Journal of Computational Mathematics
Vol.36, No.3, 2018, 351–373.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1605-m2016-0828

A TRUST-REGION-BASED ALTERNATING LEAST-SQUARES ALGORITHM FOR TENSOR DECOMPOSITIONS*

Fan Jiang and Deren Han
Jiangsu Key Laboratory of NSLSCS, School of Mathematical Sciences, Nanjing Normal University, Nanjing 210023, China
Email: 15905154902@163.com, handeren@njnu.edu.cn
Xiaofei Zhang
Information Technology Department, Chinascope, Nanjing 210023, China
Email: 879734743@qq.com

Abstract

Tensor canonical decomposition (shorted as CANDECOMP/PARAFAC or CP) decomposes a tensor as a sum of rank-one tensors, which finds numerous applications in signal processing, hypergraph analysis, data analysis, etc. Alternating least-squares (ALS) is one of the most popular numerical algorithms for solving it. While there have been lots of efforts for enhancing its efficiency, in general its convergence can not been guaranteed.

In this paper, we cooperate the ALS and the trust-region technique from optimization field to generate a trust-region-based alternating least-squares (TRALS) method for CP. Under mild assumptions, we prove that the whole iterative sequence generated by TRALS converges to a stationary point of CP. This thus provides a reasonable way to alleviate the swamps, the notorious phenomena of ALS that slow down the speed of the algorithm. Moreover, the trust region itself, in contrast to the regularization alternating least-squares (RALS) method, provides a self-adaptive way in choosing the parameter, which is essential for the efficiency of the algorithm. Our theoretical result is thus stronger than that of RALS in [26], which only proved the cluster point of the iterative sequence generated by RALS is a stationary point. In order to accelerate the new algorithm, we adopt an extrapolation scheme. We apply our algorithm to the amino acid fluorescence data decomposition from chemometrics, BCM decomposition and rank-$(L_r, L_r, 1)$ decomposition arising from signal processing, and compare it with ALS and RALS. The numerical results show that TRALS is superior to ALS and RALS, both from the number of iterations and CPU time perspectives.

Mathematics subject classification: 90C06, 90C53, 65K05
Key words: tensor decompositions, trust region method, alternating least-squares, extrapolation scheme, global convergence, regularization.

1. Introduction

The problem of decomposing a higher-order tensor into a sum of products of lower-order tensors finds more and more important applications in signal processing [3, 4], data analysis [2,32], scientific computing [5,22,24,25], biomedical engineering [1,39], machine learning [35], chemometrics [36], etc. One of the famous decompositions of higher-order tensors is CANDECOMP/PARAFAC analysis (CP, canonical polyadic decomposition), which can be dated back to the work of Hitchcock in 1927 [20,21]. However, it is out of the interests of researchers until

* Received January 5, 2017 / Revised version received March 20, 2017 / Accepted May 22, 2017 / Published online March 28, 2018 /

the study of Tucker [38], Carroll and Chang [7] and Harshman [19] in the fields of psychometries and phonetics in 1970, respectively. In signal processing, there is a generalized PARAFAC, named Block-PARAFAC [3]. The model is applied in direct-sequence code division multiple access (DS-CDMA) system, and orthogonal frequency division multiplexing (OFDM) system.

Recently, some numerical optimization algorithms were tailored for solving tensor decomposition problems [37], among which the Levenberg-Marquardt algorithm for non-linear least squares problems [15–17] attracts much attention [23]. Nevertheless, the alternating least squares (ALS) method is still the most popular one, due to its simplicity in implementation [2, 7, 19, 24, 36]. However, on one hand, ALS usually needs a large number of iterations to converge because the convergence rate of many iterations is almost null; and, on the other hand, we can not prove that the limit points of the subsequences generated by the ALS are the critical points of the least squares cost function. Many researchers thus pay a lot of attentions to enhancing the efficiency of ALS numerically by introducing skills from numerical algebra and numerical optimization. For example, the line search along the incremental direction of an old iterate to a new one is proposed by Harshman [19], and as a consequence, it needs efficient schemes in finding an 'optimal' step size. For real-valued tensors, the optimal step size can be directly calculated because it is a solution of a polynomial equation with a single argument, and the resulting algorithm is called "enhanced line search" (ELS) [10, 33]. For complex-valued tensors CP, Nion and De Lathauwer propose "enhanced line search with complex step" (ELSCS) [29,30], where a complex-valued step size factor that contains the modulus and the phase is introduced, and the optimal step size is approximated by an alternating minimization manner, i.e., find the optimal modulus for a fixed phase and then find the optimal phase for a fixed modulus. Recently, by using the classic resultant results from algebraic geometry [12], Chen, et al. [8] find that the complex-valued optimal step size can also be found by solving two single variable polynomial equations successively. Since the polynomials are with high order, they propose to solve the first polynomial equation by solving an eigenvalue problem and the algorithm is much stable. Most recently, Domanov and De Lathauwer [14] propose to reduce the CP decomposition of third-order tensors to generalized eigenvalue decomposition, which enables to use the high performance techniques from numerical algebra.

While there have been several techniques in enhancing the efficiency of ALS numerically, the results on its global convergence is in its infancy. To ensure the global convergence, Li, Kindermann, and Navasca [26] propose to use the regularization technique and name the resulting algorithm as regularized alternating least squares method, shorted as RALS method. That is, the subproblems in ALS are replaced by the new objectives which are the sum of the original least squares and the regularization terms (a quadratic term). This quadratic term, which transforms the objective function from a convex function to a strongly convex one, plays an important role. Theoretically, Li, et al. [26] prove that the limit points of the converging subsequences of the RALS are the critical points of the least squares cost function; and numerically, it speeds up ALS by successfully avoiding the swamp phenomenon, i.e., a large number of iterations with no improvements.

The RALS, though having the advantages over ALS, both from theoretical and numerical viewpoints, has some issues to be handled. Theoretically, the assertion that the limit points of the converging subsequences of the RALS are the critical points of the least squares cost function is not a satisfying result, and one wonders if the whole generated sequence converges to a critical point. Numerically or practically, the regularization parameter plays a very important role in the efficiency of the algorithm, while it is difficult to choose a proper value. To tackle

these issues, in this paper, we propose a trust-region-based alternating least-squares algorithm (TRALS), which combines the trust-region technique from numerical optimization and the ALS. Our contributions can be summarized as follows:

- Under very mild assumptions, we prove that the least-squares objective function decreases in a monotonic manner and the difference between two successive iterates tends to zero for each block. Based on this result, we successfully prove that the whole generated sequence converges to a critical point.
- The implementation of TRALS is simple. Compared to ALS, TRALS requires the additional work of determining to increase or decrease the trust region parameter; however, the rule of choosing trust region parameter in TRALS is very efficient and adds little computational burden.
- In order to accelerate TRALS, we propose an extrapolation method, i.e., after solving the trust-region least-squares problems, we use a simple linear combination of the solution of the trust-region least-squares problems and the last iterate to generate the next iterate.
- We report some numerical results and compare TRALS with ALS and RALS. The results indicate that TRALS possesses a faster rate of convergence.

The rest of this paper is organized as follows: In Section 2, we summarize some notations and notions that will be used throughout the paper. In Section 3, we introduce the CP decomposition model and describe our new method formally. In Section 4, we prove the convergence of the new method. In Section 5, we perform some numerical experiments and compare our new methods with ALS and RALS. Finally, we give some conclusions in Section 6.

2. Preliminary

Our notations are quite standard, and the readers can further refer to the survey paper [24]. We denote a scalar in $\mathbb{R}$ by a lower-case letter a, a vector by a bold lower-case letter $\mathbf{a}$. The bold upper-case letter $\mathbf{A}$ represents a matrix and the symbol of tensor is a calligraphic letter $\mathcal{A}$. The i-th element of a vector $\mathbf{a}$ is a_i, element (i,j) of a matrix $\mathbf{A}$ is denoted by a_{ij}, element (i,j,k) of a third-order tensor $\mathcal{A}$ is a_{ijk}. Here we can see that a vector is a first-order tensor and a matrix is a second-order tensor. A tensor whose order is larger than or equal to three is called a high-order tensor. Before discussing tensor decompositions, we give some basic definitions of tensors.

Definition 2.1. *Mode-n fiber: A mode-n fiber of an N-order tensor is a vector defined by fixing all indices but the n-th one.*

Definition 2.2. *Slice: A slice of an N-order tensor is a matrix defined by fixing every index but two.*

By the above definitions, a mode-1 fiber of a matrix is its column, a mode-2 fiber of a matrix is its row. A three-order tensor has three kinds of slices, which named frontal, horizontal and lateral slices. Figure 2.1 shows mode-1 (column), mode-2 (row) and mode-3 (tube) fibers of a third order tensor $\mathcal{Y} \in \mathbb{R}^{I\times J\times K}$, denoted by $y_{:jk}$, $y_{i:k}$, $y_{ij:}$, and three kinds of slice of a tensor $\mathcal{Y} \in \mathbb{R}^{I\times J\times K}$, denoted by $\mathbf{Y}_{::k}$, $\mathbf{Y}_{i::}$, $\mathbf{Y}_{:j:}$, respectively.

By the definition of slices, we can define a mode-n matricization of a tensor.

Definition 2.3. *Mode-n Matricization: Matricization is the process of reordering the elements of an N-th order tensor into a matrix. The mode-n matricization of a tensor* $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ *is denoted by* $\mathbf{Y}_{(n)}$ *and concatenates the mode-n fibers to be the columns of the resulting matrix.*

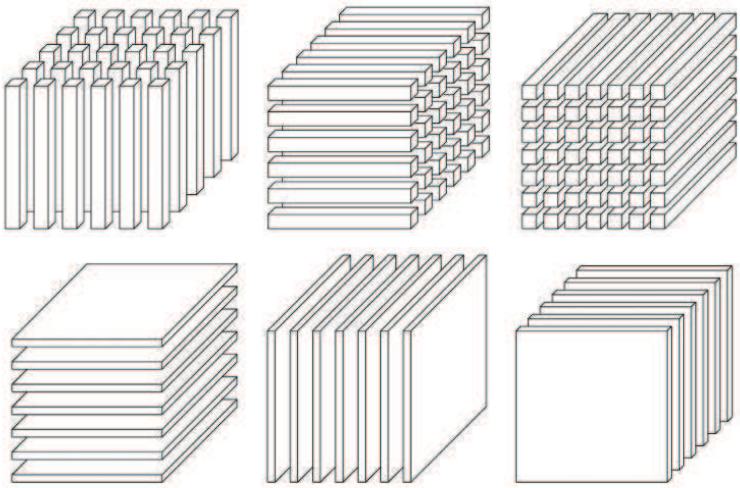

Fig. 2.1. The left one of the first row is mode-1 fibers: $y_{:jk}$, the middle one is mode-2 fibers: $y_{i:k}$, and the right one is mode-3 fibers: $y_{ij:}$. The left one of the second row is horizontal slices: $\mathbf{Y}_{i::}$, the middle one is lateral slices: $\mathbf{Y}_{:j:}$, and the right one is frontal slices: $\mathbf{Y}_{::k}$.

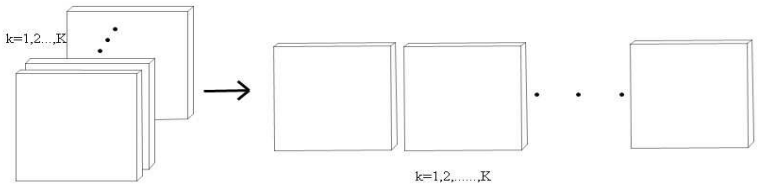

Fig. 2.2. Illustration of mode-1 matricization–the mode-1 fibers are aligned to form a matrix.

For a tensor $\mathcal{Y} \in \mathbb{R}^{I \times J \times K}$, the mode-1 matricizations of $\mathcal{Y}$ is:

$$\mathbf{Y}_{(1)} = \begin{pmatrix} y_{111} & \cdots & y_{1J1} & y_{112} & \cdots & y_{1J2} & \cdots & y_{11K} & \cdots & y_{1JK} \\ y_{211} & \cdots & y_{2J1} & y_{212} & \cdots & y_{2J2} & \cdots & y_{21K} & \cdots & y_{2JK} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ y_{I11} & \cdots & y_{IJ1} & y_{I12} & \cdots & y_{IJ2} & \cdots & y_{I1K} & \cdots & y_{IJK} \end{pmatrix}.$$

The CANDECOMP/PARAFAC (CP) decomposition of a tensor is to decompose it as the sum of some rank one tensors. An N-order tensor is *rank one*, if it can be written as the outer product of N vectors, i.e., $\mathcal{Y} = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \cdots \circ \mathbf{a}^{(N)}$, or in other words, its element is the product of N scalars, $y_{i_1,i_2\ldots i_N} = a^{(1)}_{i_1} a^{(2)}_{i_2} \cdots a^{(N)}_{i_N}$.

Definition 2.4. *CP decomposition: A CP decomposition of a third-order tensor is a decomposition of $\mathcal{Y}$ as a linear combination of a minimal number of rank-1 tensor,*

$$\mathcal{Y} = \sum_{r=1}^{R} \mathbf{a}^{(r)} \circ \mathbf{b}^{(r)} \circ \mathbf{c}^{(r)}.$$

$\mathbf{a}^{(r)}$, $\mathbf{b}^{(r)}$, $\mathbf{c}^{(r)}$ *are the r-th row of* $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, *which are factors of the tensor* $\mathcal{Y}$.

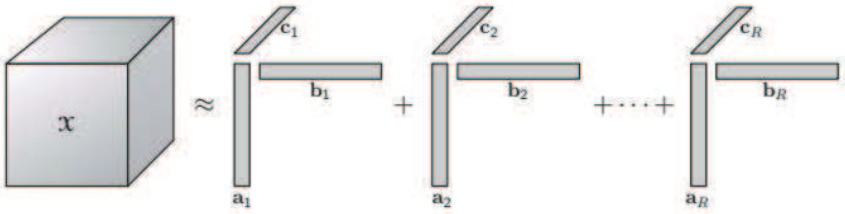

Fig. 2.3. A CP of a third-order tensor.

In designing numerical methods for dealing with CP decomposition, we may use several tensor operations. Here we list some of them.

Definition 2.5. *Kronecker Product: The Kronecker product of matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *is defined as*

$$\mathbf{A} \bigotimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}.$$

Definition 2.6. *Khatri-Rao Product: The Khatri-Rao product of matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *is defined as*

$$\mathbf{A} \bigodot_R \mathbf{B} = (\mathbf{A}_1 \bigotimes \mathbf{B}_1 \quad \cdots \quad \mathbf{A}_R \bigotimes \mathbf{B}_R),$$

where

$$\mathbf{A} = (\mathbf{A}_1 \ \cdots \ \mathbf{A}_R), \ \mathbf{B} = (\mathbf{B}_1 \ \cdots \ \mathbf{B}_R).$$

If matrices $\mathbf{A} = (\mathbf{a}_1 \quad \cdots \quad \mathbf{a}_R)$, $\mathbf{B} = (\mathbf{b}_1 \quad \cdots \quad \mathbf{b}_R)$, Khatri-Rao product of them is

$$\mathbf{A} \bigodot \mathbf{B} = (\mathbf{a}_1 \bigotimes \mathbf{b}_1 \ \ \mathbf{a}_2 \bigotimes \mathbf{b}_2 \quad \cdots \quad \mathbf{a}_R \bigotimes \mathbf{b}_R).$$

It is obvious that if $\mathbf{a}$ and $\mathbf{b}$ are vectors, then the Khatri-Rao and Kronecker Product are identical, i.e., $\mathbf{a} \bigotimes \mathbf{b} = \mathbf{a} \bigodot \mathbf{b}$.

Definition 2.7. *Frobenius norm: The Frobenius norm of a tensor* $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ *is the square root of the sum of the squares of all its elements:*

$$\|\mathcal{Y}\|_F = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} y_{i_1,i_2 \ldots i_N}^2}.$$

Besides CP decomposition, there are more general decompositions such as block component model (BCM) [3,8,28] and rank-$(L_r, L_r, 1)$ block term decomposition (BTD) [37].

Definition 2.8. *Mode-n Product: The mode-2 and mode-3 products of a third-order tensor* $\mathcal{H} \in \mathbb{R}^{I\times L\times P}$ *by the matrices* $\mathbf{S} \in \mathbb{R}^{J\times L}$ *and* $\mathbf{A} \in \mathbb{R}^{K\times P}$*, denoted by* $\mathcal{H} \bullet_2 \mathbf{S}$ *and* $\mathcal{H} \bullet_3 \mathbf{A}$*, result in a* $\mathbb{R}^{I\times J\times P}$ *tensor and a* $\mathbb{R}^{I\times L\times K}$ *tensor, respectively, with elements defined by*

$$(\mathcal{H} \bullet_2 \mathbf{S})_{ijp} := \sum_{l=1}^{L} h_{ilp}s_{jl}, \qquad (\mathcal{H} \bullet_3 \mathbf{A})_{ilk} := \sum_{p=1}^{P} h_{ilp}a_{kp}.$$

Definition 2.9. *BCM decomposition: A third-order tensor* $\mathcal{Y} \in \mathbb{R}^{I\times J\times K}$ *follows a BCM decomposition if it can be written as*

$$\mathcal{Y}=\sum_{r=1}^{R} \mathcal{H}_r \bullet_2 \mathbf{S}_r \bullet_3 \mathbf{A}_r.$$

The vectors $\mathbf{h}_r \in \mathbb{R}^{I\times 1}$, $\mathbf{s}_r \in \mathbb{R}^{J\times 1}$ and $\mathbf{a}_r \in \mathbb{R}^{K\times 1}$ of CP decomposition are now replaced by a tensor $\mathcal{H}_r \in \mathbb{R}^{I\times L\times P}$ and two matrices $\mathbf{S}_r \in \mathbb{R}^{J\times L}$ and $\mathbf{A}_r \in \mathbb{R}^{K\times P}$, respectively.

Definition 2.10. *rank*$-(L_r, L_r, 1)$ *BTD: A decomposition of a tensor* $\mathcal{Y} \in \mathbb{R}^{I\times J\times K}$ *in a sum of rank*$-(L_r, L_r, 1)$ *terms,* $1 \le r \le R$*, is a decomposition of* $\mathcal{Y}$ *of the form*

$$\mathcal{Y}=\sum_{r=1}^{R} (\mathbf{A}_r\mathbf{B}_r^\top) \circ \mathbf{c}_r.$$

The matrixs $\mathbf{A}_r \in \mathbb{R}^{I\times L_r}$ *and* $\mathbf{B}_r \in \mathbb{R}^{J\times L_r}$ *are rank-*L_r*, and the vector* $\mathbf{c}_r \in \mathbb{R}^{K\times 1}$*.*

3. ALS, RALS, and TRALS

From the numerical optimization point of view, all tensor decompositions such as CP, BCM and BCD, can be formulated as a minimization problem whose objective function is the least-squares of the equations in their definitions. We here thus focus on the CP decomposition of a third-order tensor $\mathcal{Y} \in \mathbb{R}^{I\times J\times K}$, and the corresponding optimization problem is the following least-squares minimization problem:

$$f(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \min \frac{1}{2} \left\| \mathcal{Y} - \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \right\|_F^2, \tag{3.1}$$

where $\mathbf{A} \in \mathbb{R}^{I\times R}$, $\mathbf{B} \in \mathbb{R}^{J\times R}$, $\mathbf{C} \in \mathbb{R}^{K\times R}$ and $\mathbf{a}_r$, $\mathbf{b}_r$, $\mathbf{c}_r$ are their r-th columns.

3.1. Alternating Least-Squares (ALS)

Owing to the multilinearity of CP in three factors, researchers consider one factor at a time in minimizing the cost function, while fixing the other two, i.e., solving a linear least-squares problem, an easy task. Then, three factors are found by alternately solving these three linear least squares problems, leading to the ALS algorithm.

Using Khatri-Rao product and a tensor's mode-n matricization:

$$\begin{cases} \mathbf{Y}_{(1)} \approx \mathbf{A}(\mathbf{C} \bigodot \mathbf{B})^\top, \\ \mathbf{Y}_{(2)} \approx \mathbf{B}(\mathbf{C} \bigodot \mathbf{A})^\top, \\ \mathbf{Y}_{(3)} \approx \mathbf{C}(\mathbf{B} \bigodot \mathbf{A})^\top, \end{cases} \tag{3.2}$$

the original problem (3.1) can be reformulated as

$$\begin{aligned} f(\mathbf{A},\mathbf{B},\mathbf{C}) &= \min \frac{1}{2}\left\|\mathbf{Y}_{(1)} - \mathbf{A}(\mathbf{C}\bigodot\mathbf{B})^\top\right\|_F^2 \\ &= \min \frac{1}{2}\left\|\mathbf{Y}_{(2)} - \mathbf{B}(\mathbf{C}\bigodot\mathbf{A})^\top\right\|_F^2 \\ &= \min \frac{1}{2}\left\|\mathbf{Y}_{(3)} - \mathbf{C}(\mathbf{B}\bigodot\mathbf{A})^\top\right\|_F^2, \end{aligned}$$

where $\mathbf{Y}_{(1)}$, $\mathbf{Y}_{(2)}$, $\mathbf{Y}_{(3)}$ are mode-1, mode-2, mode-3 matricizations of the tensor $\mathcal{Y}$. The recursion of ALS is as follows: at the k-th iteration with given $\mathbf{A}^k$, $\mathbf{B}^k$ and $\mathbf{C}^k$, it solves (3.3a)-(3.3c)

$$\begin{cases} \mathbf{A}^{k+1} = \arg\min f(\mathbf{A},\mathbf{B}^k,\mathbf{C}^k) = \arg\min \frac{1}{2}\|\mathbf{Y}_{(1)} - \mathbf{A}(\mathbf{C}^k\bigodot\mathbf{B}^k)^\top\|_F^2, & \text{(3.3a)} \\ \mathbf{B}^{k+1} = \arg\min f(\mathbf{A}^{k+1},\mathbf{B},\mathbf{C}^k) = \arg\min \frac{1}{2}\|\mathbf{Y}_{(2)} - \mathbf{B}(\mathbf{C}^k\bigodot\mathbf{A}^{k+1})^\top\|_F^2, & \text{(3.3b)} \\ \mathbf{C}^{k+1} = \arg\min f(\mathbf{A}^{k+1},\mathbf{B}^{k+1},\mathbf{C}) = \arg\min \frac{1}{2}\|\mathbf{Y}_{(3)} - \mathbf{C}(\mathbf{B}^{k+1}\bigodot\mathbf{A}^{k+1})^\top\|_F^2. & \text{(3.3c)} \end{cases}$$

to obtain the next iteration. These three subproblems are all linear least-squares problems, but with different coefficient matrices.

To further simplifying the analysis and implementation, we use a vectorization of a matrix, mode-n matricization of a tensor and Kronecker product to translate the objective function to the one with vector variables. Hence, we first define vectorization.

Definition 3.1. *Vectorization: The vectorization of a matrix* $\mathbf{A} = (\mathbf{a}_1\ \mathbf{a}_2\ \cdots\ \mathbf{a}_n) \in \mathbb{R}^{m\times n}$, *denoted by* $vec(\mathbf{A})$, *is a vector of size* (mn) *defined by*

$$vec(\mathbf{A}) = \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{pmatrix}.$$

Let $\mathbf{x}_1 = vec(\mathbf{A}^\top) \in \mathbb{R}^{IR}$, $\mathbf{x}_2 = vec(\mathbf{B}^\top) \in \mathbb{R}^{JR}$, and $\mathbf{x}_3 = vec(\mathbf{C}^\top) \in \mathbb{R}^{KR}$ be the three subvectors of $\mathbf{x} \in \mathbb{R}^{IR}\times\mathbb{R}^{JR}\times\mathbb{R}^{KR}$, $\mathbf{y}_{(i)} = vec(\mathbf{Y}_{(i)}^\top)$, $i = 1,2,3$. Then problem (3.1) can be converted to the following form:

$$\begin{cases} \min & f(\mathbf{x}) = f(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3) = \frac{1}{2}\sum_{i,j,k}(\mathcal{Y}_{ijk} - \sum_{r=1}^R a_{ir}b_{jr}c_{kr})^2, \\ \text{s.t.} & \mathbf{x} \in \mathbb{R}^{IR}\times\mathbb{R}^{JR}\times\mathbb{R}^{KR}. \end{cases} \tag{3.4}$$

Similarly, let $\mathbf{X}_1^k = \mathbf{I}_{I\times I}\bigotimes(\mathbf{C}^k\bigodot\mathbf{B}^k)$, $\mathbf{X}_2^k = \mathbf{I}_{J\times J}\bigotimes(\mathbf{C}^k\bigodot\mathbf{A}^{k+1})$, and $\mathbf{X}_3^k = \mathbf{I}_{K\times K}\bigotimes(\mathbf{B}^{k+1}\bigodot\mathbf{A}^{k+1})$. Then, the ALS procedure (3.3a)-(3.3c) can be rewritten as

$$\begin{cases} \mathbf{x}_1^{k+1} = \arg\min f_1^k(\mathbf{x}_1) = \arg\min f(\mathbf{x}_1,\mathbf{x}_2^k,\mathbf{x}_3^k) = \arg\min \frac{1}{2}\|\mathbf{X}_1^k\mathbf{x}_1 - \mathbf{y}_{(1)}\|_2^2, \\ \mathbf{x}_2^{k+1} = \arg\min f_2^k(\mathbf{x}_2) = \arg\min f(\mathbf{x}_1^{k+1},\mathbf{x}_2,\mathbf{x}_3^k) = \arg\min \frac{1}{2}\|\mathbf{X}_2^k\mathbf{x}_2 - \mathbf{y}_{(2)}\|_2^2, \\ \mathbf{x}_3^{k+1} = \arg\min f_3^k(\mathbf{x}_3) = \arg\min f(\mathbf{x}_1^{k+1},\mathbf{x}_2^{k+1},\mathbf{x}_3) = \arg\min \frac{1}{2}\|\mathbf{X}_3^k\mathbf{x}_3 - \mathbf{y}_{(3)}\|_2^2, \end{cases} \tag{3.5}$$

which is exactly a block nonlinear Gauss-Seidel method (GS) [18].

The convergence of GS method can be proved when the function f in (3.4) is strictly quasi-convex [33] while in general may not converge in the sense that it may produce a sequence with limit points being not the critical points of the problem. To overcome this convergence difficulty and to enhance the numerical efficiency of ALS, Li, Kindermann, and Navasca [26] propose a regularized alternating least-squares method by adding a quadratic term to each subproblem in ALS, which they called RALS.

3.2. Regularized Alternating Least-Squares (RALS)

The iterative procedure of RALS proposed by Li, Kindermann, and Navasca [26] is as follows

$$\begin{cases} \mathbf{x}_1^{k+1} = \arg\min \dfrac{1}{2}\|\mathbf{X}_1^k\mathbf{x}_1 - \mathbf{y}_{(1)}\|_2^2 + \dfrac{1}{2}\lambda_k\|\mathbf{x}_1 - \mathbf{x}_1^k\|_2^2, \\ \mathbf{x}_2^{k+1} = \arg\min \dfrac{1}{2}\|\mathbf{X}_2^k\mathbf{x}_2 - \mathbf{y}_{(2)}\|_2^2 + \dfrac{1}{2}\lambda_k\|\mathbf{x}_2 - \mathbf{x}_2^k\|_2^2, \\ \mathbf{x}_3^{k+1} = \arg\min \dfrac{1}{2}\|\mathbf{X}_3^k\mathbf{x}_3 - \mathbf{y}_{(3)}\|_2^2 + \dfrac{1}{2}\lambda_k\|\mathbf{x}_3 - \mathbf{x}_3^k\|_2^2, \end{cases} \tag{3.6}$$

where $\lambda_k > 0$ is the regularization parameter. The added regularization term drives the subproblems from convexity to strictly convexity, which further ensures the uniqueness of the solutions of the subproblems. Consequently, they can prove the following result (See Theorem 4.3 in [26]):

Theorem 3.1. *Suppose that the sequence* $(\mathbf{x}_1^k, \mathbf{x}_2^k, \mathbf{x}_3^k)$ *obtained from RALS has limit points, then every limit point* $(\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \bar{\mathbf{x}}_3)$ *is a critical point of the Problem (3.4).*

RALS not only possesses the advantage of having convergent result, but also has numerical advantage over ALS, since it can shorten the stage of swamp. However, the convergent result is built upon the assumption that the sequence $(\mathbf{x}_1^k, \mathbf{x}_2^k, \mathbf{x}_3^k)$ obtained from RALS has limit points, but it does not give any condition that ensures the whole generated sequence converges to a critical point. Also, from the numerical point of view, the regularization parameter λ_k plays an essential role, since a larger one may make the three subproblems easier while slows down the convergent speed, and a smaller one may lead to harder subproblems due to ill-conditioned subproblems. Although there are some schemes for choosing suitable parameters [27, 34], it is still a very hard and important problem in applications. We thus in the following propose an alternative, namely, the trust-region-based ALS (TRALS).

3.3. The Trust-Region-Based ALS (TRALS)

While the regularization method modifies the objective function by adding a quadratic term to ensure convergence, trust-region adds a constraint to the problem, such as [11, 31]. It is an effective method for solving optimization problems, which can guarantee the global convergence of the problem under very mild conditions. The basic idea of the trust-region method is that we can get a trial step $\mathbf{s}_k$ by solving an approximation problem (a quadratic model is usually used) of the original optimization problem within the trust-region $\{\mathbf{s}|\|\mathbf{s}\| \le \Delta_k\}$ at the k-th iteration. The trial step makes $\mathbf{x}_k + \mathbf{s}_k$ to be "best-point" in a generalized spherical $\Omega_k = \{\mathbf{x}_k + \mathbf{s}|\|\mathbf{s}\| \le \Delta_k\}$. So the trust-region method sometimes sets up $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$ or $\mathbf{x}_{k+1} = \mathbf{x}_k$ directly, and adjusts the trust-region radius according to the ratio between the predicted reduction and the actual reduction.

At iteration k with the given information, the trust-region based ALS method first solves the three trust region subproblems:

$$\begin{cases} \min\ f_1(\mathbf{x}_1^k+\mathbf{s}_1)=\dfrac{1}{2}\|\mathbf{X}_1^k(\mathbf{x}_1^k+\mathbf{s}_1)-\mathbf{y}_{(1)}\|_2^2 \\ \text{s.t.}\quad \|\mathbf{s}_1\|_2\le\Delta_1^k \\ \min\ f_2(\mathbf{x}_2^k+\mathbf{s}_2)=\dfrac{1}{2}\|\mathbf{X}_2^k(\mathbf{x}_2^k+\mathbf{s}_2)-\mathbf{y}_{(2)}\|_2^2 \\ \text{s.t.}\quad \|\mathbf{s}_2\|_2\le\Delta_2^k \\ \min\ f_3(\mathbf{x}_3^k+\mathbf{s}_3)=\dfrac{1}{2}\|\mathbf{X}_3^k(\mathbf{x}_3^k+\mathbf{s}_3)-\mathbf{y}_{(3)}\|_2^2 \\ \text{s.t.}\quad \|\mathbf{s}_3\|_2\le\Delta_3^k, \end{cases} \tag{3.7}$$

whose solutions are denoted as $(\mathbf{s}_1^k,\mathbf{s}_2^k,\mathbf{s}_3^k)$. To adjust the radius of the trust region, we thus define another ratio

$$\rho_i^k=\frac{f_i(\mathbf{x}_k+\mathbf{s}_k)}{f_i(\mathbf{x}_k)}.$$

Based on ρ_i^k, we decide whether to increase or decrease the trust-region radius Δ_i^k and update $\mathbf{x}_i^k$, for $i=1,\ 2,\ 3$. The adjustment rule is as usual, i.e., if ρ_i^k is small, the parameter Δ_i^k can be reduced, and if ρ_k is large while $\rho_i^k<\delta<1$, the parameter Δ_i^k can be enlarged. Furthermore, if ρ_i^k is close to 1, the trust-region radius Δ_i^k should be reduced too.

The subproblems in (3.7) are trust-region problems [11, 31]. Denote a general trust-region problem as

$$\min\ \Psi(\mathbf{s})=\frac{1}{2}\|\mathbf{A}(\mathbf{x}+\mathbf{s})-\mathbf{b}\|_2^2,\ \text{s.t.}\ \|\mathbf{s}\|_2\le\Delta. \tag{3.8}$$

When

$$\Delta=\|(\mathbf{G}+\mu\mathbf{I})^{-1}\mathbf{g}_{\mathbf{x}}\|_2, \tag{3.9}$$

where $\mathbf{g}_{\mathbf{x}}=\mathbf{A}^\top\mathbf{A}\mathbf{x}-\mathbf{A}^\top\mathbf{b}$ and $\mathbf{G}=\mathbf{A}^\top\mathbf{A}$, (3.8) is equivalent to solving the unconstrained problem

$$\min\ \frac{1}{2}\{\|\mathbf{A}(\mathbf{x}+\mathbf{s})-\mathbf{b}\|_2^2+\mu\|\mathbf{s}\|_2^2\}, \tag{3.10}$$

whose solution can be obtained by solving the following linear equation

$$(\mathbf{G}+\mu\mathbf{I})\mathbf{s}=-\mathbf{g}_{\mathbf{x}}. \tag{3.11}$$

In other words, for solving the trust-region problems problem (3.8), we can first solve (3.11) and then adjust the parameter so that the condition (3.9) is satisfied.

We now describe the TRALS algorithm formally in Algorithm 1.

Generally, extrapolation techniques can enhance the performance of the algorithm. Hence, in order to accelerate the convergence of TRALS, we propose to adopt the following simple extrapolation formula to generate the next iterate

$$\mathbf{x}_i^{k+1}=\gamma\tilde{\mathbf{x}}_i^{k+1}+(1-\gamma)\mathbf{x}_i^k,\qquad i=1,2,3, \tag{3.12}$$

where $\gamma\in(1,2)$ is a fixed positive constant, $\tilde{\mathbf{x}}_i^{k+1}=\mathbf{x}_i^k+\mathbf{s}_i^{k+1}$ and $\mathbf{s}_i^{k+1}$ is an approximate solution of the trust-region subproblem. If $\gamma=1$, there is no extrapolation. We denote this algorithm as trust-region-based alternating least-squares algorithm with extrapolation (ETRALS), which is now formally described in Algorithm 2.

Algorithm 3.1. ALS with Trust-Region (TRALS)

Give initial $\mathbf{A}^0,\ \mathbf{B}^0,\ \mathbf{C}^0,\ \mu_i^0 > 0,\ \mu_{\max} > \mu_{\min} > 0,\ \eta_i^1 < \eta_i^2 < 1,\ 0 < \beta_i^1 < 1, \beta_i^2 > 1,$ $i = 1, 2, 3.$

while stopping criterion is not satisfied **do**

Obtain the $\mathbf{s}_1^{k+1}$ by solving (3.10), compute ρ_1^k.

$\mathbf{x}_1^{k+1} = \mathbf{x}_1^k + \mathbf{s}_1^{k+1}$, translate $\mathbf{x}_1^{k+1}$ into $\mathbf{A}^{k+1}$.

$$\tilde{\mu}_1^{k+1} = \begin{cases} \min\{\beta_1^2 \mu_1^k, \mu_{\max}\},\ if\ \eta_1^1 < \rho_1^k \le \eta_1^2; \\ \beta_1^1 \mu_1^k,\ otherwise, \end{cases}$$

$$\mu_1^{k+1} = \max\{\tilde{\mu}_1^{k+1}, \mu_{\min}\}.$$

Use the same method as 3-4 to get $\mathbf{B}^{k+1},\ \mu_2^{k+1},\ \mathbf{C}^{k+1},\ \mu_3^{k+1}$.

end while.

Algorithm 3.2. Extrapolation TRALS (ETRALS)

Give initial $\mathbf{A}^0,\ \mathbf{B}^0,\ \mathbf{C}^0,\ \mu_i^0 > 0,\ \mu_{\max} > \mu_{\min} > 0,\ \eta_i^1 < \eta_i^2 < 1,\ 0 < \beta_i^1 < 1, \beta_i^2 > 1,$ $i = 1, 2, 3.$

while stopping criterion is not satisfied **do**

Obtain $\mathbf{s}_1^{k+1}$ by solving (3.10), compute ρ_1^k.

$\tilde{\mathbf{x}}_1^{k+1} = \mathbf{x}_1^k + \mathbf{s}_1^{k+1}$, translate $\tilde{\mathbf{x}}_1^{k+1}$ into $\tilde{\mathbf{A}}^{k+1}$.

$$\tilde{\mu}_1^{k+1} = \begin{cases} \min\{\beta_1^2 \mu_1^k, \mu_{\max}\},\ if\ \eta_1^1 < \rho_1^k \le \eta_1^2; \\ \beta_1^1 \mu_1^k,\ otherwise, \end{cases}$$

$$\mu_1^{k+1} = \max\{\tilde{\mu}_1^{k+1}, \mu_{\min}\}.$$

Use the same method as 3-4 to get $\tilde{\mathbf{B}}^{k+1},\ \mu_2^{k+1},\ \tilde{\mathbf{C}}^{k+1},\ \mu_3^{k+1}$.

Get the next iterate via a simple extrapolation

$$\mathbf{A}^{k+1} = \gamma \tilde{\mathbf{A}}^{k+1} + (1-\gamma)\mathbf{A}^k,$$
$$\mathbf{B}^{k+1} = \gamma \tilde{\mathbf{B}}^{k+1} + (1-\gamma)\mathbf{B}^k,$$
$$\mathbf{C}^{k+1} = \gamma \tilde{\mathbf{C}}^{k+1} + (1-\gamma)\mathbf{C}^k.$$

end while.

4. Convergence of ETRALS

Since TRALS is a special case of ETRALS with $\gamma = 1$, in the section, we just discuss the convergence of ETRALS. We begin our analysis with the following lemma.

Lemma 4.1. *Let* $\{\mathbf{x}_1^k,\ \mathbf{x}_2^k,\ \mathbf{x}_3^k\}$ *be the sequence generated by Algorithm* 2*, then*

1. *The sequence* $\{f(\mathbf{x}_1^k, \mathbf{x}_2^k, \mathbf{x}_3^k)\}$ *is convergent;*
2. *When* $k \to \infty$, $\|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\| \to 0,\ i = 1, 2, 3$*;*

3. *Set* $\mathbf{x}^k = (\mathbf{x}_1^k, \mathbf{x}_2^k, \mathbf{x}_3^k)$. *If*

$$\sum_{k=0}^{\infty} \|\mathbf{x}^{k+1} - \mathbf{x}^k\| < \infty, \tag{4.1}$$

then the sequence $\{\mathbf{x}^k\} = \{\mathbf{x}_1^k, \mathbf{x}_2^k, \mathbf{x}_3^k\}$ *is convergent.*

Proof. We prove the lemma item by item.

1. Note that by Algorithm 2, $\mathbf{x}_1^{k+1} - \mathbf{x}_1^k = \gamma(\tilde{\mathbf{x}}_1^{k+1} - \mathbf{x}_1^k)$. Consequently,

$$\begin{aligned}
&f_1(\mathbf{x}_1^k) - f_1(\mathbf{x}_1^{k+1}) \\
&= \frac{1}{2}\|\mathbf{X}_1^k\mathbf{x}_1^k - \mathbf{y}_{(1)}\|_2^2 - \frac{1}{2}\|\mathbf{X}_1^k\mathbf{x}_1^{k+1} - \mathbf{y}_{(1)}\|_2^2 \\
&= \frac{1}{2}\|\mathbf{X}_1^k\mathbf{x}_1^k - \mathbf{y}_{(1)}\|_2^2 - \frac{1}{2}\|\mathbf{X}_1^k\mathbf{x}_1^k + \gamma\mathbf{X}_1^k(\tilde{\mathbf{x}}_1^{k+1} - \mathbf{x}_1^k) - \mathbf{y}_{(1)}\|_2^2 \\
&= \frac{1}{2}\{-2\gamma(\mathbf{X}_1^k\mathbf{x}_1^k - \mathbf{y}_{(1)})^\top\mathbf{X}_1^k(\tilde{\mathbf{x}}_1^{k+1} - \mathbf{x}_1^k) - \gamma^2\|\mathbf{X}_1^k(\tilde{\mathbf{x}}_1^{k+1} - \mathbf{x}_1^k)\|_2^2\}.
\end{aligned}$$

Because by (3.11),

$$-(\mathbf{X}_1^k\mathbf{x}_1^k - \mathbf{y}_{(1)})^\top\mathbf{X}_1^k = (\tilde{\mathbf{x}}_1^{k+1} - \mathbf{x}_1^k)^\top((\mathbf{X}_1^k)^\top(\mathbf{X}_1^k) + \mu_1^k\mathbf{I}),$$

which leads to

$$\begin{aligned}
&f_1^k(\mathbf{x}_1^k) - f_1^k(\mathbf{x}_1^{k+1}) \\
&= \frac{1}{2}\left\{2\gamma(\tilde{\mathbf{x}}_1^{k+1} - \mathbf{x}_1^k)^\top((\mathbf{X_1}^k)^\top(\mathbf{X_1}^k) + \mu_1^k\mathbf{I})^\top(\tilde{\mathbf{x}}_1^{k+1} - \mathbf{x}_1^k) - \gamma^2\|\mathbf{X_1}^k(\tilde{\mathbf{x}}_1^{k+1} - \mathbf{x}_1^k)\|_2^2\right\} \\
&= \frac{1}{2}\left\{2\gamma\|\tilde{\mathbf{x}}_1^{k+1} - \mathbf{x}_1^k\|^2_{((\mathbf{X_1}^k)^\top(\mathbf{X_1}^k)+\mu_1^k\mathbf{I})} - \gamma^2\|\tilde{\mathbf{x}}_1^{k+1} - \mathbf{x}_1^k\|^2_{(\mathbf{X_1}^k)^\top(\mathbf{X_1}^k)+\mu_1^k\mathbf{I})}\right\} \\
&\geq \frac{1}{2}(2-\gamma)\gamma\mu_{\min}\|\tilde{\mathbf{x}}_1^{k+1} - \mathbf{x}_1^k\|_2^2.
\end{aligned} \tag{4.2}$$

Similarly,

$$f_2^k(\mathbf{x}_2^k) - f_2^k(\mathbf{x}_2^{k+1}) \geq \frac{1}{2}(2-\gamma)\gamma\mu_{\min}\|\tilde{\mathbf{x}}_2^{k+1} - \mathbf{x}_2^k\|_2^2, \tag{4.3}$$

$$f_3^k(\mathbf{x}_3^k) - f_3^k(\mathbf{x}_3^{k+1}) \geq \frac{1}{2}(2-\gamma)\gamma\mu_{\min}\|\tilde{\mathbf{x}}_3^{k+1} - \mathbf{x}_3^k\|_2^2. \tag{4.4}$$

Adding (4.2)-(4.4) yields

$$f(\mathbf{x}_1^k, \mathbf{x}_2^k, \mathbf{x}_3^k) - f(\mathbf{x}_1^{k+1}, \mathbf{x}_2^{k+1}, \mathbf{x}_3^{k+1}) \geq \frac{1}{2}(2-\gamma)\gamma\mu_{\min}\sum_{i=1}^{3}\|\tilde{\mathbf{x}}_i^{k+1} - \mathbf{x}_i^k\|_2^2, \tag{4.5}$$

meaning that $f(\mathbf{x}_1^k, \mathbf{x}_2^k, \mathbf{x}_3^k) \geq f(\mathbf{x}_1^{k+1}, \mathbf{x}_2^{k+1}, \mathbf{x}_3^{k+1})$, and $\{f(\mathbf{x}_1^k, \mathbf{x}_2^k, \mathbf{x}_3^k)\}$ is nonincreasing. As a consequence, the sequence $\{f(\mathbf{x}_1^k, \mathbf{x}_2^k, \mathbf{x}_3^k)\}$ is convergent due to its nonnegativity.

2. Summing (4.5) over k through 0 and K, we have

$$f(\mathbf{x}_1^0, \mathbf{x}_2^0, \mathbf{x}_3^0) - f(\mathbf{x}_1^K, \mathbf{x}_2^K, \mathbf{x}_3^K) \geq \frac{1}{2}(2-\gamma)\gamma\mu_{\min}\sum_{k=0}^{K}\sum_{i=1}^{3}\|\tilde{\mathbf{x}}_i^{k+1} - \mathbf{x}_i^k\|_2^2. \tag{4.6}$$

Due to $\mu_{min} > 0$ and the function f is nonnegative, $\gamma \in (1, 2)$, it follows that

$$\sum_{k=1}^{\infty}\sum_{i=1}^{3} \|\tilde{\mathbf{x}}_i^{k+1} - \mathbf{x}_i^k\|_2^2 < \infty,$$

indicating that

$$\|\tilde{\mathbf{x}}_i^{k+1} - \mathbf{x}_i^k\|_2^2 \to 0, \quad i = 1, 2, 3. \tag{4.7}$$

Noting that $\|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\| = \gamma\|\tilde{\mathbf{x}}_i^{k+1} - \mathbf{x}_i^k\|$, $i = 1, 2, 3$, (4.7) implies

$$\|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\| \to 0, \quad i = 1, 2, 3.$$

3. Using the triangle inequality, it follows from the condition (4.1) that

$$\|\mathbf{x}^k\|_2 \le \|\mathbf{x}^{k-1}\|_2 + \|\mathbf{x}^k - \mathbf{x}^{k-1}\|_2 \le \cdots \le \|\mathbf{x}^0\|_2 + \sum_{l=1}^{\infty} \|\mathbf{x}^l - \mathbf{x}^{l-1}\|_2 < \infty,$$

indicating the sequence $\{\mathbf{x}^k\} = \{\mathbf{x}_1^k, \mathbf{x}_2^k, \mathbf{x}_3^k\}$ is bounded. On the other hand, since $\{f(\mathbf{x}^k)\}$ is convergent, for any $\varepsilon > 0$, there is a positive integer K such that for all $k > K$ and for any positive integer M,

$$f(\mathbf{x}^k) - f(\mathbf{x}^{k+M}) < \frac{1}{2}(2 - \gamma)\varepsilon,$$

which, together with (4.5), means that

$$\frac{1}{2}(2 - \gamma)\gamma \sum_{l=1}^{M}\sum_{i=1}^{3} \|\tilde{\mathbf{x}}_i^{k+l} - \mathbf{x}_i^{k+l-1}\|_2^2 < \frac{1}{2}(2 - \gamma)\varepsilon,$$

or

$$\sum_{i=1}^{M}\sum_{i=1}^{3} \|\tilde{\mathbf{x}}_i^{k+l} - \mathbf{x}_i^{k+l-1}\|_2^2 < \frac{\varepsilon}{\gamma}.$$

Hence, for any $\varepsilon > 0$, there is a positive integer K, for all $k > K$ and any positive integer M,

$$\|\mathbf{x}^{k+M} - \mathbf{x}^k\|_2^2 \le \sum_{l=1}^{M} \|\mathbf{x}^{k+l} - \mathbf{x}^{k+l-1}\|_2^2 = \sum_{l=1}^{M}\sum_{i=1}^{3} \|\tilde{\mathbf{x}}_i^{k+l} - \mathbf{x}_i^{k+l-1}\|_2^2 < \frac{\varepsilon}{\gamma},$$

implying that $\{\mathbf{x}^k\}$ is a *Cauchy* sequence, and the assertion follows immediately. □

Theorem 4.1. *Suppose that* (4.1) *holds. Then* $\{\mathbf{x}^k\} = \{\mathbf{x}_1^k, \mathbf{x}_2^k, \mathbf{x}_3^k\}$ *converges to a critical point of the problem* (3.1).

Proof. In Lemma 4.1, we have proved that $\{\mathbf{x}^k\}$ converges and we here just need to prove that the limit point $\{\bar{\mathbf{x}}\} = \{\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \bar{\mathbf{x}}_3\}$ of $\{\mathbf{x}^k\}$ is a critical point of the problem (3.1), i.e.,

$$\|\nabla_{\bar{\mathbf{x}}_i} f(\bar{\mathbf{x}})\| = 0, \quad i = 1, 2, 3.$$

In Lemma 4.1, we have proved that $\|\tilde{\mathbf{x}}_i^{k+1} - \mathbf{x}_i^k\|_2^2 \to 0$, $i = 1, 2, 3$, so $\|\mathbf{s}_i^k\| \to 0$, $i = 1,\ 2,\ 3$ since $\mathbf{s}_i^k = \tilde{\mathbf{x}}_i^{k+1} - \mathbf{x}_i^k$. Note that

$$\nabla_{\mathbf{x}_i^k} f_i^k(\mathbf{x}_i^k) = \mathbf{g}_i^k = {\mathbf{X}_i^k}^{\top}\mathbf{X}_i^k\mathbf{x}_i^k - {\mathbf{X}_i^k}^{\top}\mathbf{y}_i,$$

and from (3.11),

$$\mathbf{g}_i^k = -(\mathbf{X}_i^{k\top}\mathbf{X}_i^k + \mu_i^k\mathbf{I})\mathbf{s}_i^{k+1}.$$

Since $\mu_i^k < \mu_{\max}$ and $\{\mathbf{X}_i^{k\top}\mathbf{X}_i^k\}$ is bounded, $\|(\mathbf{X}_i^{k\top}\mathbf{X}_i^k + \mu_i^k\mathbf{I})\mathbf{s}_i^{k+1}\| \to 0$. Hence,

$$\lim_{k\to\infty} \|\nabla_{\mathbf{x}_i^k} f_i(\mathbf{x}_i^k)\| = \|\nabla_{\bar{\mathbf{x}}_i} f(\bar{\mathbf{x}})\| = 0, \quad i = 1, 2, 3,$$

and the proof is complete. □

5. Numerical experiments

In this section, we will consider the models PARAFAC and BCM to test the new algorithm ETRALS by some numerical experiments, and compare it with ALS and RALS. Besides, we add the same extrapolation formula as the ETRALS to ALS (EALS) in the next numerical experiments. All codes were written in MATLAB R2009a.

5.1. Example1: Find an appropriate γ

The extrapolation parameter γ can usually enhance the performance of the algorithm and the choice of γ=1 leads ETRALS to TRALS. We thus first use a PARAFAC model [26] to observe its effect with different values. Let the matrices

$$\mathbf{A} = \begin{pmatrix} 1 & \cos(\theta) & 0 \\ 0 & \sin(\theta) & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 3 & \sqrt{2}\cos(\theta) & 0 \\ 0 & \sin(\theta) & 1 \\ 0 & \sin(\theta) & 0 \end{pmatrix}, \quad \mathbf{C} = \mathbf{I}_{3\times 3}.$$

be the three factor matrices of a third-order tensor $\mathcal{Y} \in \mathbb{R}^{2\times 3\times 3}$

$$\mathcal{Y}=\mathbf{a}_1 \circ \mathbf{b}_1 \circ \mathbf{c}_1 + \mathbf{a}_2 \circ \mathbf{b}_2 \circ \mathbf{c}_2 + \mathbf{a}_3 \circ \mathbf{b}_3 \circ \mathbf{c}_3,$$

where θ is a parameter. It is obvious that as $\theta \to 0$, the columns of $\mathbf{A}$ and $\mathbf{B}$ become collinear. In this experiment we set the maximal number of iterations 5000, and choose $\theta = \frac{\pi}{120}$. Each entry of the starting points $\mathbf{A}^0, \mathbf{B}^0$ and $\mathbf{C}^0$ is drawn from the normal distribution $\mathcal{N}(0, 1)$. The stopping criterion is $\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 \le 1 \times 10^{-5}$, where $\hat{\mathcal{Y}}$ stands for the obtained tensor after every iteration. In Figure 5.1, we give the mean number of iterations when $\gamma = 1, 1.2, 1.4, 1.6, 1.8, 2$, after 1000 random experiments. From Figure 5.1, the best result is 750 iterations with $\gamma = 1.8$. ETRALS can not reach the stopping error tolerance within 5000 iterations, with $\gamma = 2$. So in the next experiments we always choose $\gamma = 1.8$.

5.2. Example 2: Application of BCM decomposition in DS-CDMA

In this section, we examine the performance of ETRALS by applying it to BCM, i.e., to solve the problem of blind separation-equalization of convolutive DS-CDMA mixtures received by an antena array after multipath propagation, and compare them with the ALS, RALS and EALS.

The following assumptions on the model are the same as those in [28]. We assume that the signal of the r-th user is subject to inter-symbol-interference over L consecutive symbols and that this signal arrives at antenna array via P specular paths. For user r, $r = 1, \cdots, R$, the $I \times L$ frontal slice $\mathcal{H}_r(:, :, p)$ of $\mathcal{H}_r$ collects samples of the convolved spreading waveform

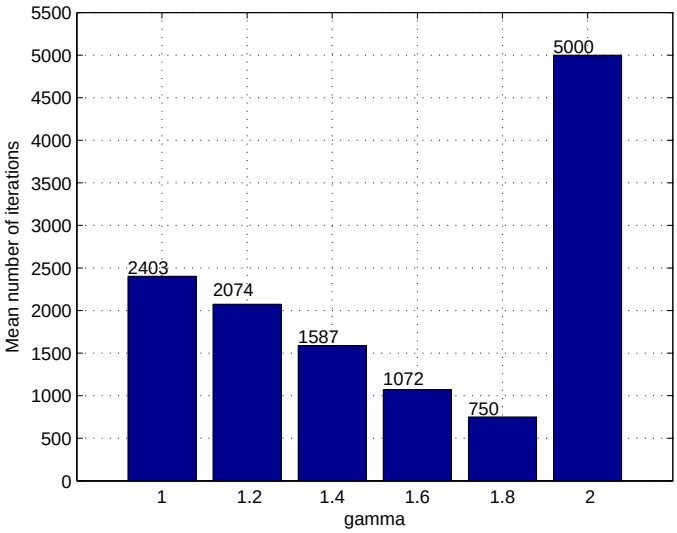

Fig. 5.1. Iterations versus γ.

associated to the p-th path, $p = 1, \cdots, P$. The $J \times L$ matrix $\mathbf{S}_r$ holds the J transmitted symbols and has a Toeplitz structure. The $K \times P$ matrix $\mathbf{A}_r$ collects the response of the K antennas according to the angles of arrival of the P paths. In our experiments, we consider $R = 4$ users, pseudorandom spreading codes of length $I = 8$, a short frame of $J = 50$ QPSK symbols, $K = 4$ antennas, $L = 2$ interfering symbols, and $P = 2$ paths per user. For this BCM tensor decomposition model, we use Mode-n matricization of a tensor and Khatri-Rao product, and the problem is formulated as the following least-squares problem

$$\min \frac{1}{2} \left\| \mathbf{Y} - (\mathbf{S} \bigodot_R \mathbf{A}) \mathbf{H}^\top \right\|_F^2,$$

where $\mathbf{Y} \in \mathbb{R}^{JL \times I}$, $\mathbf{S} \in \mathbb{R}^{J \times RL}$, $\mathbf{A} \in \mathbb{R}^{K \times RP}$, $\mathbf{H} \in \mathbb{R}^{I \times RLP}$.

In the first experiment, the signal-to-noise ratio (SNR) at the input of the BCM receiver is defined by

$$\text{SNR} = 10 \log_{10} \left(\frac{\|\mathcal{Y}\|_F^2}{\|\mathcal{N}\|_F^2} \right), \tag{5.1}$$

where $\mathcal{Y}$ is the real-valued noise-free tensor of observations and $\mathcal{N}$ is the tensor of zero-mean white (in all dimensions) Gaussian noise. We run 1000 Monte Carlo experiments. For each one, the initialization of the iterate is performed by selecting the best one from ten different random starting points, which obey the standard Gaussian distribution. We stop the algorithm when the iteration number exceeds 500 or the stopping tolerance $\|\hat{\mathcal{Y}}^{(n)} - \hat{\mathcal{Y}}^{(n-1)}\|_F \leq \epsilon$ is satisfied, and ϵ is set to be 10^{-6}.

We show mean bit error rate[1)] (BER) over all users in Figure 5.2. From Figure 5.2 we can see that EALS, RALS, TRALS, ETRALS methods work better than the ALS method, and ETRALS has the best effect among all the methods.

We then compare these five algorithms from the number of iterations and CPU time perspectives. Table 5.1 lists the number of iterations and CPU time of these methods, with different SNR. The results in Table 5.1 indicate that as the SNR increases, the number of iterations and

[1)] The bit error rate (BER) is the number of bit errors per unit time.

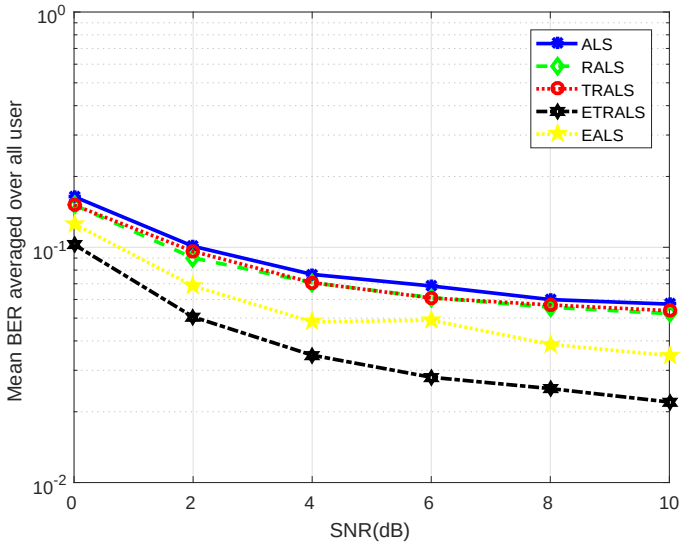

Fig. 5.2. BER versus SNR.

CPU time of all the five methods decrease. The number of iterations of RALS is less than ALS and more than TRALS. When SNR $= 0, 2, 4, 8$, the number of iterations of TRALS is more than EALS. When SNR $= 6, 10$, the number of iterations of TRALS is less than EALS. The most important is that ETRALS needs the least number of iterations. As to the CPU time of the five methods, comparing with ALS, RALS and EALS, solving every subproblems of TRALS and ETRALS will cost more time. TRALS takes more time than ALS and RALS, when SNR $= 0, 2, 4$. When SNR $= 6, 8, 10$, the CPU time of TRALS is almost as same as ALS and RALS. Because EALS and ETRALS save a lot of number of iterations, the CPU costs of them are the least among the five methods.

SNR	Iteration					CPU(s)				
	ALS	RALS	TRALS	EALS	ETRALS	ALS	RALS	TRALS	EALS	ETRALS
SNR=0	344.86	322.72	299.31	275.50	230.83	4.58	4.77	5.12	3.66	3.96
SNR=2	282.52	262.81	231.07	209.53	163.50	3.75	3.84	3.98	2.78	2.82
SNR=4	244.22	223.84	192.85	184.98	138.77	3.25	3.29	3.36	2.46	2.41
SNR=6	228.06	202.93	169.13	185.23	122.69	3.04	2.99	2.93	2.46	2.13
SNR=8	213.60	198.68	162.30	160.44	111.44	2.84	2.94	2.84	2.13	1.95
SNR=10	206.40	177.94	148.86	160.04	102.87	2.75	2.63	2.60	2.13	1.80

Table 5.1: Mean number of iterations and CPU time.

To make the results in Table 5.1 more intuitional, we plot them in Figure 5.3.

In the second experiment, we consider a nearly collinear case. When the signals of the users are nearly collinear, which may happen in practice, the convergence of ALS is very slow due to the swamp phenomenon. In the experiment, the first user's signal is random QPSK symbols, and the other users' signals differ from it in just one bit. The condition number of the factor $\mathbf{A}$ is 10, and the tensor $\mathcal{Y}$ is noise free. The typical curve of the cost function values versus number of iterations is shown in Figure 5.4. It is clear that the algorithms converge to the right decomposition with different number of iterations. The ALS takes 3903 iterations to get

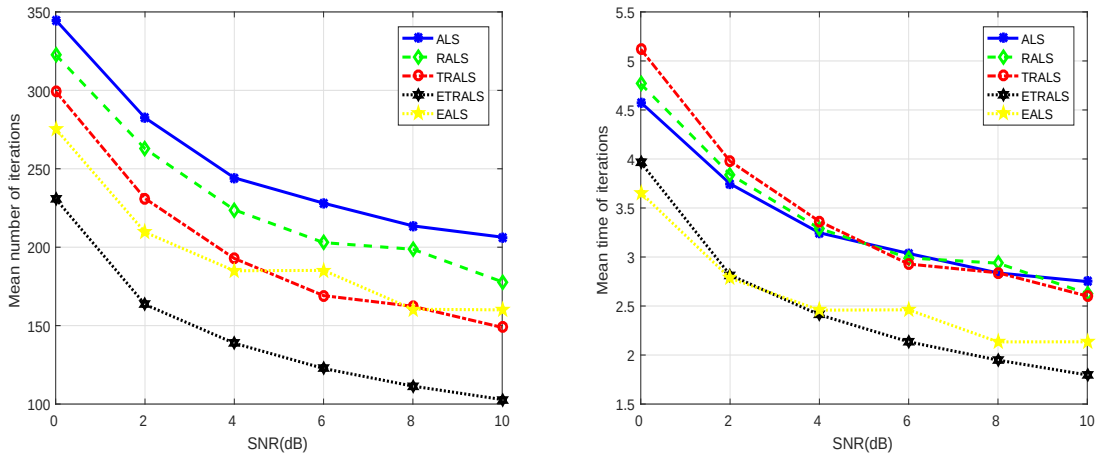

Fig. 5.3. Performance of mean number of iterations and CPU time.

the objective error 10^{-6}. RALS needs 2714 iterations and TRALS costs 2430 iterations. EALS needs 2054 iterations. The best one, ETRALS, only needs 1388 iterations.

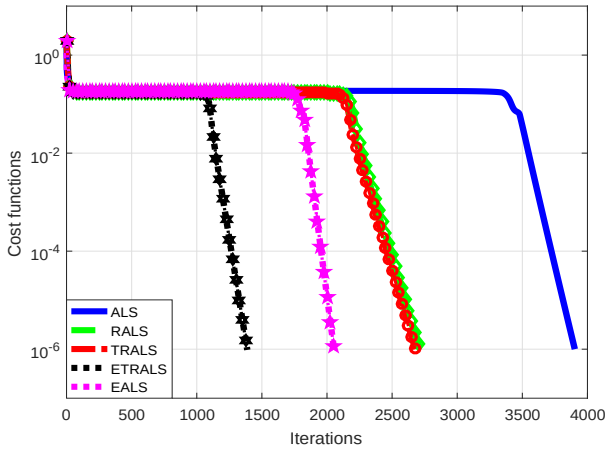

Fig. 5.4. Cost function values versus iterations.

5.3. Example 3: Application of CP in the fluorescence of amino acids

This example is about the fluorescence of amino acids. The data set on the fluorescence of amino acids comes from chemometrics, consisting of five simple laboratory-made samples of tyrosine, tryptophan and phenylalanine dissolved in phosphate buffered water. Samples $1-3$ are single amino acid: Tryptophan, Tyrosine and Phenylalanine, respectively, while $4-5$ are their mixture. The samples are measured by fluorescence (excitation $250-300$ nm, emission $250-450$ nm, 1 nm intervals) on a spectrofluorometer. The data is a tensor $\mathcal{X}$ with size of $5\times 61\times 201$. Figure 5.5 shows these five samples.

Because there are three kinds of amino acids , we first choose a reasonable rank $R=3$. Sometimes, the overestimation of the rank may happen in chemometrics, and we choose another

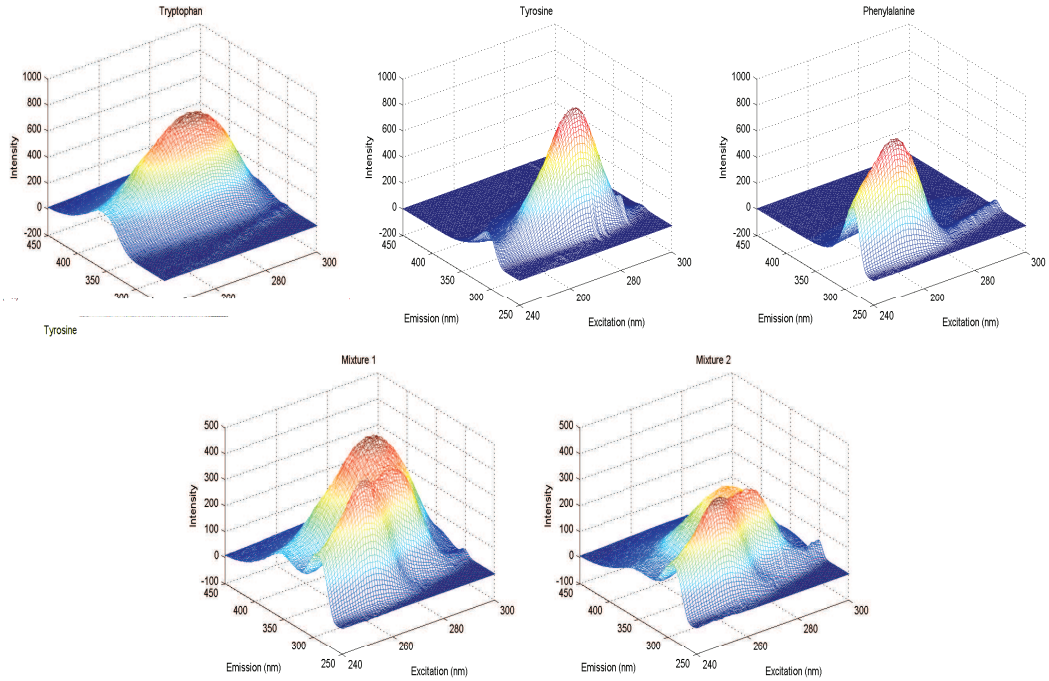

Fig. 5.5. The fluorescence landscapes of the samples.

Table 5.2: Number of iterations and CPU time.

R	Iteration					CPU(s)				
	ALS	RALS	TRALS	EALS	ETRALS	ALS	RALS	TRALS	EALS	ETRALS
R=3	103	103	103	65	65	0.40	0.47	0.63	0.25	0.40
R=4	1214	1198	726	932	392	5.09	5.94	4.38	4.02	2.42

over-rated rank $R = 4$ to test its effect. All the five algorithms, ALS, RALS, TRALS, EALS, and ETRALS, start with the same initial point, whose mode-n factor is the R leading eigenvectors of $\mathbf{X}_{(n)}\mathbf{X}_{(n)}^{\top}$, and $\mathbf{X}_{(n)}$ is the mode-n matricization of the tensor $\mathcal{X}$.

Figure 5.6 shows the excitation and emission spectra with $R = 3$. The red, blue, and magenta lines stand for the spectra of the tryptophan, the tyrosine and the phenylalanine, respectively. The results in Figure 5.6 indicate that for this case, all the algorithms can find good approximate solutions. The results for the case that $R = 4$ are shown in Figure 5.7. In Figure 5.7, we can see that there is a black dotted line in every subgraph. This black dotted line stands for an nonexistent element, due to the overestimation of the rank.

Although these four algorithms can find quite good solutions, their efficiencies are different. Table 5.2 lists the number of iterations and CPU time of these methods, with $R = 3$ and $R = 4$. We also plot the error $\|\mathcal{X}^{k+1} - \mathcal{X}^{k}\|_F^2$ versus the number of iterations used to reach the tolerance of 1×10^{-8} in Figure 5.8, where the left one is for the case when $R = 3$ and the right one for the case when $R = 4$. From the left one of Figure 5.8 and Table 5.2, we can find that ALS, RALS, TRALS have nearly the same effect of taking 103 iterations to reach the tolerance. The best are EALS and ETRALS, which takes 65 iterations. From right one of Figure 5.8 and Table 5.2, we find ALS, RALS, TRALS, EALS and ETRALS process slowly than their counterparts in the left one. ALS needs 1214 iterations to get the tolerance; RALS is slightly better than ALS,

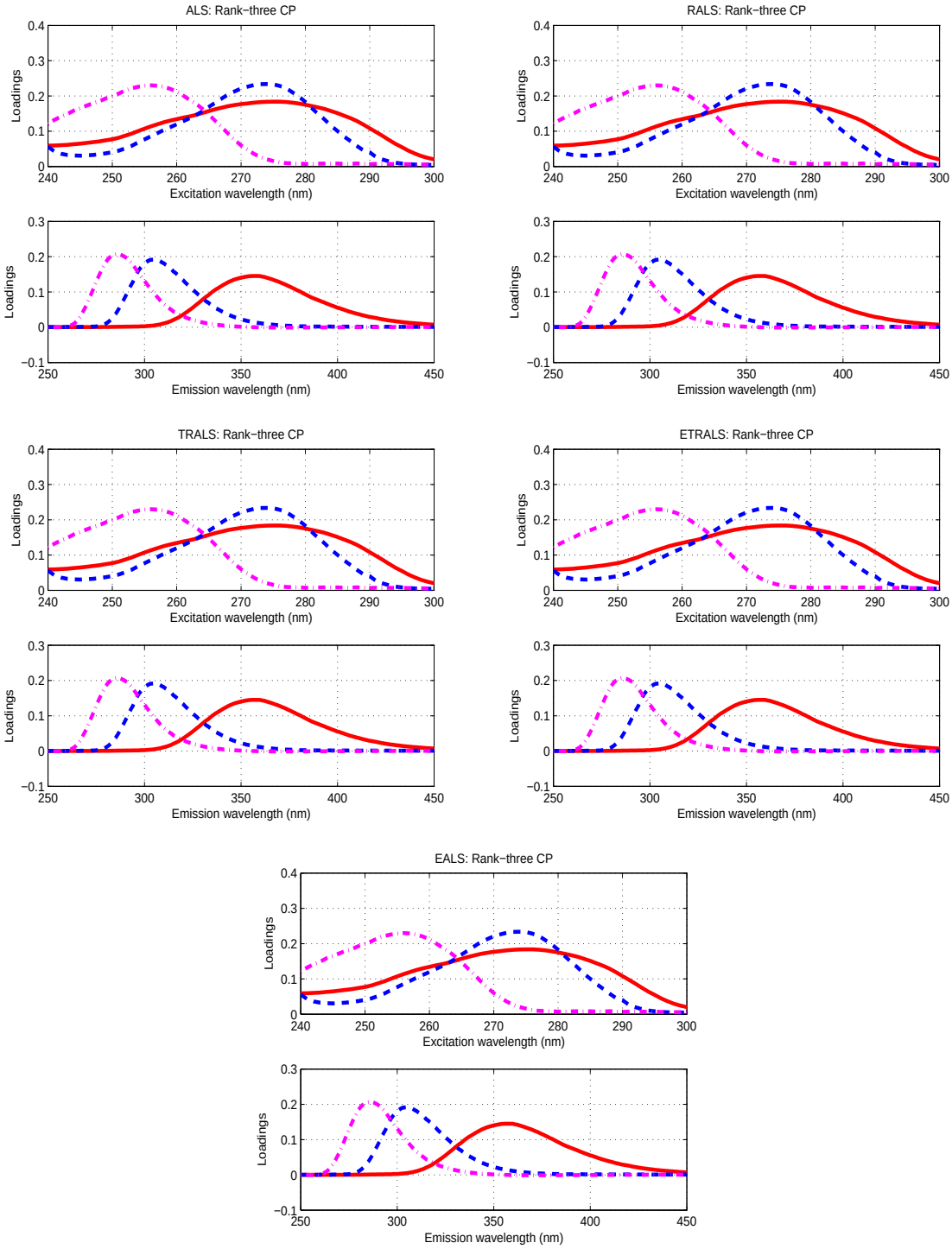

Fig. 5.6. The excitation and emission of amino acids with $R = 3$.

it takes 1198 iterations; EALS and TRALS takes 932 and 726 iterations, respectively; and the best one is still ETRALS, which needs 392 iterations.

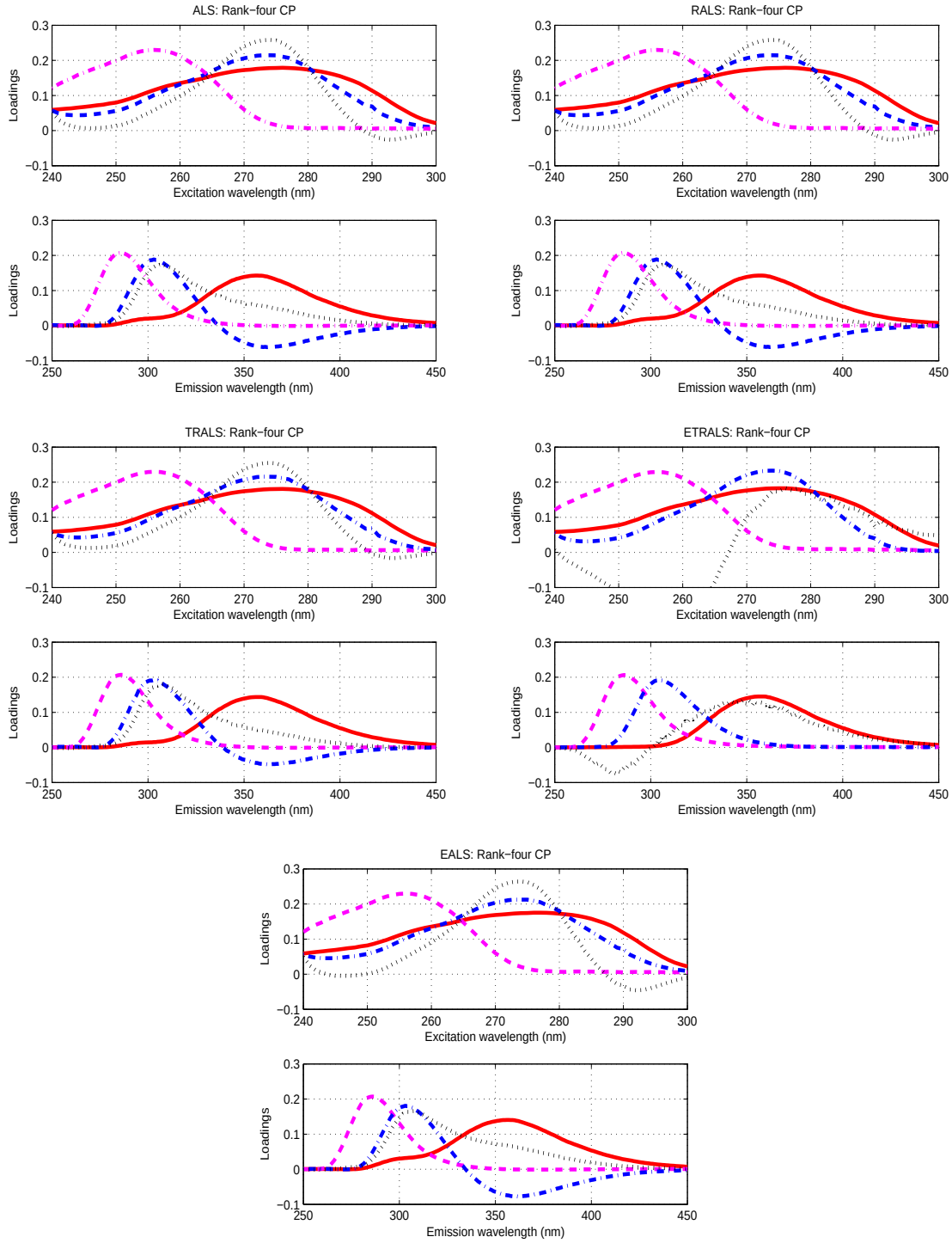

Fig. 5.7. The excitation and emission of amino acids with $R = 4$.

5.4. Example 4: Application of a rank-$(L_r, L_r, 1)$ decomposition in signal processing

We consider a rank-$(L_r, L_r, 1)$ block term decomposition. We first consider an objective tensor $\mathcal{Y}$ without noise and then a tensor with Gaussian noise. For each experiment, we generate

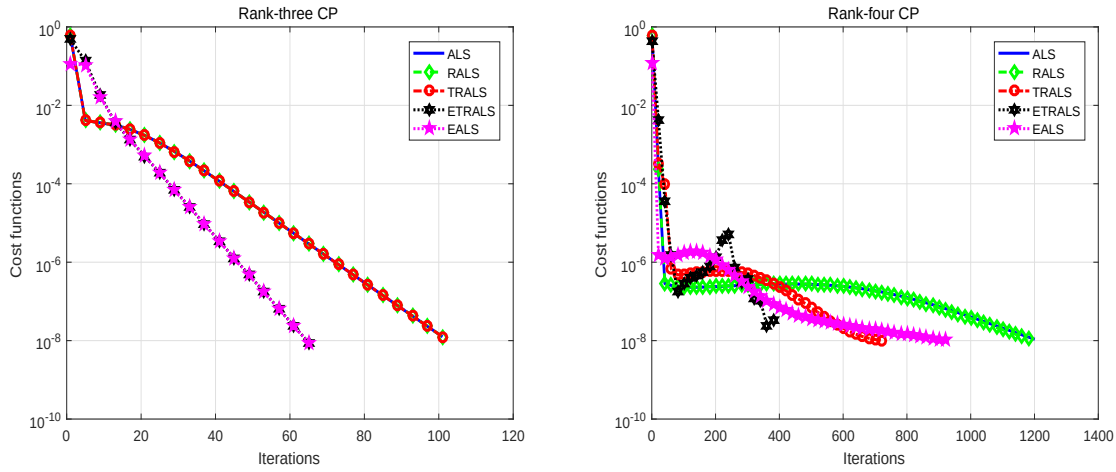

Fig. 5.8. The left one is $R = 3$, and the right one is $R = 4$.

20 objective tensors, and the factors of the tensors are drawn from a uniform distribution in the interval $(0, 1)$. For each tensor, we run 50 Monte Carlo experiments, so 1000 Monte Carlo experiments are run.

For the noise free case, we set the objective tensor $\mathcal{Y} \in \mathbb{R}^{5\times6\times6}$ and $R = 3, L_1 = L_2 = L_3 = 2$, i.e., a rank-$(2, 2, 1)$ block term decomposition. The factors $\mathbf{A}_r \in \mathbb{R}^{5\times2}$, $\mathbf{B}_r \in \mathbb{R}^{6\times2}$, $\mathbf{C}_r \in \mathbb{R}^{6\times1}$, $r = 1, 2, 3$ are generated random. The problem can be written as

$$\begin{cases} \mathbf{Y}_{(1)}^\top = (\mathbf{B} \bigodot \mathbf{C})\mathbf{A}^\top \\ \mathbf{Y}_{(2)}^\top = (\mathbf{C} \bigodot \mathbf{A})\mathbf{B}^\top \\ \mathbf{Y}_{(3)}^\top = (vec(\mathbf{B}_1 \bigodot \mathbf{A}_1), \cdots, vec(\mathbf{B}_R \bigodot \mathbf{A}_R))\mathbf{C}^\top \end{cases}$$

$\mathbf{A} = (\mathbf{A}_1, \cdots, \mathbf{A}_R)$, $\mathbf{B} = (\mathbf{B}_1, \cdots, \mathbf{B}_R)$, $\mathbf{C} = (\mathbf{c}_1, \cdots, \mathbf{c}_R)$.

Now, we illustrate how to get choose the initial points. For simplicity, we only show how to get $\mathbf{A}^0$ and the other initial points can be constructed in a similar manner. We first generate three random matrices $K_r \in \mathbb{R}^{5\times2}, r = 1, 2, 3$, whose entries are drawn from the normal distribution $\mathcal{N}(0, 1)$. Then, we apply the normalized Schmidt orthogonalization to the column of K_r and take the obtained matrices as $\mathbf{A}_r^0, r = 1, 2, 3$. Finally, we assemble the matrices $\mathbf{A}_r^0, r = 1, 2, 3$ to get the initial matrix $\mathbf{A}^0$. We stop the algorithm when the number of iterations exceeds 1000 or the stopping tolerance $\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 \leq 1 \times 10^{-5}$ is satisfied, where $\hat{\mathcal{Y}}$ stands for the obtained tensor after every iteration. If the number of iterations is 1000, and the stopping tolerance is not satisfied, we set that CPU time and iteration number infinite. We analyzed the performance data using the profiles of Dolan and Moré [13]. That is, we plot the fraction ρ of problems for which any given method is within a factor τ of the best result (iteration or CPU time). Here, the function $\rho(\tau)$ is defined by

$$\rho(\tau) = \frac{|\{a_i < 2^\tau \min(\mathbb{A})\}|}{|\mathbb{A}|},$$

where $|\mathbb{A}|$ stands the number of all the elements of the set $\mathbb{A}$, and a_i is an element of $\mathbb{A}$. Figure 5.9 is the performance profile of the four algorithms. From Figure 5.9 we can find that both from number of iterations and CPU time perspectives, ETRALS is the best one.

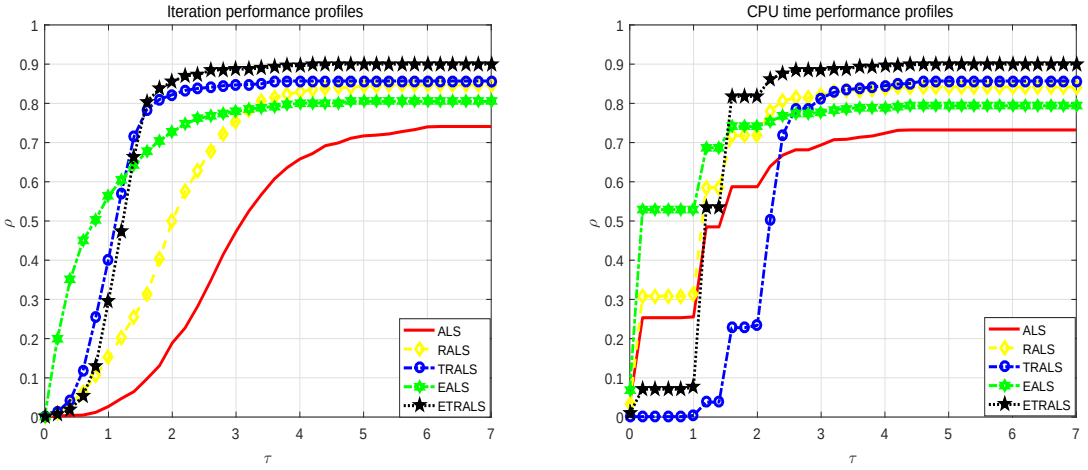

Fig. 5.9. The left one is number of iterations, and the right one is CPU time.

Then we consider a tensor with Gaussian noise, i.e., $\mathcal{Y} = \tilde{\mathcal{Y}} + \mathcal{N}$, where $\tilde{\mathcal{Y}}$ is the real-valued noise-free tensor of observations and $\mathcal{N}$ is the tensor of Gaussian noise. A suitable $\mathcal{N}$ is selected to make SNR=25. We stop the algorithm when the iteration number exceeds 1000 or the stopping tolerance $\|\hat{\mathcal{Y}}^{(n)} - \hat{\mathcal{Y}}^{(n-1)}\|_F \leq \epsilon$ is satisfied, where $\epsilon = 10^{-10}$. Figure 5.10 plots the four algorithms' performance profiles.

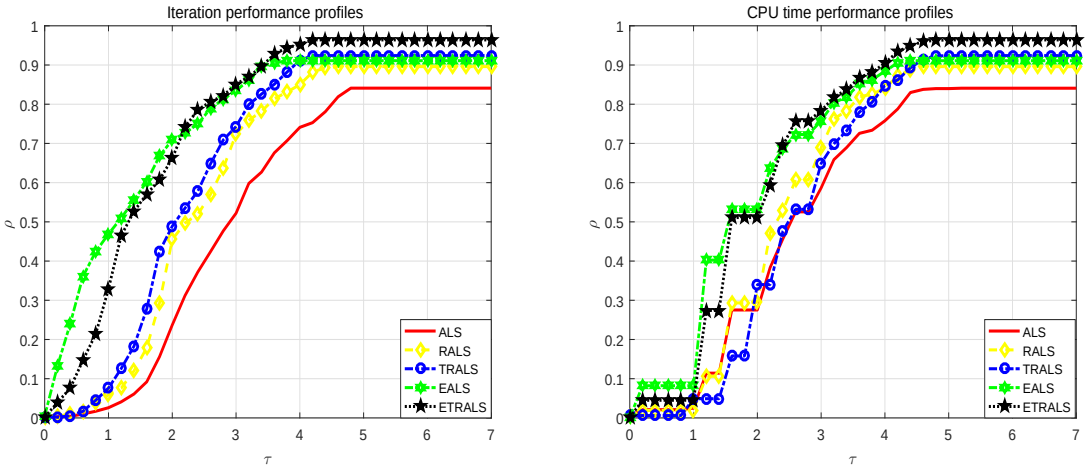

Fig. 5.10. The left one is number of iterations, and the right one is CPU time.

From Figure 5.10 we can observe that ETRALS is the best one, both for the number of iterations and CPU time, and its advantage is even more obvious than that for the noise free case.

6. Conclusions

In this paper, we proposed a new alternating least-squares type method, where we combine the classic optimization technique, the trust-region skill, to the well-known alternating least-

squares method for tensor decomposition problem, resulting in a trust-region-based alternating least-squares method (TRALS). To make the method more efficient, we further adopt the extrapolation scheme and obtain a new algorithm, ETRALS. Under mild conditions, we manage to prove the global convergence of the new method. We further apply the new algorithm to some tensor decomposition models arising from real applications, such as the tensor BCM decomposition from DS-CDMA, a problem from signal processing; CP in the fluorescence of amino acids, a problem from chemometrics; a rank-$(L_r, L_r, 1)$ decomposition, also from signal processing, and compare the new algorithm with the classic ALS and the recent modification RALS. In all these experiments, the new algorithm, ETRALS performs the best.

Acknowledgments. The authors would like to thank the reviewers and the editor for useful comments. This research is supported by a project funded by PAPD of Jiangsu Higher Education Institutions and the NSFC grants 11625105, 11371197, 11431002, 11571178.

References

[1] E. Acar, C. Aykut-Bingol, H. Bingol, R. Bro and B. Yener, Multiway analysis of epilepsy tensors, *Bioinformatics,* **23** (2007), i10-i18.

[2] E. Acar and B. Yener, Unsupervied multiway data analysis: A literature survey, *IEEE T. Knowl. Data En.,* **21** (2009), 6-20.

[3] A.L.F. De Almeida, G. Favier and J.C.M. Mota, Generalized PARAFAC model for multidimensional wireless communications with application to blind multiuser equalization, *Conference Record of the 39th Asilomar Conference on Signals, Systems and Computers,* **11** (2005), 1429-1433.

[4] A.L.F. De Almeida, G. Favier and J.C.M. Mota, Constrained tensor modeling approach to blind multiple-antenna CDMA schemes, *IEEE T. Signal Proces.,* **56** (2008), 2417-2428.

[5] G. Beylkin and M.J. Mohlenkamp, Algorithms for numerical analysis in high dimensions, *SIAM J. Sci. Comput.* **26** (2005), 2133-2159.

[6] R. Bro, Multi-way analysis in the food industry: Models, algorithms, and applications, Ph.D. dissertation, University of Amsterdam, Amsterdam, 1998.

[7] J.D. Carol and J.J. Chang, Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition, *Psychometrika*, **35** (1970), 283-319.

[8] Y. Chen, D. Han and L. Qi, New ALS methods with extrapolating search directions and optimal step size for complex-valued tensor decompositions, *IEEE T. Signal Proces.*, **59** (2011), 5888-5898.

[9] J. Coloigner, A. Karfoul, L. Albera and P. Comon, Line search and trust region strategies for canonical decomposition of semi-nonnegative semi-symmetric 3rd order tensors, *Linear Algebra Appl.*, **450** (2014), 334-374.

[10] P. Comon, X. Luciani and A.L.F. De Almeida, Tensor decompositions, alternating least squares and other tales, *J. Chemometr.*, **23** (2009), 393-405.

[11] A.R. Conn, N.I. Gould and P.L. Toint, Trust region methods, SIAM, 2000.

[12] D.A. Cox, J. Little and D. O'Shea, Using Algebraic Geometry, Springer Science & Business Media, 2006.

[13] E.D. Dolan and J.J. Moré, Benchmarking optimization software with performance profiles, *Math. Program.*, **91** (2002), 201-213.

[14] I. Domanov and L.D. Lathauwer, Canonical polyadic decomposition of third-order tensors: reduction to generalized eigenvalue decomposition, *SIAM J. Matrix Anal. A.*, **35** (2014), 636-660.

[15] J. Fan and Y. Yuan, On the quadratic convergence of the Levenberg-Marquardt method without nonsingularity assumption, *Computing*, **74** (2005), 23-39.

[16] J. Fan, The modified Levenberg-Marquardt method for nonlinear equations with cubic convergenc, *Math. Comput.*, **81** (2012), 447-466.

[17] J. Fan, Accelerating the modified Levenberg-Marquardt method for nonlinear equation, *Math. Comput.*, **83** (2014), 1173-1187.

[18] L. Grippo and M. Sciandrone, On the convergence of the block nonlinear Gauss-Seidel method under convex constraints, *Oper. Res. Lett.,* **26** (2000), 127-136.

[19] R.A. Harshman, Foundations of the PARAFAC procedure: model and conditions for an "explanatory" multi-code factor analysis, UCLA Working Papers, 1970.

[20] F.L. Hitchcock, The expression of a tensor or a polyadic as a sum of products, *J. Math. Phys.,* **6** (1927), 164-189.

[21] F.L. Hitchcock, Multiple invariants and generalized rank of a p-way matrix or tensor, *J. Math. Phys.*, **7** (1928), 39-79.

[22] W. Hackbush, B.N. Khoromskij and E.E. Tyrtyshnikov, Hierarchical Kronecker Tensor-Product Approximations, *J. Numer. Math.,* **13** (2005), 119-156.

[23] A.R. Hoy, C.G. Koay, S.R. Kecskemeti and A.L. Alexander, Optimization of a free water elimination two-compartment model for diffusion tensor imaging, *NeuroImage*, **103** (2014), 323-333.

[24] T.G. Kolda and B.W. Bader, Tensor decompositions and applications, *SIAM Rev.,* **51** (2009), 455-500.

[25] B. Khoromskij and V. Khoromskaia, Low rank tucker type tensor approximation to classical potentials, *Open Math.,* **5** (2007), 523-550.

[26] N. Li, S. Kindermann and C. Navasca, Some convergence results on the regularized alternating least-squares method for tensor decomposition, *Linear Algebra Appl.,* **2** (2013), 796-812.

[27] C. Navasca, L. De Lathauwer and S. Kindermann, Swamp reducing technique for tensor decomposition, *IEEE 16th European Signal Processing Conference*, (2008), 1-5.

[28] D. Nion and L. De Lathauwer, A block factor analysis based receiver for blind multi-user access in wireless communications, *IEEE International Conference on Acoustics, Speech and Signal Processing Proceedings,* **5** (2006), 825-828.

[29] D. Nion and L. De Lathauwer, Line search computation of the block factor model for blind multi-user access in wireless communications, *IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, (2006), 1-5.

[30] D. Nion and L. De Lathauwer, An enhanced line search scheme for complex-valued tensor decompositions. Application in DS-CDMA, *Signal Process.*, **88** (2008), 749-755.

[31] J. Nocedal and S. Wright, Numerical optimization, Springer Science & Business Media, 2006.

[32] L. Qi, W. Sun and Y. Wang, Numerical multilinear algebra and its applications, *Front. Math. China,* **2** (2004), 501-526.

[33] M. Rajih, P. Comon and R.A. Harshman, Enhanced line search: A novel method to accelerate PARAFAC, *SIAM J. Matrix Anal. A.,* **30** (2008), 1128-1147.

[34] M. Razaviyayn, M. Hong and Z. Luo, A unified convergence analysis of block successive minimization methods for nonsmooth optimization, *SIAM J. Optimiz.*, **23** (2013), 1126-1153.

[35] M. Signoretto, Q.T. Dinh, L. De Lathauwer and J.A.K. Suykens, Learning with tensors: a framework based on convex optimization and spectral regularization, *Mach. Learn.*, **94** (2014), 303-351.

[36] A. Smilde, R. Bro and P. Geladi, Multi-way analysis: applications in the chemical sciences, John Wiley & Sons, 2005.

[37] L. Sorber, M. van Barel and L. De Lathauwer, Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-$(L_r, L_r, 1)$ terms, and a new generalization, *SIAM J. Optimiz.,* **23** (2013), 695-720.

[38] L.R. Tucker, Some mathematical notes on three-mode factor analysis, *Psychometrika,* **31** (1966), 279-311.

[39] M. De Vos, A. Vergult, L. De Lathauwer, W. De Clercq, S. Van Huffel, P. Dupont, A. Palmini and W. Van Paesschen, Canonical decomposition of ictal scalp EEG reliably detects the seizure onset zone, *NeuroImage,* **37** (2007), 844-854.