The general tensor regular splitting iterative method for multilinear PageRank problem

Shuting Tang, Xiuqin Deng and Rui Zhan

School of Mathematics and Statistics, Guangdong University of Technology, Guangzhou, 510006, China

* Correspondence: Email: zhanrui@gdut.edu.cn.

Abstract: The paper presents an iterative scheme called the general tensor regular splitting iterative (GTRS) method for solving the multilinear PageRank problem, which is based on a (weak) regular splitting technique and further accelerates the iterative process by introducing a parameter. The method yields familiar iterative schemes through the use of specific splitting strategies, including fixed-point, inner-outer, Jacobi, Gauss-Seidel and successive overrelaxation methods. The paper analyzes the convergence of these solvers in detail. Numerical results are provided to demonstrate the effectiveness of the proposed method in solving the multilinear PageRank problem.

Keywords: multilinear PageRank; tensor; regular splitting; convergence

Mathematics Subject Classification: 65F10, 65H10

1. Introduction

As the most crucial algorithm for web ranking in the web search engines, Google’s PageRank has been extensively researched over the years [1–8]. Gleich et al. [9] extended PageRank to higher-order Markov chains and proposed the following multilinear PageRank problem:

$$\mathbf{x} = \alpha \mathcal{P} \mathbf{x}^{m-1} + (1 - \alpha) \mathbf{v}, \quad (1.1)$$

where $\alpha \in (0, 1)$ is a parameter, $\mathbf{v} \in \mathbb{R}^n$ is a stochastic vector, i.e., $\mathbf{v} \geq 0, \|\mathbf{v}\|_1 = 1$, $\mathbb{R}^n$ is the set of all $n$-dimensional real vectors. The stochastic solution $\mathbf{x} \in \mathbb{R}^n$ is called the multilinear PageRank vector. Here $\mathcal{P} = (p_{i_1, i_2, \ldots, i_m}) (\forall i_2, \ldots, i_m \in \langle n \rangle)$ is an $m^{th}$-order $n$-dimensional stochastic tensor representing an $(m - 1)^{th}$-order Markov chain, namely,

$$p_{i_1, i_2, \ldots, i_m} \geq 0, \sum_{i_1=1}^n p_{i_1, i_2, \ldots, i_m} = 1,$$
where $\langle n \rangle = \{1, 2, \cdots, n\}$. The tensor-vector product $P x^{m-1}$ is a vector in $\mathbb{R}^n$ and its definition is given in Section 2.

In particular, the multilinear PageRank model (1.1) can be transformed into higher-order Markov chain when we take $\alpha = 1$ (see [10] for the general case). When $m = 2$, (1.1) simplifies to the classical PageRank as described in [11].

The model (1.1) can be transformed into

$$x = \bar{P} x^{m-1},$$

where $\bar{P} = \alpha P + (1 - \alpha) V$ and $V$ is an $m$th-order $n$-dimensional tensor with $(V)_{i_1 \cdots i_m} = v_i, \forall i, i_2, \cdots, i_m \in \langle n \rangle$. Clearly, $\bar{P}$ is also a transition probability tensor.

In recent years, the multilinear PageRank problem has attracted much attention. The theory of existence and uniqueness of solutions for the multilinear PageRank problem is a fundamental basis for analyzing the convergence of algorithms. Gleich et al. [9] pointed to the fact that the multilinear PageRank vector is unique in (1.1) for $\alpha < \frac{1}{m-1}$. The above condition can be easily applied to compute the stochastic solution of (1.1) when $\alpha < \frac{1}{2}$. However, when $\frac{1}{2} < \alpha < 1$, especially when $\alpha \approx 1$, it becomes increasingly challenging to solve (1.1). This challenge inspired researchers to search for more general but stricter conditions, such as those discussed in studies by Li et al. [12, 13], Huang and Wu [14], Fasino and Tudisco [15] and Liu et al. [16]. It should be noted that we cannot theoretically compare the merits of these conditions and there is no optimal condition in numerical experiments.

Gleich et al. first proposed Newton’s method to solve this problem in [9]. Subsequently, Meini and Poloni [17] and Guo et al. [18] proposed the Perron Newton iterative algorithm and the multi-step Newton method, respectively. These algorithms mentioned above involve gradient computation, which have shown excellent performance. But when solving asymmetric tensor models, the difficulty of gradient computation can lead to expensive calculations. Unlike gradient algorithms, non-gradient algorithms typically do not require the gradient information of the objective function. Instead, they estimate the optimal solution of the objective function through other methods. Non-gradient algorithms have proven to be effective in solving this problem and have gained a significant amount of attention in recent years. For example, Gleich et al. [9] proposed the fixed-point method, the shifted fixed-point method, the inner-outer method and the inverse method. Liu et al. [19] introduced several relaxation methods. Other novel methods have been proposed by various scholars in an effort to achieve better convergence performances (see References [20–23] and their respective citations). Hence, the problem about how to design high-speed, robust and flexible non-gradient algorithms has become the key challenge in solving the multilinear PageRank problem.

As an important class of non-gradient algorithms, the splitting algorithm has significant applications in tensor computations. For example, Liu et al. [24] proposed a tensor splitting algorithm for solving tensor equations and high-order Markov chain models. Cui et al. [25] proposed an iterative refinement method by using higher-order singular value decomposition to solve general multilinear systems. Jiang and Li [26] proposed a new preconditioner for improving the AOR-type method. Cui and Zhang [27] proposed a new constructive way to estimate bounds on the H-eigenvalues of two kinds of interval tensors. This paper will explore a tensor splitting algorithm for solving the multilinear PageRank problem.

The main contributions of this paper can be outlined as follows:
(1) The general tensor regular splitting (GTRS) iterative method is proposed to address the multilinear PageRank problem.

(2) Additionally, we offer five typical splitting methods of the GTRS iteration.

(3) We give the convergence analysis for the proposed method based on the uniqueness condition provided by Li et al. [13].

(4) Several numerical examples are provided and the outcomes significantly outperform the existing ones in most cases.

The reminder of this work is organized as follows. In Section 2, we present some essential definitions and key properties. By making use of the (weak) regular splitting of the coefficient matrix $I - \alpha P$ and $x^{m-2}$, we propose the GTRS iterative method in Section 3. Section 4 is devoted to the convergence analysis of the GTRS method. In Section 5, some application experiments are presented to demonstrate the effectiveness of our method. Finally, the conclusion is given in Section 6.

2. Preliminaries

In this section, we present a brief overview of the notations and definitions that are necessary for this paper.

An $m^{th}$-order $n$-dimensional tensor $A$ with $n^m$ entries is defined as

$$A = (a_1, \ldots, a_m), a_{i_1, \ldots, i_m} \in \mathbb{R}, i_j \in \langle n \rangle, j = 1, \ldots, m,$$

where $\mathbb{R}$ is the real field. $A$ is called non-negative (positive) if $a_{i_1, \ldots, i_m} \geq 0$ ($a_{i_1, \ldots, i_m} > 0$).

For any two matrices $A = (a_{ij})_{n \times n}, B = (b_{ij})_{n \times n} \in \mathbb{R}^n$, $A \geq B$ means that $a_{ij} \geq b_{ij}$ for $i, j \in \langle n \rangle$.

**Definition 1.** [13] Let $A$ be an $m^{th}$-order $n$-dimensional tensor, $x$ be an $n$-dimensional vector and $x_i$ represent the $i^{th}$ entry of $x$. $A x^{m-r}$ is an $r^{th}$-order $n$-dimensional tensor is given by

$$(A x^{m-r})_{i_1, \ldots, i_r} = \sum_{i_{r+1}, \ldots, i_m \in \langle n \rangle} a_{i_1, \ldots, i_m} x_{i_{r+1}} \cdots x_{i_m}.$$  \hspace{1cm} (2.1)

**Definition 2.** [13] Let $A$ be an $m^{th}$-order $n$-dimensional tensor and both $x$ and $y$ be $n$-dimensional tensors. Then we get the following definition

$$A(x^{m-r} - y^{m-r}) = A x^{m-r} - A y^{m-r}.$$ \hspace{1cm} (2.2)

Particularly, when $r = 1$ and $r = 2$, we obtain $A x^{m-1}$ and $A x^{m-2}$ from (2.1). Noticed that $A x^{m-1} = A x^{m-2} x$. Then, (2.2) can be expressed as

$$A(x^{m-1} - y^{m-1}) = A x^{m-1} - A y^{m-1}, A(x^{m-2} - y^{m-2}) = A x^{m-2} - A y^{m-2}.$$  

**Definition 3.** [13] Let $A$ be an $m^{th}$-order $n$-dimensional tensor and both $x$ and $y$ be $n$-dimensional tensors. $A^{(k)}_{xy}$ is an $n \times n$ matrix whose $(i, j)^{th}$ entry is defined as

$$(A^{(k)}_{xy})_{i, j} = \sum_{i_2, \ldots, i_{k-1}, i_{k+1}, \ldots, i_m \in \langle n \rangle} a_{i_1, \ldots, i_m} x_{i_2} \cdots x_{i_{k-1}} y_{i_{k+1}} \cdots y_{i_m}.$$
It should be noted that (1.1) can be rewritten as

\[(I - \alpha \mathcal{P} x^{m-2}) x = (1 - \alpha) \nu.\]  

(2.3)

**Definition 4.** A real square matrix $\hat{A}$ is defined as an M-matrix when it satisfies the following conditions: $\hat{A}$ is a Z-matrix (i.e., all of its off-diagonal elements $\hat{a}_{ij} \leq 0$) and $\hat{A}$ can be expressed as $hI - \hat{B}$, where $\hat{B}$ is a nonnegative matrix with $\hat{b}_{ij} \geq 0$, and its spectral radius $\rho(\hat{B}) < h$.

**Definition 5.** [28] Let $O$ be a null $n \times n$ matrix and $\hat{A}$, $\hat{M}$ and $\hat{N}$ be $n \times n$ real matrices and $\hat{A} = \hat{M} - \hat{N}$ is a regular splitting of matrix $\hat{A}$ if $\hat{M}$ is nonsingular with $\hat{M}^{-1} \geq O$ and $\hat{N} \geq O$. Similarly, $\hat{A} = \hat{M} - \hat{N}$ is a weak regular splitting of matrix $\hat{A}$ if $\hat{M}$ is nonsingular with $\hat{M}^{-1} \geq O$ and $\hat{M}^{-1} \hat{N} \geq O$. $\hat{A} = \hat{M} - \hat{N}$ is a convergent splitting of matrix $\hat{A}$ if $\hat{M}$ is nonsingular with $\rho(\hat{M}^{-1} \hat{N}) < 1$.

**Definition 6.** Let $z \in \mathbb{R}^n$ be a stochastic vector, $0 \in \mathbb{R}^n$ and $z_+ = \max(z, 0)$. We define the projection $\text{proj}(\cdot)$ as follows:

$$\text{proj}(z) = \frac{z_+}{\|z_+\|_1}.$$  

It is clear that $\text{proj}(\cdot)$ is a stochastic vector.

### 3. The GTRS Iterative Method

Building upon the success of the general inner-outer iterative method (see [29]) for solving the typical PageRank problem, we introduce the GTRS iterative method as an efficient solution for computing the multilinear PageRank vector.

Let

$$I - \alpha \mathcal{P} x^{m-2} = \tilde{M}_x - \tilde{N}_x$$  

(3.1)

be a (weak) regular splitting. Based on (3.1), the iterative method, i.e., GTRS, for (1.1) is constructed as follows:

$$\left(\tilde{M}_{x_k} - \phi \tilde{N}_{x_k}\right) y_{k+1} = (1 - \phi) \tilde{N}_{x_k} x_k + b, x_{k+1} = \text{proj}(y_{k+1}), k = 0, 1, 2, \ldots$$  

(3.2)

with $0 < \phi < 1$ and $b = (1 - \alpha) \nu$. The scheme (3.2) is derived from the matrix splitting

$$I - \alpha \mathcal{P} x^{m-2} = \tilde{M}_{x_k} - \tilde{N}_{x_k},$$

where $\tilde{M}_{x_k} = \tilde{M}_{x_k} - \phi \tilde{N}_{x_k}$ and $\tilde{N}_{x_k} = (1 - \phi) \tilde{N}_{x_k}$.

Let $\tilde{U}_x \geq 0$ be the strictly upper-triangular matrix of the matrix $\mathcal{P} x^{m-2}$, $\tilde{D}_x \geq 0$ be a diagonal matrix and $\tilde{L}_x \geq 0$ be a lower-triangular matrix such that

$$\mathcal{P} x^{m-2} = \tilde{D}_x + \tilde{U}_x + \tilde{L}_x.$$  

(3.3)

Next, we give several typical practical choices of the matrices $\tilde{M}_x$ and $\tilde{N}_x$. Derived from the above matrix splitting (3.3), those well-known iterative methods for solving (1.1) are presented below:

1. Let $\tilde{M}_{x_k} = I$ and $\tilde{N}_{x_k} = \alpha \mathcal{P} x_k^{m-2}$; then, we can obtain the fixed point iteration (denote by GTRS-FP):

$$\left(I - \phi \alpha \mathcal{P} x_k^{m-2}\right) y_{k+1} = (\alpha - \phi \alpha) \mathcal{P} x_k^{m-2} x_k + b.$$  

(3.4)
4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.

4. Convergence analysis

In this section, we analyze the convergence of the GTRS iterative method. We establish the convergence of the GTRS iteration by using the uniqueness condition of the solution to the multilinear PageRank problem in [13]. At first, we give several lemmas which will be used later.
Lemma 1. [13] Let $\mathbf{x}$ and $\mathbf{y}$ be $n$-dimensional stochastic vectors, and let $\mathcal{J}^{(k)}(k = 2, 3, \cdots, m)$ be $m^{th}$-order $n$-dimensional tensors defined as

\[
\left(\mathcal{J}^{(k)}\right)_{i_1, \ldots, i_k} = \sigma^{(k)}_{i_1, i_2, \ldots, i_{k-1}, i_{k+1}, \ldots, i_m}, \quad \forall i_k \in \langle n \rangle,
\]

where $\sigma^{(k)}_{i_1, i_2, \ldots, i_{k-1}, i_{k+1}, \ldots, i_m} \in \mathbb{R}$ for any $i_k \in \langle n \rangle$, $l = 1, 2, \cdots, k - 1, k + 1, \cdots, i_m$ and $k = 2, 3, \cdots, m$; then,

\[
\mathcal{J}^{(k)}_{xy} \Delta \mathbf{x} = 0,
\]

where $\Delta \mathbf{x} = \mathbf{x} - \mathbf{y}$.

Lemma 2. [13] Let $\mathcal{P}$ be an $m^{th}$-order stochastic tensor and $\mathbf{v}$ be a stochastic vector; then, the multilinear PageRank model (1.1) has the unique solution if

\[
\alpha < \frac{1}{\min\{\mu, \nu\}},
\]

where $\mu = \min_{\mathcal{J}^{(k)}, k = 2, 3, \cdots, m} \mu(\mathcal{J}^{(2)}, \cdots, \mathcal{J}^{(m)})$ and $\nu = \min_{\mathcal{J}^{(k)}, k = 2, 3, \cdots, m} \nu(\mathcal{J}^{(2)}, \cdots, \mathcal{J}^{(m)})$.

Lemma 3. Let $\mathcal{P}$ be an $m^{th}$-order $n$-dimensional stochastic tensor, and let $\mathbf{y}$ and $\mathbf{z}$ be $n$-dimensional stochastic vectors; then, we have

\[
\|\mathcal{P}\mathbf{x}^{m-2}(\mathbf{y} - \mathbf{z})\|_1 \leq \gamma \|\mathbf{y} - \mathbf{z}\|_1,
\]

where $\gamma = \min\{\tilde{\gamma}, \tilde{\gamma}\}$ with $\tilde{\gamma} = 1 - \min_{i_3, \ldots, i_m \in \langle n \rangle} \sum_{i_1=1}^{n} \min_{i_2 \in \langle n \rangle} p_{i_1 i_2, \ldots, i_m}$ and $\tilde{\gamma} = \max_{i_3, \ldots, i_m \in \langle n \rangle} \sum_{i_1=1}^{n} \max_{i_2 \in \langle n \rangle} p_{i_1 i_2, \ldots, i_m} - 1$.

Proof. Let $\Delta \mathbf{y} = \mathbf{y} - \mathbf{z}$ and $\Delta \mathbf{y}_i$ be the $i$-th entry of $\Delta \mathbf{y}$, and let $\mathcal{J}^{(2)}$ be defined in Lemma 1; we have

\[
\|\mathcal{P}\mathbf{x}^{m-2}\Delta \mathbf{y}\|_1 = \sum_{i=1}^{n} \left| \sum_{i_{j} \in \langle n \rangle} \left( p_{i_{j} i_{j+1}, \ldots, i_m} - \sigma^{(2)}_{i_{j}, i_{j+1}, \ldots, i_m} \right) \Delta \mathbf{y}_{i_{j} i_{j+1} \cdots i_m} \right| \\
\leq \sum_{i=1}^{n} \sum_{i_{j} \in \langle n \rangle} \left| p_{i_{j} i_{j+1}, \ldots, i_m} - \sigma^{(2)}_{i_{j}, i_{j+1}, \ldots, i_m} \right| \|\Delta \mathbf{y}_{i_{j} i_{j+1} \cdots i_m}\|_1 \overset{(4.1)}{=} \sum_{i=1}^{n} \left| p_{i_{j} i_{j+1}, \ldots, i_m} - \sigma^{(2)}_{i_{j}, i_{j+1}, \ldots, i_m} \right| \|\Delta \mathbf{y}\|_1.
\]

Substituting $\sigma^{(2)}_{i_{j}, i_{j+1}, \ldots, i_m} = \min_{i_j} p_{i_{j} i_{j+1}, \ldots, i_m}$ and $\sigma^{(2)}_{i_{j}, i_{j+1}, \ldots, i_m} = \max_{i_j} p_{i_{j} i_{j+1}, \ldots, i_m}$ into (4.1) respectively, it follows that

\[
\|\mathcal{P}\mathbf{x}^{m-2}\Delta \mathbf{y}\|_1 \leq \tilde{\gamma} \|\Delta \mathbf{y}\|_1,
\]

and

\[
\|\mathcal{P}\mathbf{x}^{m-2}\Delta \mathbf{y}\|_1 \leq \tilde{\gamma} \|\Delta \mathbf{y}\|_1,
\]

where

\[
\tilde{\gamma} = 1 - \min_{i_3, \ldots, i_m \in \langle n \rangle} \sum_{i_1=1}^{n} \min_{i_2 \in \langle n \rangle} p_{i_1 i_2, \ldots, i_m} \text{ and } \tilde{\gamma} = \max_{i_3, \ldots, i_m \in \langle n \rangle} \sum_{i_1=1}^{n} \max_{i_2 \in \langle n \rangle} p_{i_1 i_2, \ldots, i_m} - 1.
\]

Let $\gamma = \min\{\tilde{\gamma}, \tilde{\gamma}\}$; we have

\[
\|\mathcal{P}\mathbf{x}^{m-2}\Delta \mathbf{y}\|_1 \leq \gamma \|\Delta \mathbf{y}\|_1.
\]

This completes the proof. \qed

AIMS Mathematics

Volume 9, Issue 1, 1443–1471.
**Lemma 4.** [13] Let \( P \) be an \( m \)-th-order \( n \)-dimensional stochastic tensor, and let \( x, y \) and \( z \) be \( n \)-dimensional stochastic vectors. For any set \( J^{(k)}(k=3,\cdots,m) \), we have
\[
\left\| P \left( x^{m-2} - y^{m-2} \right) z \right\|_1 \leq \tilde{\mu} \left( J^{(3)}, \cdots, J^{(m)} \right) \|Ax\|_1,
\]
where
\[
\tilde{\mu} \left( J^{(3)}, \cdots, J^{(m)} \right) = \max_{1 \leq i \leq (n)} \max_{k=3} \max_{i_1=1}^{n} \left| p_{i_1,i_k} - \sigma^{(k)}_{i_1,i_2,i_k,i_{k+1},\cdots,i_{m-1}} \right|.
\]

**Lemma 5.** [19] Let \( \hat{y} = (\hat{y}_i) \in \mathbb{R}^n \) with \( \sum_{i=1}^{n} \hat{y}_i = 1 \); \( z = (z_i) \) is a stochastic vector. If \( y = \text{proj}(\hat{y}) \), then \( \|y - z\|_1 \leq 2\|\hat{y} - z\|_1 \).

**Lemma 6.** Let \( \hat{y} = (\hat{y}_i) \in \mathbb{R}^n \); \( z = (z_i) \) is a stochastic vector. If \( y = \text{proj}(\hat{y}) \), then \( \|y - z\|_1 \leq 2\|\hat{y} - z\|_1 \).

**Proof.** Since \( y = \text{proj}(\hat{y}) = \frac{\hat{y}}{\|\hat{y}\|_1} \), it follows that
\[
\|y - z\|_1 = \|\text{proj}(\hat{y}) - z\|_1 = \left\| \frac{\hat{y}}{\|\hat{y}\|_1} - z \right\|_1 = \left\| \frac{\hat{y} - z}{\|\hat{y}\|_1} \right\|_1 \leq \|\hat{y} - z\|_1 + \left| 1 - \|\hat{y}\|_1 \right| \leq 2\|\hat{y} - z\|_1.
\]

The proof is completed. \( \square \)

**Theorem 1.** Let \( P \) be an \( m \)-th-order \( n \)-dimensional stochastic tensor, \( \alpha < \frac{1}{\min_{\mu,\nu}} \) and \( x \) be the exact solution of (1.1). Then, for an arbitrary initial stochastic guess \( x_0 \), the iterative sequence \( \{x_k\}_{k=0}^{\infty} \) generated by (3.2) is convergent and
\[
\|x_k - x\|_1 \leq \delta_{x_k-1} \cdots \delta_{x_0} \|x_0 - x\|_1,
\]
where \( \mu_{x_k} = \frac{(1-\phi)\|\tilde{M}_{x_k}^{-1}\tilde{S}_{x_k}\|_{1} + \alpha \mu(J^{(3)}, \cdots, J^{(m)})\|\tilde{M}_{x_k}^{-1}\|_{1}}{1 - \phi \|\tilde{M}_{x_k}^{-1}\tilde{S}_{x_k}\|_{1}} \) and \( \delta_{x_k} = 2\mu_{x_k} \).

**Proof.** By (3.2), we have
\[
(M_{x_k} - \phi\tilde{N}_{x_k}) y_{k+1} = (1 - \phi)\tilde{N}_{x_k} x_k + b. \tag{4.2}
\]
Note that \( x \) is the solution of (1.1); hence,
\[
(M_{x_k} - \phi\tilde{N}_{x_k}) x = (1 - \phi)\tilde{N}_{x_k} x + b. \tag{4.3}
\]
Let \( \hat{e}_k = y_k - x \) and \( e_k = x_k - x \). Subtracting (4.3) from (4.2) yields
\[
\hat{M}_{x_k} \hat{e}_{k+1} = \phi\tilde{N}_{x_k} \hat{e}_{k+1} + (1 - \phi)\tilde{N}_{x_k} e_k + (\tilde{N}_{x_k} - \hat{M}_{x_k}) x + (\hat{M}_{x_k} - \tilde{N}_{x_k}) x
\]
\[
= \phi\tilde{N}_{x_k} \hat{e}_{k+1} + (1 - \phi)\tilde{N}_{x_k} e_k + (\alpha P x_k^{m-2} - I) x + (1 - \alpha P x_k^{m-2}) x
\]
\[
= \phi\tilde{N}_{x_k} \hat{e}_{k+1} + (1 - \phi)\tilde{N}_{x_k} e_k + \alpha P (x_k^{m-2} - x^{m-2}) x.
\]
Thus, subtracting (4.8) from (4.7), we have
\[ (I - \phi \tilde{M}_x^{-1} \tilde{N}_x) \hat{e}_{k+1} = (1 - \phi) \tilde{M}_x^{-1} \tilde{N}_x e_k + \alpha \tilde{M}_x^{-1} \mathcal{P}(x_k^{m-2} - x^{m-2}) x. \] (4.4)

By taking the 1-norm of both sides of (4.4) and Lemma 4 we get
\[
\left( 1 - \phi \left\| \tilde{M}_x^{-1} \tilde{N}_x \right\|_1 \right) \| \hat{e}_{k+1} \|_1 \\
\leq \left[ (1 - \phi) \left\| \tilde{M}_x^{-1} \tilde{N}_x \right\|_1 + \alpha \tilde{\mu} \left( J^{(3)}, \ldots, J^{(m)} \right) \left\| \tilde{M}_x^{-1} \right\|_1 \right] \| e_k \|_1.
\]

Thus
\[
\| \hat{e}_{k+1} \|_1 \leq \frac{(1 - \phi) \left\| \tilde{M}_x^{-1} \tilde{N}_x \right\|_1 + \alpha \tilde{\mu} \left( J^{(3)}, \ldots, J^{(m)} \right) \left\| \tilde{M}_x^{-1} \right\|_1}{1 - \phi \left\| \tilde{M}_x^{-1} \tilde{N}_x \right\|_1} \| e_k \|_1 = \mu_x \| e_k \|_1. \]

From (4.5) and Lemma 6, we obtain
\[
\| e_{k+1} \|_1 \leq 2 \| \hat{e}_{k+1} \|_1 \leq \delta_x \| e_k \|_1 \leq \delta_x \delta_{x_{k-1}} \cdots \delta_{x_0} \| e_0 \|_1.
\]

Then the GTRS iteration converges linearly with the convergence rate \( \delta_x \) when \( \mu_x < \frac{1}{2} \). \( \Box \)

**Remark 1.** In Theorem 1, the value of \( \delta_x \) is associated with \( x_k \), which is difficult to prove in practical scenarios. Therefore, we will provide convergence analysis for specific splittings of the GTRS iterative method, such as (3.4)–(3.8).

**Theorem 2.** Let \( \mathcal{P} \) be an \( m \)-th-order \( n \)-dimensional stochastic tensor; \( x \) be a solution of (1.1), and \( \alpha < \min \left\{ \frac{1}{\phi \delta(J^{(3)}, \ldots, J^{(m)}) + \gamma}, \frac{1}{\min[\mu, \nu]} \right\} \); then, for an arbitrary initial stochastic guess \( x_0 \), the iterative sequence \( \{x_k\}_{k=0}^\infty \) generated by (3.4) is convergent and
\[
\| x_k - x \|_1 \leq \varsigma^k \| x_0 - x \|_1,
\]
where
\[
\varsigma = \frac{\phi \delta(J^{(3)}, \ldots, J^{(m)}) + (\alpha - \phi) \gamma}{1 - \phi \delta(J^{(3)}, \ldots, J^{(m)}) - \alpha(\alpha - \phi) \gamma}.
\]

**Proof.** It is easy to verify that \( \alpha < \frac{1}{\min[\mu, \nu]} \) when \( \alpha < \min \left\{ \frac{1}{\phi \delta(J^{(3)}, \ldots, J^{(m)}) + \gamma}, \frac{1}{\min[\mu, \nu]} \right\} \). According to Lemma 2, the solution \( x \) is unique.

Set \( \hat{e}_k = y_k - x \) and \( e_k = x_k - x \). From (3.4), it follows that
\[
y_{k+1} = \phi \alpha \mathcal{P} x_k^{m-2} y_{k+1} + (\alpha - \phi \alpha) \mathcal{P} x_k^{m-2} x_k + b.
\] (4.7)

Note that \( x \) is the solution of (1.1) yields
\[
x = \phi \alpha \mathcal{P} x^{m-2} x + (\alpha - \phi \alpha) \mathcal{P} x^{m-2} x + b.
\] (4.8)

Subtracting (4.8) from (4.7), we have
\[
\hat{e}_{k+1} = \phi \alpha \mathcal{P} x_k^{m-2} \hat{e}_{k+1} + \alpha \mathcal{P} (x_k^{m-2} - x^{m-2}) x + (\alpha - \phi \alpha) \mathcal{P} x_k^{m-2} e_k.
\] (4.9)

By (4.7), we have
\[
\sum_{i=1}^{n} ((I - \phi \alpha \mathcal{P} x_k^{m-2}) y_{k+1})_i = (\alpha - \phi \alpha) \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij_1 \ldots j_m} x_{k,i_1} \cdots x_{k,i_m} + \sum_{i=1}^{n} b_i
\]
\[
= \alpha - \phi \alpha + 1 - \alpha
\]
\[
= 1 - \phi \alpha.
\]
On the other hand

\[ 1 - \phi \alpha = \sum_{i=1}^{n} y_{k+1,i} - \phi \alpha \sum_{i=1}^{n} \sum_{l, j, 2, \ldots, m \in \mathcal{N}} P_{1, i, j, \ldots, m} y_{k+1, i, j, \ldots, m} \]

\[ = \sum_{i=1}^{n} y_{k+1,i} - \phi \alpha \sum_{i=1}^{n} y_{k+1,i}. \]

Therefore, we get

\[ \sum_{i=1}^{n} y_{k+1,i} = 1, \sum_{i=1}^{n} \hat{e}_{k+1,i} = 0. \tag{4.10} \]

Combining (4.9) and Lemmas 3 and 4 together gives

\[ \| \hat{e}_{k+1} \|_1 \leq \phi \alpha \gamma \| \hat{e}_{k+1} \|_1 + \phi \alpha \mu \left( T^{(3)}, \ldots, T^{(m)} \right) \| e_0 \|_1 + (\alpha - \phi \alpha) \gamma \| e_k \|_1. \]

It is noted that

\[ \| \hat{e}_{k+1} \|_1 \leq \frac{\phi \alpha \mu \left( T^{(3)}, \ldots, T^{(m)} \right) + (\alpha - \phi \alpha) \gamma}{1 - \phi \alpha \gamma} \| e_k \|_1 = \zeta \| e_k \|_1. \]

Then by (4.10) and Lemma 5, we get

\[ \| e_{k+1} \|_1 \leq \| \hat{e}_{k+1} \|_1 \leq \zeta \| e_k \|_1 \leq \zeta^{k+1} \| e_0 \|_1. \]

By the above proof, we have that \( \zeta < 1 \); then, the proof is completed. \( \square \)

**Theorem 3.** Let \( \mathcal{P} \) be an \( m \)th-order \( n \)-dimensional stochastic tensor; \( \mathbf{x} \) be a solution of (1.1) and \( \alpha < \min \left\{ 1, \frac{1}{\phi \mu \left( T^{(3)}, \ldots, T^{(m)} \right) + \gamma}, \frac{1}{\min \{ \mu, \gamma \}} \right\} \); then, for an arbitrary initial stochastic guess \( \mathbf{x}_0 \), the iterative sequence \( \{ \mathbf{x}_k \}_{k=0}^{\infty} \) generated by (3.5) is convergent and

\[ \| \mathbf{x}_k - \mathbf{x} \|_1 \leq \zeta^k \| \mathbf{x}_0 - \mathbf{x} \|_1, \]

where \( \zeta = \frac{(\beta + \phi \alpha - \phi \beta) \mathcal{P} x_k^{m-2} y_{k+1} + (1 - \phi) (\alpha - \beta) \mathcal{P} x_k^{m-2} \mathbf{b}}{1 - (\beta + \phi \alpha - \phi \beta) \gamma} \).

**Proof.** Clearly, we have the unique solution \( \mathbf{x} \) if \( \alpha < \min \left\{ 1, \frac{1}{\phi \mu \left( T^{(3)}, \ldots, T^{(m)} \right) + \gamma}, \frac{1}{\min \{ \mu, \gamma \}} \right\} \) from Lemma 2.

By (3.5), we have

\[ y_{k+1} = (\beta + \phi \alpha - \phi \beta) \mathcal{P} x_k^{m-2} y_{k+1} + (1 - \phi) (\alpha - \beta) \mathcal{P} x_k^{m-2} \mathbf{b}. \]

Let \( \hat{e}_k = y_k - \mathbf{x} \) and \( e_k = y_k - \mathbf{x} \). Then, we can get the following equation:

\[ \hat{e}_{k+1} = (\beta + \phi \alpha - \phi \beta) \mathcal{P} x_k^{m-2} \hat{e}_{k+1} + \alpha \mathcal{P} (x_k^{m-2} - x^{m-2}) \mathbf{x} + (1 - \phi) (\alpha - \beta) \mathcal{P} x_k^{m-2} e_k. \tag{4.11} \]

By the same proof as in Theorem 2, we get

\[ \sum_{i=1}^{n} (y_{k+1,i} - (\beta + \phi \alpha - \phi \beta) \mathcal{P} x_k^{m-2} y_{k+1}) = (1 - \phi)(\alpha - \beta) \sum_{i=1}^{n} \sum_{l, j, 2, \ldots, m \in \mathcal{N}} P_{1, i, j, \ldots, m} y_{k+1, i, j, \ldots, m} + \sum_{i=1}^{n} b_i \]

\[ = (1 - \phi)(\alpha - \beta) + 1 - \alpha \]

\[ = 1 - \beta - \phi \alpha + \phi \beta. \]
Similarly, we have
\[
1 - \beta - \phi \alpha + \phi \beta = \sum_{i=1}^{n} y_{k+1,i} - (\beta + \phi \alpha - \phi \beta) \sum_{i=1}^{n} \sum_{i,j_1,\ldots,j_m \in \mathcal{E}(n)} p_{i,j_1,\ldots,j_m} y_{k+1,i} x_{k,j_1} \cdots x_{k,j_m}
\]
\[
= \sum_{i=1}^{n} y_{k+1,i} - (\beta + \phi \alpha - \phi \beta) \sum_{i=1}^{n} y_{k+1,i}.
\]
Hence
\[
\sum_{i=1}^{n} y_{k,i} = 1, \sum_{i=1}^{n} \hat{e}_{k,i} = 0. \tag{4.12}
\]
By combining (4.11) with Lemmas 3 and 4, we get
\[
||\hat{e}_{k+1}|| \leq (\beta + \phi \alpha - \phi \beta) ||\hat{e}_{k+1}|| + (\beta + \phi \alpha - \phi \beta) \bar{\mu} (J^{(3)}, \ldots, J^{(m)}) ||e_k|| + (1 - \phi)(\alpha - \beta) \bar{\gamma} ||e_k||,
\]
which implies that
\[
||\hat{e}_{k+1}|| \leq \frac{(\beta + \phi \alpha - \phi \beta) \bar{\mu} (J^{(3)}, \ldots, J^{(m)}) + (1 - \phi)(\alpha - \beta) \bar{\gamma}}{1 - (\beta + \phi \alpha - \phi \beta)} ||e_k|| = \zeta ||e_k||.
\]
By (4.12) and Lemma 5, we have
\[
||e_{k+1}|| \leq ||\hat{e}_{k+1}|| \leq \zeta ||e_k|| \leq \zeta^{k+1} ||e_0||.
\]
It is easy to check that \(\zeta < 1\) and the proof is completed. \(\square\)

**Theorem 4.** Let \(\mathcal{P}\) be an \(m\textsuperscript{th}\)-order \(n\)-dimensional stochastic tensor, \(x\) be a solution of (1.1) and \(\alpha < \min\{\frac{1}{2(1-\phi) + 3\bar{\mu} (J^{(3)}, \ldots, J^{(m)})}, \frac{1}{\min(\mu, \nu)}\}\); then, for an arbitrary initial stochastic guess \(x_0\), the iterative sequence \(\{x_{k}\}_{k=0}^{\infty}\) generated by (3.6) is convergent and
\[
||x_{k} - x|| \leq \xi^k ||x_0 - x||,
\]
where \(\xi = 2^{(1-\phi)\gamma + \varphi \bar{\mu}(J^{(3)}, \ldots, J^{(m)})} \frac{1}{1 - (1-\phi)\gamma + \varphi \bar{\mu}(J^{(3)}, \ldots, J^{(m)})} \min(\mu, \nu)\).

**Proof.** By Lemma 2 and the condition that \(\alpha < \min\{\frac{1}{2(1-\phi) + 3\bar{\mu} (J^{(3)}, \ldots, J^{(m)})}, \frac{1}{\min(\mu, \nu)}\}\), we know that \(x\) is unique.

Let \(\hat{e}_k = y_k - x\) and \(e_k = x_k - x\). By (3.6), we obtain
\[
y_{k+1} = \alpha \bar{D}_{\alpha} y_{k+1} + \phi \alpha (\bar{L}_{\alpha} + \bar{U}_{\alpha}) y_{k+1} + \alpha (1 - \phi)(\bar{L}_{\alpha} + \bar{U}_{\alpha}) x_k + b;
\]
then we obtain
\[
\hat{e}_{k+1} = (\alpha \bar{D}_{\alpha} + \phi \alpha (\bar{L}_{\alpha} + \bar{U}_{\alpha})) \hat{e}_{k+1} + \alpha (1 - \phi)(\bar{L}_{\alpha} + \bar{U}_{\alpha}) e_k + \alpha \mathcal{P}(x_{k-2} - x_{k-2})\]
\[
\leq \alpha \mathcal{P}(x_{k-2} - x_{k-2}) \hat{e}_{k+1} + \alpha (1 - \phi) \mathcal{P}(x_{k-2} - x_{k-2}) e_k + \alpha \mathcal{P}(x_{k-2} - x_{k-2}) x.
\]
Combining this with Lemmas 3 and 4 leads to
\[
||\hat{e}_{k+1}|| \leq \alpha \gamma ||\hat{e}_{k+1}|| + (1 - \phi) \alpha \gamma ||e_k|| + \alpha \bar{\mu} (J^{(3)}, \ldots, J^{(m)}) ||e_k||.
\]
Thus\[
\|\hat{e}_{k+1}\|_1 \leq \frac{(1 - \phi)\alpha y + \alpha \mu \left(\mathcal{J}^{(3)}, \cdots, \mathcal{J}^{(m)}\right)}{1 - \alpha \mu \left(\mathcal{J}^{(3)}, \cdots, \mathcal{J}^{(m)}\right)} \|e_k\|_1 = \frac{1}{2} \xi \|e_k\|_1 .
\] (4.13)

By (4.13) and Lemma 6, we have\[
\|e_{k+1}\|_1 \leq 2 \|\hat{e}_{k+1}\|_1 \leq \xi \|e_k\|_1 \leq \xi^{k+1} \|e_0\|_1 .
\]

Notice that $\xi < 1$. This completes the proof of the theorem. \[\]

**Theorem 5.** Let $\mathcal{P}$ be an $m^{th}$-order $n$-dimensional stochastic tensor, $x$ be a solution of (1.1) and $\alpha < \min\left\{\frac{2\omega + (1 - \omega)\phi}{\min\{1, \nu\}}\right\}$; then, for an arbitrary initial stochastic guess $x_0$, the iterative sequence $\{x_k\}_{k=0}^{\infty}$ generated by (3.7) is convergent and

\[
\|x_k - x\|_1 \leq \xi^k \|x_0 - x\|_1 .
\]

**Proof.** It is similar to the proof of Theorem 4. \[\]

**Theorem 6.** Let $\mathcal{P}$ be an $m^{th}$-order $n$-dimensional stochastic tensor, $x$ be a solution of (1.1), $\alpha < \min\left\{\frac{2\omega + (1 - \omega)\phi}{\min\{1, \nu\}}\right\}$ and $\omega \in \left(\frac{1 - \phi}{2 - \phi}, 1\right)$; then, for an arbitrary initial stochastic guess $x_0$, the iterative sequence $\{x_k\}_{k=0}^{\infty}$ generated by (3.8) is convergent and

\[
\|x_k - x\|_1 \leq \vartheta^k \|x_0 - x\|_1 ,
\]

where $\vartheta = \frac{2\omega \alpha \mu \left(\mathcal{J}^{(3)}, \cdots, \mathcal{J}^{(m)}\right) + (1 - \phi)\alpha (1 - \omega)\nu + \omega \alpha \nu}{1 - \phi (1 - \omega) - \omega \alpha \nu}$. \[\]

**Proof.** According to Lemma 2, it is obvious that the solution $x$ is unique when $\alpha < \min\left\{\frac{2\omega + (1 - \omega)\phi}{\min\{1, \nu\}}\right\}$ and $\omega \in \left(\frac{1 - \phi}{2 - \phi}, 1\right)$.

By (3.8), we have

\[
y_{k+1} = (\alpha D_{x_k} + \omega a L_{x_k} + \phi (1 - \omega)(I - \alpha D_{x_k}) + \omega a U_{x_k})y_{k+1} + (1 - \phi)\left[(1 - \omega)(I - \alpha D_{x_k}) + \omega a U_{x_k}\right]x_k + \omega b .
\]

Taking $\hat{e}_k = y_k - x$ and $e_k = x_k - x$, by a proof that is analogous to that for the above theorems, we get

\[
\hat{e}_{k+1} = (\alpha D_{x_k} + \omega a L_{x_k} + \phi (1 - \omega)(I - \alpha D_{x_k}) + \omega a U_{x_k})\hat{e}_{k+1} + (1 - \phi)\left[(1 - \omega)(I - \alpha D_{x_k}) + \omega a U_{x_k}\right]e_k + \omega a P(x_k^{m-2} - x^{m-2})x
\]

\[
= [\phi (1 - \omega)I + \alpha (1 - \phi)D_{x_k} + \omega a L_{x_k} + \omega a U_{x_k}]\hat{e}_{k+1} + (1 - \phi)\left[(1 - \omega)(I - \alpha D_{x_k}) + \omega a U_{x_k}\right]e_k + \omega a P(x_k^{m-2} - x^{m-2})x
\]

\[
\leq [\phi (1 - \omega)I + \alpha (1 - \phi)D_{x_k} + \omega a P(x_k^{m-2} - x^{m-2})]e_{k+1} + \omega a P(x_k^{m-2} - x^{m-2})e_k
\]

\[
+ (1 - \phi)\left[(1 - \omega)(I - \alpha D_{x_k}) + \omega a U_{x_k}\right]e_k .
\]

Due to Lemma 4, the above inequality has the following estimation

\[
\|\hat{e}_{k+1}\|_1 \leq \|\phi (1 - \omega) + \alpha (1 - \phi) + \omega a \nu \left(\mathcal{J}^{(3)}, \cdots, \mathcal{J}^{(m)}\right)\|e_{k+1}\|_1 + \omega a \mu \left(\mathcal{J}^{(3)}, \cdots, \mathcal{J}^{(m)}\right)\|e_{k+1}\|_1
\]

\[
+ \left[(1 - \phi)(1 - \omega)(1 - \alpha) + \omega a (1 - \phi)\right]\|e_{k+1}\|_1 .
\]
Hence
\[ \|e_{k+1}\|_1 \leq \frac{\omega \alpha \bar{\mu} (J^{(3)} \ldots J^{(m)}) + (1 - \phi)(1 - \omega)(1 + \alpha) + \omega \alpha (1 - \phi)}{1 - \phi(1 - \omega) - \alpha(1 - \phi) - \omega \alpha \gamma} \|\hat{e}_{k+1}\|_1 = \frac{1}{2} \vartheta \|e_k\|_1. \] (4.15)

According to (4.15) and Lemma 6, we have
\[ \|e_{k+1}\|_1 \leq 2\|\hat{e}_{k+1}\|_1 \leq \vartheta \|e_k\|_1 \leq \vartheta^{k+1} \|e_0\|_1. \]

and \( \vartheta < 1 \). This completes the proof. \( \square \)

**Remark 2.** For the uniqueness conditions stated in other papers, such as those presented in [12, 13], we can also offer corresponding convergence analysis for our proposed algorithm. Here, we omit the detailed description.

5. Numerical experiments

In this section, we present numerical experiments to exhibit the asset of the algorithm we are proposing.

All experiments were performed by using MATLAB R2016a on a Windows 10 64 bit computer equipped with a 1.00 GHz Intel® Core™ i4-29210M CPU processor and 8.00 GB of RAM. We chose to use three parameters to evaluate the effectiveness of the proposed method, which are the iteration number (denoted by IT), the computing time in seconds (denoted by CPU) and the error rate (denoted by \( \text{err} \)), respectively. The \( \text{err} \) parameter is defined by
\[ \text{err} = \|x_k - \alpha P x_k^{m-1} - b\|_1. \]

We have opted to employ large damping factors during the computation process, and in this case, iterative methods typically face major convergence problems when solving multilinear PageRank problems. Therefore, we tested the following values for the damping parameter \( \alpha = 0.95, 0.99, 0.995, 0.999 \), respectively. We selected that the maximum iterative number to be 1,000, the initial value \( x_0 = v = \frac{1}{n} e \), where \( e \) is an \( n \)-dimensional vector with all entries being 1, and the termination tolerance as \( \varepsilon = 10^{-10} \). The stopping criterion for all test methods was set as \( \|x_k - \alpha P x_k^{m-1} - b\|_1 < \varepsilon \). We searched the parameter \( \phi \) (or \( \omega \)) from 0.1 to 1 with the step size 0.01 and \( \beta \in (0, \alpha) \) from 0.1 to 0.9 with the interval of 0.01; also, we set
\[ \tilde{D}_x = \frac{1}{2} \text{diag}(P x^{m-2}), \tilde{L}_x = \text{tril}(P x^{m-2}, -1) + \tilde{D}_x, \tilde{U}_x = \text{triu}(P x^{m-2}, 1). \]

Next, we provide the numerical analysis for the GTRS iteration discussed in Section 3, using four examples. We compare our approach with the relaxation methods presented by Liu et al. [19] and the TARS method. The TARS iterative method is described in Algorithm 2. To clarify, we denote Algorithms 1–4 in the work of Liu et al. as Al₁, Al₂, Al₃ and Al₄ respectively.

**Example 1.** In this example, we considered three tensors from the practical problems in [30, 31]. Example (i) involves DNA sequence data, while Example (ii) applies interpersonal relationship data, and Example (iii) involves physicists’ occupational mobility data. The numerical results are listed in Tables 1–3, where the minimum CPU times in each row are indicated in bold font.
Algorithm 2: The TARS iterative method

Require: \( \tilde{M}_x, \tilde{N}_x, \alpha, \beta, \omega, \nu, \) relaxation parameter \( \hat{\gamma}, \) maximum \( k_{\text{max}}, \) termination tolerance \( \varepsilon, \) and an initial stochastic vector \( x_0 \)

Ensure: \( x \)

1: \( k \leftarrow 1 \)
2: while \( x_k - \alpha P x^{m-1}_k - b < \varepsilon \) do
3: \( \text{if } k < k_{\text{max}} \text{ then} \)
4: \( y_{k+1} \leftarrow \tilde{M}_x^{-1} \tilde{N}_x x_k + \tilde{M}_x^{-1} b \)
5: \( \hat{x}_{k+1} \leftarrow \hat{\gamma} y_{k+1} + (1 - \hat{\gamma}) x_k \)
6: \( x_{k+1} \leftarrow \text{proj}(\hat{x}_{k+1}) \)
7: \( k = k + 1 \)
8: \( \text{end if} \)
9: \( \text{end while} \)

(i)
\[
P(:, :, 1) = \begin{pmatrix} 0.6000 & 0.4083 & 0.4935 \\ 0.2000 & 0.2568 & 0.2426 \\ 0.2000 & 0.3349 & 0.2639 \end{pmatrix}, \quad P(:, :, 2) = \begin{pmatrix} 0.5217 & 0.3300 & 0.4152 \\ 0.2232 & 0.2800 & 0.2658 \\ 0.2551 & 0.3900 & 0.3190 \end{pmatrix},
\]
\[
P(:, :, 3) = \begin{pmatrix} 0.5565 & 0.3648 & 0.4500 \\ 0.2174 & 0.2742 & 0.2600 \\ 0.2261 & 0.3610 & 0.2900 \end{pmatrix};
\]

(ii)
\[
P(:, :, 1) = \begin{pmatrix} 0.5810 & 0.2432 & 0.1429 \\ 0 & 0.4109 & 0.0701 \\ 0.4190 & 0.3459 & 0.7870 \end{pmatrix}, \quad P(:, :, 2) = \begin{pmatrix} 0.4708 & 0.1330 & 0.0327 \\ 0.1341 & 0.5450 & 0.2042 \\ 0.3951 & 0.3220 & 0.7631 \end{pmatrix},
\]
\[
P(:, :, 3) = \begin{pmatrix} 0.4381 & 0.1003 & 0 \\ 0.0229 & 0.4338 & 0.0930 \\ 0.5390 & 0.4659 & 0.9070 \end{pmatrix};
\]

(iii)
\[
P(:, :, 1) = \begin{pmatrix} 0.9000 & 0.3340 & 0.3106 \\ 0.0690 & 0.6108 & 0.0754 \\ 0.0310 & 0.0552 & 0.6140 \end{pmatrix}, \quad P(:, :, 2) = \begin{pmatrix} 0.6700 & 0.1040 & 0.0805 \\ 0.2892 & 0.8310 & 0.2956 \\ 0.0408 & 0.0650 & 0.6239 \end{pmatrix},
\]
\[
P(:, :, 3) = \begin{pmatrix} 0.6604 & 0.0945 & 0.0710 \\ 0.0716 & 0.6133 & 0.0780 \\ 0.2680 & 0.2922 & 0.8501 \end{pmatrix}.
\]

From Tables 1–3, we have the following observations and remarks:

(1) Although our algorithms require more iterative steps, the computational times are typically shorter than those of \( Al_1, Al_2, Al_3 \) and \( Al_4 \) and TARS, provided that the parameters are chosen appropriately.
<table>
<thead>
<tr>
<th>GTRS-FP</th>
<th>GTRS-IO</th>
<th>GTRS-JCB</th>
<th>GTRS-GS</th>
<th>GTRS-SOR</th>
<th>TARS-IO</th>
<th>TARS-JCB</th>
<th>TARS-GS</th>
<th>TARS-SOR</th>
<th>AL1</th>
<th>AL2</th>
<th>AL3</th>
<th>AL4</th>
</tr>
</thead>
<tbody>
<tr>
<td>α φ</td>
<td>CPU</td>
<td>IT β φ</td>
<td>CPU</td>
<td>IT φ</td>
<td>CPU</td>
<td>IT φ</td>
<td>CPU</td>
<td>IT α φ</td>
<td>CPU</td>
<td>IT</td>
<td>CPU</td>
<td>IT</td>
</tr>
<tr>
<td>0.9</td>
<td>0.49</td>
<td>0.000067</td>
<td>12 0.3 0.41</td>
<td>0.000059</td>
<td>12 0.94</td>
<td>0.000066</td>
<td>11 0.92</td>
<td>0.000066</td>
<td>11 0.6 1</td>
<td>0.000067</td>
<td>11 0.000073</td>
<td>10 0.000097</td>
</tr>
<tr>
<td>0.95</td>
<td>0.34</td>
<td>0.000067</td>
<td>13 0.7 0.51</td>
<td>0.000059</td>
<td>12 0.29</td>
<td>0.000066</td>
<td>10 0.96</td>
<td>0.000066</td>
<td>12 0.7 0.78</td>
<td>0.000065</td>
<td>15 0.000067</td>
<td>10 0.000068</td>
</tr>
<tr>
<td>0.99</td>
<td>0.39</td>
<td>0.000067</td>
<td>13 0.9 0.19</td>
<td>0.000062</td>
<td>12 0.97</td>
<td>0.000066</td>
<td>12 1</td>
<td>0.000064</td>
<td>29 0.000066</td>
<td>10 0.000073</td>
<td>9 0.000071</td>
<td>11 0.000071</td>
</tr>
<tr>
<td>0.999</td>
<td>0.86</td>
<td>0.000067</td>
<td>12 0.8 0.75</td>
<td>0.000063</td>
<td>12 0.58</td>
<td>0.000065</td>
<td>9 0.76</td>
<td>0.000066</td>
<td>13 0.6 1</td>
<td>0.000067</td>
<td>12 0.000070</td>
<td>9 0.000072</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GTRS-FP</th>
<th>GTRS-IO</th>
<th>GTRS-JCB</th>
<th>GTRS-GS</th>
<th>GTRS-SOR</th>
<th>TARS-IO</th>
<th>TARS-JCB</th>
<th>TARS-GS</th>
<th>TARS-SOR</th>
<th>AL1</th>
<th>AL2</th>
<th>AL3</th>
<th>AL4</th>
</tr>
</thead>
<tbody>
<tr>
<td>α φ</td>
<td>CPU</td>
<td>IT β φ</td>
<td>CPU</td>
<td>IT φ</td>
<td>CPU</td>
<td>IT φ</td>
<td>CPU</td>
<td>IT α φ</td>
<td>CPU</td>
<td>IT</td>
<td>CPU</td>
<td>IT</td>
</tr>
<tr>
<td>0.9</td>
<td>0.92</td>
<td>0.000068</td>
<td>25 0.8 0.13</td>
<td>0.000060</td>
<td>25 0.24</td>
<td>0.000064</td>
<td>24 0.36</td>
<td>0.000068</td>
<td>32 0.000077</td>
<td>17 0.000063</td>
<td>17 0.000062</td>
<td>20 0.000134</td>
</tr>
<tr>
<td>0.95</td>
<td>0.29</td>
<td>0.000069</td>
<td>31 0.9 0.99</td>
<td>0.000057</td>
<td>27 0.35</td>
<td>0.000066</td>
<td>26 0.19</td>
<td>0.000066</td>
<td>29 0.8 0.18</td>
<td>0.000065</td>
<td>37 0.000066</td>
<td>18 0.000061</td>
</tr>
<tr>
<td>0.99</td>
<td>0.93</td>
<td>0.000070</td>
<td>29 0.9 0.29</td>
<td>0.000058</td>
<td>28 0.6 0.000067</td>
<td>28 0.32</td>
<td>0.000068</td>
<td>31 1</td>
<td>0.000068</td>
<td>30 0.000071</td>
<td>19 0.000072</td>
<td>21 0.000064</td>
</tr>
<tr>
<td>0.999</td>
<td>0.72</td>
<td>0.000069</td>
<td>30 0.9 0.87</td>
<td>0.000059</td>
<td>29 0.52</td>
<td>0.000069</td>
<td>28 0.17</td>
<td>0.000068</td>
<td>31 0.3 0.32</td>
<td>0.000065</td>
<td>94 0.000257</td>
<td>20 0.000072</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GTRS-FP</th>
<th>GTRS-IO</th>
<th>GTRS-JCB</th>
<th>GTRS-GS</th>
<th>GTRS-SOR</th>
<th>TARS-IO</th>
<th>TARS-JCB</th>
<th>TARS-GS</th>
<th>TARS-SOR</th>
<th>AL1</th>
<th>AL2</th>
<th>AL3</th>
<th>AL4</th>
</tr>
</thead>
<tbody>
<tr>
<td>α φ</td>
<td>CPU</td>
<td>IT β φ</td>
<td>CPU</td>
<td>IT φ</td>
<td>CPU</td>
<td>IT φ</td>
<td>CPU</td>
<td>IT α φ</td>
<td>CPU</td>
<td>IT</td>
<td>CPU</td>
<td>IT</td>
</tr>
<tr>
<td>0.9</td>
<td>0.98</td>
<td>0.000071</td>
<td>51 0.6 0.12</td>
<td>0.000061</td>
<td>55 0.98</td>
<td>0.000072</td>
<td>50 0.88</td>
<td>0.000071</td>
<td>52 0.9 0.9</td>
<td>0.000066</td>
<td>52 0.000082</td>
<td>39 0.000093</td>
</tr>
<tr>
<td>0.95</td>
<td>0.55</td>
<td>0.000069</td>
<td>71 0.7 1</td>
<td>0.000061</td>
<td>61 0.96</td>
<td>0.000068</td>
<td>61 0.96</td>
<td>0.000067</td>
<td>62 0.7</td>
<td>0.000045</td>
<td>48 0.000068</td>
<td>48 0.000069</td>
</tr>
<tr>
<td>0.99</td>
<td>0.98</td>
<td>0.000067</td>
<td>74 0.6 0.54</td>
<td>0.000079</td>
<td>78 0.33</td>
<td>0.000067</td>
<td>72 0.28</td>
<td>0.000066</td>
<td>88 0.5 0.7</td>
<td>0.000065</td>
<td>88 0.000068</td>
<td>56 0.000101</td>
</tr>
<tr>
<td>0.999</td>
<td>0.45</td>
<td>0.000066</td>
<td>91 0.2 0.25</td>
<td>0.000060</td>
<td>93 0.94</td>
<td>0.000067</td>
<td>76 0.35</td>
<td>0.000066</td>
<td>90 0.3 0.74</td>
<td>0.000065</td>
<td>132 0.000070</td>
<td>58 0.000063</td>
</tr>
</tbody>
</table>
(2) Considering the least CPU times obtained from GTRS-FP, GTRS-IO, GTRS-JCB, GTRS-GS and GTRS-SOR, we observe that they account for approximately 0%, 91.7%, 0%, 0% and 8.3%, respectively. Therefore, in this example, GTRS-IO appears to be the most efficient algorithm among the five tested ones.

(3) Among the five types of splitting in the GTRS iteration, GTRS-JCB may result in higher CPU time usage compared to other methods, but it exhibits fewer iterative steps. This indicates that in cases with fewer iterations, GTRS-JCB can converge faster, while other methods may require more iterations to achieve the same convergence performance.

Next, we demonstrate the relationship between the iteration count and different values of the parameters $\alpha$ and $\phi$ in the GTRS iteration based on the inner-outer splitting approach, using Example 1(ii). By choosing $\beta = \{0.2, 0.4, 0.6, 0.8\}$ and $\phi$ from 0.1 to 1 in the interval of 0.01, we obtain Figure 1. It is shown that the GTRS iteration requires fewer iteration steps for a larger $\phi$, which is more evident for a larger $\alpha$, such as when $\alpha = 0.999$. It is clear that the iteration number changes significantly depending on the chosen $\beta$, especially with $\beta = 0.2$.

![Figure 1. The iterative steps for Example 1(ii) with different values of $\alpha$ and $\phi$.](image-url)
Example 2. [12, 19] Let $\Gamma$ be a directed graph with node set $\mathcal{V} = \langle n \rangle = \mathbb{D} \cup \mathbb{P}$, where $\mathbb{D}$ and $\mathbb{P}$ denote the sets of dangling nodes and pairwise connected ones, respectively. Denote $n_p$ ($n_p \geq 2$) as the number of nodes in $\mathbb{P}$, that is, $\mathbb{P} = \{1, 2, \ldots, n_p\}$. A nonnegative tensor $\mathcal{A}$ is constructed as follows:

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

where $a_{i_1 i_2 \cdots i_l}$ is taken randomly in $(0, 1)$. Normalizing the entries with $p_{i_1 i_2 \cdots i_l} = \frac{a_{i_1 i_2 \cdots i_l}}{\sum_{i_1=1}^n a_{i_1 i_2 \cdots i_l}}$ yields a stochastic tensor $\mathcal{P} = p_{i_1 i_2 \cdots i_l}$.

Based on the data presented in Table 4, there are observations that can be made as follows:

(1) The GTRS iteration demonstrates significantly shorter CPU time than those for $Al_1$–$Al_4$ and TARS. This suggests that the GTRS iterative method demonstrates higher efficiency in terms of computational time when appropriate parameters are chosen.

(2) In the case of all tested algorithms for a larger $n_p$, such as the case of $n_p = 120$, GTRS-SOR exhibits shorter CPU times than other methods.

(3) For $n = 60$, in terms of the CPU times, GTRS-IO achieves the best performance when $\alpha=0.9$ or 0.95, while GTRS-FP, GTRS-JCB,GTRS-GS and GTRS-SOR outperform GTRS-IO when $\alpha=0.99$ or 0.999. This implies that the behavior and reliability of these methods do not exhibit a monotonic relationship with $\alpha$.

Furthermore, we display the test results of the relationship between CPU time and $err$ at each step in Figure 2 with $n = 150$ and $n_p = 120$. The values of $err$ are taken as the logarithm of base-10. It is known from Figure 2 that the GTRS iterative method performs better than $Al_1$–$Al_4$ and TARS in terms of both CPU time and error measures.
Figure 2. The relationship between CPU time and \textit{err} for different values of \(\alpha\) in Example 2.

**Example 3.** \cite{13} We generated third-order stochastic tensors with dimension \(n\) of 100, 150 and 200 using the function \texttt{rand} of MATLAB. The results are reported in Table 5.

From Table 5, we observe the following:

(1) If the parameters are chosen suitably, the GTRS iterative method can achieve shorter CPU times than \(A_{l_1} - A_{l_4}\) and TARS.

(2) For GTRS-JCB, the optimal performance is achieved when \(\alpha = 0.99\) or 0.999, and \(n = 100\) or 150. Similarly, GTRS-GS performs best when \(n = 150\), while GTRS-SOR achieves the best results when \(n = 200\). Notably, GTRS-IO shows the best performance when \(\alpha = 0.95\). GTRS-FP requires the least iterative steps.

(3) In terms of CPU times, GTRS-IO outperforms GTRS-JCB when \(\alpha = 0.9\) or 0.95. However, GTRS-JCB, GTRS-GS and GTRS-SOR outperform GTRS-IO when \(\alpha = 0.99\) or 0.999.
Table 4. Comparison of GTRS iterative method with $A_1 - A_4$ and TARS for Example 2.

<table>
<thead>
<tr>
<th>GTRS-FP</th>
<th>GTRS-IO</th>
<th>GTRS-JCB</th>
<th>GTRS-GS</th>
<th>GTRS-SOR</th>
<th>TARS-IO</th>
<th>TARS-JCB</th>
<th>TARS-GS</th>
<th>TARS-SOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>$\phi$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\omega$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\beta$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\gamma$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>IT</td>
<td>CPU</td>
<td>IT</td>
<td>CPU</td>
<td>IT</td>
<td>CPU</td>
<td>IT</td>
<td>CPU</td>
</tr>
<tr>
<td>60</td>
<td>0.9</td>
<td>0.026</td>
<td>0.015</td>
<td>0.014</td>
<td>0.023</td>
<td>0.012</td>
<td>0.012</td>
<td>0.011</td>
</tr>
<tr>
<td>200</td>
<td>0.9</td>
<td>0.026</td>
<td>0.015</td>
<td>0.014</td>
<td>0.023</td>
<td>0.012</td>
<td>0.012</td>
<td>0.011</td>
</tr>
</tbody>
</table>

Table 5. Comparison of GTRS iterative method with $A_1 - A_4$ and TARS for Example 3.

<table>
<thead>
<tr>
<th>GTRS-FP</th>
<th>GTRS-IO</th>
<th>GTRS-JCB</th>
<th>GTRS-GS</th>
<th>GTRS-SOR</th>
<th>TARS-IO</th>
<th>TARS-JCB</th>
<th>TARS-GS</th>
<th>TARS-SOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>$\phi$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\omega$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\beta$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\gamma$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>IT</td>
<td>CPU</td>
<td>IT</td>
<td>CPU</td>
<td>IT</td>
<td>CPU</td>
<td>IT</td>
<td>CPU</td>
</tr>
<tr>
<td>100</td>
<td>0.9</td>
<td>0.026</td>
<td>0.015</td>
<td>0.014</td>
<td>0.023</td>
<td>0.012</td>
<td>0.012</td>
<td>0.011</td>
</tr>
<tr>
<td>200</td>
<td>0.9</td>
<td>0.026</td>
<td>0.015</td>
<td>0.014</td>
<td>0.023</td>
<td>0.012</td>
<td>0.012</td>
<td>0.011</td>
</tr>
</tbody>
</table>
Similarly, we present the test results for the correlation between CPU times and $err$ at each iterative step for $n = 200$ in Figure 3.

As depicted in Figure 3, it is evident that the GTRS iterative method consistently outperforms $Al_1$–$Al_4$ and TARS in terms of both CPU time and error measurement.

**Example 4.** [13] Let $t$ denote the density of zero entries in a tensor. $P$ is a positive tensor or a zero tensor when $t = 0$ or $t = 1$ respectively. Test sparse tensors with different $t$ are generated by the function `randsample` of MATLAB.

In Example 4, we generated third-order tensors of dimension 200 with densities ranging from 0.1 to 0.9 in intervals of 0.1. The numerical results are presented in Table 6. The conclusions drawn from Table 6 are as follows:

1. When suitable parameters are found, our proposed algorithm consistently outperforms the methods presented in [19] and the TARS method.
### Table 6. Comparison of GTRS iterative method with $A_l$ to $A_4$ and TARS for Example 4.

<table>
<thead>
<tr>
<th>n</th>
<th>$\phi$</th>
<th>CPU</th>
<th>IT $\beta$</th>
<th>GTRS-FP</th>
<th>GTRS-IO</th>
<th>GTRS-JCB</th>
<th>GTRS-GS</th>
<th>GTRS-SOR</th>
<th>TARS-FP</th>
<th>TARS-IO</th>
<th>TARS-JCB</th>
<th>TARS-GS</th>
<th>TARS-SOR</th>
<th>$A_l$</th>
<th>$A_l$</th>
<th>$A_l$</th>
<th>$A_l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.9</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>0.9</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>0.9</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>0.9</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>0.9</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>0.9</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>0.9</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>0.9</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>0.9</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td>0.14462</td>
<td>0.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** The table presents the comparison of GTRS iterative methods with $A_l$ to $A_4$ and TARS for Example 4. The values represent the performance metrics, such as CPU time and iteration counts, for each method.
(2) Out of the 36 test cases, in terms of the minimum CPU times, GTRS-SOR, GTRS-GS, GTRS-IO, GTRS-JCB and GTRS-FP occupy about 36.1%, 0%, 44.5%, 19.4% and 0%, respectively. Based on these results, GTRS-SOR appears to be the most competitive algorithm, particularly when $t=0.4$ or 0.6.

(3) When $t$ is less than 0.5, GTRS-JCB has better convergence performance than GTRS-GS and GTRS-FP. In most cases, GTRS-IO demonstrates a shorter CPU time than the other splitting methods of the GTRS iterative type, especially when $t=0.8$.

Next, we discuss the relationship between the iterative steps and $\phi$ with different values of $\alpha$. Given the superior convergence properties of the GTRS iterative method with successive overrelaxation splitting, and the significant variation in the number of iterations in Example 4, we selected this splitting for the experiment. Specifically, we conducted the experiment by using $\omega = \{0.3, 0.5, 0.7, 0.9\}$, $t = \{0.4, 0.6\}$ and varying $\phi$ from 0.1 to 1 in increments of 0.01. We have plotted the associated comparison results in Figures 4 and 5. The results indicate that increasing $\phi$ leads to better performance of the GTRS iterative method, as evidenced by the reduction in the number of iterations required.

![Figure 4](image-url)

**Figure 4.** Taking $t=0.4$, the iterative steps with different values of $\alpha$ and $\phi$ for Example 4.
Figures 6–9 show the convergence history of the GTRS iteration under different densities of tensors and values of $\alpha$. From Figures 6–9, we find that the proposed algorithms converge faster than the existing methods in the most cases, while GTRS-IO performs the best. It is noticed that GTRS-FP, GTRS-IO and GTRS-JCB always require less computing time than the existing methods. This is due to the fact that they compute smaller errors, which reduce the number of iterations required for convergence.

Different parameter settings can lead to optimal performance for each GTRS iteration. By selecting the appropriate values for $\alpha$ and $n$, the computational efficiency and accuracy of the GTRS iteration can be improved. It is important to note that these optimal parameter settings may vary depending on the specific problem and dataset being analyzed. In general, the GTRS iteration can achieve better convergence performance than $A_{l_1} - A_{l_4}$ and TARS in terms of solving the multilinear PageRank problem.
Figure 6. The relationship between CPU time and err for different values of $t$ with $\alpha = 0.90$ for Example 4.
Figure 7. The relationship between CPU time and err for different values of \( t \) with \( \alpha = 0.95 \) for Example 4.
Figure 8. The relationship between CPU time and $err$ for different values of $t$ with $\alpha = 0.99$ for Example 4.
Figure 9. The relationship between CPU time and err for different values of $t$ with $\alpha = 0.999$ for Example 4.

6. Conclusions

In this paper, based on the (weak) regular splitting of the matrix $I - \alpha \mathcal{P} x^{m-2}$, we presented the GTRS iterative method for solving (1.1) from the perspective of the multilinear PageRank problem, and we have proved its overall convergence. In addition, we provided several splits of the GTRS iteration and constructed the corresponding convergence theorems. Numerical experiments indicated that the GTRS iterative method is superior to the existing ones in [19] and the TARS method as a result of choosing appropriate parameters. Overall, this research contributes to the advancement of non-gradient algorithms and their practical implications in tensor computations. In future work, we plan to investigate the latest uniqueness conditions to establish better convergence properties.
Use of AI tools declaration

The authors declare that they have not used artificial intelligence tools in the creation of this paper.

Acknowledgments

This work was supported in part by the Postgraduate Educational Innovation Program of Guangdong (No. 2021SFKC030) and Guangzhou Basic and Applied Basic Research Project (No. 2023A04J1322). The authors gratefully acknowledge the anonymous reviewers for their constructive and valuable suggestions, which greatly improved this paper.

Conflict of interest

The authors declare no conflict of interest.

References


