Decomposition of large nonnegative tensors using memetic algorithms with application to environmental data analysis

Sylvain Maire, Xuan Vu, Caroline Chaux, Cyril Prissette, Nadège Thirion-Moreau

Decomposition of large nonnegative tensors using memetic algorithms with application to environmental data analysis

Sylvain Maire[a], Xuan Thanh Vu[*,a], Caroline Chaux[b], Cyril Prissette[a], Nadège Thirion-Moreau[**,a]

[a] *Université de Toulon, Aix Marseille Université, CNRS, LIS, Toulon, France, UMR 7020, F-83957 La Garde, France*
[b] *CNRS, Centrale Marseille, I2M, UMR 7373, F-13453 Marseille, France*

Abstract

In this article, we address the problem of the Canonical Polyadic decomposition (or Candecomp/Parafac Decomposition) of large N-way tensors under nonnegativity constraints. This problem is usually carried out using algebraic or iterative (alternating or all at once) deterministic optimization algorithms. Here, we focus on stochastic approaches and more precisely on memetic algorithms. Different variants of these algorithms are suggested and compared. A partial cost function and other optimization tools are introduced to reduce the complexity of the problem at hand. Different (either deterministic or stochastic) strategies concerning the local search step are also considered. This leads to new algorithms which are analysed thanks to computer simulations and compared with state of the art algorithms. When the tensor rank is unknown, we also propose a solution to estimate it. Finally, our approach is tested on a real experimental water monitoring application where the tensor rank is unknown.

Key words: Nonnegative Tensor Factorization (NTF); multi-linear algebra; Candecomp/Parafac (CP) decomposition; stochastic optimization; evolutionary algorithms; Big data tensors; missing data

1. Introduction

The tensor decompositions problem has emerged from various application fields where the data sets are often arranged in multi-way arrays or tensors. Multilinear analysis and tensor decompositions have received an increasing attention from different scientific communities (including statistics, numerical linear algebra, numerical analysis, (audio) signal processing, telecommunications, data mining, computer vision, linguistics, psychometrics, neuroscience, chemometrics, biomedical engineering, and so on), see for example [2][26][15][18] for an overview. The Canonical Polyadic Decomposition (a.k.a. CPD) is one of the most well-known decompositions of tensors. It consists of representing a tensor as a minimal sum of rank-one tensors hence the term "Canonical". It dates back to the work of Hitchcock [22] in 1927 and was later regarded as a

*This work has been carried out in the framework of the Labex Archimède (ANR-11-LABX-0033).
**Corresponding author.
Email addresses: maire@univ-tln.fr (Sylvain Maire), vuthanh.xuan.k4@gmail.com (Xuan Thanh Vu), caroline.chaux@univ-amu.fr (Caroline Chaux), prissette@univ-tln.fr (Cyril Prissette), thirion@univ-tln.fr (Nadège Thirion-Moreau)

generalization to higher-orders of the Singular Value Decomposition (SVD) of a matrix (2-way array).

Since the statement of the CPD problem, numerous numerical methods have been suggested to handle it, including i) direct methods (e.g. the GRAM-DTLD method [52][55]), ii) deterministic (alternating or all-at-once) optimization methods (notably the ALS or HALS method [15], the gradient-based methods [21], Newton and Quasi-Newton methods [62][48][49], among others) and more recently iii) stochastic methods [66][63]. While a large number of articles fall into the first two categories, this article is, however, among the few works which cover the latter. The algorithms suggested herein are based on a pure stochastic method, namely memetic algorithms [38] [39], which rely on a local search and the evolution of a population (or a family of tentative solutions). The efficiency of these algorithms to solve the CPD problem as well as their flexibility will be pointed out.

In one of the leading application fields of tensors, for instance 3D-fluorescence spectroscopy data analysis [59], the CPD enables to solve an inverse problem. In fact, the fluorescent chemical compounds (also known as fluorophores) dissolved in water samples can be separated and their characteristics can be recovered. In this application, the latent variables (the desired solutions of the CPD problem) are intrinsically non-negative since they stand for emission spectra, excitation spectra and concentrations of the unknown number of chemical constituents of the mixtures. The study of CPD under non-negativity constraints is also motivated by recent theoretical researches showing that the existence of a low rank approximation for CPD is ensured under such constraints, which may not be true in the general case [32]. Among existing algorithms addressing non-negative CPD, two main techniques have been developed. They are either applying a projection onto the feasible set [15], or using a change of variables by a squared function [48][49] or an exponential function [17]. The technique used in this paper is closer to the former but instead of a projection we propose a "selective" reflection.

Another important aspect is the continually increasing volume of data due to fast development of technological tools to acquire and to store information. This leads to both tensors of larger dimensions (sizes) and to higher-order tensors (Time-Resolved Fluorescence Spectroscopy [31] and the possibility to acquire 4D datasets). Therefore, it is both challenging and a real necessity to efficiently process and analyse large data sets in tolerable elapsed time (see [54] for an overview). Few strategies have been suggested to handle large-scale and (or) high-order tensors. To the best of our knowledge, very few one stage solutions have been suggested. They rely on an adaptative scheme [41]. Most investigated solutions share a 2-stage scheme in which the first stage aims at reducing the difficulties raised by Big Data, and the second stage consists of iterating an optimization algorithm. Regarding the first stage, we classify these methods into three groups. The first category involves compression-based algorithms using complete tensors [54][16]. The compression step can be repeated several times [54] or only once by using Tucker3 decomposition [16]. In the second group, sub-tensors are employed in the place of the whole tensor. The sub-tensors are extracted for example, in a random way while ensuring each unknown is involved at the same (expected) frequency, or by partitioning the original tensor into non-overlapping blocks [63][65][43]. The extraction can be done once over all iterations, or it can be repeated randomly at every iteration. Finally, in the third class of solution, the use of a subset of randomly chosen entries of the tensors is suggested. The idea is close to that of missing data. One of the main advantage is that there is no requirement for a "tensor-form" concerning the extracted data.

Our work falls in the last category. However rather different from methods in the first two classes, in which the core algorithms in the second stage are deterministic and quite classic (e.g. ALS, conjugate gradient, Levenberg-Marquardt, among others), memetic algorithms are suggested here. This double-stochastic scheme makes our method standing out from the rest of the literature. This article is a continuation of our previous works [66] but also an extension to the general Nth-order tensor case. The ability of the suggested algorithms to handle the nonnegative tensor decomposition is pointed out. Moreover, we generalize the loss function involved in the optimization problem to the pth-power of l_p-norm (with $p \geq 1$), in which the ordinary squared Euclidean function, often used in iterative methods for CPD is a particular case where p is equal to two. As compared to [66], the algorithm suggested here is fully optimized: pre-stocking is introduced in addition to the economic computation of the loss function; some strategies regarding the partial cost function are also studied. Especially, in the case of the squared Euclidean norm (corresponding to p equal to two), a closed form for the locally optimal step size can be derived which helps to accelerate the algorithm. Then, we introduce some hybrid scenarios by coupling these two strategies. Computer simulations are then performed in the 3D-fluorescence spectroscopy context. The efficiency and robustness of our algorithms are emphasized. Finally, an experimental water monitoring application is also considered.

In the balance of this article, the following notations will be used. Scalars are denoted in (lowercase or capital) italic letters, vectors in bold lowercase letters, matrices in bold capital letters. Tensors, cost functions and equations sets are in calligraphic uppercase letters. The article is organized as follows. After an introduction, the second section is dedicated to the statement of the non-negative CPD problem. The objective function is introduced as well as a partial cost function. In the third section, the general principal of memetic algorithms is recalled. Then, a special case of these genetic algorithms called Collaborative Evolution of Population (CEP) is described. In the fourth section, this method is adapted to the CPD problem. The principle of all its stages is presented and discussed. An algorithm is derived and different improvements are also suggested and studied (partial ℓ_p cost function, stochastic or locally optimal step size but also hybrid strategies combining both optimal and stochastic step size). In Section V, computer simulations are performed to illustrate the behaviour of our algorithms in different contexts considering synthetic 3D fluorescence spectroscopy like data. Experimental water monitoring data are then considered in section VI. Finally, a conclusion is drawn and perspectives are delineated.

2. Problem statement

2.1. CP decomposition of N-way tensors

We consider tensors $\overline{\mathcal{T}} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ of order N. Nevertheless, in most practical applications, third or fourth-order tensors are often considered. Such N-way tensors can be represented by a N-mode array in a chosen basis. The order N corresponds to the number of indices of the associated array (*ways* or *modes* [55]). Each entry of $\overline{\mathcal{T}}$ is then denoted by $\bar{t}_{i_1,\ldots,i_N}$. In this article, fluorescence spectroscopy applications are targeted [30]. It is the reason why we are interested in the problem of the decomposition of tensors into a sum of R rank-1 terms [60][59]. Such a tensor decomposition is known as the Polyadic Decomposition (PD) of a tensor. We focus on that particular tensor decomposition since in fluorescence spectroscopy, at low concentrations, the non-linear model predicted by the Beer-Lambert law can be linearized [30], involving

that the fluorescence intensity rather accurately follows a three-way (or trilinear) model [55][51]. We recall that a rank-one tensor of order N is defined as the outer product, denoted by $\otimes$, of N vectors $\bar{\mathbf{a}}^{(n)}$. The polyadic decomposition of $\overline{\mathcal{T}}$ thus writes

$$\overline{\mathcal{T}} = \sum_{r=1}^{R} \bar{\mathbf{a}}_r^{(1)} \otimes \bar{\mathbf{a}}_r^{(2)} \otimes \ldots \otimes \bar{\mathbf{a}}_r^{(N)} = [\![\bar{\mathbf{A}}^{(1)}, \bar{\mathbf{A}}^{(2)}, \ldots, \bar{\mathbf{A}}^{(N)}]\!], \tag{1}$$

where the N matrices $\bar{\mathbf{A}}^{(n)} = (\bar{a}_{i_n r}^{(n)}) = [\bar{\mathbf{a}}_1^{(n)}, \bar{\mathbf{a}}_2^{(n)}, \ldots, \bar{\mathbf{a}}_R^{(n)}] \in \mathbb{R}^{I_n \times R}$, with $n \in \{1, \ldots, N\}$, are the so-called *loading (or factor) matrices* or *low-rank latent factors* or more simply *factors*, whose R columns $\bar{\mathbf{a}}_r^{(n)}$, with $r \in \{1, \ldots, R\}$, are the *loading vectors* of the n-th loading matrix $\bar{\mathbf{A}}^{(n)}$. We also introduce the element-wise form of (1) that will be useful in the next sections, i.e. for all $(i_1, i_2, \ldots, i_N) \in \mathcal{I} = \{1, \ldots, I_1\} \times \{1, \ldots, I_2\} \times \ldots \times \{1, \ldots, I_N\}$,

$$\bar{t}_{i_1 i_2 \ldots i_N} = \sum_{r=1}^{R} \bar{a}_{i_1 r}^{(1)} \bar{a}_{i_2 r}^{(2)} \ldots \bar{a}_{i_N r}^{(N)}. \tag{2}$$

The minimum value of R such that (1) is valid is called the *tensor rank* [28][53] and the corresponding decomposition (1) the Canonical Polyadic Decomposition (CPD) or the Tensor Rank Decomposition. It was first introduced by Hitchcock [22]. In the following, given an observed tensor $\mathcal{T}$, our purpose is to find its best rank-one approximation (i.e. $\mathcal{T} \sim \overline{\mathcal{T}}$) and thus to estimate all the hidden variables *i.e.* all the elements $a_{i_n r}^{(n)}$ (for all $r = 1, \ldots R$, $i_n = 1, \ldots I_n$, $n = 1, \ldots N$) of the N loading matrices $\mathbf{A}^{(n)}$. We also emphasize that $\mathcal{T}$ can be a noisy version of $\overline{\mathcal{T}}$ (i.e. $\mathcal{T} \sim \overline{\mathcal{T}} + \mathcal{B}$ where $\mathcal{B}$ stands for the noise tensor). This low-rank approximation problem, is an optimization problem and more precisely a minimization one. The estimated solution will be denoted by $\widehat{\mathcal{T}}$.

2.2. A minimisation problem to estimate the latent variables

The entrywise error, for all $(i_1, \ldots, i_N) \in \mathcal{I}$ is denoted by

$$e_{i_1 i_2 \ldots i_N} = t_{i_1 \ldots i_N} - \sum_{r=1}^{R} a_{i_1 r}^{(1)} \ldots a_{i_N r}^{(N)}. \tag{3}$$

A way to address the aforementioned low-rank approximation problem, is to minimize a cost function $\mathcal{H}(\mathbf{x})$ that measures how the observed tensor fits the CP model:

$$\underset{\mathbf{x} \in \mathbb{R}^L}{\text{minimize}} \;\; \mathcal{H}(\mathbf{x}), \tag{4}$$

where $\mathbf{x}$ stands for the $L = (I_1 + \ldots + I_N) \times R$ unknowns written in the vector form

$$\mathbf{x} = \begin{pmatrix} \text{vec}\{\mathbf{A}^{(1)}\} \\ \vdots \\ \text{vec}\{\mathbf{A}^{(N)}\} \end{pmatrix} \in \mathbb{R}^L$$

where the operator vec$\{\cdot\}$ vertically stacks the columns of a matrix into a vector.
One possible choice for $\mathcal{H}(\mathbf{x})$ is the p^{th}-power l_p function defined as

$$\mathcal{F}^p(\mathbf{x}) = \sum_{(i_1, \ldots, i_N) \in \mathcal{I}} |t_{i_1 \ldots i_N} - \sum_{r=1}^{R} a_{i_1 r}^{(1)} \ldots a_{i_N r}^{(N)}|^p = \sum_{(i_1, \ldots, i_N) \in \mathcal{I}} |e_{i_1 i_2 \ldots i_N}|^p \tag{5}$$

where p is a real number such that $p \geq 1$. Most iterative methods designed for CPD problems rely on optimization schemes where the fidelity term involved in the objective function $\mathcal{H}(\mathbf{x})$ is a particular case of Eq. (5), for instance the squared Euclidean cost function obtained with $p = 2$

$$\mathcal{F}^2(\mathbf{x}) = \sum_{(i_1,\ldots,i_N)\in\mathcal{I}} e^2_{i_1 i_2 \ldots i_N}. \tag{6}$$

When performing the CP decomposition, the tensor rank R is assumed to be known and consequently has to be estimated first. Finally, the nonnegativity constraints impose that $a^{(n)}_{i_n r} \geq 0$ for all $n \in \{1,\ldots,N\}, i_n \in \{1,\ldots,I_n\}, r \in \{1,\ldots,R\}$.

2.3. Partial cost function built by random sampling

In practice, using the whole tensor $\mathcal{T}$ is not always possible or even convenient. Due to some reasons (among which are the lack, the alteration or the delation of some measurements), missing data may occur during the data acquisition process or after their pre-processing. Furthermore, when the dimensions I_n and especially the order N of the tensor become significantly large, handling the whole cost function becomes expensive. For such reasons, we suggest to use partial cost functions instead of the aforementioned cost function $\mathcal{F}^p$ in (5). We define an integer M corresponding to the M equations among the $I_1 \times I_2 \times \ldots \times I_N$ available ones that will be selected in (2). It enables us to define a partial cost function, denoted by $\mathcal{F}^p_M$, constituted of the M p^{th}-power residuals in (5)

$$\mathcal{F}^p_M(\mathbf{x}) = \sum_{(i_1,i_2,\ldots,i_N)\in\mathcal{I}_M} |e_{i_1 i_2 \ldots i_N}|^p, \tag{7}$$

where $\mathcal{I}_M \subset \mathcal{I}$ denotes the index set $(i_1,\ldots,i_N)$ associated to the M selected equations: $\mathcal{I}_M = \{(i_1,i_2,\ldots,i_N) \in \mathcal{I} \mid \text{equation } (i_1,i_2,\ldots,i_N) \text{ is chosen}\}$. In practice, the set $\mathcal{I}_M$ can be chosen randomly according to some distribution (e.g. uniform law, weighted random distribution). We also notice that the standard cost function $\mathcal{F}^p$ corresponds to a special case of $\mathcal{F}^p_M$ where $M = I_1 \times I_2 \times \ldots I_N$ (all available equations are selected). In practice, we often choose M such that its value is few times greater than the number of unknowns L and satisfies $0 < M \leq I_1 \times I_2 \times \ldots I_N$. Another significant advantage of such a technique is that it also enables to treat naturally the missing data problem without further developments and variants of the algorithms like those traditionally required by deterministic approaches [61][1][50].

3. Memetic algorithms

3.1. Introduction

Stochastic optimization algorithms enable to handle inherent system noise and highly nonlinear, high dimensional models or systems. Notable stochastic algorithms include stochastic approximation algorithms [29], stochastic gradient descent [7], evolutionary algorithms [5], simulated annealing [25], Genetic Algorithms (GAs) [23] and so on. A short overview can be found in [57] (for further details see for example [56]).
Among the existing stochastic algorithms, we are interested in a particular class, namely memetic algorithms, which are an extension of the traditional GAs [39]. The term "memetic" was first proposed by Moscato in the late 1980s [36]. Memetic algorithms,

like GAs, are based on the evolution of a population of candidate solutions which converges to a solution of an optimization problem. In memetic algorithms a local search step is added to avoid the convergence to a local extremum which may happen with GAs. For such a reason, memetic algorithms are also called hybrid genetic algorithms or hybrid evolutionary algorithms. They have found a wide number of applications notably in NP-hard problems [10] [14] [19], machine learning [3] [27] [37], robotics [20] [42] [47], electronics [4] [6] [24], engineering [46] [58] [67], image and speech processing and computer vision [11][35], to cite a few (for an overview see e.g. [39] and the reference therein).

3.2. General description of memetic algorithms

As a metaheuristic technique, memetic algorithms are iterative. Each iteration, also called a *generation*, contains three steps: a selection, a search and a replacement. The principle of memetic algorithms for the minimization problem $\min_{\mathbf{x}\in\Omega} f(\mathbf{x})$ where $\Omega \subset \mathbb{R}^L$, $f : \Omega \to \mathbb{R}$ can be summed up as follows

Principle of memetic algorithms:

- **Step 1.** At iteration $k = 1$, create a random population of $N \geq 2$ candidates $\{\mathbf{x}^1[1], \ldots, \mathbf{x}^N[1]\} \subset \Omega$.
  Compute $f(\mathbf{x}^1[1]), \ldots, f(\mathbf{x}^N[1])$ to sort the population.
- **Step 2.** At iteration $k > 1$,
  - *Selection stage*: generate a new population from the present generation (e.g. at random, or using a fraction of the best candidates or keeping some of the best candidates and some of the worst, etc.)
  - *Search stage*: use a random or deterministic local search to improve the population.
  - *Replacement*: create the new generation $\{\mathbf{x}^1[k], \ldots, \mathbf{x}^N[k]\}$ of the population.
- **Step 3.** If none of the stopping criteria are met then set $k \leftarrow k + 1$, return to **Step 2** else stop.

3.3. Special case of CEP algorithms

We suggest here a new memetic algorithm originally designed to tackle CPD problems [66]. The developed method is inspired by the idea of the Collaborative Evolution of Population (CEP) algorithm [40] which relies on a duel between two random candidates of the population and a cloning step. The selection step of this algorithm consists first of choosing uniformly at random a pair of two different candidates among the population. By comparing their corresponding cost function values, the candidate with smaller cost function value will be selected whereas the other one will be discarded. The local search stage consists in searching in a neighbourhood of the selected candidate to make a proposition for a new tentative solution. The general principle of the proposed method can be summarized as follows

Principle of the CEP algorithm:

- **Step 1.** At iteration $k = 1$, select a random population of $N \in \mathbb{N}$ elements $\{\mathbf{x}^1[1], \ldots, \mathbf{x}^N[1]\} \subset \Omega$. Compute $f(\mathbf{x}^1[1]), \ldots, f(\mathbf{x}^N[1])$.
- **Step 2.** At iteration $k > 1$,
  - *Selection stage*: choose uniformly at random a pair of different candidates in the population. Keep the candidate with smaller cost function value and

discard the other from the population.

- *Search stage*: search around in a neighbourhood of the selected candidate to find a new candidate.
- *Replacement*: add this new candidate to the population.

- **Step 3.** If none of stopping criteria are met then set $k \leftarrow k+1$, return to **Step 2** else stop.

The CEP algorithm and memetic algorithms in general do not require computation of derivatives (e.g. gradient, Hessian matrices) but only the computation of the cost function. This property is a great advantage of the method over deterministic methods (e.g. gradient-type or Newton-type methods). For this reason, other objective functions than the quadratic function (6), like the ℓ_1-norm or more generally ℓ_p-norms but also the Kullback-Leibler divergence [15] become quite simple to handle.

Finally, a few stages of the CEP algorithm are displayed in Fig. 1 on a very simple didactic example. In this example, we search for a solution in a 2D disk by minimizing the Euclidean distance to the solution (symbolized by a bigger red point). A population of four candidate solutions is used.

Figure 1: From left to right: illustration of the different steps of two iterations of the CEP method: selection search and replacement applied to a simple problem: find a point (in red) within a 2D disk (bounded by the blue circle). A population of 4 candidate solutions (points in black, magenta, blue, and green) is used.

4. Application of memetic algorithms to the non-negative CPD problem

As already explained in the second Section, the non-negative CPD problem is formulated as an optimization problem which relies on a well-chosen cost function. This cost function can involve either a quadratic norm (6) (as in most approaches dealing with tensor decompositions) or more generally ℓ_p functions (5). More precisely, in this article, a partial ℓ_p cost function of randomly chosen equations (see (7)) is suggested to reduce even more the complexity of the cost function computation but also to be able to address the missing data problem with exactly the same algorithm. We notice that the idea of a random sampling has already been used in [63][12] where tensor "fibers" or blocks of sub-tensors are selected randomly which can be regarded as a particular case of the partial cost function suggested herein. Now, we are going to explain and discuss all the elements specific to memetic algorithms.

4.1. Selection and local search

Assume that at generation k, two candidate solutions are selected, namely $\mathbf{x}^{N_1}$ and $\mathbf{x}^{N_2}$ ($N_1 \neq N_2 \in \{1,\ldots,N\}$) and that the cost function value using $\mathbf{x}^{N_1}$ is smaller than the one using $\mathbf{x}^{N_2}$. The candidate $\mathbf{x}^{N_2}$ is discarded and we proceed to a local search which means that we now seek for a new candidate solution $\tilde{\mathbf{x}}$ in a neighborhood of $\mathbf{x}^{N_1}$ in order to replace $\mathbf{x}^{N_2}$. From now on, to simplify the notations, we omit the population index in the selected candidate and simply denote $\mathbf{x}^{N_1}$ by $\mathbf{x} = (x_1, x_2, \ldots, x_L)^\top \in \mathbb{R}^L$. Our search strategy is very simple. Only one component $x_{\bar{l}}$ of $\mathbf{x}$, chosen uniformly at random in $\{1,\ldots,L\}$, is modified by simply

adding a quantity $\widehat{\mu}_{\bar{l}}$ to its value. If the value of $x_{\bar{l}} + \widehat{\mu}_{\bar{l}}$ is negative then we use its absolute value instead. This corresponds to a reflection. Letting $\boldsymbol{\mu} = [0, \ldots, 0, \mu_{\bar{l}}, 0, \ldots, 0]$, we have

$$\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\mu}, \tag{8}$$

or, at the $k+1$-th iteration

$$\tilde{\mathbf{x}}[k+1] = \mathbf{x}[k] + \boldsymbol{\mu}[k], \tag{9}$$

where $\tilde{\mathbf{x}}[k+1] = (\tilde{x}_1[k+1], \tilde{x}_2[k+1], \ldots, \tilde{x}_L[k+1])^\top$ and $\boldsymbol{\mu}[k] = [0, \ldots, 0, \mu_{\bar{l}}[k], 0, \ldots, 0]$, with $\forall l \in \{1, \ldots, L\}$:

$$\tilde{x}_l[k+1] = \begin{cases} x_{\bar{l}}[k] + \mu_{\bar{l}}[k] = |x_{\bar{l}}[k] + \widehat{\mu}_{\bar{l}}[k]| & \text{if } l = \bar{l}, \\ x_l[k] & \text{otherwise.} \end{cases} \tag{10}$$

The use of the absolute function $|\cdot|$ ensures that $\tilde{\mathbf{x}}[k+1]$ is non-negative. The modification of a single component instead of a set of components is motivated by the possibility of pre-stocking and of a low-cost computation for the cost function $\mathcal{F}_M^p$ (see the next subsection 4.2). The crucial question of the choice of $\widehat{\mu}_{\bar{l}}[k]$ is discussed in subsection 4.3. In the general context of non-negative Nth-order tensor decompositions, we propose stochastic local steps. Besides, in the particular case of a quadratic cost function ($p = 2$), a possible (locally) optimal step and some hybrid variants can be introduced (see subsection 4.5).

4.2. Cheap cost function computation

Our goal is now to compute the value of the cost function $\mathcal{F}_M^p$ at the new candidate solution $\tilde{\mathbf{x}}[k+1] = \mathbf{x}[k] + \boldsymbol{\mu}[k]$ accurately and efficiently, knowing that $\mathcal{F}_M^p(\mathbf{x}[k])$ has already been computed. To perform this task, a simple economic computation of $\mathcal{F}_M^p(\tilde{\mathbf{x}}[k+1])$ is proposed. We also show later in Subsection 4.4 that the computational expense can be even more reduced using pre-stocking techniques when the size of the population is equal to 2.

We denote by $\mathcal{I}_M^{\bar{l}} \subset \mathcal{I}_M$ the set of all the indices involving the variable $x_{\bar{l}}$ (see subsection 2.3 for the definition of $\mathcal{I}_M$) i.e.

$$\mathcal{I}_M^{\bar{l}} = \{(i_1, i_2, \ldots, i_N) \in \mathcal{I}_M \mid x_{\bar{l}} \text{ appears in the equation}\}. \tag{11}$$

By considering $I_0 = 0$, we can notice that if

$$1 + \sum_{i=0}^{n-1} RI_i \leq \bar{l} \leq \sum_{i=1}^{n} RI_i \tag{12}$$

then the n-th mode was selected in subsection 4.1. Moreover, it is also possible to detect which variable $a^{(n)}_{i_{\bar{l}}, r_{\bar{l}}}$ is in fact considered when the variable $x_{\bar{l}}$ is chosen:

$$r_{\bar{l}} = \begin{cases} \lfloor \frac{(\bar{l} - \sum_{i=0}^{n-1} I_i R)}{I_n} \rfloor + 1 & \text{if} \quad (\bar{l} - \sum_{i=0}^{n-1} I_i R) \bmod I_n \neq 0 \\ \lfloor \frac{(\bar{l} - \sum_{i=0}^{n-1} I_i R)}{I_n} \rfloor & \text{else} \end{cases} \tag{13}$$

$$i_{\bar{l}} = \begin{cases} (\bar{l} - \sum_{i=0}^{n-1} I_i R) \bmod I_n & \text{if} \quad (\bar{l} - \sum_{i=0}^{n-1} I_i R) \bmod I_n \neq 0 \\ I_n & \text{else} \end{cases} \tag{14}$$

where mod stands for the modulo and $\lfloor \cdot \rfloor$ is the floor operator. For all $(i_1, \ldots, i_N) \in \mathcal{I}_M^{\bar{l}}$, we have

$$\begin{aligned} e_{i_1 \ldots i_N}[k+1] = t_{i_1 \ldots i_N} - \sum_{r \neq r_{\bar{l}}} a^{(1)}_{i_1 r}[k] \ldots a^{(N)}_{i_N r}[k] - a^{(1)}_{i_1 r_{\bar{l}}}[k] \ldots \\ a^{(n-1)}_{i_{\bar{l}-1} r_{\bar{l}}}[k] \left(a^{(n)}_{i_{\bar{l}} r_{\bar{l}}}[k] + \mu_{\bar{l}}[k] \right) a^{(n+1)}_{i_{\bar{l}+1} r_{\bar{l}}}[k] \ldots a^{(N)}_{i_N r_{\bar{l}}}[k], \end{aligned} \tag{15}$$

Hence, we have

$$e_{i_1\dots i_N}[k+1] = e_{i_1\dots i_N}[k] - \mu_{\bar{l}}[k] \underbrace{a^{(1)}_{i_1 r_{\bar{l}}}[k] \dots a^{(n-1)}_{i_{\bar{l}-1} r_{\bar{l}}}[k] a^{(n+1)}_{i_{\bar{l}+1} r_{\bar{l}}}[k] \dots a^{(N)}_{i_N r_{\bar{l}}}[k]}_{z^{(-x_{\bar{l}})}_{i_1\dots i_N, r_{\bar{l}}}[k]} \tag{16}$$

where

$$z^{(-x_{\bar{l}})}_{i_1\dots i_N, r_{\bar{l}}}[k] = a^{(1)}_{i_1 r}[k] \dots a^{(N)}_{i_N r}[k] / a^{(n)}_{i_{\bar{l}} r_{\bar{l}}}[k] \quad \text{if} \quad a^{(n)}_{i_{\bar{l}} r_{\bar{l}}}[k] \neq 0$$

Letting

$$\mathcal{G}_M(\tilde{\mathbf{x}}[k+1], \mathbf{x}[k]) = \mathcal{F}^p_M(\tilde{\mathbf{x}}[k+1]) - \mathcal{F}^p_M(\mathbf{x}[k]), \tag{17}$$

we finally have

$$\mathcal{G}_M(\tilde{\mathbf{x}}[k+1], \mathbf{x}[k]) = \sum_{(i_1,\dots,i_N)\in\mathcal{I}^{\bar{l}}_M} \left[|e_{i_1\dots i_N}[k] - \mu_{\bar{l}}[k] \cdot z^{(-x_{\bar{l}})}_{i_1\dots i_N, r_{\bar{l}}}[k]|^p - |e_{i_1\dots i_N}[k]|^p \right]. \tag{18}$$

Since $\mathrm{card}(\mathcal{I}^{\bar{l}}_M)$ is often by far smaller than $\mathrm{card}(\mathcal{I}_M) = M$, the calculation of $\mathcal{G}_M(\tilde{\mathbf{x}}[k+1], \mathbf{x}[k])$ is rather economic. This leads to a cheap computation of $\mathcal{F}^p_M(\tilde{\mathbf{x}}[k+1])$ since $\mathcal{F}^p_M(\tilde{\mathbf{x}}[k+1]) = \mathcal{F}^p_M(\mathbf{x}[k]) + \mathcal{G}_M(\tilde{\mathbf{x}}[k+1], \mathbf{x}[k])$.

4.3. Stochastic step

We now have to address the important question of the choice of the step size $\mu_{\bar{l}}[k]$. In iterative optimization algorithms, the choice of the step size is often crucial. A too large step size may lead to a divergence while a too small step size may lead to a slow convergence rate. For standard stochastic algorithms like the stochastic gradient with decaying step size, it is proven under convexity assumptions that if the rate of decay is chosen a priori and verifies $\sum_{k=1}^{\infty} \mu[k] = \infty$ and $\sum_{k=1}^{\infty} \mu^2[k] < \infty$ the convergence to a local minimum is granted [33]. A frequently used step size is $\mu[k] = 1/k$.

In our case, we suggest the use of a stochastic step size $\mu_{\bar{l}}[k]$ drawn from the uniform law $\mathcal{U}(-b[k], b[k])$[1]. The parameter $b[k]$ does not depend directly on the value of $\mathbf{x}[k]$ but is adapted to $\mathcal{F}^p_M(\mathbf{x}[k])$. More precisely the parameter $\boldsymbol{\mu}[k]$ decreases when the cost function $\mathcal{F}^p_M(\mathbf{x}[k])$ decreases.

We now explain the heuristics used to build this step size. We target a step $\boldsymbol{\mu}[k]$ such that the corresponding residuals in (3) at $\tilde{\mathbf{x}}[k+1] = \mathbf{x}[k] + \boldsymbol{\mu}[k]$ are close to 0 which involves that for all $(i_1, \dots, i_N) \in \mathcal{I}^{\bar{l}}_M$, $e_{i_1\dots i_N}[k+1] \simeq 0$. Thus, Eq. (15) (or Eq. (16)) can be rewritten as

$$e_{i_1\dots i_N}[k] = t_{i_1\dots i_N} - \sum_{r=1}^{R} a^{(1)}_{i_1 r}[k] \dots a^{(N)}_{i_N r}[k] \simeq \mu_{\bar{l}}[k] z^{(-x_{\bar{l}})}_{i_1\dots i_N, r_{\bar{l}}}[k]. \tag{19}$$

The estimation of our step size $\mu_{\bar{l}}[k]$ therefore rely on two heuristics.

1. First, the quantity $\sqrt[p]{\frac{\mathcal{F}^p_M(\mathbf{x}[k])}{M}}$ indicates the mean error that we have on each term $|t_{i_1\dots i_N} - \sum_{r=1}^{R} a^{(1)}_{i_1 r}[k] \dots a^{(N)}_{i_N r}[k]| = |e_{i_1\dots i_N}[k]|$ belonging to $\mathcal{F}^p_M(\mathbf{x}[k])$.

[1] Other distributions could be used, however we choose the uniform distribution for its simplicity

2. The second heuristic is based on an approximation of
$z^{(-x_{\overline{l}})}_{i_1\dots i_N,r_{\overline{l}}}[k] = a^{(1)}_{i_1 r_{\overline{l}}}[k]\dots a^{(n-1)}_{i_{\overline{l}-1},r_{\overline{l}}}[k]a^{(n+1)}_{i_{\overline{l}+1}r_{\overline{l}}}[k]\dots a^{(N)}_{i_N r_{\overline{l}}}[k]$.
If we assume that all the components of $\mathbf{x}[k]$ are equal, i.e. $x_1[k] = \dots = x_L[k] = \tau[k]$, we have

$$\sum_{(i_1,\dots,i_N)\in\mathcal{I}_M} t_{i_1\dots i_N} = \sum_{(i_1,\dots,i_N)\in\mathcal{I}_M}\sum_{r=1}^{R} a^{(1)}_{i_1 r}[k]\dots a^{(N)}_{i_N r}[k] = MR\tau^N[k]. \quad (20)$$

leading to

$$\tau[k] = \sqrt[N]{\frac{\sum_{(i_1,\dots,i_N)\in\mathcal{I}_M} t_{i_1\dots i_N}}{MR}} = \tau \quad (21)$$

This heuristic τ, is also useful for the initialization of the algorithm. Under this coarse approximation of equal values components, $z^{(-n)}_{i_1\dots i_N,r}[k]$ can be approximated by τ^{N-1}.

And finally, thanks to the two heuristics above, the stochastic step size denoted by $\mu^{sto}[k]$, can be drawn from the following uniform distribution

$$\mu^{sto}[k] \sim \mathcal{U}\left(-b[k], b[k]\right),$$
$$\text{where} \quad b[k] = \sqrt[p]{\frac{\mathcal{F}^p_M(\mathbf{x}[k])}{M}} \Big/ \tau^{N-1}. \quad (22)$$

4.4. Choice of the population size

Genetic and memetic algorithms usually involve a large population in order to ensure the convergence to a global extremum in presence of many local extrema. However, if the convergence is still ensured with a small population, this convergence is more likely to be faster. So we will now test our algorithm with different sizes of the population starting from a population of $N = 2$ candidates until a population of $N = 50$ candidates. Numerical experiments are performed on a non-negative $100 \times 100 \times 100$ tensor generated from randomly chosen loading matrices drawn from a uniform law $\mathcal{U}(0,1)$ with $N \in \{2, 10, 20, 50\}$. To be able to compare the results, an error index is needed. One common error measure (usually used in CPD problems), is the Relative Reconstruction Error (RRE) which writes

$$\text{RRE} = \frac{\|\widehat{\mathcal{T}} - \mathcal{T}\|_F^2}{\|\mathcal{T}\|_F^2} \qquad \text{RRE}_{\text{dB}} = 10\log_{10}(\text{RRE}), \quad (23)$$

where $\widehat{\mathcal{T}}$ stands for the reconstructed tensor. The resulting RRE and computation time are provided in Table 1. The obtained results are averaged over 10 runs.

	$N=2$	$N=10$	$N=20$	$N=50$
Mean CPU time (s)	**1014**	2283	5187	16189
Mean RRE (dB)	-70.03	-70.05	-70.01	-70.01

Table 1: Performances in terms of RRE and computation time versus the size of the population.

We can observe that although all our numerical tests reach the same performance (RRE $\simeq -70$ dB), the use of a population of size $N = 2$ candidates reduces by a factor 2 (resp. 5 or 16) the computer running time compared to that needed with a population of $N = 10$ (resp. $N = 20, 50$) candidates. When solving the CPD problem, the use of two candidates happens to be optimal in the convergence speed of the proposed algorithm. This property seems to be quite general based on the other cases we have tested. From now on, we will focus on the version of the algorithm involving only two candidates.

Another advantage of using a candidate population of size two is that it becomes easy to reduce the computational complexity by pre-stocking previously computed terms. Indeed, we suggest to store the values of residual errors $e_{i_1 \dots i_N}$ and to update, at each iteration, only the $\mathrm{card}(\mathcal{I}_M^{\bar{l}})$ modified ones (see 4.2). We recall that the residual error $e_{i_1 \dots i_N}[k+1]$ is given by (16), *i.e.* for any $(i_1, \dots, i_N) \in \mathcal{I}_M^{\bar{l}}$, $e_{i_1 \dots i_N}[k+1] = e_{i_1 \dots i_N}[k] - \mu_{\bar{l}}[k] \cdot z^{(-x_{\bar{l}})}_{i_1 \dots i_N, r_{\bar{l}}}[k]$.

4.5. New versions of the algorithm in the quadratic case

In the specific case of quadratic cost functions, new versions of the algorithm can be derived. They rely on a choice of the step size that is different from the preceding stochastic step.

i) Locally optimal step

We propose herein a new way to compute the step $\boldsymbol{\mu}[k] = (0, \dots, 0, \mu_{\bar{l}}[k], 0, \dots, 0)$. In the particular case of a quadratic cost function, $\mathcal{F}_M^2(\mathbf{x}[k] + \boldsymbol{\mu}[k])$ can be expressed as

$$\begin{aligned}\mathcal{F}_M^2(\mathbf{x}[k] + \boldsymbol{\mu}[k]) = & \sum_{(i_1,\dots,i_N) \in \mathcal{I}_M^{\bar{l}}} (e_{i_1 \dots i_N}[k] - \mu_{\bar{l}}[k] \cdot z^{(-x_{\bar{l}})}_{i_1 \dots i_N, r_{\bar{l}}}[k])^2 + \sum_{(i_1,\dots,i_N) \in \mathcal{I}_M \setminus \mathcal{I}_M^{\bar{l}}} e^2_{i_1 \dots i_N}[k] \\ = & \mu_{\bar{l}}^2[k]\alpha_0[k] - 2\mu_{\bar{l}}[k]\alpha_1[k] + \alpha_2[k],\end{aligned}$$

where

$$\begin{aligned}\alpha_0[k] &= \sum_{(i_1,\dots,i_N) \in \mathcal{I}_M^{\bar{l}}} (z^{(-x_{\bar{l}})}_{i_1 \dots i_N, r_{\bar{l}}}[k])^2, \\ \alpha_1[k] &= \sum_{(i_1,\dots,i_N) \in \mathcal{I}_M^{\bar{l}}} e_{i_1 \dots i_N}[k] \cdot z^{(-x_{\bar{l}})}_{i_1 \dots i_N, r_{\bar{l}}}[k], \\ \alpha_2[k] &= \sum_{(i_1,\dots,i_N) \in \mathcal{I}_M} e^2_{i_1 \dots i_N}[k].\end{aligned}$$

$$\frac{\partial \mathcal{F}_M^2(\mathbf{x}[k] + \boldsymbol{\mu}[k])}{\partial \mu_{\bar{l}}} = 2\mu_{\bar{l}}[k]\alpha_0[k] - 2\alpha_1[k] = 0 \Rightarrow \mu_{\bar{l}}^{opt}[k] = \frac{\alpha_1[k]}{\alpha_0[k]},$$

This optimization stage is similar to one step of the ALS method except that, here, we compute the solution of a least square problem involving one single variable selected at random instead of blocks of variables selected cyclically.

ii) Hybrid strategies

Stochastic optimization algorithms usually exhibit a slow convergence rate when compared to deterministic optimization algorithms. However, they are, in general, more robust since more likely to find a global solution to optimization problems than deterministic algorithms which may converge to a local minimum. Furthermore, when the initialization is close enough to the global minimum, deterministic methods converge really fast. Starting from this observation, we propose three possible combinations of the stochastic search strategy with the locally optimal search strategy in order to better exploit the advantages of each of these two strategies:

- **(H1)** Randomly choose between s_μ consecutive stochastic steps μ^{sto} and one locally optimal step μ^{opt} ($s_\mu \in \mathbb{N}^*$),
- **(H2)** Initially use μ^{sto} then when the estimate $\mathbf{x}[k]$ is close enough to the solution (e.g. in the sense that the RRE $\leq \epsilon_{RRE}$) then choose μ^{opt} until convergence,
- **(H3)** First, use $K \in \mathbb{N}^*$ iterations with μ^{sto}, then alternate between K iterations using μ^{opt} or μ^{sto} if the relative diminishing rate of the quadratic criterion $\frac{\|\mathcal{F}_M^2[\cdot+K] - \mathcal{F}_M^2[\cdot]\|}{\mathcal{F}_M^2[\cdot]}$ is smaller than $\epsilon_K \in]0, 1[$ (this alternative technique avoids a too slow convergence rate).

4.6. Resulting algorithm

The proposed algorithm is summarized in Algorithm 1.

Algorithm 1 NTF-STO

Let $\mathbf{x}^1[1], \mathbf{x}^2[1], \ldots, \mathbf{x}^N[1] \in \mathbb{R}_+^L$ and $(N, R, M, K) \in \mathbb{N}^4, N \geq 2$. {Initialization}
Select $\mathcal{F}_M^p$ at random. {Choice of partial cost function}
Calculate $\mathcal{F}_M^p(\mathbf{x}^1[1]), \ldots, \mathcal{F}_M^p(\mathbf{x}^N[1])$.
for $k = 1, 2, ..., k_{\max}$ **do** {k-th iteration}
  if $p = 2$ **then** {Storing Stage}
    store the residual values $e_{i_1 \ldots i_N}[k]$, Eq. (3);
  end if
  Select randomly $(N_1, N_2) \subset \{1, \ldots, N\}$, $N_1 \neq N_2$.
  if $\mathcal{F}_M^p(\mathbf{x}^{N_1}[k]) \leq \mathcal{F}_M^p(\mathbf{x}^{N_2}[k])$ **then** {Selection stage}
    $\mathbf{x}[k] = \mathbf{x}^{N_1}[k]$.
  else
    $\mathbf{x}[k] = \mathbf{x}^{N_2}[k]$.
  end if
  Select a component $x_{\bar{l}}$ of $\mathbf{x}$ randomly. {Search stage}
  Compute $\widehat{\mu}_{\bar{l}}[k]$ based on $\mu^{sto}[k]$, Eq. (22) (or $\mu^{opt}[k]$, Eq. (24) or hybrid strategies (Section 4.5) if $p = 2$). {Step size calculation}
  Update the population: $\mathbf{x}^{N_1}[k+1] = \mathbf{x}[k], \mathbf{x}^{N_2}[k+1] = \tilde{\mathbf{x}}[k+1] = \mathbf{x}[k] + \mu[k]$, Eq. (10), other candidates are unchanged. {Replacement stage}
  Calculate $\mathcal{F}_M^p(\mathbf{x}^{N_2}[k+1])$ using Eq. (18).{Cost function computation}
end for

5. Computer simulations on synthetic data

We now study the behavior of the proposed memetic algorithm and of its variants on numerically generated data. For simplicity, we will denote by:

- **(S)**, the version of the memetic algorithm based on stochastic steps only,
- **(O)**, the version of the memetic algorithm based on locally optimal steps only,
- **(H1), (H2), (H3)**, the three hybrid versions based on an alternation between stochastic and locally optimal steps and on the following choice of parameters:
  - $s_\mu = 10$ in **(H1)**
  - $\epsilon_{RRE} = 0.01$ in **(H2)**
  - $\epsilon_K = 0.01$ in **(H3)** with $K = 4 \times 10^4$.

The same initialization or set of initializations is used for all the considered algorithms. For all simulations, three stopping conditions are considered:

1. maximum number of iterations $k_{\max} = 6.10^8$ is reached,
2. relative reconstruction error RRE is less than 10^{-7},
3. relative diminishing rate of $\mathcal{F}_M^p$ reads $\frac{\|\mathcal{F}_M^p[\cdot+l] - \mathcal{F}_M^p[\cdot]\|}{\mathcal{F}_M^p[\cdot]} < 10^{-7}$ with $l = 4 \times 10^4$.

We will now present the results obtained with the different versions of the suggested algorithm in different contexts and we will also perform a comparison with other well-established methods of the literature (e.g. Bro's N-way [8], fast HALS [45] algorithms).

5.1. *Description of the data*

The main targeted application is chemometry and environmental data analysis thanks to 3D fluorescence spectroscopy data sets. In most simulations (except in subsection 5.7 where the studied tensors are randomly generated), the considered tensors are simulated synthetic fluorescence spectroscopy data. The simulated (non-negative) third-order tensor is composed of 3 loading matrices $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}$, which respective sizes are $I_1 = 100$, $I_2 = 47, I_3 = 100$ rows and $\overline{R} = 5$ columns. The dimensions of the resulting tensor are thus $100 \times 47 \times 100$. The columns of matrices $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}$ stand for the emission and the excitation spectra of the fluorescent chemical components. Spectra are modeled using shifted generalized Gaussian density functions (see [66][65]). Each fluorescent chemical component is characterized by a so-called Fluorescence Excitation-Emission Matrix (FEEM), resulting from the outer product of two vectors, the corresponding emission and excitation spectrum. The concentration $\mathbf{A}^{(3)}$ of all components is drawn from a uniform distribution in $[0, 10]$. The considered FEEM and the (normalized) spectra and concentrations are displayed in Fig. 2 and Fig. **??** respectively.

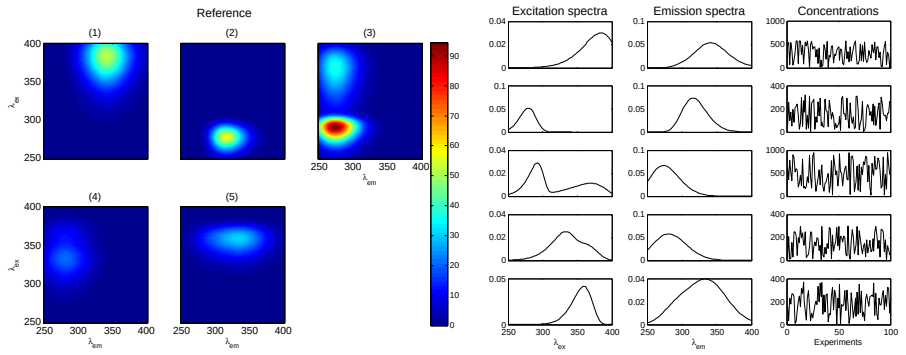

Figure 2: Ground truth FEEM (left). Ground truth excitation spectra, emission spectra and concentrations for each of the 5 components (right).

5.2. *Error indexes*

When we tackle the problems of overfactoring (*i.e.* $R > \overline{R}$) and want to assess their influence on the CP Decomposition algorithms, the RRE index is no more sufficient That is why we need to introduce two other error indexes that are used to evaluate more accurately the quality of the estimation. The first one, denoted by $\mathbf{E}_1$, measures the relative error of estimation but discarding the over-factoring part. The second one, denoted by $\mathbf{E}_2$ measures the error induced only by the over-factoring part. In this case, the inherent indetermination of the CPD problem (*i.e.* permutation and scaling ambiguities) also has to be taken into account: if we apply to a given solution a permutation of their loading factors or an appropriate scaling of their loading matrices, the obtained result is also a solution and defines the same decomposition. Permutation and scaling-independent measurements of the errors are therefore necessary.
For N-way tensors, the loading matrices $\mathbf{A}^{(n)}$, for all $n \in \{1, \ldots, N\}$, are first normalized in such a way that each column of $\mathbf{A}^{(n)}$ for $n \in \{1, ..., N-1\}$ is normalized in ℓ_1 and each column of $\mathbf{A}^{(N)}$ carries the weight. The estimated normalized solutions of the CPD algorithm are denoted by $\widehat{\mathbf{A}}^{(n)}$. Then, the R column vectors of $\widehat{\mathbf{A}}^{(n)}$ are permuted such that their Euclidean distance to $\bar{\mathbf{A}}^{(n)}$ is minimized. The permuted normalized estimates are denoted by $\widehat{\mathbf{A}}_{\sigma}^{(n)}$ while $\widehat{\mathbf{A}}_{\sigma}^{(n)}(1 : \overline{R})$ means that only its $\overline{R}$ first columns are considered and σ is the considered

permutation. Thus, the estimation error $\mathbf{E}_1$ is defined by

$$\mathbf{E}_1(\sigma) = \frac{\sum_{n=1}^{N} \|\widehat{\mathbf{A}}_{\sigma}^{(n)}(1:\overline{R}) - \mathbf{A}^{(n)}\|_1}{\sum_{n=1}^{N} \|\mathbf{A}^{(n)}\|_1}$$
$$\Rightarrow \begin{cases} \mathbf{E}_1 = \min_{\sigma} \mathbf{E}_1(\sigma) \quad \text{or} \quad \mathbf{E}_{1\text{dB}} = 10\log_{10}(\mathbf{E}_1) \\ \sigma_{opt} = \arg\min_{\sigma} \mathbf{E}_1(\sigma) \end{cases} \tag{24}$$

The error $\mathbf{E}_2$ concerning the over-factoring part, *i.e.* the remaining components $\overline{R}+1, \ldots, R$, is thus computed as follows

$$\mathbf{E}_2 = \| \sum_{r=\overline{R}+1}^{R} \mathbf{a}_{\sigma^{opt},r}^{(1)} \circ \cdots \circ \mathbf{a}_{\sigma^{opt},r}^{(N)} \|_1 \quad \text{or} \quad \mathbf{E}_{2\text{dB}} = 10\log_{10}(\mathbf{E}_2). \tag{25}$$

5.3. Equation selection in the cost function (noiseless case)

Our first objective is to study the behaviour of the proposed algorithms with respect to the number of random equations M involved in $\mathcal{F}_M^2$ (see Eq. 7). The obtained results are presented in Fig. 3. The tested values of M are $\{2L, 4L, 6L, 10L, I_1I_2I_3\}$ where $L = (I_1 + I_2 + I_3)R$ which, in terms of percentage m of involved constituent terms in $\mathcal{F}_M^2$ ($m = \frac{100\times M}{I_1I_2I_3}$), corresponds to $m = \{0.5\%, 1\%, 1.5\%, 2.6\%, 100\%\}$ respectively. For each value of M, we use 10 randomly chosen partial cost functions $\mathcal{F}_M^p$ while the algorithms are all started from the same random initialization. In Fig. 3, we observe that the suggested algorithms converge whatever

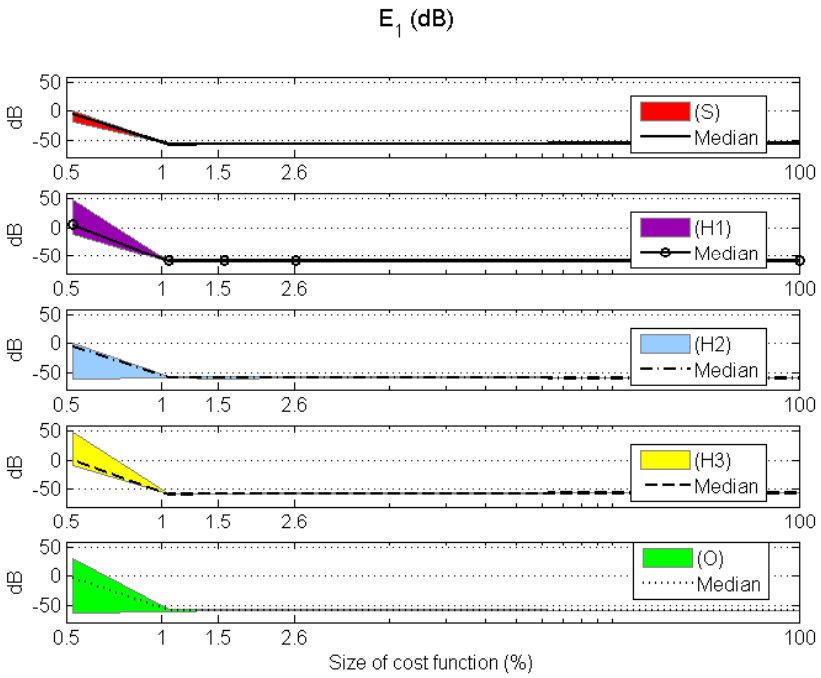

Figure 3: Performance of the different variants of the algorithm with respect to the number M of equations involved in the partial cost function $\mathcal{F}_M^2$ ($M = 2, 4, 6, 10$ times the number of unknowns, and finally the total cost function).

the considered configuration as soon as $m \geq 1\%$. In other words, only 1% of the available information seems to be sufficient for convergence in these tests. The use of a larger value of

M ensures the convergence but is time-consuming, while a too small value of M may lead to a divergence. These conclusions remain true for all the proposed variants of the algorithm.
Our next objective is to study the robustness of the proposed algorithms with respect to the choices of the set of M random equations involved in $\mathcal{F}_M^2$ (see Eq. 7). To that aim, the size M of $\mathcal{F}_M^2$ is set to $M = 4L$, since the convergence is ensured for all the versions of the algorithm in such a case. For each algorithm, the simulations are performed over 10 different sets of M randomly chosen constituent terms involved in the partial cost functions $\mathcal{F}_M^2$. A unique initialization drawn from the uniform distribution $\mathcal{U}(0, 2\tau)$ is used for all the experiments where τ is defined in Eq. 21.

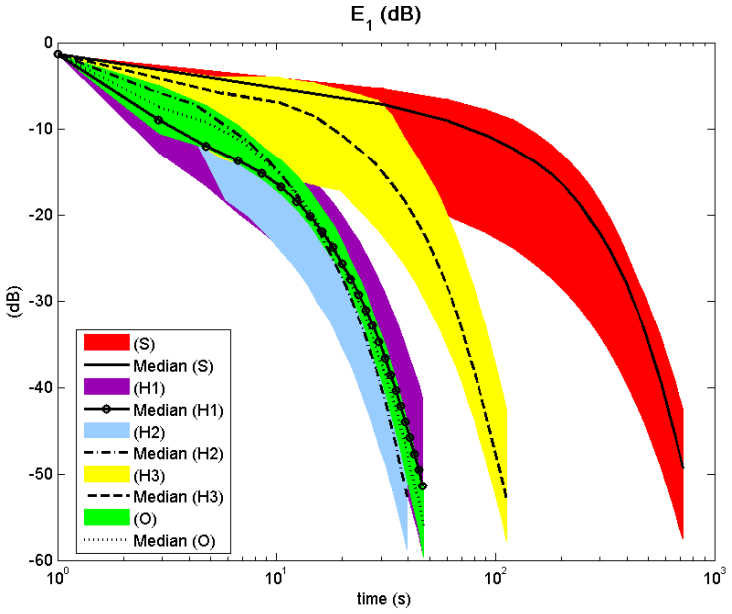

Figure 4: Behavior of the proposed memetic algorithms using ten different choices of $M = 4L$ equations for the calculation of the partial cost function $\mathcal{F}_M^2$.

The Fig. 4 illustrates the evolution of the error indexes $\mathbf{E}_1$ with respect to each algorithms over time. The range in which they vary and the corresponding median value are traced. When $M = 4L$, we can observe that all algorithms are able to converge and to attain a good level of performance (-60dB in this case). Among the different variants of the algorithm, the version with locally optimal steps **(O)** and the hybrid versions **(H1, H2)** converge the fastest. Considering the median value, **(H2)** happens to be the fastest of the three algorithms. Moreover, their computation times are relatively close to each other and by far faster than the stochastic step version **(S)** which remains the slowest. The hybrid versions **(H3)** falls between **(H1, H2, O)** and **(S)**.

5.4. Robustness with respect to initialization

We now test the robustness of all the considered algorithms versus the initialization. To study that problem, we draw ten initial values of the loading matrices from the uniform distribution $\mathcal{U}(0, 2\tau)$ (and respectively from $\mathcal{U}(0, \frac{\tau}{50})$, $\mathcal{U}(0, \frac{\tau}{5})$, $\mathcal{U}(0, 20\tau)$, $\mathcal{U}(0, 200\tau)$), thus involving a total of 50 different initializations. The same partial cost function $\mathcal{F}_M^2$ of size $M = 4L$, is used for all the experiments. Then the resulting errors $\mathbf{E}_1$ are arranged in ascending order. They are shown in Fig. 5.

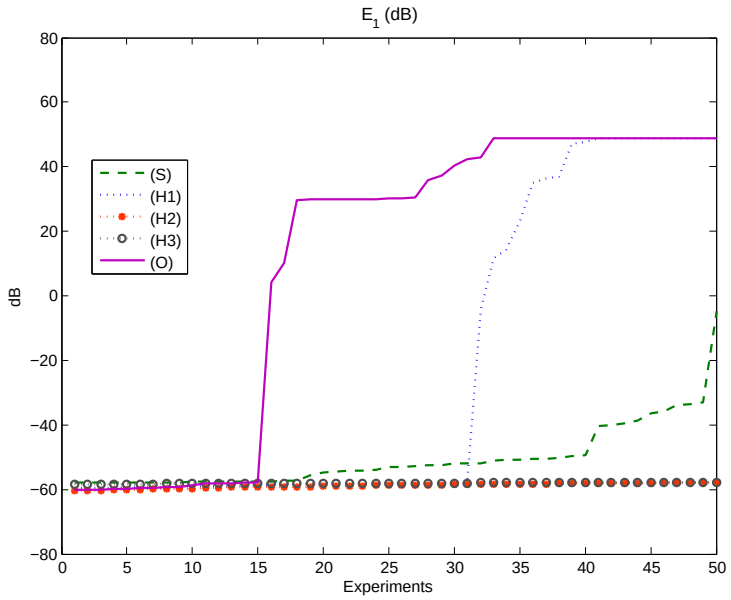

Figure 5: Estimation error $\mathbf{E}_1$ obtained with the memetic algorithms using 50 different initial values drawn from $\mathcal{U}(0, \frac{\tau}{50})$, $\mathcal{U}(0, \frac{\tau}{5})$, $\mathcal{U}(0, 2\tau)$, $\mathcal{U}(0, 20\tau)$, $\mathcal{U}(0, 200\tau)$. Noise-free case and the cost function size $M = 4L$.

We can observe that the versions **(S, H2, H3)** converge in all experiments (but sometimes **(S)** is too slow too reach the same level of performance as **(H2, H3)**) while the versions **(O, H1)** only converge in respectively 70% and 38% of the cases. Indeed, this phenomenon happens when the initialization is either relatively too close to 0 (drawn from $\mathcal{U}(0, \frac{\tau}{50})$) or too large (drawn from $\mathcal{U}(0, 200\tau)$).

5.5. Robustness with respect to noise

We suppose now that the observed tensor $\mathcal{T}$ is corrupted by noise. We consider an additive uniform noise $\mathcal{U}(0, b), (b > 0)$ and different SNR values: 10 dB, 50 dB, 100 dB, 150 dB and 200 dB respectively. The Fig. 6 illustrates the performance of the five variants of the suggested algorithm using 10 different random initializations drawn from $\mathcal{U}(0, 2\tau)$. The partial cost function $\mathcal{F}_M^2$ is fixed ($M = 4L$).
We observe that all the algorithms behave in a quite similar way with regards to noise. The weaker the noise is the smaller the error is. The median values of the error $\mathbf{E}_1$ are rather close whatever the considered version of the algorithm and the results hardly depend on the initial values.

To conclude this first set of experiments: Using a partial cost function instead of the whole cost function enables to considerably accelerate the algorithms. Moreover, if

1. **(S)** is robust with respect to noise and initialisation, it remains the slowest of the five algorithms,
2. **(H3)** is robust with respect to noise and initialisation, but remains relatively slow,
3. **(H1) (O)** converge rapidly, yet they are more sensitive to a good initialisation choice,
4. **(H2)** seems to be the best of the five algorithms since it is both fast and robust.

We will now study another type of cost function, namely the ℓ_p-norm function.

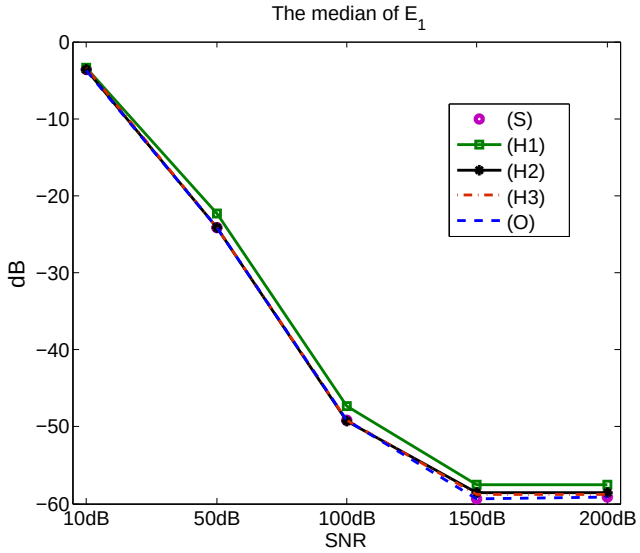

Figure 6: Median of the estimation errors versus the level of noise for the five variants of the suggested algorithms.

5.6. Choice of the ℓ_p-norm

Only the standard stochastic version **(S)** of the algorithm is tested since the calculations of the locally optimal step size are generally more complicated to conduct.

5.6.1. Special case of ℓ_1 cost function

First, we focus on the ℓ_1 cost function. Fig. 7 presents the behaviour of our memetic algorithm using the ℓ_1 cost function $\mathcal{F}_M^1$. Various values of M are tested, $M \in \{4L, 20L, 40L, 60L, 100L, 200L, I_1I_2I_3\}$, which correspond to the following percentage of constituent terms involved in $\mathcal{F}_M^p$: $m = 1\%, 5\%, 10\%, 15\%, 25\%, 50\%$ and 100% . A single initialization is drawn from $\mathcal{U}(0, 2\tau)$. While at $M = 4L$, the algorithm converges in the case of the ℓ_2 cost function $\mathcal{F}_M^2$ (see Fig. 3 at the bottom), this is no more true for $\mathcal{F}_M^1$ under exactly the same conditions (*i.e.* same initialization, same corresponding index set $\mathcal{I}_M$ and same stopping conditions), see in Fig. 7 the black line at 1%. Indeed, the convergence occurs for larger values of M: $M = 20L$ (*i.e.* $m = 5\%$) or greater. Thus, the rates of convergence are approximately linear and the shortest computation time is observed when $m = 10\%$.

5.6.2. Choice of the ℓ_p norm for the partial cost function

We now examine the behavior of **(S)** using ℓ_p cost functions with $p \in \{1, 1.5, 2, 3, 4\}$. Note that a same setting and $M = 4L$ is fixed for all tests. Fig. 8 shows the evolution of $\mathbf{E}_1$ versus time. We observe that the convergence occurs in all cases except when $p = 1$. The fastest convergence is obtained with the ℓ_2 cost function ($p = 2$).

5.7. Comparison of different algorithms

Finally, we compare the computation time obtained with the different suggested algorithms **(S, O, H1, H2, H3)** and with some popular methods e.g. the NTF-ALS [15], the fast NTF-HALS [45][44], and the N-way algorithms [9]. A unique configuration is set up for all experiments. Indeed, we use herein randomly generated data sets (*i.e.* all the loading matrices are drawn from uniform distributions). For simplicity, the tested tensors are chosen in cubic form, i.e. $I_1 = I_2 = I_3 = I$, where I takes value in $\{100, 200, 400\}$. We consider two values for the

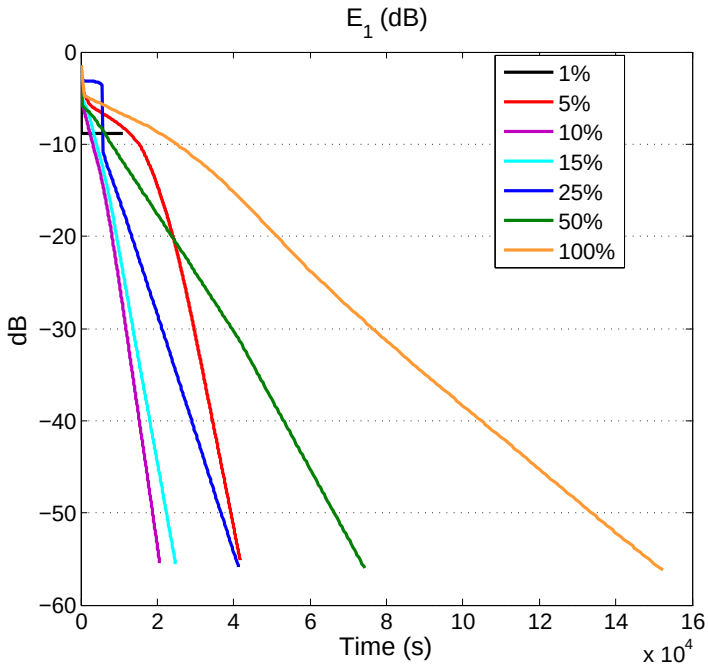

Figure 7: The computation time for the NTF-STO algorithm **(S)** w.r.t. the size M of the cost function $\mathcal{F}_M^1$ (in % of constituent terms involved in $\mathcal{F}_M^1$).

tensor rank $R = \overline{R} = 5$ or 10 but for $R = 10$, we will consider only $I = 100$. The initializations are drawn from the uniform law $U(0, 2\tau)$.
In Table 2, we can see that the new versions of our algorithms **(O, H1, H2)** are the fastest algorithms to converge among our proposed methods. In addition, we also observe a significant improvement in terms of computation time when a pre-stocking is used. Furthermore, the versions (**O, H1, H2**) outperform the NTF-ALS and the Bro's N-way when I becomes large, and remains competitive with these methods even when the tensor rank $\overline{R}$ increases for small scale tensors. This is not the case of the stochastic version **(S)**. We could not make a fair comparison with the fast NTF-HALS because the algorithm does not attain the same performance (RRE$= 10^{-8}$) than other methods. The computation time of the fast NTF-HALS is given as an indication only. Nevertheless, we note that version (**O**) of our algorithm is close to the fast HALS in terms of computation time on the hardest problems. As compared to the method suggested in [66] **(S)** without pre-stoking, we were able to gain a significant factor in the overall speed of convergence (ranging from 15 to 45).

5.8. Estimation of the tensor rank

In the previous simulations of this section, the tensor rank $\overline{R}$ was assumed to be known. In this subsection however, we target a more general tensor decomposition problem where the rank is unknown. To that aim, different approaches have been developed. Most of them are based on the addition of regularization terms [48][65] in the cost function. Here, we suggest the use of the NTF-STO algorithms together with a strategy to estimate $\overline{R}$ by progressively increasing the value of R. This idea has already been used as an acceleration method [34] in order to compute gradually the coefficients of the development of a function on an orthogonal basis. The technique can be summarized as follows:

Step 1. We assume a low tensor rank $R \leq \overline{R}$.

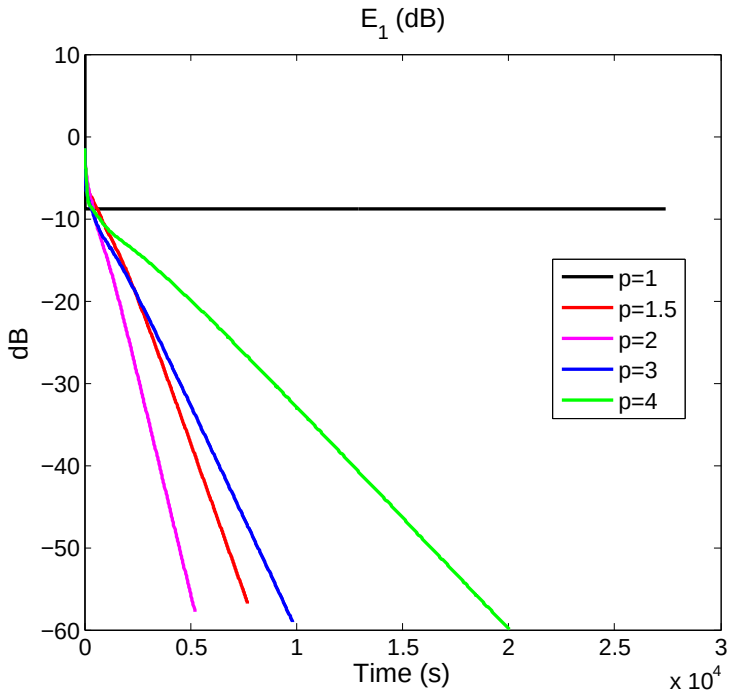

Figure 8: Evolution of the error index $\mathbf{E}_1$ versus time with the NTF-STO algorithm **(S)** for different ℓ_p cost functions, $p \in \{1, 1.5, 2, 3, 4\}$, $M = 4L$.

Step 2. We estimate the corresponding R loading factors of each component using one of the variants **(S, O, H1, H2, H3)** of the algorithm,

Step 3. If a stopping criterion is met (e.g. when $\mathcal{F}_M^p[k] \leq \epsilon_{\mathcal{F}}$) then we break. If not, we increase the rank by 1 unit, *i.e.* $R \leftarrow R + 1$. We then form a new initialization: $R + 1$ loading factors for each factor matrices formed by the previously estimated R loading vectors and a loading vector whose terms are all initialized using a constant (non-null) value. This value is computed using the same kind of heuristic as the one used to initialize the algorithm except that this time the tensor is replaced by the model error obtained at the previous step. We then return to Step 1.

We now illustrate the principle of this technique on noiseless fluorescence spectroscopy like data presented at the beginning of this section. We use a random initialization with a small starting rank e.g. $R = 2$, $\epsilon_{RRE} = 0.1$. The stopping conditions performed with rank R are $k_{\max} = 24 \times 10^7$, RRE$< 10^{-8}$, $\epsilon_K < 10^{-7}$ with $K = 4 \times 10^4$. The obtained FEEM are shown in Fig. 9 for each tested value of R. The process indeed ended when $R = \overline{R} = 5$. The obtained RRE at the end of each phase ($R \in \{2, 3, 4, 5\}$) are given Table 3.

	$R = 2$	$R = 3$	$R = 4$	$R = 5$
RRE	0.199	0.109	5.62×10^{-5}	7×10^{-9}

Table 3: Obtained RRE at the end of each phase ($R \in \{2, 3, 4, 5\}$) using the proposed method, the tensor rank is increased progressively.

We observe that the tensor rank can be estimated and that the FEEM (and all the loading factors even if they are not displayed here) are well recovered.

	$I = 100$ $R = \overline{R} = 5$	$I = 200$ $R = \overline{R} = 5$	$I = 400$ $R = \overline{R} = 5$	$I = 100$ $R = \overline{R} = 10$
NTF-ALS	15s	113s	957s	32s
fast HALS	**2.7s**	**8.7s**	**46s**	**15s**
N-way	11s	52s	388s	*20s*
(S) without stocking	216s	383s	1283s	2508s
(S) with stocking	85s	183s	430s	847s
NTF-STO **(O)**	*10s*	23s	*50s*	55s
NTF-STO **(H1)**	13s	26s	70s	63s
NTF-STO **(H2)**	11s	*21s*	65s	61s
NTF-STO **(H3)**	86s	182s	396s	393s

Table 2: Computer running time (s) of the different methods (best in bold and second best in italic), stopping conditions: relative reconstruction error RRE $\leq 10^{-8}$ except for fast HALS where RRE $\leq 7 \times 10^{-7}$. Noise-free randomly generated data.

6. Computer simulations on experimental data: a water monitoring campaign

We now study a real experimental data set corresponding to a water monitoring campaign [13]. The data in this experiment were acquired automatically every 5 minutes, during a 9 days monitoring campaign performed on water samples collected from an urban river. The size of the baseline data set considered here is $36 \times 111 \times 2592$. The excitation wavelengths range from 225nm to 400nm with a 5nm bandwidth, whereas the emission wavelengths range from 280nm to 500nm with a 2nm bandwidth. The FEEMs have been pre-processed using the Zepp's method [68] to eliminate the Rayleigh and Raman scattering[2]. All remaining negative values (representing 0.34%) of the data were set to 0. One example of a FEEM before and after pre-processing is shown in Fig. 10.

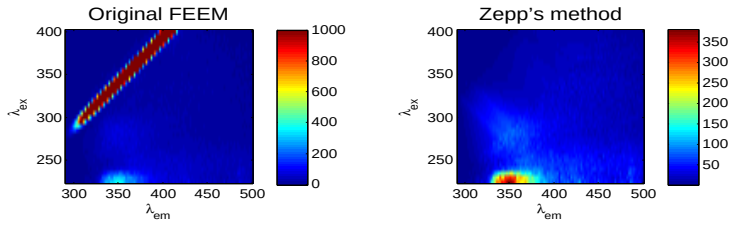

Figure 10: One of the 2594 original FEEM acquired during the monitoring campaign and the same FEEM after pre-processing using the Zepp's method.

Additional pre-processing were applied: the data corresponding to the first 6 emission slits were removed since most of the negative values in the original data occur here. To increase the readability of charts we have chosen here to discard the first 1200 FEEMs where no significant change seems to occur. Two other FEEMs (corresponding to the 1737th and 1738th acquisition

[2] Indeed, other pre-processing techniques among which are mathematical morphology [51] or other filtering techniques, calibration thanks to the use of a blank, reset or shifts can be applied. We underline (though these results are not presented here) that the estimations of the loading matrices performed thanks to the optimization algorithms presented here (i.e. N-way, NLCG, NTF-STO) can significantly differ from the application of one pre-processing technique to another.

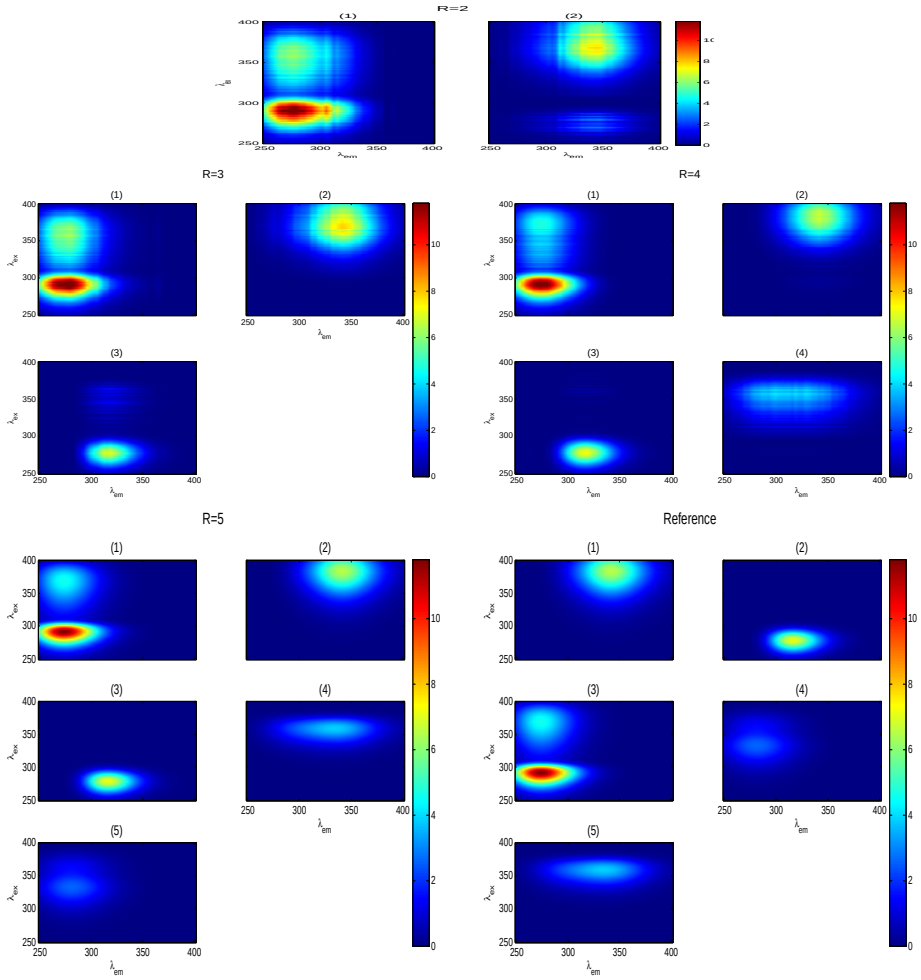

Figure 9: True FEEM (bottom right) and estimated FEEM obtained at the end of each stage when $R = 2$, $R = 3$, $R = 4$ and for the last stage $R = 5$.

time) were suppressed due to noticeable acquisition problems[3]. Finally, the size of data that are processed in this example is $36 \times 105 \times 1392$ (from the 5th day to the 9th day). The number of fluorescent substances present in this experiment is unknown. However by observation of the acquired FEEM, we can notice that an abnormal pattern appears at the seventh day whereas the FEEM obtained before this day appear to be quite stable versus time. The Fig. 11 shows some FEEM acquired before, during, and after that event.

[3]The positions of the scattering effect in these two FEEMs do not correspond to the description in the literature.

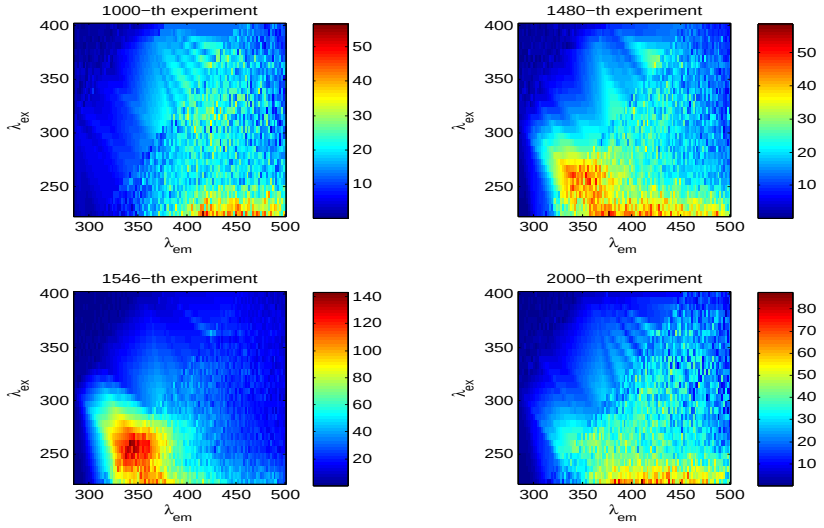

Figure 11: Example of four (pre-processed) FEEM acquired before (top left), during (top right, bottom left), and after (bottom right) day 7.

6.1. Results

Different tensor ranks (i.e. different estimates of the number of chemical compounds) are tested: $R = 4/6$. In the NLCG method [49], a ℓ_1 regularization is used. The regularization parameters are set at $\alpha = 5 \times 10^5$ for $R = 4$ (resp. $\alpha = 3 \times 10^5$ for $R = 6$) (see [64][4]).

In the memetic method, however, no regularization was performed and the hybrid strategy **(H2)** was tested since it seems to be the best variant of our algorithms in terms of robustness and rapidity. The estimation obtained with this method when $R = 6$ was derived by progressively increasing the rank with the starting estimated rank $R = 4$. Finally, the N-way algorithm with non-negativity constraints is tested on the data of interest using the same initialization as the previously mentioned methods.

The reconstructed Fluorescence Excitation Emission Matrices obtained by all the methods are displayed on the left of Fig. 12 (resp. 13) for $R = 4$ compounds (resp. for $R = 6$ compounds). Their concentrations are given on the right of Fig. 12 for $R = 4$ compounds, and in Fig. 14 for $R = 6$ compounds.

[4]To guide the choice of regularization parameters, we considered the ratio between the value of the fidelity term and that of the regularization terms. Various values for this regularization parameter were tested, yet we found that the best results (in terms of low quadratic criterion and of regularization effectiveness) were obtained with these choice of parameters

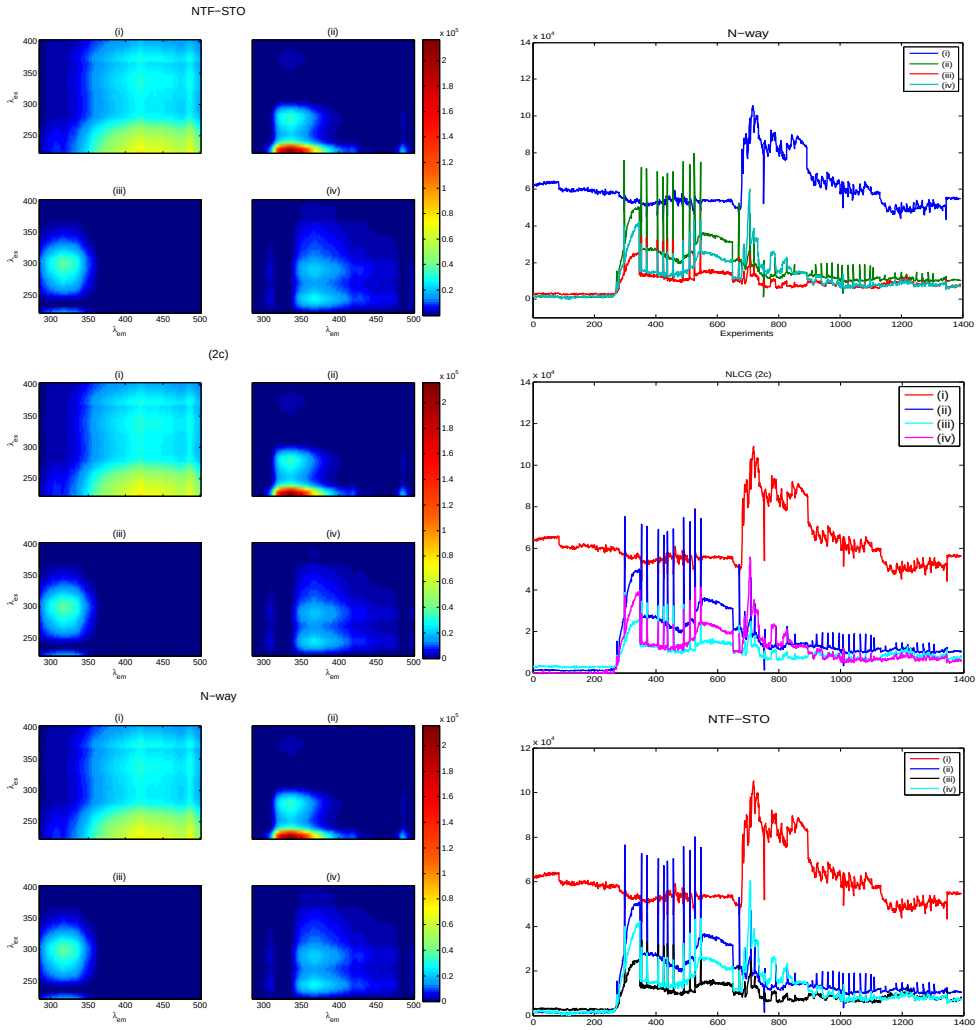

Figure 12: Estimated FEEMs (left) and concentrations (right) using N-way algorithm (top), NLCG algorithm **(2c)** (middle), memetic algorithm **(H2)** (bottom), case $R = 4$.

We observe that for $R = 4$, the three algorithms exhibit very close results, yet, we observe differences for $R = 6$. With the N-way and the NTF-STO algorithms, we obtain very close results again, which is not surprising since these two algorithms do not involve regularisation terms to enforce the sparsity of the solution. On the contrary, the NLCG algorithm, which is regularized tends to enforce sparsity and, in this case, leads us to conclude that there are probably only four fluorescent compounds in the studied collected samples of water. On this experimental data set, the incremental technique suggested in the previous section to both estimate the tensor rank and the factor matrices did not seem to be able to stop the algorithm when the true rank is reached. It may have many causes: model errors, noise, and so on. So, possible improvements will be now to consider regularization terms in our partial cost functions too.

7. Conclusion & perspectives

In this article, we have presented a new stochastic method based on a memetic algorithm in order to solve the CPD problem under non-negativity constraints. This algorithm relies on the

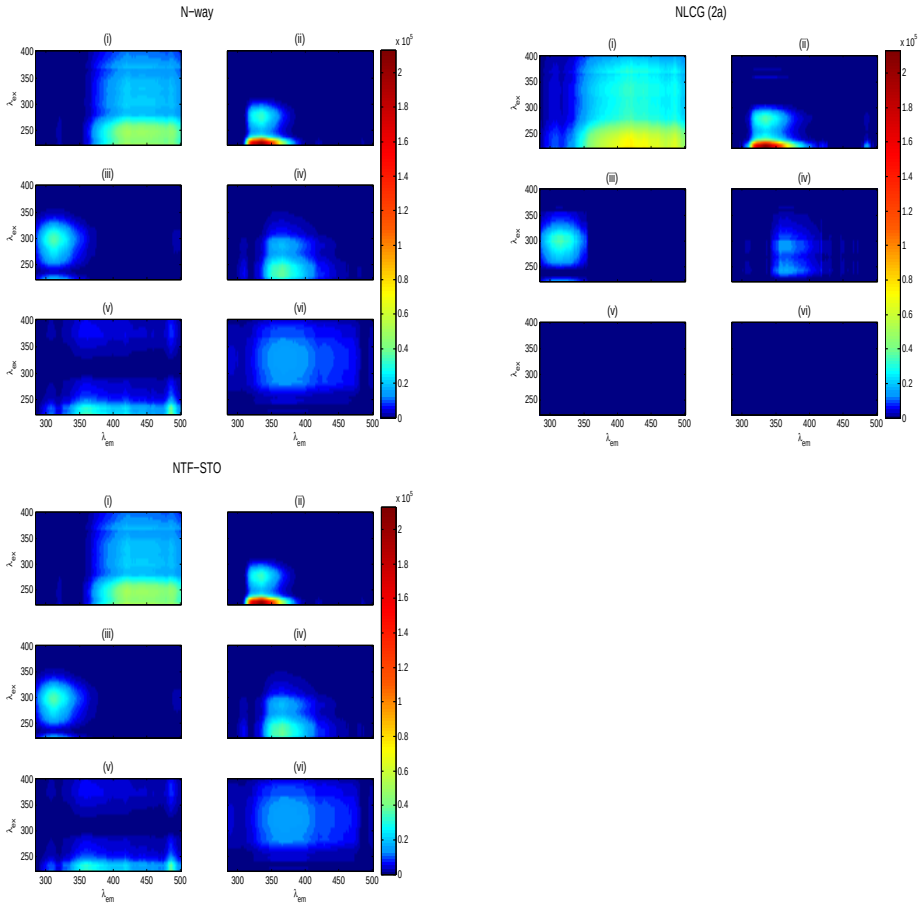

Figure 13: Estimated FEEM using the N-way (top, left), the NLCG ((**2a**), top, right), the memetic algorithm (bottom), case $R = 6$.

choice of a population of candidates, on a search stage involving the calculation of a step size and on an ℓ_p cost function.

First, we have shown that in the case of a stochastic step size adapted to the value of the ℓ_p cost function, the algorithm makes it possible to solve the CPD problem. Furthermore we have remarked that a population of two candidates was sufficient and optimal in terms of computational time to solve this problem. Then, we have tried to reduce the size of the problem by only keeping a random fraction of the involved equations. This idea was efficient too because we were able to recover the hidden loading matrices by keeping only 1% of the original equations. Another improvement was the cheap computation of the ℓ_2 cost function thanks to a computation of only the difference between its values at two consecutive steps. This was particularly efficient because our search stage consists of changing one component at a time of the vector of unknowns. The last trick to improve the speed of convergence was the introduction of a locally "optimal" step size which was obtained by solving a least square problem on one component of the vector of unknowns. The algorithm based on this new step happens to be much more efficient than the one based on the stochastic step, yet, it is less robust versus initialization. When the initialization was too far from the solution, it may fail to converge. This lead us to propose a hybrid version of the algorithm taking advantages of the robustness of the stochastic step size method and the efficiency of the optimal step size method. The combination of all these tools increased

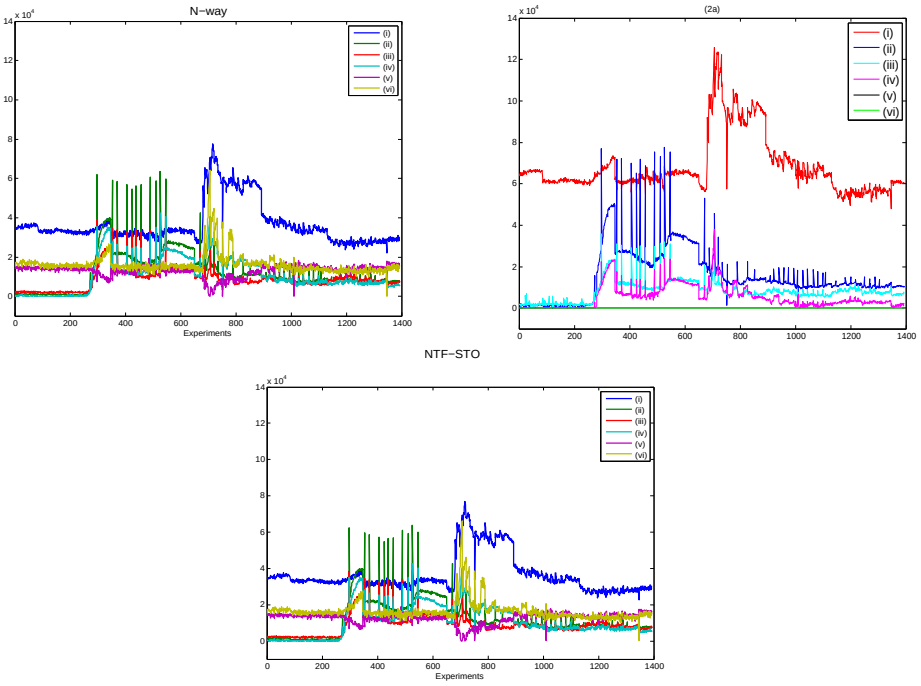

Figure 14: Estimated concentrations using the N-way (top, left), NLCG ((**2a**), top, right), and memetic algorithm (bottom), case $R = 6$.

the efficiency of the crude stochastic algorithm by several degrees of magnitude. We have also made some comparison with standard deterministic algorithms. Compared to Bro's N-way and the fast HALS algorithms, the hybrid versions of our proposed method is highly competitive in terms of computation time, especially when the tensor dimensions are large. Moreover, the approach is quite efficient since the same algorithm can also tackle the CPD problem with missing data which is not the case of classical deterministic methods for which a weighting matrix has to be introduced. Finally, we made a final attempt to improve the algorithm by changing the ℓ_2 cost function into a general ℓ_p cost function. The numerical results seem to show that the choice $p = 2$ is close to optimality. The different variant of these memetic algorithms were tested on synthetic and real experimental data. The hybrid variant **(H2)** seems to offer the best compromise ensuring good performances, a fast convergence and robustness versus noise and initialization. Finally, we also tested an incremental technique to stop the algorithm when the tensor rank is unknown. If good results were obtained on synthetic noiseless data, this way to proceed does not seem to work on real experimental data. That is why we will have to test modified cost functions involving regularization terms.

References

[1] E. Acar, T. G. Kolda, D. M. Dunlavy, and M. Morup, "Scalable tensor factorizations for incomplete data," *Chemometr. Intell. Lab. Syst.*, vol. 106, no. 1, pp. 41–56, Mar. 2011.

[2] E. Acar and B. Yener, "Unsupervised multiway data analysis: a literature survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 1, pp. 6–20, Jan. 2009.

[3] J. Aguilar and A. Colmenares, "Resolution of pattern recognition problems using a hybrid

genetic/random neural network learning algorithm," *Pattern Anal Applic*, vol. 1, no. 1, pp. 52–61, 1998.

[4] K. Aygun, D. Weile, and E. Michielssen, "Design of multilayered periodic strip gratings by genetic algorithms," *Microw. Opt. Technol. Lett.*, vol. 14, no. 2, pp. 81–85, 1997.

[5] T. Back and H.-P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," *Evol. Comput.*, vol. 1, no. 1, pp. 1–23, 1993.

[6] B. Becker and R. Drechsler, "Ofdd based minimization of fixed polarity reed-muller expressions using hybrid genetic algorithms," in *Proceedings IEEE International Conference on Computer Design: VLSI in Computers and Processor*, 1994.

[7] L. Bottou, "Stochastic learning," *Lect. Notes Comput. Sci.*, vol. 3176, 2004.

[8] R. Bro, "Parafac: tutorial and applications," *Chemometr. Intell. Lab. Syst.*, vol. 38, no. 2, pp. 149–171, Oct. 1997.

[9] R. Bro and S. D. Jong, "A fast non-negativity constrained least squares algorithm," *J. Chemometrics*, vol. 11, no. 5, pp. 393–401, 1997.

[10] T. N. Bui and B. R. Moon, "Genetic algorithm and graph partitioning," *IEEE Trans. Comput.*, vol. 45, no. 7, pp. 841–855, 1996.

[11] S. Cadieux, N. Tanizaki, and T. Okamura, "Time efficient and robust 3d brain image centering and realignment using hybrid genetic algorithm," in *Proceedings of the 36th SICE Annual Conference*, 1997, pp. 1279–1284.

[12] C. F. Caiafa and A. Cichocki, "Generalizing the column-row matrix decomposition to multi-way arrays," *Linear Algebra Appl.*, vol. 433, no. 3, pp. 557–573, Sep. 2010.

[13] E. M. Carstea, S. Mounier, R. Redon, X. Luciani, C. Gadio, and A. Baker, "On-line parafac analysis of fluorescence spectra," in *International Workshop on Organic Matter Spectroscopy (WOMS'13)*, Toulon, France, Jul. 2013.

[14] R. Cheng and M. Gen, "Parallel machine scheduling problems using memetic algorithms." in *Proc. Int. Conf. Syst., Man, Cybern. Syst.*, vol. 4, 1996, pp. 2665–2670.

[15] A. Cichocki, R. Zdunek, A. H. Phan, and S. I. Amari, *Non negative matrix and tensor factorizations: Application to exploratory multi-way data analysis and blind separation*. Wiley, 2009.

[16] J. Cohen, R. C. Farias, and P. Comon, "Fast decomposition of large nonnegative tensors," *IEEE Signal Process. Lett.*, vol. 22, no. 7, pp. 862–866, July 2015, online since Nov. 2014.

[17] J. Coloigner, A. Karfoul, L. Alberra, and P. Comon, "Line search and trust region strategies for canonical decomposition of semi-nonnegative semi-symmetric 3rd order tensors," *Lin. Algebra Appl.*, no. 450, pp. 334–374, June 2014.

[18] P. Comon, "Tensor: a brief introduction," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 44–53, May 2014.

[19] D. Costa, N. Dubuis, and A. Hertz, "Embedding of a sequential procedure within an evolutionary algorithm for coloring problems in graphs," *Journal of Heuristics*, vol. 1, no. 1, pp. 105–128, 1995.

[20] C. Cotta and J. Troya, "Using a hybrid evolutionary-a* approach for learning reactive behaviors," in *Real-World Applications of Evolutionary Computation*, vol. 1803, 2000.

[21] A. Franc, "Etude algébrique des multi-tableaux : apport de l'algèbre tensorielle," PhD thesis, University of Montepellier II, Montpellier, France, 1992.

[22] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *J. Math. and Phys.*, vol. 6, pp. 165–189, 1927.

[23] J. H. Holland, *Adaptation in natural and artificial systems*. The MIT Press, 1992.

[24] T. Kim and G. May, "Intelligent control of via formation by photosensitive bcb for mcm-l/d applications," *IEEE Trans. Semicond. Manuf.*, vol. 12, pp. 503–515, 1999.

[25] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[26] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *Siam Review*, vol. 51, no. 3, pp. 455–500, Sept. 2009.

[27] K. Krishna and M. Narasimha-Murty, "Genetic k-means algorithm," *IEEE Trans. Syst. Man. Cybern. B Cybern.*, vol. 29, no. 3, pp. 433–439, 1999.

[28] J. B. Kruskal, *Rank, decomposition and uniqueness for 3-way and n-way arrays*. North-Holland Publishing Co., Apr. 1989, pp. 7–18.

[29] H. J. Kushner and G. G. Yin, *Stochastic approximation algorithms and applications*, ser. Applications of Mathematics. New-York: Springer, 1997, vol. 35.

[30] J. R. Lakowicz, *Principle of fluorescence spectroscopy*, K. Academic, Ed. Plenum Publishers, 1999.

[31] J. R. Lakowicz, H. Szmacinski, K. Nowaczyk, K. W. Berndt, and M. Johnson, "Fluorescence lifetime imaging," *Anal. Biochem.*, vol. 202, no. 2, pp. 316–330, May 1992.

[32] L. H. Lim and P. Comon, "Nonnegative approximations of nonnegative tensors," *J. Chemometrics*, vol. 23, pp. 432–441, Aug. 2009.

[33] L. Ljung, "Analysis of stochastic gradient algorithms for linear regression problems," *IRE Trans. Inform. Theor.*, vol. 30, no. 2, pp. 151–160, Mar. 1984.

[34] S. Maire, "Réduction de variance pour l'intégration numérique et pour le calcul critique en transport neutronique," Ph.D. dissertation, University of Toulon, 2001.

[35] K. Mathias and L. Whitley, "Noisy function evaluation and the delta coding algorithm," in *Proceedings of the SPIE-The International Society for Optical Engineering*, 1994, pp. 53–64.

[36] P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts - towards memetic algorithms," 1989.

[37] ——, "An introduction to population approaches for optimization and hierarchical objective functions: The role of tabu search," *Ann Oper Res*, vol. 41, no. 1-4, pp. 85–121, 1993.

[38] ——, *Memetic algorithms: A short introduction*. Maidenhead, Berkshire, England, UK: McGraw-Hill, 1999, vol. New Ideas in Optimization, pp. 219–234.

[39] P. Moscato and C. Cotta, *A gentle introduction to memetic algorithms*. Boston, MA: Kluwer Academic Publishers, 2003, vol. Handbook of Metaheuristics, pp. 105–144.

[40] T. Q. G. Nguyen, "Méthodes de monte-carlo pour les diffusions discontinues: application à la tomographie par impédance électrique," Ph.D. dissertation, Aix-Marseille Université, Oct. 2015.

[41] D. Nion and N. D. Sidiropoulos, "Adaptive algorithms to track the parafac decomposition of a third-order tensor," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2299–2310, 2009.

[42] P. Osmera, "Hybrid and distributed genetic algorithms for motion control," in *Proceedings of the Fourth International Symposium on Measurement and Control in Robotics*, V. Chundy and E. Kurekova, Eds., 1995, pp. 297–300.

[43] A. H. Phan and A. Cichocki, "Parafac algorithms for large scale problems," *Neurocomputing*, vol. 74, pp. 1970–1984, June 2011.

[44] A. H. Phan, P. Tichavskỳ, and A. Cichocki, "Fast alternating ls algorithms for high order candecomp/parafac tensor factorizations," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4834–4846, June 2013.

[45] ——, "TENSORBOX: a Matlab package for tensor decomposition," 2013. [Online]. Available: http://www.bsp.brain.riken.jp/~phan/tensorbox.php

[46] D. Quagliarella and A. Vicini, "Hybrid genetic algorithms as tools for complex optimisation problems," in *New Trends in Fuzzy Logic II. Proceedings of the Second Italian Workshop on Fuzzy Logic*, 1998, pp. 300–307.

[47] M. Ridao, J. Riquelme, E. Camacho, and M. Toro., "An evolutionary and local search algorithm for planning two manipulators motion," in *Tasks and Methods in Applied Artifcial Intelligence*, A. D. Pobil, J. Mira, and M. Ali, Eds., vol. 1416, 1998.

[48] J.-P. Royer, P. Comon, and N. Thirion-Moreau, "Computing the nonnegative 3-way tensor factorization using tikhonov regularization," in *International Conference on Acoustic Speech and Signal Processing (ICASSP'2011)*, Prague, Czech Republic, May 2011, pp. 2732–2735.

[49] J.-P. Royer, N. Thirion-Moreau, and P. Comon, "Computing the polyadic decomposition of nonnegative third order tensors," *Signal Process.*, vol. 91, no. 9, pp. 2159–2171, Sept. 2011.

[50] ——, "Non-negative 3-way tensor factorization taking into account possible missing data," in *Proc. Eur. Sig. Image Proc. Conf.*, Bucharest, Romania, Aug. 2012.

[51] J.-P. Royer, N. Thirion-Moreau, P. Comon, R. Redon, and S. Mounier, "A regularized nonnegative canonical polyadic decomposition algorithm with preprocessing for 3d fluorescence spectroscopy," *J. Chemometrics*, vol. 29, pp. 253–265, March 2015.

[52] E. Sanchez and B. Kowalski, "Tensorial resolution: a direct trilinear decomposition," *J. Chemometrics*, vol. 4, pp. 29–45, 1990.

[53] N. Sidiropoulos and R. Bro, "On the uniqueness of multilinear decomposition of n-way arrays," *J. Chemometrics*, vol. 14, no. 3, pp. 229–239, May 2000.

[54] N. Sidiropoulos, E. E. Papalexakis, and C. Faloutsos, "Parallel randomly compressed cubes," *IEEE Signal Process. Mag., Special Issue on Big Data*, vol. 31, no. 5, pp. 57–70, 2014.

[55] A. Smilde, R. Bro, and P. Geladi, *Multi-Way Analysis with applications in the chemical sciences*. Wiley, 2004.

[56] J. C. Spall, *Introduction to stochastic search and optimization: estimation, simulation, and control*, R. L. Graham, J. K. Lenstra, and J. H. Spencer, Eds. Wiley, 2003.

[57] ——, *Stochastic Optimization*. Springer, 2012, ch. 7, pp. 173–202.

[58] D. Srinivasan, R. Cheu, Y. Poh, and A. Ng, "Development of an intelligent technique for traffic network incident detection," *Engineering Applications of Artificial Intelligence*, vol. 13, no. 3, pp. 311–322, 2000.

[59] C. A. Stedmon and S. Markager, "Behaviour of the optical properties of coloured dissolved organic matter under conservative mixing," *Estuarine, Coastal and Shelf Science*, vol. 57, no. 5-6, pp. 973–979, 2003.

[60] ——, "Resolving the variability in dissolved organic matter fluorescence in a temperate estuary and its catchment using parafac analysis," *Limnology and Oceanography*, vol. 50, no. 2, pp. 686–697, 2005.

[61] G. Tomasi and R. Bro, "Parafac and missing values," *Chemometr. Intell. Lab. Syst.*, vol. 75, no. 2, pp. 163–180, Feb. 2005.

[62] ——, "A comparison of algorithms for fitting the Parafac model," *Comput. Stat. Data Anal.*, vol. 50, no. 7, pp. 1700–1734, Apr. 2006, elsevier Sciences Publishers B. V.

[63] N. Vervliet and L. D. Lathauwer, "A randomized block sampling approach to canonical polyadic decomposition of large-scale tensors," *IEEE J. Sel. Top. Signal Process.*, vol. 10, no. 2, pp. 284–295, 2016.

[64] X. T. Vu, C. Chaux, S. Maire, and N. Thirion-Moreau, "Study of different strategies for the canonical polyadic decomposition of non-negative third order tensors with application to the separation of spectra in 3D fluorescence spectroscopy," in *IEEE Int. Workshop Mach. Learn. Signal Process.*, Reims, France, Sep 2014.

[65] X. T. Vu, C. Chaux, N. Thirion-Moreau, and S. Maire, "A new penalized nonnegative third order tensor decomposition using a block coordinate proximal gradient approach: application to 3D fluorescence spectroscopy," *J. Chemometrics*, vol. 31, no. 4, pp. 1–17, April 2017.

[66] X. T. Vu, S. Maire, C. Chaux, and N. Thirion-Moreau, "A new stochastic optimization algorithm to decompose large nonnegative tensors," *Signal Process. Lett.*, vol. 22, no. 10, pp. 1713–1717, Oct. 2015.

[67] L. Wang and J. Yen, "Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and kalman filter," *Fuzzy Sets and Systems*, vol. 101, no. 3, pp. 353–362, 1999.

[68] R. G. Zepp, W. M. Sheldon, and M. A. Moran, "Dissolved organic fluorophores in south-eastern us coastal waters: correction method for eliminating rayleigh and raman scattering peaks in excitation-emission matrices," *Marine Chemistry*, vol. 89, no. 1-4, pp. 15–36, 2004.