0.1766 & 0.2131\end{pmatrix};$$

(iv)

$$\mathcal{P}(:,:,1)=\begin{pmatrix}0.5810 & 0.2432 & 0.1429\\ 0 & 0.4109 & 0.0701\\ 0.4190 & 0.3459 & 0.7870\end{pmatrix},\mathcal{P}(:,:,2)=\begin{pmatrix}0.4708 & 0.1330 & 0.0327\\ 0.1341 & 0.5450 & 0.2042\\ 0.3951 & 0.3220 & 0.7631\end{pmatrix},$$

$$\mathcal{P}(:,:,3)=\begin{pmatrix}0.4381 & 0.1003 & 0\\ 0.0229 & 0.4338 & 0.0930\\ 0.5390 & 0.4659 & 0.9070\end{pmatrix};$$

(v)

$$\mathcal{P}(:,:,1)=\begin{pmatrix}0.9000 & 0.3340 & 0.3106\\ 0.0690 & 0.6108 & 0.0754\\ 0.0310 & 0.0552 & 0.6140\end{pmatrix},\mathcal{P}(:,:,2)=\begin{pmatrix}0.6700 & 0.1040 & 0.0805\\ 0.2892 & 0.8310 & 0.2956\\ 0.0408 & 0.0650 & 0.6239\end{pmatrix},$$

$$\mathcal{P}(:,:,3)=\begin{pmatrix}0.6604 & 0.0945 & 0.0710\\ 0.0716 & 0.6133 & 0.0780\\ 0.2680 & 0.2922 & 0.8501\end{pmatrix};$$

To show the convergence theory applied in practice, in the following remark, we take Example (i) to report the convergence range of parameters in Theorem 3.

Remark 17. Because the complementary set of $\mathbb{S}_{\mathbf{x}}$ is a finite set, the sets of α in **(1)–(4)** of Theorem 3 are always nonempty. For given α, the value of the right interval point for γ is always strictly larger than the one of left interval point if $\delta_m > \frac{m-2}{m-1}$, that is, the corresponding interval of γ is nonempty. For Example (i), it is easy to compute that $\delta_m = 0.7300$ and the convergence range **(1)–(4)** given in Theorem 3 as follows:

(1) $\alpha \in (0, 1)$ and $\gamma \in \left(0, \frac{1.2987}{\alpha}\right)$;
(2) $\alpha \in (1, 1.4259) \cap \mathbb{S}_{\mathbf{x}}$ and $\gamma \in (0.7013, \frac{1}{\alpha}]$;
(3) $\alpha \in (1.2987, 1.4259) \cap \mathbb{S}_{\mathbf{x}}$ and $\gamma \in [\frac{1}{\alpha}, \frac{100-54\alpha}{23\alpha})$;
(4) $\alpha \in (1, 1.2987) \cap \mathbb{S}_{\mathbf{x}}$ and $\gamma \in [1, \frac{1.2987}{\alpha})$.

We also plot Figures 1 and 2 to show the relationship between α and γ. Here, the green area denotes the range of α and γ. The values of γ are taken as the base-10 logarithm for Figure 1.

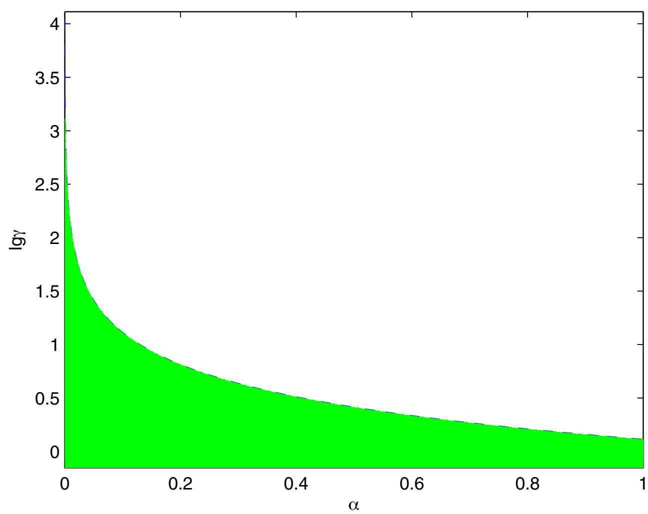

FIGURE 1 The range of γ and α when $\alpha \in (0, 1)$

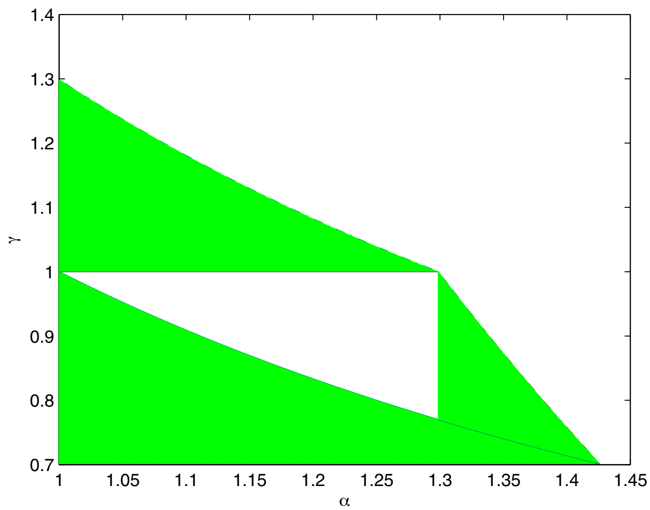

FIGURE 2 The range of γ and α when $\alpha > 1$

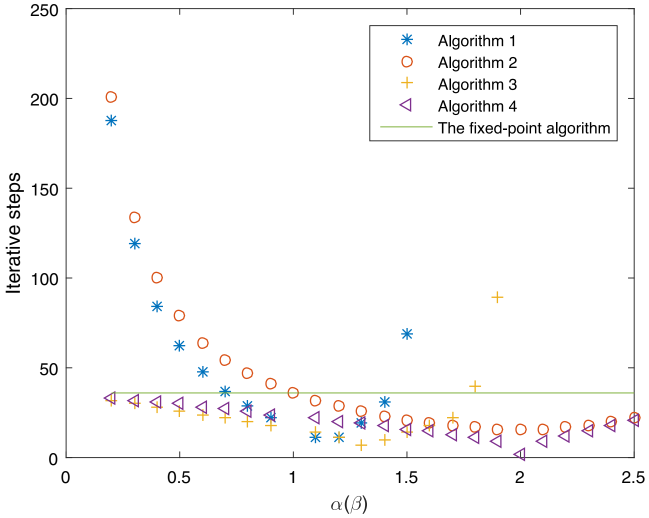

FIGURE 3 The relationship between iterative steps and $\alpha(\beta)$ by Example (iv)

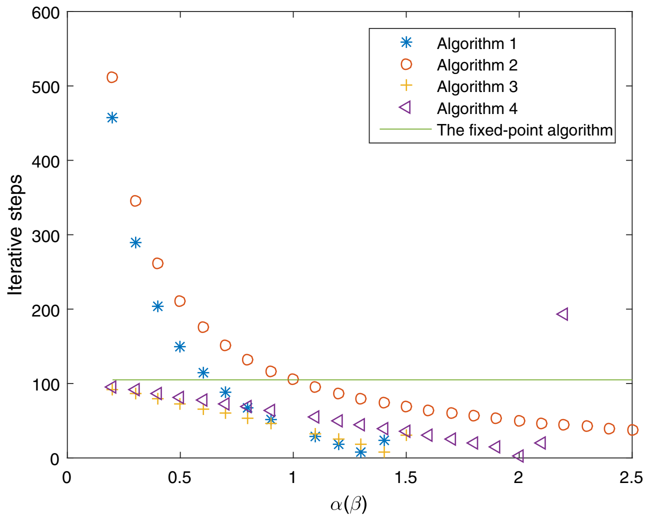

FIGURE 4 The relationship between iterative steps and $\alpha(\beta)$ by Example (v)

Firstly, taking $\gamma = \frac{1}{\alpha}$ in Algorithms 3 and 4, and α (or β) are from 0.2 to 2 in the interval of 0.2, we test the proposed algorithms by Examples (i)–(v). The numerical results are reported in Tables 2–7, in which it can be seen that the proposed algorithms are faster than the existing methods in the works of Li et al.[3] and Liu et al.[8] if the parameter is taken suitably. In particular, Algorithm 4 performs well when $\alpha = 2$ for $m = 3$. Furthermore, we explore the parameter's influence on the proposed algorithms by Examples (iv) and (v). We plot Figures 3 and 4 with the following manner: the corresponding algorithm diverges if there is no point for given $\alpha(\beta)$. It is seen that the Algorithm 4 has larger convergence domain for α than ones in Algorithm 3. Actually, this phenomenon is also verified by examples produced randomly by MATLAB.

Remark 18. From Tables 2–6, one may know that Algorithms 1, 3, and 4 can perform better than the fixed-point algorithm when the parameter is chosen suitably, and Algorithm 2 performs well except Examples (ii) and (iii). Furthermore, Algorithms 1, 3, and 4 have faster convergence than the fixed-point algorithm for some parameters in (0, 1) for all Examples (i)–(v).

Comparing the proposed algorithms with the successive overrelaxation (SOR) method (see Algorithm 5.1 and Table 1 in the work of Liu et al.,[8]) it is known from Tables 2–6 that Algorithms 3 and 4 perform better than the SOR method for some parameter α, in particular, $\alpha \in (0, 1)$.

TABLE 2 Comparison between the proposed algorithms and the existing ones in the works of Li et al.[3] and Liu et al.[8] by Example (i)

	Algorithm 1			**Algorithm 2**			**Algorithm 3**			**Algorithm 4**		
α (or β)	**CPU**	**IT**	$\boldsymbol{r}$	**CPU**	**IT**	$\boldsymbol{r}$	**CPU**	**IT**	$\boldsymbol{r}$	**CPU**	**IT**	$\boldsymbol{r}$
0.2	0.120132	113	0.833884	0.130716	116	0.838395	0.016371	14	0.169255	0.014437	14	0.176077
0.4	0.053818	52	0.658179	0.058154	55	0.676765	0.013534	13	0.145312	0.013006	13	0.159637
0.6	0.031415	30	0.472034	0.035865	34	0.515135	0.012232	12	0.11995	0.013277	13	0.142525
0.8	0.017859	18	0.274468	0.023388	22	0.353511	0.01134	11	0.093036	0.011876	12	0.124705
1.2	0.015569	15	0.207866	0.016875	16	0.226265	0.008426	8	0.033950	0.010011	10	0.086742
1.4	0.026597	26	0.397647	0.028915	27	0.430645	0.005075	5	0.001423	0.010155	10	0.066498
1.6	0.054888	54	0.653450	0.050904	48	0.635021	0.008662	8	0.033372	0.009337	9	0.045338
1.8	0.315640	302	0.927216	0.134237	124	0.839396	0.010544	10	0.070677	0.007131	7	0.023194
2.0	-	-	-	-	-	-	0.012099	12	0.110777	**0.002450**	2	0

TABLE 3 Comparison between the proposed algorithms and the existing ones in the works of Li et al.[3] and Liu et al.[8] by Example (ii)

	Algorithm 1			**Algorithm 2**			**Algorithm 3**			**Algorithm 4**		
α (or β)	**CPU**	**IT**	r	**CPU**	**IT**	r	**CPU**	**IT**	r	**CPU**	**IT**	r
0.2	0.097306	95	0.805880	0.104109	98	0.812075	0.008073	8	0.034039	0.009215	9	0.058676
0.4	0.044031	43	0.600396	0.048780	46	0.624639	0.005509	5	0.001219	0.009133	9	0.052498
0.6	0.024982	24	0.397088	0.028765	27	0.437683	0.008111	8	0.033911	0.008436	8	0.046241
0.8	0.015278	15	0.203094	0.018899	17	0.251358	0.011289	10	0.071603	0.008861	8	0.039899
1.2	0.025613	25	0.386964	0.016045	15	0.205510	0.013744	13	0.155878	0.007095	7	0.026959
1.4	0.061713	60	0.684357	0.026936	25	0.406396	0.016986	15	0.203191	0.008132	7	0.020358
1.6	-	-	-	0.048316	44	0.607469	0.017659	17	0.254541	0.007206	7	0.013667
1.8	-	-	-	0.109174	103	0.808445	0.021580	20	0.310469	0.006901	6	0.006882
2.0	-	-	-	-	-	-	0.024510	24	0.371618	**0.003090**	3	4.39E−06

TABLE 4 Comparison between the proposed algorithms and the existing ones in the works of Li et al.[3] and Liu et al.[8] by Example (iii)

	Algorithm 1			Algorithm 2			Algorithm 3			Algorithm 4		
α (or β)	CPU	IT	r	CPU	IT	r	CPU	IT	r	CPU	IT	r
0.2	0.132754	127	0.866731	0.140614	132	0.871482	0.019434	19	0.335336	0.020024	19	0.334794
0.4	0.060231	59	0.722855	0.070456	64	0.743092	0.018615	18	0.308876	0.018275	18	0.309006
0.6	0.035436	35	0.565824	0.043615	41	0.614833	0.017615	17	0.278213	0.017541	17	0.281159
0.8	0.024466	22	0.391244	0.030322	28	0.486707	0.015516	15	0.240994	0.016545	16	0.250959
1.2	0.013258	13	0.181607	0.030737	29	0.468403	0.010464	10	0.057283	0.013315	13	0.182264
1.4	0.022862	22	0.329385	0.066159	63	0.712263	0.012258	12	0.188615	0.012208	12	0.142958
1.6	0.039381	39	0.589081	0.491501	465	0.955879	0.013705	13	0.152016	0.010258	10	0.099685
1.8	0.096232	91	0.788927	-	-	-	0.015396	15	0.214785	0.008216	8	0.051808
2.0	-	-	-	-	-	-	0.018176	16	0.252133	**0.005790**	5	0.001438

TABLE 5 Comparison between the proposed algorithms and the existing ones in the works of Li et al.[3] and Liu et al.[8] by Example (iv)

	Algorithm 1			Algorithm 2			Algorithm 3			Algorithm 4		
α (or β)	CPU	IT	r	CPU	IT	r	CPU	IT	r	CPU	IT	r
0.2	0.193160	188	0.906361	0.218924	206	0.907919	0.032845	32	0.43866	0.034311	33	0.451464
0.4	0.086901	84	0.790657	0.107784	102	0.801173	0.028757	28	0.415926	0.034861	31	0.426236
0.6	0.051876	48	0.648900	0.069468	65	0.682040	0.024770	24	0.385502	0.028351	28	0.417021
0.8	0.029908	29	0.468552	0.049790	47	0.613119	0.020694	20	0.269149	0.026336	26	0.385343
1.2	0.011539	11	0.086687	0.029979	28	0.416600	0.011204	11	0.097653	0.020753	20	0.271837
1.4	0.031728	31	0.478126	0.024430	23	0.305175	0.010210	10	0.062385	0.018903	18	0.228158
1.6	-	-	-	0.020675	19	0.293196	0.018151	18	0.246595	0.015431	15	0.189342
1.8	-	-	-	0.016985	16	0.242344	0.041114	40	0.699896	0.011052	11	0.102844
2.0	-	-	-	0.015955	15	0.188725	-	-	-	**0.002240**	2	2.82E−16

TABLE 6 Comparison between the proposed algorithms and the existing ones in the works of Li et al.[3] and Liu et al.[8] by Example (v)

	Algorithm 1			**Algorithm 2**			**Algorithm 3**			**Algorithm 4**		
α (or β)	**CPU**	**IT**	***r***	**CPU**	**IT**	***r***	**CPU**	**IT**	***r***	**CPU**	**IT**	***r***
0.2	0.479097	457	0.960321	0.545470	516	0.964975	0.094903	92	0.801615	0.099437	96	0.809135
0.4	0.212161	203	0.908481	0.279272	264	0.929952	0.080908	79	0.771205	0.089435	87	0.790282
0.6	0.116418	114	0.837876	0.187184	177	0.894929	0.071528	66	0.729795	0.082056	78	0.767294
0.8	0.071293	68	0.736068	0.142030	132	0.859903	0.054436	53	0.670086	0.071298	69	0.738647
1.2	0.019277	19	0.290556	0.092069	87	0.789857	0.025864	25	0.408784	0.050838	50	0.653276
1.4	0.024127	24	0.371517	0.078297	74	0.754835	0.007051	7	0.021119	0.041125	40	0.585594
1.6	-	-	-	0.068250	64	0.719809	0.229989	223	0.909657	0.032685	31	0.485079
1.8	-	-	-	0.061870	56	0.684785	0.256674	249	0.914812	0.020564	20	0.320181
2.0	-	-	-	0.050987	49	0.649763	0.242860	237	0.919399	**0.003120**	3	8.10E−10

TABLE 7 Numerical results for the existing algorithms

	Fixed-point method[3]			SOR[8]			Newton's method (54)		
	CPU	IT	r	CPU	IT	r	CPU	IT	r
(i)	0.122742	15	0.191887	0.009588	9	0.006039	0.058142	1	-
(ii)	0.010256	9	0.064773	0.009852	10	0.020821	-	-	-
(iii)	0.022678	21	0.358721	0.035383	34	0.596896	-	-	-
(iv)	0.041827	36	0.579276	0.030631	29	0.403790	0.004890	1	-
(v)	0.111799	105	0.824880	0.014351	14	0.331060	0.023087	6	9.03E−10

Furthermore, from Tables 5 and 6, the proposed algorithms may also perform well in the case that the uniqueness condition $\delta_3 > 1/2$ is not available but the solution is still unique.

In summary, through searching the best parameter for the proposed algorithms, Algorithm 4 performs best among those algorithms for solving the tensor equation (1) if the parameters are chosen suitably.

For testing the convergence of the proposed algorithms, the stopping criterion $\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_1 < \varepsilon$ is replaced by $\|\mathcal{P}\mathbf{x}_k^{m-1} - \mathbf{x}_k\|_1 < \varepsilon$ for all the tested algorithms. We also compare the proposed algorithms with the Newton method (54). We take $\varepsilon = 10^{-9}$ and 10^{-12}, respectively and we search the optimal parameter α (or β) from 0.1 to 2.4 except 1 in the interval of 0.1 for Algorithm 1 (or Algorithm 2), and γ from 0.1 to 3.0 in the interval of 0.1 for Algorithms 3 and 4. We use Examples (iv) and (v), and report the numerical results in Tables 8–10, in which one may see that the proposed algorithms also perform better than the existing ones.

Next, we show the relationship between parameters α and γ in Algorithm 4 by Examples (iv) and (v). Taking α from 0.2 to 2 except 1 in the interval of 0.1 and γ from 0.1 to 10 in the interval of 0.1, we get Figures 5, 6, 7, and 8, from which we can see that there always exists an optimal parameter γ for given α. Therefore, we draw Figures 9 and 10, which show the variational cure between α and the optimal parameter γ. Here, the optimal γ refers to the value of γ when Algorithm 4 has the least amount of iteration steps for a given α.

4.2 | Numerical experiments for the multilinear PageRank

In this subsection, we present some numerical experiments for solving the multilinear PageRank (5) by the proposed algorithms. We take 29 stochastic tensors in the work of Gleich et al.[6] and an example given in the work of Li et al.[12] for our tests. For the sake of fairness, we do not use the code proposed by Gleich online, we rewrite the codes by using the function *ttv* of the package Tensor Toolbox 2.6 with moving out the MATLAB's objective-oriented features, which slow down the original function. We replace the stopping criteria 10^{-8} (in the work of Gleich et al.[6]) by 10^{-10} for all the tested algorithms. The maximum iterative number is taken as 1,000 and $\mathbf{v} = \frac{1}{n}\mathbf{e}$. We set the initial value $\mathbf{x}_0 = \mathbf{v}$ for the proposed algorithms. For the algorithms in the work of Gleich et al.,[6] the corresponding parameters are the same as in that Subsection 6.3 of the work of Gleich et al.[6] All the optimal parameters for the proposed algorithms are given by numerical experiments. For simplicity, we use some abbreviations given by Table 11.

The following example is given by Test 4 in the work of Li et al.[12] Let Γ be a directed graph with node set $\mathbb{V} = \{1, 2, ...n\}$. Then, (i, j) is said to be an arc of Γ. Let $\mathbb{P}$ be the subset of $\mathbb{V}$ for which arbitrary two different nodes i, j, (i, j) is an arc in Γ, and let $\mathbb{D}$ be the set of all dangling nodes in $\mathbb{V}$, that is, for any $i \in \mathbb{D}$, both (i, j) and (j, i) are not arcs of Γ for any $j \in \mathbb{V}$ (e.g., see other works[6,22,23]). By n_p ($n_p \geq 2$) we denote the number of nodes in the subset $\mathbb{P}$. Suppose that a directed graph Γ is given by $\mathbb{V} = \mathbb{D} \bigcup \mathbb{P}$. Then, we may construct a corresponding nonnegative tensor $\mathcal{A}$ as follows:

$$a_{i_1 i_2 \cdots i_m} = \begin{cases} \gamma_{i_1 i_2 \cdots i_m}, & i_1 \neq i_2, i_k \neq i_{k+1}, i_1, i_k \in \mathbb{P}, k = 2, \ldots, m-1, \\ 0, & i_1 = i_2 (or\ i_1 \in \mathbb{D}), i_k \neq i_{k+1}, i_k \in \mathbb{P}, k = 2, \ldots, m-1, \\ \frac{1}{n}, & \text{else}, \end{cases}$$

where $\gamma_{i_1 i_2 \cdots i_m} \in (0, 1)$, and then normalizing the entries with $\hat{p}_{i_1 i_2 \cdots i_m} = \frac{a_{i_1 i_2 \cdots i_m}}{\sum_{i_1=1}^{n} a_{i_1 i_2 \cdots i_m}}$ gets a stochastic tensor $\hat{\mathcal{P}} = (\hat{p}_{i_1 i_2 \cdots i_m})$. It is noted that in the following numerical test, $\gamma_{i_1 i_2 \cdots i_m}$ is taken in $(0, 1)$ by randomly and independently. For example, let Γ be a directed graph with $\mathbb{V} = \{1, 2, 3, 4, 5\}$, where $\mathbb{P} = \{1, 2\}$ and $\mathbb{D} = \{3, 4, 5\}$. Then, by the above definition,

TABLE 8 Numerical results for the proposed algorithms in different ε by Examples (iv) and (v)

	Algorithm 1				Algorithm 2				Algorithm 3				Algorithm 4			
	10^{-9}		10^{-12}		10^{-9}		10^{-12}		10^{-9}		10^{-12}		10^{-9}		10^{-12}	
	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT
(iv)	0.015643	8	0.021458	11	0.013951	13	0.018045	17	0.01173	6	0.013696	7	**0.001979**	1	**0.001986**	1
(v)	0.011727	6	0.015576	8	0.035917	34	0.047261	46	0.011709	6	0.015671	8	**0.003959**	2	**0.003977**	2

TABLE 9 Numerical results for the existing methods in different ε by Examples (iv) and (v)

	Fixed-point method[3]				SOR[8]				Newton's method (54)			
	10^{-9}		10^{-12}		10^{-9}		10^{-12}		10^{-9}		10^{-12}	
	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT
(iv)	0.720639	32	0.047430	43	0.047181	24	0.066253	33	0.059744	1	0.004183	1
(v)	0.094213	93	0.130327	129	0.021327	10	0.033859	17	0.015311	4	0.013588	4

TABLE 10 The corresponding parameters for the proposed algorithms in Table 8

	Algorithm 1		Algorithm 2		Algorithm 3		Algorithm 4	
	10^{-9}	10^{-12}	10^{-9}	10^{-12}	10^{-9}	10^{-12}	10^{-9}	10^{-12}
	α	α	β	β	(α, γ)	(α, γ)	(α, γ)	(α, γ)
(iv)	1.1	1.1	2.0	2.1	(1.2, 0.9)	(1.3, 0.8)	(2.0, 0.5)	(2.0, 0.5)
(v)	1.3	1.3	2.4	2.4	(1.3, 1.0)	(1.3, 1.0)	(2.0, 0.5)	(2.0, 0.5)

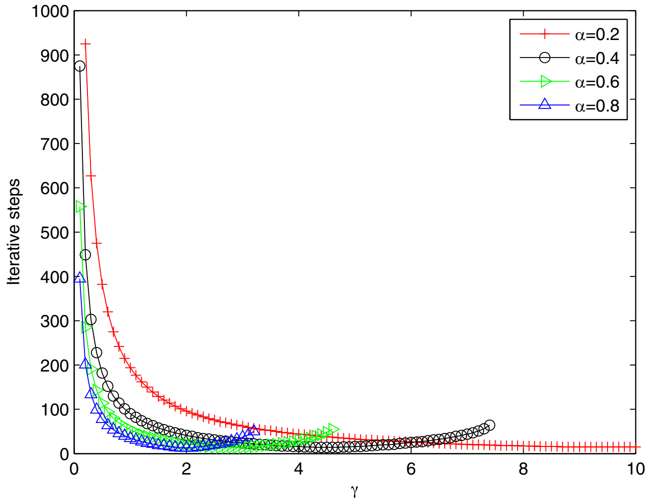

FIGURE 5 The relationship between iterative steps and γ for different α by Example (iv)

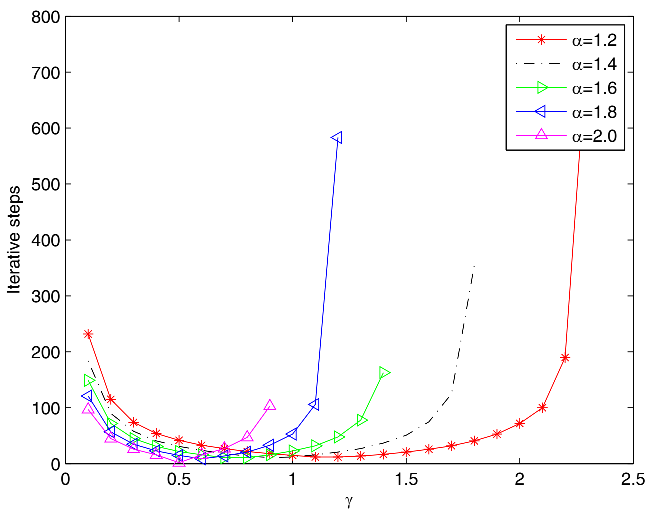

FIGURE 6 The relationship between iterative steps and γ for different α by Example (iv)

we can get an order-3 dimension 5 tensor $\mathcal{A}$ with $a_{121}, a_{212} \in (0, 1)$, $a_{221} ={}_{321}= a_{421} = a_{521} = a_{112} = a_{312} = a_{412} = a_{512} = 0$ and other entries being $1/5$. After normalizing, the corresponding stochastic tensor $\hat{\mathcal{P}}$ is given, that is, $\hat{p}_{221} = \hat{p}_{321} = \hat{p}_{421} = \hat{p}_{521} = \hat{p}_{112} = \hat{p}_{312} = \hat{p}_{412} = \hat{p}_{512} = 0$, $\hat{p}_{121} = \frac{a_{121}}{\sum_{i=1}^{5} a_{i21}} = 1$, $\hat{p}_{212} = \frac{a_{212}}{\sum_{i=1}^{5} a_{i12}} = 1$, and other entries are $1/5$.

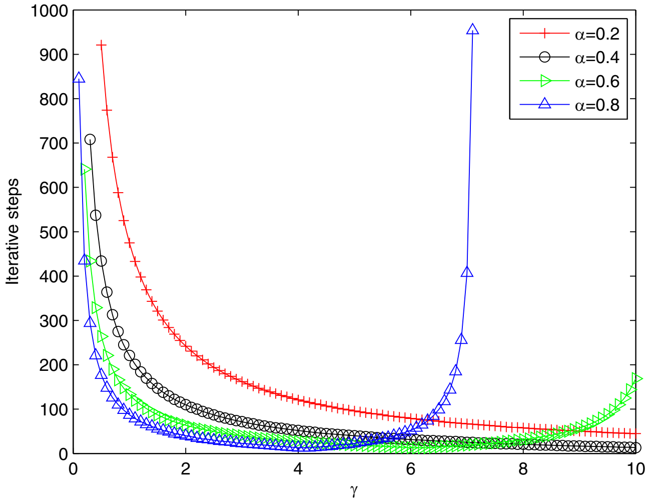

FIGURE 7 The relationship between iterative steps and γ for different α by Example (v)

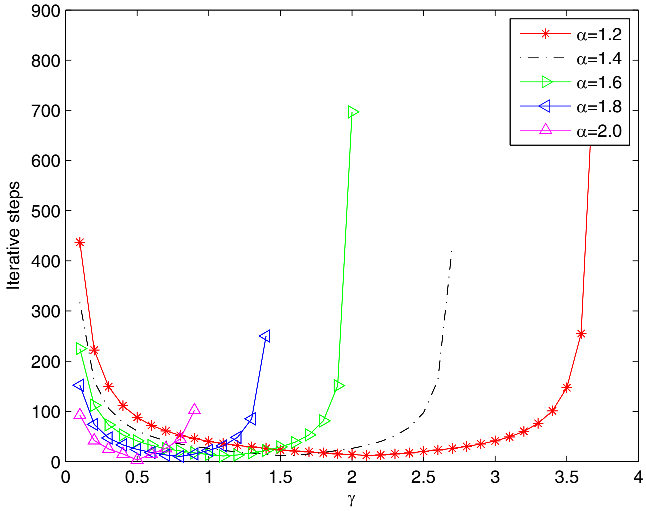

FIGURE 8 The relationship between iterative steps and γ for different α by Example (v)

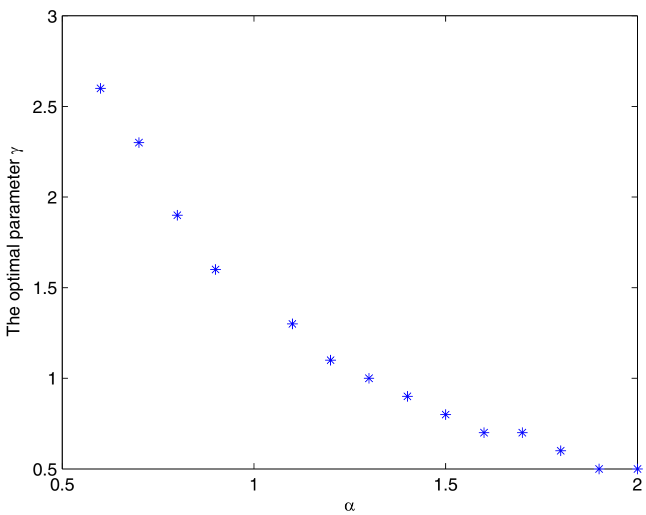

FIGURE 9 The relationship between α and the optimal γ by Example (iv)

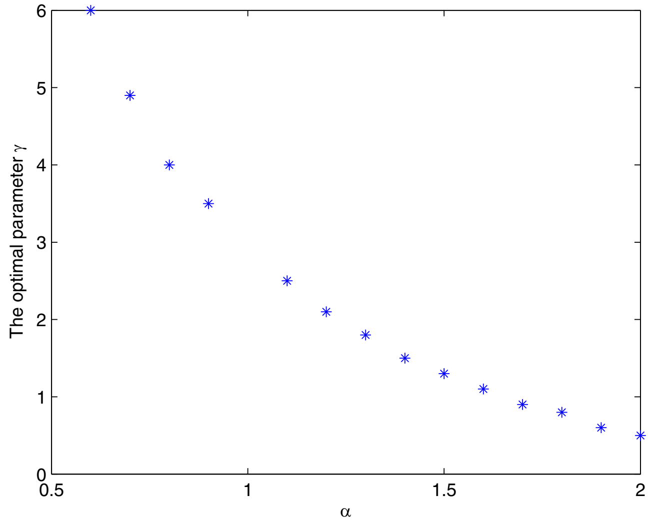

FIGURE 10 The relationship between α and the optimal γ by Example (v)

TABLE 11 Abbreviations and full explanations of the algorithms

Abbreviations	Full explanations of the algorithms
Ag1	Algorithm 1
Ag2	Algorithm 2
Ag3	Algorithm 3
Ag4	Algorithm 4
F	The fixed-point iteration[6]
S	The shifted fixed-point iteration[6]
IO	The inner–outer iteration[6]
INV	The inverse iteration[6]
N	Newton's method[6]

For 29 stochastic tensors in the work of Gleich et al.,[6] we list the numerical results in Tables 12–16 for different θ as that in the work of Gleich et al.,[6] from which we can see that Algorithms 1–4 have faster convergence than the existing ones in the work of Gleich et al.,[6] where '-' means that the corresponding algorithm is not available. The corresponding parameters in Algorithms 1–4 are reported in Table 17.

Let $m = 3$ and θ be 0.70, 0.85, 0.90, 0.95, and 0.99, respectively. For every value of θ, we let the dimension $n = 200$ and $\mathbb{P} = \{1, 2, \dots, n_p\}$ with n_p being 80, 100, 150, and 180, respectively. The function *symmetrize* of the package *Tensor Toolbox 2.6* is used to improve the efficiency of semi-symmetrizing the tensor $\mathcal{P}$ by (51). We report the numerical results in Table 18, and the corresponding parameters for the proposed algorithms are given in Table 19.

Remark 19. From Tables 12–17, we know that Algorithm 4 performs best in all tested algorithms for 29 examples given in the work of Gleich et al.[6] for different parameter θ. However, for larger size n, Algorithm 3 is the best by Table 18 because for a larger n, it will take more CPU times to do the semi-symmetrization of the tensor $\mathcal{P}$.

TABLE 12 $\theta = 0.70$

	Ag1		Ag2		Ag3		Ag4		F		S		IO		INV		N	
	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT
$R_{3,1}$	0.010380	14	0.022376	28	0.007982	9	**0.004292**	4	0.056206	52	0.051394	52	0.086218	68	0.025423	16	0.012261	5
$R_{3,2}$	0.015580	21	0.010732	14	0.009827	11	**0.004869**	5	0.014808	15	0.045859	44	0.083192	59	0.019242	12	0.014655	6
$R_{3,3}$	0.010578	14	0.010630	14	0.008702	10	**0.006934**	5	0.013228	15	0.039930	44	0.071957	59	0.019781	12	0.014975	6
$R_{3,4}$	0.010740	13	0.013808	18	0.008740	10	**0.004032**	4	0.038952	43	0.028351	30	0.062313	47	0.030430	16	0.012686	5
$R_{3,5}$	0.033358	43	0.073348	96	0.013184	15	**0.004619**	5	0.075137	86	0.166655	175	0.095845	181	0.085880	50	0.014960	6
$R_{4,1}$	0.023930	29	0.049155	62	0.012810	15	**0.004896**	5	0.048785	55	0.110354	117	0.097193	129	0.064006	33	0.016795	6
$R_{4,2}$	0.029310	33	0.050298	64	0.021338	26	**0.005028**	5	0.071994	77	0.132875	138	0.090466	127	0.067665	39	0.013303	5
$R_{4,3}$	0.030075	38	0.048041	51	0.020373	24	**0.004728**	5	0.045887	50	0.101156	104	0.084673	113	0.050214	30	0.015331	6
$R_{4,4}$	0.027346	28	0.049841	65	0.013171	16	**0.004577**	5	0.090603	77	0.133461	114	0.090406	117	0.055987	33	0.012630	5
$R_{4,5}$	0.031278	38	0.031991	40	0.018762	23	**0.004889**	5	0.067197	53	0.129168	108	0.103760	114	0.052046	31	0.015036	6
$R_{4,6}$	0.019729	25	0.049503	51	0.014774	17	**0.005074**	5	0.057386	64	0.115833	120	0.087086	116	0.059042	34	0.015506	6
$R_{4,7}$	0.024596	31	0.041129	53	0.014106	16	**0.005143**	5	0.044623	51	0.094831	98	0.110854	108	0.064175	29	0.022574	6
$R_{4,8}$	0.023153	27	0.048587	61	**0.004630**	5	0.004664	5	0.064061	68	0.111880	118	0.087952	111	0.058188	35	0.015575	6
$R_{4,9}$	0.023014	29	0.036100	45	0.013023	15	**0.004754**	5	0.046253	52	0.097584	104	0.090571	109	0.049314	30	0.015332	6
$R_{4,10}$	0.027565	35	0.044359	57	0.013155	16	**0.004139**	4	0.079699	90	0.099578	106	0.085038	110	0.051207	31	0.013012	5
$R_{4,11}$	0.019822	26	0.037714	49	0.021163	26	**0.004986**	5	0.049739	54	0.119508	107	0.089840	113	0.049062	29	0.015837	6
$R_{4,12}$	0.019306	25	0.030441	39	0.011172	13	**0.004574**	5	0.035200	38	0.081271	84	0.087640	95	0.039763	24	0.014958	6
$R_{4,13}$	0.021485	27	0.052704	64	0.015839	19	**0.005806**	5	0.051582	58	0.110100	115	0.091139	124	0.054341	33	0.015637	6
$R_{4,14}$	0.020595	27	0.031547	41	0.020170	25	**0.004587**	5	0.051129	56	0.107160	109	0.087248	110	0.050608	31	0.015638	6
$R_{4,15}$	0.023294	28	0.047461	59	0.017968	22	**0.004557**	5	0.062329	70	0.122927	128	0.089694	121	0.060293	36	0.013003	5
$R_{4,16}$	0.032206	32	0.048366	62	0.016359	19	**0.004841**	5	0.049283	56	0.104168	112	0.087370	117	0.053339	32	0.014611	6
$R_{4,17}$	0.023207	30	0.044486	56	0.017147	21	**0.005688**	5	0.047793	53	0.096580	103	0.083105	111	0.048977	29	0.015634	6
$R_{4,18}$	0.027918	36	0.043643	55	0.018805	23	**0.005003**	5	0.047172	52	0.103929	104	0.087931	115	0.050374	30	0.015879	6
$R_{4,19}$	0.022034	27	0.042632	54	0.009791	11	**0.005124**	5	0.050001	49	0.107043	96	0.087629	102	0.046368	28	0.015658	6
$R_{6,1}$	0.016840	19	0.019234	24	0.016442	18	**0.005169**	5	0.041934	40	0.096312	87	0.085963	99	0.044435	25	0.016385	6
$R_{6,2}$	0.024656	29	0.034858	44	0.014957	16	**0.005123**	5	0.044838	49	0.082122	85	0.084223	99	0.044248	25	0.016442	6
$R_{6,3}$	0.022912	27	0.031182	40	0.017534	19	**0.005104**	5	0.031023	35	0.072898	77	0.078867	88	0.038916	22	0.016059	6
$R_{6,4}$	0.026321	30	0.018958	25	0.023691	26	**0.005134**	5	0.037266	40	0.086505	89	0.096384	103	0.059205	26	0.016388	6
$R_{6,5}$	0.019425	23	0.019963	26	0.017231	19	**0.005112**	5	0.030696	33	0.072379	74	0.073681	87	0.040394	23	0.016075	6

Note. Ag1 = Algorithm 1; Ag2 = Algorithm 2; Ag3 = Algorithm 3; Ag4 = Algorithm 4; F = the fixed-point iteration; S = the shifted fixed-point iteration; IO = the inner–outer iteration; INV = the inverse iteration; N = Newton's method.

TABLE 13 $\theta = 0.85$

	Ag1		Ag2		Ag3		Ag4		F		S		IO		INV		N	
	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT
$R_{3,1}$	0.012674	17	0.032641	40	0.008376	10	**0.004882**	5	0.117967	130	0.056065	60	0.084043	59	0.031208	19	0.015869	6
$R_{3,2}$	0.019451	26	0.012840	16	0.010081	11	**0.004490**	5	0.019098	21	0.049391	51	0.086110	54	0.023861	15	0.014549	6
$R_{3,3}$	0.011492	15	0.012640	16	0.009645	12	**0.004637**	5	0.018596	21	0.048251	51	0.080946	54	0.024408	15	0.015690	6
$R_{3,4}$	0.011385	14	0.014313	18	0.008934	10	**0.004203**	4	0.064275	70	0.029558	31	0.062903	38	0.029010	18	0.012614	5
$R_{3,5}$	0.036597	47	0.092172	114	0.012857	16	**0.006058**	7	0.093738	103	0.202136	208	0.115005	159	0.095449	58	0.017776	7
$R_{4,1}$	0.029384	37	0.113832	140	0.012314	15	**0.006668**	7	0.117703	134	0.234156	244	0.108704	165	0.109654	66	0.020182	8
$R_{4,2}$	0.048894	60	0.201710	266	0.030206	39	**0.004512**	5	0.240304	261	0.322254	343	0.111015	157	0.133734	82	0.015285	6
$R_{4,3}$	0.054812	73	0.087182	112	0.029296	34	**0.005637**	6	0.101663	101	0.223818	204	0.108290	146	0.095821	57	0.017733	7
$R_{4,4}$	0.035974	44	0.145592	181	0.012012	14	**0.004646**	5	0.232945	262	0.276807	279	0.111352	166	0.152102	74	0.015685	6
$R_{4,5}$	0.053470	68	0.110264	140	0.024663	30	**0.006070**	6	0.147746	162	0.268626	275	0.111144	159	0.124296	71	0.017990	7
$R_{4,6}$	0.057087	76	0.180393	225	0.021861	27	**0.005867**	6	0.205940	225	0.303225	314	0.107417	154	0.130318	77	0.015920	6
$R_{4,7}$	0.031546	38	0.104741	132	0.010420	12	**0.005356**	6	0.105346	120	0.215962	214	0.107474	140	0.097427	58	0.018179	7
$R_{4,8}$	0.048700	59	0.219268	278	0.007905	6	**0.005364**	6	0.270008	300	0.321488	329	0.122766	135	0.162076	77	0.026930	7
$R_{4,9}$	0.031945	41	0.106860	132	0.014916	18	**0.006158**	7	0.157920	144	0.277450	242	0.122872	143	0.105235	64	0.017946	7
$R_{4,10}$	0.044837	57	0.102582	130	0.020952	26	**0.004821**	5	0.326023	364	0.216595	221	0.118098	153	0.108255	61	0.016289	6
$R_{4,11}$	0.033968	44	0.089066	112	0.027211	34	**0.005362**	6	0.119177	133	0.230692	245	0.100632	173	0.110362	67	0.015346	6
$R_{4,12}$	0.023941	32	0.062112	80	0.013611	15	**0.007613**	6	0.077762	88	0.172145	179	0.105514	139	0.083366	50	0.018087	7
$R_{4,13}$	0.036817	44	0.130619	162	0.016448	20	**0.006031**	6	0.138845	151	0.267400	274	0.116852	176	0.122474	73	0.019526	7
$R_{4,14}$	0.029575	38	0.090002	117	0.026912	34	**0.007767**	9	0.122936	134	0.209764	219	0.111333	134	0.097921	59	0.020850	8
$R_{4,15}$	0.046539	52	0.186737	238	0.024867	31	**0.005333**	6	0.216099	238	0.313278	325	0.126297	153	0.163832	79	0.021554	6
$R_{4,16}$	0.042157	52	0.107218	138	0.017868	22	**0.005988**	6	0.131097	149	0.254372	262	0.107931	166	0.121090	70	0.017573	7
$R_{4,17}$	0.046722	53	0.109676	137	0.023108	27	**0.005365**	6	0.189172	208	0.235574	240	0.101168	176	0.105479	63	0.018065	7
$R_{4,18}$	0.077556	92	0.159098	156	0.024150	28	**0.005768**	6	0.131210	141	0.269672	280	0.113948	208	0.126246	77	0.018347	7
$R_{4,19}$	0.032089	39	0.094804	113	0.011998	14	**0.005979**	6	0.131429	120	0.230683	200	0.112350	124	0.088119	54	0.017606	7
$R_{6,1}$	0.018941	22	0.048239	59	0.020024	22	**0.006268**	6	0.094195	106	0.209194	212	0.101393	161	0.104719	59	0.018519	7
$R_{6,2}$	0.031236	37	0.068890	88	0.021173	24	**0.005882**	6	0.107276	98	0.206156	161	0.114092	129	0.079171	46	0.018673	7
$R_{6,3}$	0.040145	46	0.059083	74	0.024838	29	**0.005247**	5	0.059422	67	0.127045	135	0.092299	107	0.066318	38	0.016216	6
$R_{6,4}$	0.035605	41	0.040474	51	0.034097	38	**0.005907**	6	0.077942	85	0.168509	173	0.101942	128	0.089406	49	0.018901	7
$R_{6,5}$	0.026117	31	0.036038	46	0.027406	31	**0.006400**	6	0.060321	66	0.134906	139	0.091975	112	0.073251	41	0.019053	7

Note. Ag1 = Algorithm 1; Ag2 = Algorithm 2; Ag3 = Algorithm 3; Ag4 = Algorithm 4; F = the fixed-point iteration; S = the shifted fixed-point iteration; IO = the inner–outer iteration; INV = the inverse iteration; N = Newton's method.

TABLE 14 $\theta = 0.90$

	Ag1		Ag2		Ag3		Ag4		F		S		IO		INV		N	
	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT
$R_{3,1}$	0.013616	17	0.029156	36	0.008061	9	**0.004618**	5	0.241998	245	0.071654	64	0.087443	57	0.031335	19	0.014716	6
$R_{3,2}$	0.020093	25	0.012996	17	0.010634	13	**0.007286**	6	0.020992	23	0.053882	56	0.093523	53	0.029588	17	0.019066	7
$R_{3,3}$	0.014395	16	0.013422	17	0.010699	12	**0.005418**	6	0.021179	23	0.055074	56	0.081742	53	0.028608	17	0.016928	7
$R_{3,4}$	0.011217	14	0.015011	19	0.008403	10	**0.003794**	4	0.080830	88	0.029375	31	0.063613	35	0.032182	19	0.012485	5
$R_{3,5}$	0.025782	32	0.060257	75	0.011254	13	**0.006433**	7	0.060817	68	0.142224	140	0.140943	103	0.077176	39	0.029212	9
$R_{4,1}$	0.035208	45	0.224662	283	0.012236	15	**0.014043**	16	0.263282	298	0.445391	461	0.123742	215	0.195706	114	0.020256	8
$R_{4,2}$	0.067459	84	0.564897	667	0.037489	48	**0.005992**	6	0.818459	906	0.631440	652	0.121701	180	0.214451	127	0.019024	6
$R_{4,3}$	0.085253	109	0.139854	174	0.035141	41	**0.005471**	6	0.143761	157	0.291671	300	0.118239	196	0.144280	84	0.017924	7
$R_{4,4}$	0.056914	67	0.342893	432	0.010096	11	**0.005906**	6	0.638019	660	0.657787	558	0.135700	212	0.202404	126	0.017421	7
$R_{4,5}$	0.114485	138	0.406364	512	0.029332	35	**0.006644**	7	0.490831	537	0.581350	592	0.121658	202	0.255473	127	0.024499	8
$R_{4,6}$	0.144955	175	0.489006	621	0.023035	29	**0.005921**	6	0.818433	873	0.622387	645	0.121156	181	0.209638	126	0.017443	7
$R_{4,7}$	0.040781	49	0.181148	226	0.016163	15	**0.005875**	6	0.215100	243	0.371563	382	0.120192	181	0.181182	94	0.018473	7
$R_{4,8}$	0.073670	98	0.592709	745	0.005971	6	**0.005542**	6	-	-	0.711740	745	0.123982	151	0.201221	121	0.017198	7
$R_{4,9}$	0.050443	66	0.260126	337	0.014221	17	**0.006654**	7	0.298505	329	0.415506	438	0.123039	172	0.166509	100	0.019529	8
$R_{4,10}$	0.089565	115	0.216206	235	0.027966	33	**0.004740**	5	-	-	0.479846	376	0.135382	201	0.158938	96	0.015340	6
$R_{4,11}$	0.053840	58	0.166748	204	0.025243	31	**0.005827**	6	0.205153	233	0.411063	424	0.107705	250	0.188086	111	0.017792	7
$R_{4,12}$	0.028102	36	0.123750	126	0.019397	24	**0.008542**	10	0.136520	151	0.280346	299	0.113220	201	0.136147	82	0.021633	8
$R_{4,13}$	0.047876	56	0.260216	328	0.019397	24	**0.006712**	7	0.288608	314	0.473218	490	0.124278	225	0.201858	119	0.020785	8
$R_{4,14}$	0.040031	52	0.228174	271	0.033803	42	**0.015659**	19	0.253251	272	0.389840	372	0.146514	160	0.148419	90	0.121035	47
$R_{4,15}$	0.054849	72	0.467789	599	0.021126	26	**0.005784**	6	0.805225	799	0.608365	629	0.122749	177	0.234186	124	0.021640	7
$R_{4,16}$	0.055246	73	0.226846	285	0.018716	21	**0.005613**	6	0.292033	299	0.485235	466	0.136945	215	0.189111	114	0.018038	7
$R_{4,17}$	0.053128	69	0.210995	258	0.027373	33	**0.012297**	15	-	-	0.476362	445	0.117215	286	0.188932	115	0.024375	10
$R_{4,18}$	0.070404	87	0.110930	140	0.024756	30	**0.006840**	8	0.152277	154	0.246678	256	0.116617	160	0.117297	69	0.026000	10
$R_{4,19}$	0.040874	53	0.170252	213	0.012062	14	**0.005578**	6	0.196877	218	0.306995	314	0.114841	140	0.130330	77	0.017272	7
$R_{6,1}$	0.031310	36	0.095362	123	0.026614	30	**0.006760**	7	0.210535	232	0.444666	454	0.113133	299	0.212332	124	0.020838	8
$R_{6,2}$	0.035698	42	0.094291	118	0.022696	25	**0.005881**	6	0.104861	118	0.204443	210	0.108863	142	0.104269	59	0.019350	7
$R_{6,3}$	0.059373	63	0.084022	106	0.028397	33	**0.006400**	6	0.085425	95	0.190616	189	0.105157	132	0.091700	52	0.016189	6
$R_{6,4}$	0.046621	55	0.072944	96	0.032529	36	**0.006277**	6	0.139141	152	0.273753	274	0.111435	163	0.130902	74	0.018973	7
$R_{6,5}$	0.040608	42	0.049639	64	0.034126	40	**0.005975**	6	0.090046	98	0.196313	200	0.096973	141	0.097525	57	0.018376	7

Note. Ag1 = Algorithm 1; Ag2 = Algorithm 2; Ag3 = Algorithm 3; Ag4 = Algorithm 4; F = the fixed-point iteration; S = the shifted fixed-point iteration; IO = the inner–outer iteration; INV = the inverse iteration; N = Newton's method.

TABLE 15 $\theta = 0.95$

	Ag1		Ag2		Ag3		Ag4		F		S		IO		INV		N	
	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT
$R_{3,1}$	0.013132	17	0.035764	45	0.007516	9	**0.005124**	5	-	-	0.065299	68	0.090311	55	0.032816	20	0.014785	6
$R_{3,2}$	0.019555	24	0.013247	17	0.010514	12	**0.005994**	6	0.025807	28	0.062030	65	0.086978	54	0.030816	18	0.017707	7
$R_{3,3}$	0.011610	15	0.013508	17	0.010749	12	**0.005282**	6	0.024962	28	0.061418	65	0.085845	54	0.030232	18	0.018061	7
$R_{3,4}$	0.010798	14	0.014899	19	0.008282	10	**0.004387**	4	0.104364	114	0.029491	31	0.063363	33	0.031845	19	0.012658	5
$R_{3,5}$	0.017730	23	0.038204	48	0.013161	15	**0.011713**	14	0.041515	43	0.087358	93	0.125090	68	0.043473	26	-	-
$R_{4,1}$	0.109426	138	-	-	0.013929	16	**0.012102**	15	-	-	-	-	0.142890	330	0.554911	327	0.022843	9
$R_{4,2}$	0.121392	133	-	-	0.040238	48	**0.005395**	6	-	-	-	-	0.139848	219	0.464150	280	0.017873	7
$R_{4,3}$	0.143042	181	0.352262	454	0.040344	49	**0.006487**	7	0.376207	410	0.699324	739	0.128501	356	0.315685	190	0.020503	8
$R_{4,4}$	0.088388	111	-	-	0.011549	14	**0.005475**	6	-	-	-	-	0.132553	322	0.673785	397	0.018189	7
$R_{4,5}$	-	-	-	-	0.027859	34	**0.006090**	7	-	-	-	-	0.140836	281	0.792088	419	0.024774	8
$R_{4,6}$	0.702179	875	-	-	0.021269	26	**0.005622**	6	-	-	-	-	0.131164	223	0.517128	309	0.018230	7
$R_{4,7}$	0.076400	90	-	-	0.015985	18	**0.005850**	6	-	-	-	-	0.130703	262	0.375286	222	0.018581	7
$R_{4,8}$	0.185825	238	-	-	0.006078	6	**0.005888**	6	-	-	-	-	0.140103	173	0.491808	288	0.017847	7
$R_{4,9}$	0.165522	201	-	-	0.010537	12	**0.007220**	7	-	-	-	-	0.139870	245	0.417134	249	0.020753	8
$R_{4,10}$	0.551989	695	-	-	0.064399	81	**0.007067**	6	-	-	-	-	0.129596	325	0.398684	238	0.017447	7
$R_{4,11}$	0.076920	99	0.685059	873	0.027375	32	**0.006446**	6	0.853841	945	-	-	0.121698	567	0.577167	343	0.017978	7
$R_{4,12}$	0.078836	95	0.145378	174	0.033782	42	**0.018502**	23	0.191487	209	0.377806	390	0.132220	201	0.189856	102	-	-
$R_{4,13}$	0.116501	141	-	-	0.019453	24	**0.006521**	7	-	-	-	-	0.144563	339	0.502831	302	0.019881	8
$R_{4,14}$	0.066500	85	-	-	0.045821	54	**0.019451**	22	-	-	-	-	0.160366	216	0.389617	204	0.422868	159
$R_{4,15}$	0.079746	102	-	-	0.022415	26	**0.007077**	7	-	-	-	-	0.136633	220	0.479416	283	0.017967	7
$R_{4,16}$	0.143910	167	-	-	0.023334	29	**0.005359**	6	-	-	-	-	0.151653	325	0.463552	277	0.017888	7
$R_{4,17}$	0.158898	203	-	-	0.118625	149	**0.023748**	28	-	-	-	-	-	-	0.868437	528	0.255576	100
$R_{4,18}$	0.106007	136	0.639720	812	0.025294	28	**0.016607**	21	0.770642	867	0.906721	937	0.124804	279	0.326195	195	0.063693	25
$R_{4,19}$	0.060719	74	0.491840	626	0.012830	15	**0.005571**	6	0.862100	866	0.712359	651	0.132393	162	0.215236	128	0.017661	7
$R_{6,1}$	0.090823	87	0.149957	185	0.063424	71	**0.019046**	20	0.234144	257	0.444956	465	0.129061	239	0.208007	121	2.416085	907
$R_{6,2}$	0.136816	162	0.424286	543	0.033943	39	**0.015846**	17	0.559135	528	0.872897	749	0.124879	284	0.307440	174	0.996321	375
$R_{6,3}$	0.096232	109	0.130436	159	0.048525	57	**0.006120**	6	0.152159	172	0.326427	334	0.112839	203	0.161939	91	0.018991	7
$R_{6,4}$	0.078723	90	0.317175	405	0.032201	36	**0.005820**	6	0.381507	426	0.603919	639	0.120936	247	0.260653	152	0.018130	7
$R_{6,5}$	0.062824	71	0.089274	116	0.048997	57	**0.010240**	7	0.169461	185	0.362912	368	0.106285	225	0.182709	103	0.021216	8

Note. Ag1 = Algorithm 1; Ag2 = Algorithm 2; Ag3 = Algorithm 3; Ag4 = Algorithm 4; F = the fixed-point iteration; S = the shifted fixed-point iteration; IO = the inner–outer iteration; INV = the inverse iteration; N = Newton's method.

TABLE 16 $\theta = 0.99$

	Ag1		Ag2		Ag3		Ag4		F		S		IO		INV		N	
	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT
$R_{3,1}$	0.012962	17	0.037523	47	0.006874	8	**0.004732**	5	-	-	0.066675	71	0.091730	53	0.033874	21	0.014672	6
$R_{3,2}$	0.021082	28	0.010736	14	0.011983	14	**0.005718**	6	0.031134	34	0.069176	72	0.099148	59	0.040449	22	0.021322	7
$R_{3,3}$	0.015598	16	0.010545	14	0.010021	11	**0.008202**	9	0.030195	34	0.066740	72	0.093858	59	0.037729	22	0.018171	7
$R_{3,4}$	0.011545	14	0.016603	21	0.008563	10	**0.004432**	4	0.179701	148	0.030357	31	0.070080	31	0.033808	20	0.012636	5
$R_{3,5}$	0.013518	17	0.024258	31	**0.010070**	12	0.010824	13	0.026724	27	0.070943	65	0.152088	49	0.027969	17	-	-
$R_{4,1}$	-	-	-	-	0.015869	19	**0.007742**	8	-	-	-	-	0.154630	666	-	-	0.030274	8
$R_{4,2}$	-	-	-	-	0.042394	55	**0.005336**	6	-	-	-	-	0.148023	282	-	-	0.017642	7
$R_{4,3}$	-	-	-	-	0.043767	52	**0.006777**	7	-	-	-	-	-	-	-	-	0.019942	8
$R_{4,4}$	0.261858	348	-	-	0.012937	16	**0.005680**	6	-	-	-	-	0.140468	583	-	-	0.017549	7
$R_{4,5}$	-	-	-	-	0.029781	37	**0.007513**	7	-	-	-	-	0.146022	385	-	-	0.020466	8
$R_{4,6}$	-	-	-	-	0.020127	25	**0.005628**	6	-	-	-	-	0.143965	275	-	-	0.017759	7
$R_{4,7}$	0.276104	340	-	-	0.017252	22	**0.005393**	6	-	-	-	-	0.145635	426	-	-	0.018349	7
$R_{4,8}$	-	-	-	-	**0.006342**	7	0.006418	7	-	-	-	-	0.148228	190	-	-	0.01693	7
$R_{4,9}$	-	-	-	-	0.015546	15	**0.006812**	7	-	-	-	-	0.148237	390	-	-	0.020712	8
$R_{4,10}$	-	-	-	-	0.212090	255	**0.005950**	6	-	-	-	-	0.141244	659	-	-	0.017965	7
$R_{4,11}$	-	-	-	-	0.032216	40	**0.005854**	6	-	-	-	-	-	-	-	-	0.018121	7
$R_{4,12}$	-	-	-	-	0.076739	97	**0.019697**	25	-	-	-	-	-	-	-	-	-	-
$R_{4,13}$	0.504584	626	-	-	0.021834	27	**0.006626**	7	-	-	-	-	0.150156	623	-	-	0.020531	8
$R_{4,14}$	0.127060	163	-	-	0.054102	69	**0.020833**	26	-	-	-	-	0.144276	333	-	-	0.889177	349
$R_{4,15}$	0.124578	160	-	-	0.014857	18	**0.006335**	7	-	-	-	-	0.148551	284	-	-	0.019995	8
$R_{4,16}$	-	-	-	-	0.023307	29	**0.005759**	6	-	-	-	-	0.146490	637	-	-	0.017769	7
$R_{4,17}$	0.132089	164	0.402073	510	0.064605	79	**0.006736**	7	-	-	0.812450	845	-	-	-	-	0.017729	7
$R_{4,18}$	0.287723	359	-	-	0.021791	27	**0.019516**	25	-	-	-	-	0.138603	528	-	-	2.621964	-
$R_{4,19}$	0.107104	133	-	-	0.012042	15	**0.005963**	6	-	-	-	-	0.130878	184	0.382338	240	0.017483	7
$R_{6,1}$	-	-	-	-	0.035338	41	**0.026518**	29	-	-	-	-	0.136431	901	-	-	1.074289	408
$R_{6,2}$	-	-	-	-	0.046163	55	**0.014883**	17	-	-	-	-	0.138186	764	-	-	-	-
$R_{6,3}$	-	-	-	-	0.083127	98	**0.034250**	35	-	-	-	-	-	-	-	-	-	-
$R_{6,4}$	0.207293	246	-	-	0.062366	74	**0.006916**	7	-	-	-	-	0.131543	419	0.853969	503	0.020506	8
$R_{6,5}$	0.087063	102	-	-	0.042934	50	**0.010104**	10	-	-	-	-	-	-	-	-	0.029563	11

Note. Ag1 = Algorithm 1; Ag2 = Algorithm 2; Ag3 = Algorithm 3; Ag4 = Algorithm 4; F = the fixed-point iteration; S = the shifted fixed-point iteration; IO = the inner–outer iteration; INV = the inverse iteration; N = Newton's method.

TABLE 17 The corresponding parameters of Algorithms 1–4 for the 29 examples in the work of Gleich et al.[6]

	$\theta = 0.70$				$\theta = 0.85$				$\theta = 0.90$				$\theta = 0.95$				$\theta = 0.99$			
	Ag1	**Ag2**	**Ag3**	**symAg3**	**Ag1**	**Ag2**	**Ag3**	**Ag4**	**Ag1**	**Ag2**	**Ag3**	**Ag4**	**Ag1**	**Ag2**	**Ag3**	**Ag4**	**Ag1**	**Ag2**	**Ag3**	**Ag4**
	α	β	(α,γ)	(α,γ)	α	β	(α,γ)	(α,γ)	α	β	(α,γ)	(α,γ)	α	β	(α,γ)	(α,γ)	α	β	(α,γ)	(α,γ)
$R_{3,1}$	1.1	0.8	(2.4,0.5)	(2.0,0.5)	1.1	0.7	(2.4,0.5)	(2.0,0.5)	1.2	0.8	(2.0,0.6)	(2.0,0.5)	1.2	0.7	(2.2,0.6)	(2.0,0.5)	1.2	0.8	(2.2,0.6)	(2.0,0.5)
$R_{3,2}$	0.9	1.2	(0.4,2.8)	(2.0,0.5)	0.9	1.3	(0.4,3.0)	(2.0,0.5)	1.1	1.4	(0.5,2.6)	(2.0,0.5)	1.1	1.6	(0.5,2.8)	(2.0,0.5)	1.1	1.7	(0.5,3.0)	(2.0,0.5)
$R_{3,3}$	0.9	1.2	(1.8,0.4)	(2.0,0.5)	0.9	1.3	(1.6,0.4)	(2.0,0.5)	0.9	1.3	(1.6,0.4)	(2.0,0.5)	0.9	1.5	(1.5,0.4)	(2.0,0.5)	0.9	1.7	(1.1,0.8)	(1.8,0.6)
$R_{3,4}$	0.9	0.7	(1.9,0.5)	(2.0,0.5)	0.8	0.7	(1.9,0.5)	(2.0,0.5)	0.8	0.7	(1.5,0.6)	(2.0,0.5)	0.8	0.7	(1.5,0.6)	(2.0,0.5)	0.8	0.7	(1.5,0.6)	(2.0,0.5)
$R_{3,5}$	1.1	0.9	(2.0,0.4)	(2.0,0.5)	1.1	0.9	(2.0,0.4)	(2.0,0.5)	1.1	0.9	(1.9,0.5)	(2.0,0.5)	1.1	0.9	(1.8,0.6)	(1.5,0.7)	1.1	0.9	(1.7,0.6)	(1.8,0.6)
$R_{4,1}$	1.1	0.9	(2.0,0.4)	(2.0,0.5)	1.2	1.1	(2.0,0.4)	(2.0,0.5)	1.3	1.1	(2.0,0.4)	(2.0,0.4)	1.2	0.8	(2.0,0.5)	(2.1,0.5)	0.4	0.3	(2.0,0.5)	(2.0,0.5)
$R_{4,2}$	1.1	1.4	(1.4,0.7)	(2.0,0.5)	1.1	1.1	(1.4,0.5)	(2.0,0.5)	1.1	0.8	(1.4,0.4)	(2.0,0.5)	1.1	0.7	(1.3,0.5)	(2.0,0.5)	0.5	1.3	(1.3,0.4)	(2.0,0.5)
$R_{4,3}$	0.9	1.1	(1.7,0.5)	(2.0,0.5)	0.9	0.9	(1.7,0.5)	(2.0,0.5)	0.9	0.9	(1.9,0.4)	(2.0,0.5)	1.1	0.9	(1.9,0.3)	(2.0,0.5)	0.7	0.2	(1.8,0.4)	(2.0,0.5)
$R_{4,4}$	1.1	0.9	(1.6,0.6)	(2.0,0.5)	1.2	0.9	(2.0,0.5)	(2.0,0.5)	1.3	0.9	(2.0,0.5)	(2.0,0.5)	1.3	0.7	(2.0,0.5)	(2.0,0.5)	1.3	0.4	(2.0,0.5)	(2.0,0.5)
$R_{4,5}$	0.9	1.3	(1.8,0.4)	(2.0,0.5)	1.1	1.5	(1.7,0.4)	(2.0,0.5)	1.2	0.8	(1.6,0.5)	(2.0,0.5)	0.2	0.9	(1.7,0.3)	(2.0,0.5)	0.2	1.2	(1.6,0.3)	(2.0,0.5)
$R_{4,6}$	1.2	1.4	(1.5,0.6)	(2.0,0.5)	1.1	1.1	(1.5,0.5)	(2.0,0.5)	1.1	0.7	(2.2,0.3)	(2.0,0.5)	0.7	1.8	(1.6,0.4)	(2.0,0.5)	0.8	0.8	(2.0,0.3)	(2.0,0.5)
$R_{4,7}$	0.9	0.9	(2.2,0.3)	(2.0,0.5)	1.1	0.9	(2.1,0.4)	(2.0,0.5)	1.2	1.1	(2.1,0.4)	(2.0,0.5)	1.2	0.9	(2.1,0.4)	(2.0,0.5)	1.1	0.1	(2.2,0.5)	(2.0,0.5)
$R_{4,8}$	1.2	1.2	(2.0,0.5)	(2.0,0.5)	1.3	0.8	(2.0,0.5)	(2.0,0.5)	1.3	0.5	(2.0,0.5)	(2.0,0.5)	1.3	0.1	(2.0,0.5)	(2.0,0.5)	0.7	0.1	(2.0,0,5)	(2.0,0.5)
$R_{4,9}$	1.1	1.2	(2.0,0.4)	(2.0,0.5)	1.2	1.2	(1.5,0.6)	(2.0,0.5)	1.3	0.8	(1.5,0.6)	(2.0,0.5)	1.2	0.1	(1.5,0.5)	(2.0,0.5)	0.6	0.5	(2.0,0.4)	(2.0,0.5)
$R_{4,10}$	1.1	0.9	(1.6,0.6)	(2.0,0.5)	1.1	0.9	(1.6,0.4)	(2.0,0.5)	1.1	0.9	(1.6,0.3)	(2.0,0.5)	0.7	0.6	(1.5,0.3)	(2.0,0.5)	0.9	0.2	(1.4,0.2)	(2.0,0.5)
$R_{4,11}$	1.3	1.1	(2.1,0.6)	(2.0,0.5)	1.5	1.2	(1.6,0.9)	(2.0,0.5)	1.6	1.2	(1.9,0.7)	(2.0,0.5)	1.7	1.2	(2.0,0.7)	(2.0,0.5)	0.2	0.6	(2.1,0.4)	(2.0,0.5)
$R_{4,12}$	1.1	1.1	(1.7,0.5)	(2.0,0.5)	1.1	1.1	(1.6,0.5)	(2.0,0.5)	1.1	1.2	(1.5,0.6)	(2.0,0.5)	1.2	1.2	(1.5,0.4)	(2.1,0.3)	1.1	1.3	(1.4,0.4)	(2.0,0.3)
$R_{4,13}$	1.1	0.9	(1.7,0.6)	(2.0,0.5)	1.2	0.9	(1.6,0.6)	(2.0,0.5)	1.3	0.9	(1.5,0.7)	(2.0,0.5)	1.2	0.6	(1.5,0.6)	(2.0,0.5)	1.2	0.5	(1.5,0.4)	(2.0,0.5)
$R_{4,14}$	1.1	1.5	(1.3,0.8)	(2.0,0.5)	1.2	1.5	(1.4,0.7)	(2.0,0.5)	1.2	1.1	(1.4,0.7)	(1.8,0.5)	1.2	1.1	(1.5,0.5)	(1.8,0.5)	1.2	0.8	(1.4,0.6)	(1.8,0.4)
$R_{4,15}$	1.1	1.3	(1.5,0.6)	(2.0,0.5)	1.2	1.1	(1.5,0.5)	(2.0,0.5)	1.2	0.6	(2.1,0.4)	(2.0,0.5)	1.1	1.6	(2.0,0.4)	(2.0,0.5)	1.1	0.7	(2.0,0.5)	(2.0,0.5)
$R_{4,16}$	1.1	0.9	(1.7,0.6)	(2.0,0.5)	1.2	1.1	(1.8,0.5)	(2.0,0.5)	1.3	1.1	(1.8,0.5)	(2.0,0.5)	1.4	0.7	(1.7,0.5)	(2.0,0.5)	0.3	0.5	(1.7,0.4)	(2.0,0.5)
$R_{4,17}$	1.1	0.9	(1.7,0.7)	(2.0,0.5)	1.4	0.9	(2.2,0.6)	(2.0,0.5)	1.5	0.9	(2.3,0.6)	(2.0,0.4)	1.7	0.2	(2.1,0.8)	(1.9,0.7)	1.4	0.8	(2.2,0.8)	(2.0,0.5)
$R_{4,18}$	0.9	0.9	(2.2,0.3)	(2.0,0.5)	0.9	0.9	(1.9,0.4)	(2.0,0.5)	1.1	0.9	(1.8,0.5)	(2.0,0.5)	1.2	0.8	(1.9,0.5)	(1.9,0.4)	1.2	0.6	(1.9,0.4)	(2.0,0.3)
$R_{4,19}$	1.1	0.9	(2.0,0.5)	(2.0,0.5)	1.2	1.1	(2.0,0.5)	(2.0,0.5)	1.3	1.1	(2.0,0.5)	(2.0,0.5)	1.3	0.6	(1.9,0.5)	(2.0,0.5)	1.2	0.4	(2.1,0.5)	(2.0,0.5)
$R_{6,1}$	1.1	1.5	(1.1,0.9)	(2.0,0.5)	1.1	1.8	(1.2,0.8)	(2.0,0.5)	1.1	1.9	(1.2,0.8)	(2.0,0.5)	1.1	1.6	(1.2,0.6)	(1.9,0.5)	0.2	0.3	(1.9,0.3)	(1.8,0.5)
$R_{6,2}$	0.9	0.9	(1.6,0.6)	(2.0,0.5)	1.1	0.9	(1.9,0.5)	(2.0,0.5)	1.1	0.9	(1.5,0.7)	(2.0,0.5)	1.2	0.9	(1.5,0.5)	(2.0,0.4)	0.5	0.7	(1.5,0.3)	(2.0,0.4)
$R_{6,3}$	0.9	0.9	(1.7,0.5)	(2.0,0.5)	0.9	0.9	(1.7,0.5)	(2.0,0.5)	0.9	0.9	(1.7,0.5)	(2.0,0.5)	0.9	1.1	(1.8,0.3)	(2.0,0.5)	0.1	0.1	(1.5,0.8)	(1.8,0.8)
$R_{6,4}$	1.1	1.4	(0.5,2.6)	(2.0,0.5)	1.1	1.8	(1.3,0.8)	(2.0,0.5)	1.2	1.9	(1.8,0.5)	(2.0,0.5)	1.3	1.4	(1.9,0.6)	(2.0,0.5)	1.3	0.2	(1.9,0.4)	(2.0,0.5)
$R_{6,5}$	1.1	1.3	(0.6,2.0)	(2.0,0.5)	1.1	1.4	(0.9,1.4)	(2.0,0.5)	1.1	1.5	(1.1,1.1)	(2.0,0.5)	1.1	1.6	(1.2,0.9)	(2.0,0.5)	1.3	1.5	(1.8,0.5)	(2.0,0.5)

Note. Ag1 = Algorithm 1; Ag2 = Algorithm 2; Ag3 = Algorithm 3; Ag4 = Algorithm 4; F = the fixed-point iteration; S = the shifted fixed-point iteration; IO = the inner–outer iteration; INV = the inverse iteration; N = Newton's method.

TABLE 18 Numerical results for Test 4 in the work of Li et al.[12]

		Ag1		Ag2		Ag3		Ag4		F		S		IO		INV		N	
θ	n_p	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT
0.70	80	0.061744	10	0.073873	14	**0.037444**	5	0.302956	5	0.134881	25	0.323270	60	0.456618	75	0.557869	17	0.230405	6
	100	0.066884	11	0.084555	16	**0.038077**	5	0.303750	5	0.156366	27	0.354584	64	0.455653	79	0.626494	19	0.230894	6
	150	0.057755	9	0.059043	11	**0.036301**	5	0.307075	4	0.114683	20	0.279581	50	0.428358	66	0.466539	14	0.194080	5
	180	0.042494	7	0.041612	8	**0.035262**	5	0.299558	4	0.066365	12	0.214920	37	0.396252	53	0.303601	9	0.196900	5
0.85	80	0.077120	12	0.110396	21	**0.040905**	6	0.308074	5	0.200262	37	0.467886	82	0.531195	74	0.799087	24	0.233092	6
	100	0.077878	13	0.124308	23	**0.040900**	6	0.304961	5	0.234548	43	0.520655	93	0.548934	81	0.898732	27	0.233686	6
	150	0.056449	9	0.078234	14	**0.034575**	5	0.305904	5	0.135527	24	0.341861	59	0.479661	58	0.564158	17	0.234453	6
	180	0.040685	7	0.039527	7	**0.036678**	5	0.301053	4	0.074065	13	0.223737	39	0.413694	44	0.330959	10	0.194039	5
0.90	80	0.085672	14	0.128474	24	**0.039918**	6	0.308417	5	0.247514	44	0.535984	96	0.549532	76	0.933209	28	0.236574	6
	100	0.087616	15	0.166085	31	**0.041878**	6	0.313655	6	0.293068	53	0.647108	113	0.581858	87	1.104906	33	0.27395	7
	150	0.061762	10	0.082532	15	**0.041037**	6	0.306222	5	0.148754	27	0.346769	63	0.512203	56	0.608352	18	0.23542	6
	180	0.041069	7	0.044005	8	**0.034665**	5	0.303131	4	0.076789	14	0.224434	40	0.424948	41	0.367329	11	0.19678	5
0.95	80	0.093706	15	0.165734	31	**0.044009**	6	0.313973	6	0.311243	56	0.680685	118	0.598488	83	1.141271	34	0.277314	7
	100	0.113560	19	0.216767	41	**0.046937**	7	0.311981	6	0.418902	75	0.872187	154	0.641057	104	1.474419	44	0.271705	7
	150	0.063469	11	0.089535	17	**0.039757**	6	0.305145	5	0.162391	29	0.379718	67	0.528886	54	0.638188	19	0.235021	6
	180	0.040562	7	0.043697	8	**0.03441**7	5	0.300909	4	0.079728	14	0.225704	41	0.424916	39	0.369382	11	0.198371	5
0.99	80	0.113253	18	0.210311	40	**0.048344**	7	0.312860	6	0.407344	73	0.879781	150	0.638754	95	1.440216	43	0.27571	7
	100	0.155137	25	0.386693	71	**0.04896**9	7	0.322150	7	0.741852	132	1.528124	264	0.695860	155	2.485348	74	0.315349	8
	150	0.073223	12	0.090264	17	**0.042362**	6	0.306312	5	0.179351	31	0.402528	72	0.548413	53	0.700927	21	0.234184	6
	180	0.041406	7	0.042319	8	**0.034414**	5	0.303174	4	0.076592	14	0.219931	41	0.433867	37	0.369884	11	0.195264	5

Note. Ag1 = Algorithm 1; Ag2 = Algorithm 2; Ag3 = Algorithm 3; Ag4 = Algorithm 4; F = the fixed-point iteration; S = the shifted fixed-point iteration; IO = the inner–outer iteration; INV = the inverse iteration; N = Newton's method.

TABLE 19 The corresponding parameters of Algorithms 1–4 for Test 4 in the work of Li et al.[12]

	$\theta = 0.70$				$\theta = 0.85$				$\theta = 0.90$				$\theta = 0.95$				$\theta = 0.99$			
	Ag1	**Ag2**	**Ag3**	**Ag4**	**Ag1**	**Ag2**	**Ag3**	**Ag4**	**Ag1**	**Ag2**	**Ag3**	**Ag4**	**Ag1**	**Ag2**	**Ag3**	**Ag4**	**Ag1**	**Ag2**	**Ag3**	**Ag4**
n_p	α	β	(α, γ)	(α, γ)	α	β	(α, γ)	(α, γ)	α	β	(α, γ)	(α, γ)	α	β	(α, γ)	(α, γ)	α	β	(α, γ)	(α, γ)
80	1.2	1.5	(2.0,0.5)	(2.0,0.5)	1.3	1.5	(2.0,0.5)	(2.0,0.5)	1.3	1.6	(2.0,0.5)	(2.0,0.5)	1.4	1.7	(2.0,0.5)	(2.0,0.5)	1.4	1.7	(2.0,0.5)	(2.0,0.5)
100	1.3	1.4	(2.0,0.5)	(2.0,0.5)	1.3	1.6	(2.0,0.5)	(2.0,0.5)	1.4	1.7	(2.0,0.5)	(2.0,0.5)	1.5	1.7	(2.0,0.5)	(2.0,0.5)	1.5	1.8	(2.0,0.5)	(2.0,0.5)
150	1.2	1.3	(2.0,0.5)	(2.0,0.5)	1.2	1.4	(2.0,0.5)	(2.0,0.5)	1.2	1.4	(2.0,0.5)	(2.0,0.5)	1.2	1.5	(2.0,0.5)	(2.0,0.5)	1.3	1.5	(2.0,0.5)	(2.0,0.5)
180	1.1	1.2	(1.7,0.6)	(2.0,0.5)	1.1	1.2	(2.0,0.5)	(2.0,0.5)	1.1	1.2	(2.0,0.5)	(2.0,0.5)	1.1	1.2	(2.0,0.5)	(2.0,0.5)	1.1	1.2	(2.0,0.5)	(2.0,0.5)

Note. Ag1 = Algorithm 1; Ag2 = Algorithm 2; Ag3 = Algorithm 3; Ag4 = Algorithm 4; F = the fixed-point iteration; S = the shifted fixed-point iteration; IO = the inner–outer iteration; INV = the inverse iteration; N = Newton's method.

5 | CONCLUDING REMARKS

In this paper, we develop some algorithms for solving Equation (1) arising from the higher-order Markov Chains and the multilinear PageRank (4), and give their convergence analysis. By choosing suitable parameters, the proposed algorithms have faster convergence than the existing ones in the works of Li et al.[3] and Liu et al.[8] In the future, we will explore the optimal parameter for some structured tensors.

ACKNOWLEDGEMENTS

The authors would like to thank the referees for their helpful comments and suggestions. The first author was supported by the Opening Project of Guangdong Province Key Laboratory of Computational Science at the Sun Yat-sen University (Grant 2018008). The second author is supported in part by National Natural Science Foundation of China (Grants 11671158, U1811464, and 11771159), and Major Project (Grant 2016KZDXM025) and Innovation Team Project (Grant 2015KCXTD007) of Guangdong Provincial Universities. The third author was supported by the University of Macau (Grant MYRG2017-00098-FST, MYRG2018-00047-FST) and the Macao Science and Technology Development Fund (050/2017/A).

CONFLICT OF INTEREST

This work does not have any conflicts of interest.

ORCID

Dongdong Liu https://orcid.org/0000-0001-5621-2975

REFERENCES

1. Berman A, Plemmons JR. Nonnegative matrices in the mathematical science. Philadelphia, PA: SIAM; 1994.
2. Ching W-K, Ng MK. Markov chains: Models, algorithms and applications. New York, NY: Springer; 2006. International series on operations research and management science.
3. Li W, Ng MK. On the limiting probability distribution of a transition probability tensor. Linear Multilinear Algebra. 2014;62:362–385.
4. Chang KC, Zhang T. On the uniqueness and non-uniqueness of the positive Z-eigenvector for transition probability tensors. J Math Anal Appl. 2013;408(2):525–540.
5. Li W, Cui L-B, Ng MK. The perturbation bound for the Perron vector of a transition probability tensor. Numer Linear Algebra Appl. 2013;20(6):985–1000.
6. Gleich DF, Lim L-H, Yu Y. Multilinear PageRank. SIAM J Matrix Anal Appl. 2015;36:1507–1541.
7. Qi L. Eigenvalues of a real supersymmetric tensor. J Symb Comput. 2005;40:1302–1324.
8. Liu D, Li W, Vong S-W. The tensor splitting with application to solve multi-linear systems. J Comput Appl Math. 2018;330:75–94.
9. Parikh N, Boyd S. Proximal algorithms. Found Trends Optim. 2014;1(3):127–239.
10. Bertsekas DP. Projected Newton methods for optimization problems with simple constraints. SIAM J Control Optim. 1982;20(2):221–246.
11. Stewart GW, Sun J-G. Matrix perturbation theory. Cambridge, MA: Academic Press; 1990.
12. Li W, Liu D, Ng MK, Vong S-W. The uniqueness of multilinear PageRank vectors. Numer Linear Algebra Appl. 2017;24(6):e2017.
13. Lim LH. Singular values and eigenvalues of tensors: a variational approach. Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 05); 2005 Dec 13–15; Puerto Vallarta, Mexico. Piscataway, NJ: IEEE.
14. Varga RS. On diagonal dominance arguments for bounding $\|a^{-1}\|_\infty$. Linear Algebra Appl. 1976;14:211–217.
15. Ni Q, Qi L. A quadratically convergent algorithm for finding the largest eigenvalue of a nonnegative homogeneous polynomial map. J Glob Optim. 2015;61(4):627–641.
16. Kolda TG, Bader BW. Tensor decompositions and applications. SIAM Review. 2009;51:455–500.
17. Kolda TG. Multilinear operators for higher-order decompositions. Albuquerque, NM: Sandia National Laboratories; 2006. Technical Report SAND2006-2081.
18. Bader BW, Kolda TG. Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. ACM Trans Math Softw. 2006;32(4):635–653.
19. Raftery AE. A model for high-order Markov chains. J Royal Stat Soc Ser B Methodol. 1985;47:528–539.
20. Raftery A, Tavaré S. Estimation and modelling repeated patterns in high order Markov chains with the mixture transition distribution model. J Royal Stat Soc Ser C Appl Stat. 1994;43:179–199.

21. Berchtold A, Raftery A. The mixture transition distribution model for high-order Markov chains and non-Gaussian time series. Statistical Science. 2002;7:328–356.
22. Li X, Ng MK, Ye Y. Multicomm: finding local community structure in multi-dimensional networks. IEEE Trans Knowl Data Eng. 2014;26:929–941.
23. Page L, Brin S, Motwani R, Winograd T. The PageRank citation ranking: bringing order to the web. Stanford, CA: Stanford University; 1999. Technical report.

How to cite this article: Liu D, Li W, Vong S-W. Relaxation methods for solving the tensor equation arising from the higher-order Markov chains. *Numer Linear Algebra Appl*. 2019;e2260. https://doi.org/10.1002/nla.2260