CANONICAL DECOMPOSITION OF EVEN ORDER HERMITIAN POSITIVE SEMI-DEFINITE ARRAYS

A. KARFOUL[1,2], *L. ALBERA*[1,2], *P. COMON*[3,4]

[1]Inserm, UMR 642, Rennes, F-35000, France
[2]LTSI, University of Rennes 1, Rennes, F-35000, France
[3]CNRS, UMR 6070, Sophia-Antipolis
[4]I3S, University of Nice, 2000 route des Lucioles, BP 121, 06903 Sophia-Antipolis, France

ABSTRACT

Most of the algorithms today available to compute the canonical decomposition of higher order arrays are either computationally very heavy, or are not guaranteed to converge to the global optimum. The solution we propose in order to keep the numerical complexity moderate is i) to stop the latter algorithms once the solution belongs to the convergence region of the global optimum, and ii) to refine the solution with a mere gradient descent algorithm. The case of fourth order hermitian positive semi-definite arrays with complex entries is considered. In fact, the hermitian symmetry constraint is taken into account by optimizing a higher order multivariate polynomial criterion. A compact matrix form of the gradient is then computed based on an appropriate framework allowing for derivation in $\mathbb{C}$ whereas the cost function is not complex analytic. This compact expression is perfectly suitable for matrix-based programming environments such as MATLAB where loops are to be avoided at all costs. Eventually, computer results show a good performance of the proposed approach.

1. INTRODUCTION

Since its first appearance (around 1927), the CANonical Decomposition (CAND) of Higher Order (HO) arrays raised an increasing interest, first in psychometrics and independently in phonetics [15], and later in chemistry and signal processing. Indeed, its range of applications was extended to widespread research areas such and biomedical engineering [9]. The goal of a CAND is to decompose a given HO array into a sum of rank-1 HO arrays provided that the decomposition is essentially unique (i.e. unique up to scale and permutation), extending by this fact both the concepts of rank and Singular Value Decomposition (SVD) of matrices to HO arrays, but without any orthogonal constraint.

Many tools were developed in order to find the CAND of a given HO array. Among them, iterative algorithms such as the Levenberg-Marquardt method [6] and the well-known Alternate Least Square (ALS) [15], or one of its variant versions [11,14,15], just to cite a few. On the other hand, and under some conditions on ranks, the CAND can be transformed into a simultaneous joint diagonalization problem giving rise to a new class of semi-algebraic algorithms [2, 7]. The latter methods are well valued since they avoid the convergence problems of the fully iterative approaches (slow convergence, local minima, etc.). Some of them [2] focus on the decomposition of even HO hermitian positive semi-definite arrays often encountered in practice especially when resorting to HO cumulants [3], for instance to achieve an independent component analysis [1]. But since a relatively small number of steps is generally sufficient to reach a neighborhood of the global optimum (the convergence region), the execution of additional iterations to reach the optimal solution with a good accuracy is generally prohibitive in terms of numerical complexity.

Therefore, and in order to keep the numerical complexity relatively low, we propose to firstly stop the considered semi-algebraic algorithm as soon as its current solution belongs to the convergence region, and secondly to refine the solution with a gradient descent algorithm, much cheaper in terms of numerical complexity. The case of Fourth Order (FO) hermitian positive semi-definite arrays with complex entries is studied. In fact, the hermitian symmetry constraint is taken into account by optimizing a higher order multivariate polynomial criterion. A compact matrix form of the gradient is then computed based on an appropriate framework allowing for derivation in $\mathbb{C}$ whereas the cost function is not complex analytic. This compact expression is perfectly suitable for matrix-based programming environments such as MATLAB where loops are to be avoided at all costs. Eventually, computer results show a good performance of the proposed approach.

2. MATHEMATICAL TOOLS AND PROBLEM FORMULATION

This section is devoted to some basic definitions related to multilinear algebra operations. In the sequel, vectors, matrices and arrays with more than two indices will be denoted in bold lowercase, in bold uppercase and in bold calligraphic uppercase, respectively. Plain uppercases will be mainly used to denote dimensions.

2.1 Basic definitions in multilinear algebra

Definition 1 *A rank-1 q-way array ($q \geq 1$) $\mathcal{T} \in \mathbb{C}^{N_1 \times \cdots \times N_q}$ is equal to the outer product $\mathbf{a}^{(1)} \circ \cdots \circ \mathbf{a}^{(q)}$ of q vectors $\mathbf{a}^{(i)} \in \mathbb{C}^{N_i}$ ($1 \leq i \leq q$) where each element of $\mathcal{T}$ is defined by:*

$$\mathcal{T}_{i_1,\cdots,i_q} = a^{(1)}_{i_1} \cdots a^{(q)}_{i_q} \tag{1}$$

The rank of a HO array always exists and it is defined as follows:

Definition 2 *The rank of a q-way ($q \geq 3$) array $\mathscr{T}$, denoted by* $\mathrm{rk}(\mathscr{T})$*, is the minimal number of rank-1 q-way arrays that yield $\mathscr{T} \in \mathbb{C}^{N_1 \times \cdots \times N_q}$ in a linear combination.*

Thus, contrary to the matrix case, the rank of a HO array can exceed the dimensions of the latter. Using both definitions 1 and 2, CAND of a HO array is given by:

Definition 3 *CAND of a q-way ($q \geq 3$) array $\mathscr{T}$ is the linear combination of $P = \mathrm{rk}(\mathscr{T})$ rank-1 q-way arrays:*

$$\mathscr{T} = \sum_{p=1}^{P} \lambda_p \mathbf{a}_p^{(1)} \circ \cdots \circ \mathbf{a}_p^{(q)} \tag{2}$$

where $\lambda_p \in \mathbb{C}$ and $\mathbf{a}_p^{(i)}$ stands for the p-th column of the well called loading matrix $\mathbf{A}^{(i)} \stackrel{\text{def}}{=} [\mathbf{a}_1^{(i)}, \cdots, \mathbf{a}_P^{(i)}]$ associated to the i-th ($1 \leq i \leq q$) direction of $\mathscr{T}$.

When $\mathscr{T}$ is of Hermitian symmetry, then its loading matrices are equal up to a complex conjugate and its CAND can then be defined by:

Definition 4 *A complex q-way ($q > 3$) array $\mathscr{T}$ is called Hermitian if it admits a CAND of the following form:*

$$\mathscr{T} = \sum_{p=1}^{P} \lambda_p \mathbf{a}_p^{\xi(1)} \circ \cdots \circ \mathbf{a}_p^{\xi(q)} \tag{3}$$

where $\xi(i) = \pm 1$ is defined such that $\mathbf{a}^1 = \mathbf{a}$ and $\mathbf{a}^{-1} = \mathbf{a}^*$ with * the complex conjugate and where the P scalar values λ_p are real.

It is obvious from definition 3 that the different rank-1 terms can be permuted and scaled without modifying the sum. In such a case, CAND is considered to be unique up to these trivial indeterminacies. Several studies were achieved in order to find the aproppriate conditions for CAND's uniqueness. Some of the latter [13] stated that CAND of symmetric HO array is essentially unique with probability one when the dimension of the latter does not exceed its order. Generally, it can be stated that each HO array has a unique CAND if its rank falls between its Kruskal rank defined by Kruskal's condition [12] and its generic rank defined as the rank which occurs with probability one (see [6] and the references therein).

On the other hand, HO arrays can be easily manipulated when they are transformed into matrices, for instance using the following transformation:

Definition 5 *Let $\mathscr{T}$ be a square 2q-way ($q \geq 2$) array of dimensions N. Let $\lfloor q/2 \rfloor$ and $\lceil q/2 \rceil$ be the lower and the upper integer part of $q/2$, repectively. Then the (i,j)-th component of the $(N^q \times N^q)$ unfolding matrix $\mathbf{T} = \mathrm{mat}(\mathscr{T})$ is given by:*

$$T_{i,j} = \mathscr{T}_{n_1,\cdots,n_{\lceil q/2 \rceil},n_{\lceil q/2 \rceil+1},\cdots,n_q,n_{q+1},\cdots,n_{q+\lceil q/2 \rceil},n_{q+\lceil q/2 \rceil+1},\cdots,n_{2q}}$$

where:

$$\begin{aligned} i = {} & (n_1 - 1)N^{q-1} + \cdots + (n_{\lceil q/2 \rceil} - 1)N^{\lfloor q/2 \rfloor} + \\ & (n_{q+\lceil q/2 \rceil+1} - 1)N^{\lfloor q/2 \rfloor - 1} + \cdots + (n_{2q} - 1)N + n_{2q} \end{aligned}$$

and:

$$\begin{aligned} j = {} & (n_{q+1} - 1)N^{q-1} + \cdots + (n_{q+\lceil q/2 \rceil} - 1)N^{\lfloor q/2 \rfloor} + \\ & (n_{\lceil q/2 \rceil+1} - 1)N^{\lfloor q/2 \rfloor - 1} + \cdots + (n_q - 1)N + n_q \end{aligned}$$

It's noteworthy that according to definitions 5 and 4 using $\xi(i) = 1$ for $i \in \{1, \cdots, q\}$ and $\xi(i) = -1$ for $i \in \{q+1, \cdots, 2q\}$, the matrix $\mathbf{T}$ is related to the $(N \times P)$ loading matrix $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_P]$ in the following way:

$$\mathbf{T} = (\mathbf{A}^{\odot q - \lfloor q/2 \rfloor} \odot \mathbf{A}^{*\odot \lfloor q/2 \rfloor}) \mathbf{\Lambda} (\mathbf{A}^{\odot q - \lfloor q/2 \rfloor} \odot \mathbf{A}^{*\odot \lfloor q/2 \rfloor})^{\mathsf{H}} \tag{4}$$

where $\odot$ stands for the Khatri-Rao product (column-wise Kronecker product) [2], $^{\mathsf{H}}$ is the transpose conjugate operator and $\mathbf{\Lambda}$ is a diagonal matrix whose diagonal is the vector $[\lambda_1, \cdots, \lambda_P]$. Note that, for positive semi-definite arrays, the P values λ_p are strictly positive, then we can drop $\mathbf{\Lambda}$ in (4) since its entries λ_p can be involved in the columns of $\mathbf{A}$.

Based on the latter definitions, the problem we tackle here can be defined as follows:

Problem 1 *Let $\mathscr{T}$ be an even HO positive semi-definite array of Hermitian symmetry, find its CAND with a lower computational cost than semi-algebraic methods.*

3. TOWARD A LOWER COST CAND

The solution proposed hereafter to solve problem 1 requires two stages. First, minimum number of steps of a semi-algebraic method [2, 7] is used to compute a solution in the convergence region of the global optimum. Second, the latter solution is then used as an initial point for a gradient descent algorithm, which will then easily attain the optimal solution. This strategy is two-fold. Not only, a global optimum is guaranteed, but the numerical complexity is reduced compared with fully semi-algebraic schemes. For the sake of simplicity, only FO arrays are considered in the sequel.

3.1 A simple Gradient descent algorithm

Gradient descent algorithms are the simplest approaches to solve optimization problems. Besides their simplicity, they are known to be cheap in terms of numerical complexity. Let $\mathbf{\Psi}$ be a real scalar to be minimized with respect to a complex matrix $\mathbf{Z}$ and let $\nabla_{\mathbf{Z}} \mathbf{\Psi}(\mathbf{Z}_0)$ be its gradient at point $\mathbf{Z}_0$, then each iteration of the gradient algorithm is given by:

$$\mathbf{Z}^{(it+1)} = \mathbf{Z}^{(it)} - \mu^{(it)} \nabla_{\mathbf{Z}} \mathbf{\Psi}(\mathbf{Z}^{(it)}) \tag{5}$$

where parameter $\mu^{(it)}$ stands for the step size at the it-th iteration. The main drawback of the gradient descent algorithm is its sensitivity to initialization, especially when the cost function is not convex, because the algorithm may converge to the closest local minimum. Therefore a good initial guess is important in initializing the algorithm. As far as the step size is concerned, its choice should be done in a clever way. In fact, a small step size will lead to a slow convergence, while an overshooting and instability preventing the convergence occur when the step size is too large. A simple way to avoid such situations consists of defining an optimal step computed in an automatic way locally.

3.2 Gradient computation

Since the computation of the gradient of the objective function $\mathbf{\Psi}$ with respect to the considered complex variable $\mathbf{Z}$ is necessary in the gradient descent rule (5), it is useful to give insights into this computation. In the FO array case, a natural

cost function $\mathbf{\Psi}$ derived from equation (4) for $q=2$ is given by:

$$\mathbf{\Psi}(\mathbf{A},\mathbf{A}^*) = \left\|\mathbf{T} - (\mathbf{A}\odot\mathbf{A}^*)(\mathbf{A}\odot\mathbf{A}^*)^{\mathsf{H}}\right\|_F^2 \quad (6)$$

where $\mathbf{T}=\mathrm{mat}(\mathscr{T}),\mathbf{A}$ are two complex matrices of size $(N^2\times N^2)$ and $(N\times P)$, repectively and $\|.\|_F$ is the Frobenius norm.

As is well known, it is legitimate to consider $\mathbf{\Psi}$ as a scalar function of two independent complex matrices $\mathbf{A}$ and $\mathbf{A}^*$, being understood that this view gives a rule of thumb to calculate the derivatives. In fact, considering $\mathbf{\Psi}$ as a function of the real and imaginary parts of $\mathbf{A}$ would be mathematically sound, but would lead to exactly the same result, if the definition below of the complex derivative of a real function is assumed:

$$\frac{\partial\Psi}{\partial z} = \frac{\partial\Psi}{\partial\Re(z)} + j\,\frac{\partial\Psi}{\partial\Im(z)} \quad (7)$$

Note that this definition is necessary, since a real function of a complex variable is not holomorphic (i.e. the Cauchy-Riemann conditions are not satisfied). Then the differential of $\mathbf{\Psi}$ can be written as:

$$\begin{aligned}&\mathrm{d}\mathbf{\Psi}(\mathbf{A},\mathbf{A}^*) = \\ &D_{\mathbf{A}}\mathbf{\Psi}(\mathbf{A},\mathbf{A}^*)\,\mathrm{dvec}(\mathbf{A}) + D_{\mathbf{A}^*}\mathbf{\Psi}(\mathbf{A},\mathbf{A}^*)\,\mathrm{dvec}(\mathbf{A}^*)\end{aligned} \quad (8)$$

where vec is the matrix-to-vector transformation, unvec is its inverse such that $\mathrm{unvec}(\mathrm{vec}(\mathbf{A}))=\mathbf{A}$ and $D_{\mathbf{A}}\mathbf{\Psi}(\mathbf{A},\mathbf{A}^*) = \mathrm{vec}(\partial\mathbf{\Psi}/\partial\mathbf{A})^{\mathsf{T}}\in\mathbb{C}^{1\times NP}$ is defined as a vector presentation of the matrix derivative $\partial\mathbf{\Psi}(\mathbf{A},\mathbf{A}^*)/\partial\mathbf{A}$. It is worth mentioning that the stationary points of $\mathbf{\Psi}$ can be found using the proposition below [5] [10], which is a direct consequence of definition (7):

Proposition 1 *Let $h:\mathbb{C}^{N\times P}\times\mathbb{C}^{N\times P}\longrightarrow\mathbb{R}$ and $g:\mathbb{R}^{N\times P}\times\mathbb{R}^{N\times P}\rightarrow\mathbb{R}$ be defined as $h(\mathbf{Z},\mathbf{Z}^*)=g(\mathbf{X},\mathbf{Y})$, where $\mathbf{Z}=\mathbf{X}+j\mathbf{Y}$. In other words, g and h represent the same map, but in a different coordinate system. A stationary point of function h, or equivalently of g is found by one of the following three equivalent conditions:*

$$\begin{aligned}&(i)\quad D_{\mathbf{X}}g(\mathbf{X},\mathbf{Y})=\mathbf{0}\cap D_{\mathbf{Y}}g(\mathbf{X},\mathbf{Y})=\mathbf{0}\\ &(ii)\quad D_{\mathbf{Z}}h(\mathbf{Z},\mathbf{Z}^*)=\mathbf{0}\\ &(iii)\quad D_{\mathbf{Z}^*}h(\mathbf{Z},\mathbf{Z}^*)=\mathbf{0}\end{aligned}$$

where $\mathbf{0}$ is an NP-dimensional row vector of zeros.

Now, from (8) proposition 1, it is sufficient to compute $D_{\mathbf{A}}\mathbf{\Psi}(\mathbf{A},\mathbf{A}^*)$ in order to find the global minimum of $\mathbf{\Psi}$. Since matrix $\mathbf{T}$ is Hermitian, the objective function $\mathbf{\Psi}$ (6) can be rewritten as:

$$\mathbf{\Psi}(\mathbf{A},\mathbf{A}^*) = \mathrm{Tr}(\mathbf{T}^{\mathsf{H}}\mathbf{T}) - 2f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*) + g(\mathbf{A},\mathbf{A}^*) \quad (9)$$

where $\mathrm{Tr}(\mathbf{Z})$ is the trace of the square matrix $\mathbf{Z}$ and where $f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*)=\mathrm{Tr}(\mathbf{A}_2{}^{\mathsf{H}}\mathbf{T}\mathbf{A}_2)$, $\mathbf{A}_2=\mathbf{A}\odot\mathbf{A}^*$ and $g(\mathbf{A},\mathbf{A}^*)=\mathrm{Tr}(\mathbf{A}_2\mathbf{A}_2{}^{\mathsf{H}}\mathbf{A}_2\mathbf{A}_2{}^{\mathsf{H}})$. Then we get:

$$\mathrm{d}\mathbf{\Psi}(\mathbf{A},\mathbf{A}^*) = -2\mathrm{d}f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*) + \mathrm{d}g(\mathbf{A},\mathbf{A}^*) \quad (10)$$

and consequently:

$$D_{\mathbf{A}}\mathbf{\Psi}(\mathbf{A},\mathbf{A}^*) = -2D_{\mathbf{A}}f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*) + D_{\mathbf{A}}g(\mathbf{A},\mathbf{A}^*) \quad (11)$$

In order to compute the right hand side of (11), useful properties below are recalled:

$$\mathrm{vec}(\mathbf{FZ}) = (\mathbf{I}_P\otimes\mathbf{F})\,\mathrm{vec}(\mathbf{Z}) \quad (12)$$

$$\mathrm{vec}(\mathbf{FZ}) = (\mathbf{Z}^{\mathsf{T}}\otimes\mathbf{I}_P)\,\mathrm{vec}(\mathbf{F}) \quad (13)$$

$$\mathrm{vec}(\mathbf{F}^{\mathsf{T}}) = \mathbf{U}_{PN}\,\mathrm{vec}(\mathbf{F}) \quad (14)$$

$$\begin{aligned}\mathrm{vec}(\mathbf{F}\odot\mathbf{G}) = \mathrm{diag}[\mathrm{vec}(\mathbf{F})\otimes\mathbf{1}_L](\mathbf{U}_{NP}\otimes\mathbf{I}_L)\times\\ (\mathbf{1}_P\otimes\mathbf{I}_{NL})\,\mathrm{vec}(\mathbf{G})\end{aligned} \quad (15)$$

$$\mathrm{vec}(\mathbf{F}\odot\mathbf{G}) = \mathrm{diag}[\mathrm{vec}(\mathbf{1}_P\otimes\mathbf{G})](\mathbf{I}_{PN}\otimes\mathbf{1}_L)\,\mathrm{vec}(\mathbf{F}) \quad (16)$$

$$\mathrm{Tr}(\mathbf{FZ}) = \mathrm{vec}(\mathbf{F}^{\mathsf{H}})^{\mathsf{H}}\mathrm{vec}(\mathbf{Z}) \quad (17)$$

$$\mathrm{Tr}(\mathbf{FZ}) = \mathrm{vec}(\mathbf{F}^{\mathsf{T}})^{\mathsf{T}}\mathrm{vec}(\mathbf{Z}) \quad (18)$$

$$(\mathbf{F}\otimes\mathbf{D})^{\mathsf{T}} = \mathbf{F}^{\mathsf{T}}\otimes\mathbf{D}^{\mathsf{T}} \quad (19)$$

where $\mathbf{F}\in\mathbb{C}^{P\times N},\mathbf{Z}\in\mathbb{C}^{N\times P},\mathbf{G}\in\mathbb{C}^{L\times N},\mathbf{D}\in\mathbb{C}^{L\times M}$, $\otimes$ is the Kronecker product operator and $\mathbf{U}_{PN}$ is a permutation matrix of size $(PN\times PN)$ defined by:

$$\mathbf{U}_{PN} = \sum_{p=1}^{P}\sum_{n=1}^{N}\mathbf{E}_{pn}^{(P\times N)}\otimes\mathbf{E}_{np}^{(N\times P)} \quad (20)$$

with $\mathbf{E}_{pn}^{(P\times N)}$ a $(P\times N)$ elementary matrix of zeros except for the (p,n)-th position which is set to one. In addition, $\mathrm{diag}(\mathbf{v})$ is a diagonal matrix whose diagonal is vector $\mathbf{v}$, $\mathbf{I}_N$ is the $(N\times N)$ identity matrix and $\mathbf{1_N}$ is an N-dimensional column vector of ones.

Let's begin to compute $D_{\mathbf{A}}f_{\mathbf{T}}$, which appears in the right hand side of (11). This requires to compute the differential of $f_{\mathbf{T}}$ given by:

$$\begin{aligned}\mathrm{d}f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*) = D_{\mathbf{A}_2}f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*)\,\mathrm{dvec}(\mathbf{A}_2)\\ + D_{\mathbf{A}_2^*}f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*)\,\mathrm{dvec}(\mathbf{A}_2^*)\end{aligned} \quad (21)$$

where:

$$\mathrm{dvec}(\mathbf{A}_2) = D_{\mathbf{A}}\mathbf{A}_2\,\mathrm{dvec}(\mathbf{A}) + D_{\mathbf{A}^*}\mathbf{A}_2\,\mathrm{dvec}(\mathbf{A}^*) \quad (22)$$

$$\mathrm{dvec}(\mathbf{A}_2^*) = D_{\mathbf{A}}\mathbf{A}_2^*\mathrm{dvec}(\mathbf{A}) + D_{\mathbf{A}^*}\mathbf{A}_2^*\mathrm{dvec}(\mathbf{A}^*) \quad (23)$$

That is to say, inserting (22) and (23) into (21):

$$\begin{aligned}\mathrm{d}f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*) = (D_{\mathbf{A}_2}f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*)D_{\mathbf{A}}\mathbf{A}_2 +\\ D_{\mathbf{A}_2^*}f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*)D_{\mathbf{A}}\mathbf{A}_2^*)\,\mathrm{dvec}(\mathbf{A}) +\\ (D_{\mathbf{A}_2}f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*)D_{\mathbf{A}^*}\mathbf{A}_2 +\\ D_{\mathbf{A}_2^*}f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*)D_{\mathbf{A}^*}\mathbf{A}_2^*)\,\mathrm{dvec}(\mathbf{A}^*)\end{aligned} \quad (24)$$

Now since the differential of $f_{\mathbf{T}}$ is related to $D_{\mathbf{A}}f_{\mathbf{T}}$ in the following way:

$$\begin{aligned}\mathrm{d}f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*) = D_{\mathbf{A}}f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*)\,\mathrm{dvec}(\mathbf{A}) +\\ D_{\mathbf{A}^*}f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*)\,\mathrm{dvec}(\mathbf{A}^*)\end{aligned} \quad (25)$$

we get from (24):

$$\begin{aligned}D_{\mathbf{A}}f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*) = D_{\mathbf{A}_2}f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*)D_{\mathbf{A}}\mathbf{A}_2 +\\ D_{\mathbf{A}_2^*}f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*)D_{\mathbf{A}}\mathbf{A}_2^*\end{aligned} \quad (26)$$

As a result, we need to compute the four derivatives $D_{\mathbf{A}_2}f_{\mathbf{T}}, D_{\mathbf{A}_2}\mathbf{A}_2, D_{\mathbf{A}_2^*}\mathbf{A}_2$ and $D_{\mathbf{A}_2}\mathbf{A}_2^*$ in order to know exactly $D_{\mathbf{A}}f_{\mathbf{T}}$. Using (17), (13), (14), (19), $f_{\mathbf{T}}$ can be written by:

$$\begin{aligned}f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*) = \mathrm{vec}(\mathbf{A}_2)^{\mathsf{T}}\mathbf{U}_{N^2P}^{\mathsf{T}}(\mathbf{T}^{\mathsf{T}}\otimes\mathbf{I}_P)\,\mathbf{U}_{N^2P}\,\mathrm{vec}(\mathbf{A}_2^*)\end{aligned} \quad (27)$$

Then (21) can be expressed as follows:

$$\begin{aligned} \mathrm{d}f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*) = \\ (\mathrm{vec}(\mathbf{A}_2^*)^{\mathsf{T}}\mathbf{U}_{N^2P}^{\mathsf{T}}(\mathbf{T}\otimes\mathbf{I}_P)\mathbf{U}_{N^2P})\mathrm{dvec}(\mathbf{A}_2) + \\ (\mathrm{vec}(\mathbf{A}_2)^{\mathsf{T}}\mathbf{U}_{N^2P}^{\mathsf{T}}(\mathbf{T}^{\mathsf{T}}\otimes\mathbf{I}_P)\,\mathbf{U}_{N^2P})\mathrm{dvec}(\mathbf{A}_2^*) \end{aligned} \quad (28)$$

So, comparing (28) with (21) and using (13) and (14), we have:

$$\begin{aligned} D_{\mathbf{A}_2} f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*) &= \mathrm{vec}(\mathbf{T}^{\mathsf{T}}\mathbf{A}_2^*) \\ D_{\mathbf{A}_2^*} f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*) &= \mathrm{vec}(\mathbf{T}\mathbf{A}_2) \end{aligned} \quad (29)$$

Moreover, from (15) and (16), we get:

$$D_{\mathbf{A}}\mathbf{A}_2 = \mathrm{diag}(\mathrm{vec}(\mathbf{1}_N\otimes\mathbf{A}^*))(\mathbf{I}_{NP}\otimes\mathbf{1}_N) \quad (30)$$

$$D_{\mathbf{A}}\mathbf{A}_2^* = \mathrm{diag}(\mathrm{vec}(\mathbf{A}^*)\mathbf{1}_N)(\mathbf{U}_{PN}\mathbf{I}_N)(\mathbf{I}_{NP}\otimes\mathbf{1}_N) \quad (31)$$

Thus, according to the latter equations, we obtain:

$$\begin{aligned} D_{\mathbf{A}} f_{\mathbf{T}}(\mathbf{A},\mathbf{A}^*) = (\mathrm{vec}(\mathbf{T}^{\mathsf{T}}\mathbf{A}_2^*)^{\mathsf{T}}\mathrm{diag}(\mathrm{vec}(\mathbf{1}_N\otimes\mathbf{A}^*))\times \\ (\mathbf{I}_{NP}\otimes\mathbf{1}_N) + \mathrm{vec}(\mathbf{T}\mathbf{A}_2)^{\mathsf{T}}\mathrm{diag}(\mathrm{vec}(\mathbf{A}^*)\otimes\mathbf{1}_N)\times \\ (\mathbf{U}_{PN}\otimes\mathbf{I}_N)(\mathbf{1}_N\otimes\mathbf{I}_{NP})) \end{aligned} \quad (32)$$

Now, we have just to compute the derivative $D_{\mathbf{A}}g$ to get the entier expression of the gradient (11) of $\mathbf{\Psi}$ with respect to $\mathrm{vec}(\mathbf{A})$. We derive from (9) and (18):

$$g(\mathbf{A},\mathbf{A}^*) = \mathrm{vec}(\mathbf{A}_2\mathbf{A}_2{}^{\mathsf{H}})^{\mathsf{H}}\,\mathrm{vec}(\mathbf{A}_2\mathbf{A}_2{}^{\mathsf{H}}) \quad (33)$$

Now, following the same procedure used to derive (24) from (25)-(23), we get, if we provisionnally denote $\mathbf{Y} = \mathbf{A}_2\mathbf{A}_2{}^{\mathsf{H}}$:

$$\begin{aligned} D_{\mathbf{A}}g(\mathbf{A},\mathbf{A}^*) = D_{\mathbf{Y}}g(\mathbf{A},\mathbf{A}^*)D_{\mathbf{A}_2}\mathbf{Y}D_{\mathbf{A}}\mathbf{A}_2 + \\ D_{\mathbf{Y}^*}g(\mathbf{A},\mathbf{A}^*)D_{\mathbf{A}_2}\mathbf{Y}^*D_{\mathbf{A}}\mathbf{A}_2 + \\ D_{\mathbf{Y}}g(\mathbf{A},\mathbf{A}^*)D_{\mathbf{A}_2^*}\mathbf{Y}D_{\mathbf{A}}\mathbf{A}_2 + \\ D_{\mathbf{Y}^*}g(\mathbf{A},\mathbf{A}^*)D_{\mathbf{A}_2^*}\mathbf{Y}^*D_{\mathbf{A}}\mathbf{A}_2 \end{aligned} \quad (34)$$

Consequently, it is easy to deduce the following results:

$$D_{\mathbf{Y}}g(\mathbf{A},\mathbf{A}^*) = \mathrm{vec}(\mathbf{Y})^{\mathsf{H}} \quad (35)$$

$$D_{\mathbf{Y}^*}g(\mathbf{A},\mathbf{A}^*) = \mathrm{vec}(\mathbf{Y})^{\mathsf{T}} \quad (36)$$

Also, using equations (12), (13), we have:

$$D_{\mathbf{A}_2}\mathbf{Y} = (\mathbf{A}_2^*\otimes\mathbf{I}_{N^2}) \quad (37)$$

$$D_{\mathbf{A}_2^*}\mathbf{Y} = (\mathbf{I}_{N^2}\otimes\mathbf{A}_2)\mathbf{U}_{N^2P} \quad (38)$$

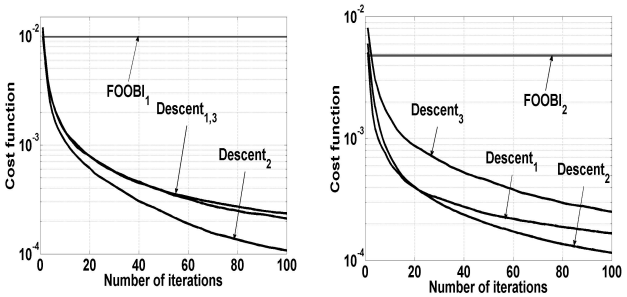
and:

$$D_{\mathbf{A}_2}\mathbf{Y}^* = (\mathbf{I}_{N^2}\otimes\mathbf{A}_2^*)\mathbf{U}_{N^2P} \quad (39)$$

$$D_{\mathbf{A}_2^*}\mathbf{Y}^* = (\mathbf{A}_2^*\otimes\mathbf{I}_{N^2}) \quad (40)$$

Then using equations (34)-(40) and equations (15), (16), we obtain the following derivative of g:

$$\begin{aligned} D_{\mathbf{A}}g(\mathbf{A},\mathbf{A}^*) = 2\mathrm{vec}(\mathbf{A}_2^*\mathbf{A}_2^{\mathsf{T}}\mathbf{A}_2^*)^{\mathsf{T}}\mathrm{diag}(\mathrm{vec}(\mathbf{1}_N\otimes\mathbf{A}^*)) \\ \times(\mathbf{I}_{NP}\otimes\mathbf{1}_N) + 2\mathrm{vec}(\mathbf{A}_2\mathbf{A}_2^{\mathsf{H}}\mathbf{A}_2)^{\mathsf{T}}\mathrm{diag}(\mathrm{vec}(\mathbf{A}^*)\otimes\mathbf{1}_N) \\ \times(\mathbf{U}_{PN}\otimes\mathbf{I}_N)(\mathbf{1}_N\otimes\mathbf{I}_{NP}) \end{aligned} \quad (41)$$



(a) Using FOOBI_1 (b) Using FOOBI_2

Figure 1: The median of the objective function $\mathbf{\Psi}$ over 1000 realizations, as a function of the number of iterations when a gradient descent is used to refine the convergence, starting from an initial guess estimated by a FOOBI method.

4. COMPUTER RESULTS

The performance of the proposed approach is studied hereafter. A matrix $\mathbf{A}$ of size $(N\times P)$ is generated modeling the propagation of $P=4$ angularly well-separated sources impinging on a uniform linear array of $N=3$ sensors. A FO positive semi-definite array with hermitian symmetry is computed, based on equation (3) for $q=4$ and $\lambda_1=\cdots=\lambda_4=1$. A noise array uniformly distributed in $[0,1]$ is added to the resulting FO array in order to produce a noisy hermitian positive semi-definite FO array $\mathscr{T}$. As pointed out in section 3, our approach uses the first steps of a semi-algebraic method as an initial guess to attain the convergence region of the global minimum. We propose to use either the first implementation of FOOBI [8], namely FOOBI_1 or its second implementation, called FOOBI_2, as semi-algebraic method. The different steps of FOOBI_1 are summarized below:

Step1 Build the $(N^2\times N^2)$ hermitian positive semi-definite matrix $\mathbf{T}=\mathrm{mat}(\mathscr{T})$.

Step2 Compute the EVD $\mathbf{T}=\mathbf{E}\mathbf{\Lambda}\mathbf{E}^{\mathsf{H}}$ and a square root $\mathbf{H}=\mathbf{E}\mathbf{\Lambda}^{1/2}$. Next, normalize the eigenvectors such that the P matrices $\mathbf{H}_p=\mathrm{unvec}(\mathbf{h}_p)$ of size $(N\times N)$ are hermitian.

Step3 Compute the $P(P+1)/$ FO arrays $\mathscr{L}_{st}=\Phi(\mathbf{H}_s,\mathbf{H}_t)$ for $1\le s\le t\le P$ where $\mathbf{\Phi}^{(1)}(\mathbf{X},\mathbf{Y})\in\mathbb{C}^{N\times N\times N\times N}$ is a rank-1 detecting device defined for matrices $\mathbf{X}$ and $\mathbf{Y}$ of size $(N\times N)$ in element-way manner as follows:

$$\mathbf{\Phi}^{(1)}(\mathbf{X},\mathbf{Y})_{ijk\ell} = X_{ij}Y_{k\ell}^* + Y_{ij}X_{k\ell}^* - X_{ik}Y_{i\ell}^* - X_{ik}X_{j\ell}^* \quad (42)$$

and stack the results in matrix $\mathbf{M}=[\mathrm{vec}(\mathscr{L}_{11}),\mathrm{vec}(\mathscr{L}_{22}),\cdots,\mathrm{vec}(\mathscr{L}_{PP}),\mathrm{vec}(\mathscr{L}_{12}),\mathrm{vec}(\mathscr{L}_{13}),\cdots,\mathrm{vec}(\mathscr{L}_{1P}),\mathrm{vec}(\mathscr{L}_{23}),\cdots,\mathrm{vec}(\mathscr{L}_{2P}),\cdots,\mathrm{vec}(\mathscr{L}_{P-1P})]$ of size $(N^4\times P(P+1)/2)$.

Step4 Compute the P right singular vectors $\mathbf{u}_p$ of $\mathbf{M}$ that correspond to the smallest singular values. Stack these vectors in P upper triangular matrices $\mathbf{U}_p$ such that $\mathbf{u}_p=[U_{p,11},U_{p,22},\cdots,U_{p,PP},U_{p,12},U_{p,13},\cdots,U_{p,1P},U_{p,23},\cdots,U_{p,2P},\cdots,U_{p,P-1P}]^{\mathsf{T}}$ where $U_{p,ij}$ is the (i,j)-th entry of matrix $\mathbf{U}_p$. Then compute the P matrices $\mathbf{W}_p=(\mathbf{U}_p+\mathbf{U}_p{}^{\mathsf{T}})/2$.

Step5 Compute the orthogonal matrix $\mathbf{Q}$ that simultaneously diagonalizes the P matrices $\mathbf{W}_p$, for instance using the JAD algorithm [4].

Step6 Compute an estimate, $\widehat{\mathbf{A}}_2$, of matrix $\mathbf{A}_2$ as $\widehat{\mathbf{A}}_2 = \mathbf{E}\,\mathbf{\Lambda}^{1/2}\,\mathbf{Q}$.

Step7 Estimate one column vector of matrix $\mathbf{A}$ as the dominant left singular vector of unvec$(\widehat{\mathbf{a}}_{2,p})$ where $\widehat{\mathbf{a}}_{2,p}$ is one of the P column vectors of $\widehat{\mathbf{A}}_2$.

As far as FOOBI$_2$ is concerned, it follows the previous steps but with a different rank-1 detecting device noted $\mathbf{\Phi}^{(2)}(\mathbf{X},\mathbf{Y}) \in \mathbb{C}^{N\times N}$ defined by:

$$\mathbf{\Phi}^{(2)}(\mathbf{X},\mathbf{Y}) = \mathbf{XY} - \mathrm{Tr}(\mathbf{X})\mathbf{Y} + \mathbf{YX} + \mathrm{Tr}(\mathbf{Y})\mathbf{X} \qquad (43)$$

and using a variant of JAD in order to identify $\mathbf{Q}$. More precisely, we propose i) to run steps 1 to 6 using either FOOBI$_1$ or FOOBI$_2$ in order to compute an estimate, $\widehat{\mathbf{A}}_2$, of the $(N^2 \times P)$ matrix $\mathbf{A}_2$ and then ii) to use one of the N matrix blocks $\mathbf{\Sigma}_n^* = \mathbf{A}\mathbf{D}_n\mathbf{\Pi}$ of size $(N \times P)$ obtained from $\widehat{\mathbf{A}}_2 = [\mathbf{\Sigma}_1{}^{\mathsf{T}}, \mathbf{\Sigma}_2{}^{\mathsf{T}}, \cdots, \mathbf{\Sigma}_N{}^{\mathsf{T}}]^{\mathsf{T}}$, where $\mathbf{D}_n, \mathbf{\Pi}$ are diagonal and permutation matrices, respectively.

Figures 1(a) and 1(b) display the variations of the median of the objective function $\mathbf{\Psi}$ calculated over 1000 independent realizations as a function of the number of iterations. In fact, each figure shows the performance of a gradient descent named "Descent$_n$", initialized with the n-th block $\mathbf{\Sigma}_n^*$. These descents are compared to the performance of a complete FOOBI method. As already pointed out, only the first six steps of FOOBI$_1$ and FOOBI$_2$ are run to initialize our three descents in figures 1(a) and 1(b), respectively. Note that whatever the FOOBI method, only some FOOBI steps followed by a gradient descent are necessary, and hence should be preferred instead of a complete execution of FOOBI.

5. CONCLUSION

A novel approach to compute the CAND of FO hermitian positive semi-definite arrays has been proposed. It proceeds in two main stages. The first one consists of a rough calculation of the CAND using some steps of an appropriate semi-algebraic method so that the solution obtained lies in the neighborhood of the global optimum. The second one consists of using the latter solution as an initial guess for a gradient descent algorithm in order to reach the global optimum, with no significant additional computational burden. Expressions of the gradients have been provided; the difficulty to obtain their expressions was mainly coming from the fact that variables were complex, and that the optimization criterion enjoyed some symmetry properties. Computer results showed the good performance of the proposed method. A generalization to higher orders will be given in a longer version of this paper. Moreover, our approach will be compared with other CAND methods in terms of computational complexity, which is more meaningful than the number of iterations.

Acknowledgment

This work has been supported in part by contract ANR-06-BLAN-0074 "Decotes", and in part by contract ANR-BLAN07-0314-02 "mv-EMD".

REFERENCES

[1] L. ALBERA, A. FERREOL, P. CHEVALIER, and P. COMON, "ICAR, a tool for blind source separation using fourth order statistics only," *Trans. On Sig. Proces.*, vol. 53, no. 10, pp. 2986–2997, October 2005.

[2] L. ALBERA, A. FERREOL, P. COMON, and P. CHEVALIER, "Blind Identification of Overcomplete Mixtures of sources (BIOME)," *Linear Algebra Applications*, vol. 391C, pp. 3–30, November 2004.

[3] P. O. AMBLARD, M. GAETA, and J. L. LACOUME, "Statistics for complex variables and signals-part I: Variables," *Signal Processing*, vol. 53, pp. 1–13, 1996.

[4] J.-F. CARDOSO and A. SOULOUMIAC, "Jacobi angles for simultaneous diagonalization," *SIAM Jour. Mat. Anal. Applic.*, vol. 17, no. 1, pp. 161–164, 1996.

[5] P. COMON, "Estimation multivariable complexe," *Traitement du Signal*, vol. 3, no. 2, pp. 97–101, 1986.

[6] P. COMON, X. LUCIANI, and A. L. F. de ALMEIDA, "Tensor decomposition, alternating least squares and other tales," *Jour. Chemometrics*, special issue, 2009.

[7] L. DeLATHAUWER, "A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization," *SIAM Jour. Mat. Anal. Applic.*, vol. 28, no. 3, pp. 642–666, 2006.

[8] L. DeLATHAUWER, J. CASTAING, and J. F. CARDOSO, "Fourth-order cumulant-based blind identification of underdetermined mixtures," *IEEE Trans. Sig. Proc.*, vol. 55, no. 6, pp. 2965–2973, June 2007.

[9] C. S. HERMANN, J. PARNAS, M. MORUP, L. K. HANSEN, and S. M. ARNFRED, "Parallel factor analysis as an exploratory tool for wavelet transformed eventrelated EEG," *NeuroImage*, vol. 29, no. 3, pp. 938–947, 2006.

[10] A. HJORUNGNES and D. GESBERT, "Complex-valued matrix differentiation: Techniques and key results," *Trans. On Sig. Proces.*, vol. 55, no. 6, pp. 2740–2746, June 2007.

[11] A. KARFOUL, L. ALBERA, and L. D. LATHAUWER, "Canonical decomposition of even higher order cumulant arrays for blind underdetermined mixture identification," in *the IEEE SAM workshop 2008*, Darmstatdt, Germany, July 21-23 2008, pp. 501–505.

[12] J. B. KRUSKAL, "Three-way arrays : Rank and uniqueness of trilinear decompositions," *Linear Algebra and Applications*, vol. 18, pp. 95–138, 1977.

[13] M. MELLA, "Singularities of linear systems and the waring problem," *Trans. Am. Math. Soc.*, vol. 358, no. 12, pp. 5523–5538, December 2006.

[14] D. NION and L. DeLATHAUWER, "An enhanced line search scheme for complex-valued tensor decompositions. application in ds-cdma," *Signal Processing*, no. 88, pp. 749–755, 2008.

[15] M. RAJIH, P. COMON, and R. HARSHMAN, "Enhanced line search : A novel method to accelerate PARAFAC," *SIAM Journal on Matrix Analysis Appl.*, vol. 30, no. 3, pp. 1148–1171, Sep. 2008.