Double Coupled Canonical Polyadic Decomposition of Third-Order Tensors: Algebraic Algorithm and Relaxed Uniqueness Conditions

Xiao-Feng Gong, *Member, IEEE*, Qiu-Hua Lin, *Member, IEEE*, Feng-Yu Cong, *Senior Member, IEEE*, and Lieven De Lathauwer, *Fellow, IEEE*

Abstract **—** Double coupled canonical polyadic decomposition (DC-CPD) decomposes multiple tensors with coupling in the first two modes, into minimal number of rank-1 tensors that also admit the double coupling structure. It has a particular interest in joint blind source separation (J-BSS) applications. In a preceding paper, we proposed an algebraic algorithm for underdetermined DC-CPD, of which the factor matrices in the first two modes of each tensor may have more columns than rows. It uses a pairwise coupled rank-1 detection mapping to transform a possibly underdetermined DC-CPD into an overdetermined DC-CPD, which can be solved algebraically via generalized eigenvalue decomposition (GEVD). In this paper, we generalize the pairwise or second-order coupled rank-1 detection mapping to an arbitrary order $K \geq 2$. Based on this generalized coupled rank-1 detection mapping, we propose a broad framework for the algebraic computation of DC-CPD, which consists of a series of algorithms with more relaxed working assumptions, each corresponding to a fixed order $K \geq 2$. Deterministic and generic uniqueness conditions are provided. We will show

This research is funded by: (1) National natural science foundation of China (nos. 61671106, 61331019, 61871067, 81471742); (2) Fundamental Research Funds for the Central Universities (nos. DUT16QY07). (3) Research Council KU Leuven: C1 project C16/15/059-nD. (4) FWO: project G.0830.14N, G.0881.14N, EOS project no. 30468160 (SeLMA); (5) EU: The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Advanced Grant: BIOTENSORS (no. 339804). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information.

Xiao-Feng Gong and Qiu-Hua Lin are with the School of Information and Communication Engineering, Dalian University of Technology, Dalian, China, 116024. (e-mails: xfgong@dlut.edu.cn, qhlin@dlut.edu.cn).

Feng-Yu Cong is with the Department of Biomedical Engineering, Dalian University of Technology, Dalian, China, 116024 (e-mail: cong@dlut.edu.cn).

Lieven De Lathauwer is with the STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics; Department of Electrical Engineering (ESAT), KU Leuven, BE-3001 Leuven, Belgium; and Group Science, Engineering and Technology, KU Leuven, Kulak 8500 Kortrijk, Belgium (e-mail: Lieven.DeLathauwer@kuleuven.be).

through analysis and numerical results that our new uniqueness conditions for DC-CPD are more relaxed than the existing results for DC-CPD and CPD. We will further show, through simulation results, the performance of the proposed algebraic DC-CPD framework in approximate DC-CPD and a J-BSS application, in comparison with existing DC-CPD and CPD algorithms.

Index Terms — Tensor; Canonical polyadic decomposition; Double coupled; Algebraic algorithm; Uniqueness.

I. Introduction

Double coupled canonical polyadic decomposition (DC-CPD) is the minimal decomposition of $M \times M$ third-order tensors $\{\mathcal{T}^{(m,n)} \in \mathbb{C}^{N \times N \times T}, m,n = 1,...,M\}$ with coupling in the first two modes, into $M \times M$ sets of rank-1 tensors with the same double coupling structure in the first two modes, as shown in Fig. 1. DC-CPD is of particular interest in joint blind source separation (J-BSS) [2]–[7], [23], which has emerged recently as a data-driven technique for multi-set data fusion. The main idea of J-BSS is to exploit the dependence across datasets (inter-set dependence) and independence of latent sources within a dataset (intra-set independence) to perform BSS, with indication of correspondences among decomposed components across datasets.

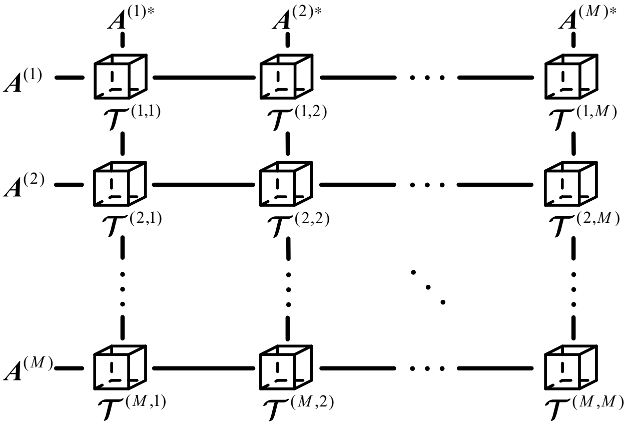


Fig. 1. Illustration of DC-CPD (picture taken from [23]). The tensors are placed at different nodes of a grid according to their indices. The tensor at node (m, n) admits $\mathcal{T}^{(m,n)} = [\![\boldsymbol{A}^{(m)}, \boldsymbol{A}^{(n)*}, \boldsymbol{C}^{(m,n)}]\!]_R$, m, n = 1,…,M. Tensors in the mth row of the grid are coupled in the first mode by $\boldsymbol{A}^{(m)}$. Tensors in the mth column of the grid are coupled in the second mode by $\boldsymbol{A}^{(m)*}$.

The idea of DC-CPD originates from work on multi-set canonical correlation analysis (MCCA) [1]. MCCA constructs a cross-covariance matrix between each pair of datasets, and thus for all the pairs of datasets the constructed cross-covariance matrices together admit a double coupled matrix decomposition. In [2]–[5], a covariance tensor is constructed between each pair of datasets, which holds several covariance matrices at different time-lags or time frames as its frontal slices, and thus all the constructed tensors admit a DC-CPD. In [6], a second-order statistics based method has been proposed to construct tensors that admit a DC-CPD, without imposing additional diversities of the source signals, e.g., nonstationarity, correlation among samples, etc. However, the methods in [2]–[6] are limited to the over-determined case where there are more observation channels than source signals in each dataset, i.e., the mixing matrix in each dataset has full column rank. In [7], real-valued underdetermined J-BSS is considered. In the constructed DC-CPD the factor matrices in the first two modes of each tensor may not have full column rank. The framework of structured data fusion (SDF) [8] can be used to implement iterative DC-CPD methods based on numerical optimization techniques including quasi-Newton and nonlinear least squares (NLS). The latest version of SDF was embedded in the Tensorlab 3.0 software package [9].

Here we note that our study of DC-CPD is in essence different from the coupled CPD (C-CPD) works in the literature [10]–[21], in the sense that these C-CPD methods mostly assume that the common factor matrix has full column rank, while in DC-CPD the common factor matrices in the first two modes may not have full column rank. In [21], a Bayesian framework for incorporating flexible coupling has been introduced for coupled tensor decomposition. It considers the single (flexible) coupling between two tensor datasets and thus is also different from the DC-CPD model.

On the other hand, in the DC-CPD works mentioned above [2]–[5], [7], the algebraic aspects of DC-CPD, including the algebraic computation and uniqueness conditions, are not studied. We formally call an algorithm algebraic if it relies only on arithmetic operations, overdetermined sets of linear equations, matrix singular value decomposition (SVD) and generalized eigenvalue decomposition (GEVD). In particular, an algebraic computation does not involve a numerical optimization in which the cost function may have multiple local optima, or of which the convergence to the exact solution is not guaranteed. In [6], the DC-CPD identifiability issue is studied for the case in which at least two factor matrices of each tensor have full column rank (namely the first and the second factor matrix in (2) below). In a preceding manuscript [23], we have considered the case in which only one factor matrix of each tensor is assumed to have full column rank (namely the third factor matrix), and have proposed an algebraic DC-CPD algorithm. This algorithm uses a coupled rank-1 detection mapping to convert the decomposition of each pair of coupled tensors in DC-CPD into an overdetermined CPD. By applying the coupled rank-1 detection to all pairs of coupled tensors, overall an overdetermined DC-CPD is obtained. The latter can be solved algebraically via GEVD. In [23], the uniqueness conditions for DC-CPD are also studied, which are shown to be more relaxed than those of its CPD counterpart.

In this paper, we provide a more thorough study of the algebraic aspects of DC-CPD, for the case where only the third factor matrix of each tensor is assumed to have full column rank, including a broad framework for the algebraic computation of DC-CPD and uniqueness conditions that are shown to be more relaxed than those in [23]. The main contributions of this work are summarized as follows.

(*i*) We propose a new coupled rank-1 detection mapping of order $K \geq 2$, which exploits the coupling structure among K coupled tensors each time. As a basic mathematical tool for algebraic DC-CPD, it generalizes the second-order coupled rank-1 detection mapping proposed in [23], which exploits the pairwise coupling structure between two coupled tensors at a time.

(*ii*) Based on the new coupled rank-1 detection mapping, we propose a broad framework for the algebraic computation of DC-CPD, which consists of a series of algebraic algorithms, each corresponding to a fixed order $K \geq 2$. For any selected value of K, the corresponding algorithm converts all the K-combinations of coupled tensors of an underdetermined DC-CPD into a set of tensors that admits a (K+1)th-order overdetermined C-CPD with coupling in the first K modes, a problem that admits an algebraic solution via GEVD. The advantage of using the coupled rank-1 detection mapping of order $K \geq 2$ over its second-order counterpart [23], is that the working assumptions for the proposed algorithm are more relaxed.

(*iii*) We present new deterministic and generic uniqueness conditions for DC-CPD, which are shown to be more relaxed than those presented in [23] and those for its CPD counterpart [25]–[29]. Due to the more relaxed uniqueness conditions, the proposed algorithm can handle difficult underdetermined DC-CPD problems, for which existing algorithms, not even optimization based, do not generate correct results, as will be shown in simulations.

Although the above contributions may be considered as generalizations of results in [23], the study is far from easy. To provide a brief reading help, we note that the basic idea behind the development from [23] to the present paper is to some extent analogous to how the results for algebraic CPD have evolved over the years in [24]–[29]. More precisely, in [24] the first rank-1 detection mapping was introduced. Based on a variant of the mapping in [24], an algebraic CPD algorithm was developed and corresponding uniqueness results derived [25]. Then, the concept of rank-1 detection was generalized to rank-R detection through the use of compound matrices, to compute the CPD of a third-order tensor, none of the three factor matrices may have full column rank [26]–[28]. In [29], the rank-1 or rank-R detection mapping was further generalized to higher order, in an implicit way, for the construction of the matrix in equation (2.8) of [29]. By exploiting the structure of the null space of this matrix, new algebraic CPD algorithms and more relaxed uniqueness conditions can be derived [29]. A review that attempts to demystify CPD uniqueness results can be found in *Section IV* of the recent review paper [30], and may be a good starting point for laymen.

For DC-CPD, the coupled rank-1 detection mapping in [23] can be considered as the coupled version of the rank-1 detection mapping in [25], while the *K*th-order coupled rank-1 detection mapping, as will be introduced in (8), is a generalization of it to higher order. Therefore, by following a line of thought that has proven successful in the study of CPD, in this paper we aim to provide more insights into the uniqueness and algebraic computation of DC-CPD, a new coupled tensor decomposition model that has not yet been fully understood.

The rest of the paper is organized as follows. In Section II, we explain the DC-CPD formulation and basic assumptions. In Section III, we review the algebraic DC-CPD algorithm and the corresponding uniqueness conditions in [23]. In Section IV, we present the new deterministic and generic DC-CPD uniqueness conditions. In Section V we present the new framework for the algebraic computation of DC-CPD. In Section VI, simulation results are provided to demonstrate the performance of the proposed DC-CPD algorithm, in comparison with other DC-CPD and CPD algorithms. Section VII concludes this paper.

Notation: Vectors, matrices and tensors are denoted by lowercase boldface, uppercase boldface and uppercase calligraphic letters, respectively. The rth column vector and the (i, j)th entry of $\boldsymbol{A}$ are denoted by $\boldsymbol{a}_r$ and $a_{i,j}$, respectively. The identity matrix, all-zero matrix and all-zero vectors are denoted by $\boldsymbol{I}_M \in \mathbb{C}^{M\times M}$, $\boldsymbol{0}_{M,N} \in \mathbb{C}^{M\times N}$, $\boldsymbol{0}_M \in \mathbb{C}^M$, respectively. Subscripts are omitted if there is no ambiguity. The right null space of a matrix $\boldsymbol{M}$ is denoted as ker($\boldsymbol{M}$). The dimension of a vector space $\mathfrak{I}$ is denoted as dim($\mathfrak{I}$). Transpose, conjugate, conjugated transpose, Moore-Penrose pseudo-inverse, Frobenius norm and matrix determinant are denoted as $(\cdot)^T$, $(\cdot)^*$, $(\cdot)^H$, $(\cdot)^\dagger$, $\|\cdot\|_F$, and $|\cdot|$, respectively.

The symbols ‘ $\otimes$ ’, ‘ $\odot$ ’ and ‘ $\circledast$ ’ denote Kronecker product, Khatri-Rao product, and tensor outer product, respectively, defined as follows:

$$\boldsymbol{A}\otimes\boldsymbol{B} \triangleq \begin{bmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \cdots \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, \boldsymbol{A}\odot\boldsymbol{B} \triangleq [\boldsymbol{a}_1\otimes\boldsymbol{b}_1, \boldsymbol{a}_2\otimes\boldsymbol{b}_2, \cdots], \ (\boldsymbol{a}\circledast\boldsymbol{b}\circledast\boldsymbol{c})_{i,j,k} \triangleq a_i b_j c_k .$$

We denote $\odot_{u\in\Omega}\boldsymbol{B}^{(u)}$ and $\otimes_{u\in\Omega}\boldsymbol{B}^{(u)}$ as the Khatri-Rao product and Kronecker product of all the matrices $\boldsymbol{B}^{(u)}$ with $u\in\Omega$, respectively. MATLAB notation will be used to denote submatrices of a tensor. For instance, we use $(\mathcal{T})_{(:,:,k)}$ or $\boldsymbol{T}_{(:,:,k)}$ to denote the frontal slice of tensor $\mathcal{T}$ obtained by fixing the third index to k. A polyadic decomposition (PD) of $\mathcal{T}$ expresses $\mathcal{T}$ as the sum of rank-1 terms:

$$\mathcal{T} = [\![\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}]\!]_R \triangleq \sum\nolimits_{r=1}^{R} \boldsymbol{a}_r \otimes \boldsymbol{b}_r \otimes \boldsymbol{c}_r \in \mathbb{C}^{I\times J\times K}, \tag{1}$$

where $\boldsymbol{A} \triangleq [\boldsymbol{a}_1,...,\boldsymbol{a}_R] \in \mathbb{C}^{I\times R}, \boldsymbol{B} \triangleq [\boldsymbol{b}_1,...,\boldsymbol{b}_R] \in \mathbb{C}^{J\times R}$, and $\boldsymbol{C} \triangleq [\boldsymbol{c}_1,...,\boldsymbol{c}_R] \in \mathbb{C}^{K\times R}$. We call (1) a canonical PD (CPD) if R is minimal.

For an Nth-order tensor $\mathcal{T} \in \mathbb{C}^{I_1\times\cdots\times I_N}$, $\text{vec}(\mathcal{T}) \in \mathbb{C}^{I_1\cdots I_N}$ denotes the vector representation of $\mathcal{T}$ defined by $[\text{vec}(\mathcal{T})]_{\tilde{i}} \triangleq t_{i_1,...,i_N}$, with $\tilde{i} = \sum_{n=1}^{N}(i_n - 1)\prod_{m=1}^{N-n} I_m + 1$, while unvec($\cdot$) performs the inverse. The mode-i matrix representation of a third-order tensor $\mathcal{T} \in \mathbb{C}^{I\times J\times K}$ is denoted as $\boldsymbol{T}_i$, i = 1, 2, 3, and defined by:

$$\left(\boldsymbol{T}_1\right)_{(j-1)K+k,i} = \left(\boldsymbol{T}_2\right)_{(i-1)K+k,j} = \left(\boldsymbol{T}_3\right)_{(i-1)J+j,k} = t_{i,j,k}.$$

We define $\text{Ten}(\boldsymbol{T},[I_1,...,I_{K-1},J]) = \mathcal{T}$ as the operation to reshape a $\prod_{k=1}^{K-1} I_k \times J$ matrix $\boldsymbol{T}$ into a Kth-order tensor $\mathcal{T}$ of size $I_1 \times\cdots\times I_{K-1} \times J$, such that:

$$t_{i_1,...,i_{K-1},j} = \left(\boldsymbol{T}\right)_{\tilde{i},j}, \text{where } \tilde{i} \triangleq \sum\nolimits_{k=1}^{K-1}(i_k - 1)\prod\nolimits_{m=1}^{K-k-1} I_m + 1.$$

The operator $\text{perm}_{\boldsymbol{p}}(\mathcal{T})$ permutes the index of a tensor $\mathcal{T}$ according to a permutation vector $\boldsymbol{p}$, such that the $\boldsymbol{p}(i)$th index of $\mathcal{T}$ is permuted to the ith index of $\text{perm}_{\boldsymbol{p}}(\mathcal{T})$. For instance, $\text{perm}_{(2,1,3)}(\mathcal{T})$ permutes the first and second indices of $\mathcal{T}$. Concatenation of tensors $\mathcal{T}_1 \in \mathbb{C}^{I\times J\times K_1},...,\mathcal{T}_N \in \mathbb{C}^{I\times J\times K_N}$ in the last mode is denoted as $\mathcal{T} \triangleq \text{cat}(\mathcal{T}_1,...,\mathcal{T}_N) \in \mathbb{C}^{I\times J\times(K_1+\cdots+K_N)}$.

The number of k-permutations from a set of m elements is defined as $\mathrm{P}_m^k \triangleq m\cdot(m-1)\cdots(m-k+1)$, and the number of k-combinations from a set of m elements is defined as $\mathrm{C}_m^k \triangleq \mathrm{P}_m^k / \mathrm{P}_k^k$. The cardinality of a set is denoted as card($\cdot$).

II. Problem Formulation

A. Double Coupled Canonical Polyadic Decomposition (DC-CPD)

We say that a set of third-order tensors with varying superscripts m and n, $\{\mathcal{T}^{(m,n)} \in \mathbb{C}^{N\times N\times L}, m,n = 1,...,M\}$, together admits an R-term coupled PD (DC-PD) if the following equation holds for each tensor $\mathcal{T}^{(m,n)}$:

$$\mathcal{T}^{(m,n)} = \sum\nolimits_{r=1}^{R} \boldsymbol{a}_r^{(m)} \otimes \boldsymbol{a}_r^{(n)*} \otimes \boldsymbol{c}_r^{(m,n)} = [\![\boldsymbol{A}^{(m)},\boldsymbol{A}^{(n)*},\boldsymbol{C}^{(m,n)}]\!]_R, \tag{2}$$

where $\boldsymbol{A}^{(m)} \triangleq [\boldsymbol{a}_1^{(m)},...,\boldsymbol{a}_R^{(m)}] \in \mathbb{C}^{N\times R}$, $\boldsymbol{C}^{(m,n)} \triangleq [\boldsymbol{c}_1^{(m,n)},...,\boldsymbol{c}_R^{(m,n)}] \in \mathbb{C}^{L\times R}$. The rank-1 terms with fixed r and varying m and n, $\{\boldsymbol{a}_r^{(m)} \otimes \boldsymbol{a}_r^{(n)*} \otimes \boldsymbol{c}_r^{(m,n)},\ m,n = 1,...,M\}$, are together denoted as a double coupled rank-1 term. The term "double coupled" is used to indicate the fact that each PD in a DC-PD shares factor matrices with other PDs in the first two modes. That is to say, if we place the tensors $\mathcal{T}^{(m,n)}$ at different nodes of a grid according to their indices (see Fig. 1), the tensors in the same "row" (e.g. the *m*th row) are coupled in the first mode (by $\boldsymbol{A}^{(m)}$), and all the tensors in the same "column" (e.g. the *n*th column) are coupled in the second mode (by $\boldsymbol{A}^{(n)*}$). In addition, if the number of double coupled rank-1 terms in (2), *R*, is minimal, then the DC-PD is denoted as DC-CPD. The parameter *R* is then defined as the double coupled rank of tensors $\{\boldsymbol{\mathcal{T}}^{(m,n)} \in \mathbb{C}^{N\times N\times L}, m,n=1,...,M\}$.

The double coupled rank-1 terms $\{\boldsymbol{a}_r^{(m)} \otimes \boldsymbol{a}_r^{(n)*} \otimes \boldsymbol{c}_r^{(m,n)},\ m,n=1,...,M\}$ in (2) can be arbitrarily permuted and the vectors $\boldsymbol{a}_r^{(m)}, \boldsymbol{a}_r^{(n)*}$ and $\boldsymbol{c}_r^{(m,n)}$ with fixed *r*, *m*, *n* can be arbitrarily scaled provided the overall double coupled rank-1 term remains the same. We say that the DC-CPD is unique when it is only subject to these trivial indeterminacies.

DC-CPD can be obtained from multi-set data in J-BSS via the use of second-order statistics [2]–[7], [23]. More precisely, we consider the following multi-set data model:

$$\boldsymbol{x}^{(m)}(t) = \boldsymbol{A}^{(m)}\boldsymbol{s}^{(m)}(t), \qquad m=1,...,M, \tag{3}$$

where $\boldsymbol{x}^{(m)}(t) \in \mathbb{C}^N$, $\boldsymbol{s}^{(m)}(t) \in \mathbb{C}^R$, $\boldsymbol{A}^{(m)} \in \mathbb{C}^{N\times R}$ denote the observed signal, source signal and mixing matrix for the *m*th dataset, respectively. We assume that the sources are temporally nonstationary with zero mean and unit variance, and that $s_r^{(m)}(t)$ and $s_u^{(n)}(t)$ are independent for $1 \le r \ne u \le R$ (intra-set independence) and dependent for $1 \le r = u \le R$ (inter-set dependence), regardless of the value of *m* and *n*. Then we can construct a set of covariance tensors $\{\boldsymbol{\mathcal{T}}^{(m,n)} \in \mathbb{C}^{N\times N\times L}, m,n=1,...,M\}$ as follows:

$$(\boldsymbol{\mathcal{T}}^{(m,n)})_{(:,:,l)} = \mathrm{E}\{\boldsymbol{x}^{(m)}(t)[\boldsymbol{x}^{(n)}(t)]^H\} = \boldsymbol{A}^{(m)}\,\mathrm{E}\{\boldsymbol{s}^{(m)}(t)[\boldsymbol{s}^{(n)}(t)]^H\}\boldsymbol{A}^{(n)H}, \tag{4}$$

where the frontal slice $(\boldsymbol{\mathcal{T}}^{(m,n)})_{(:,:,l)}$ is the cross-covariance matrix between the *m*th and *n*th dataset at time instant *l*, and *L* denotes the number of time frames for which such a cross-covariance is computed. Since $\mathrm{E}\{\boldsymbol{s}^{(m)}(t)\cdot[\boldsymbol{s}^{(n)}(t)]^H\}$ is diagonal under the assumption of intra-set independence and inter-set dependence, we can rewrite (4) as:

$$\boldsymbol{\mathcal{T}}^{(m,n)} = \sum_{r=1}^{R} \boldsymbol{a}_r^{(m)} \otimes \boldsymbol{a}_r^{(n)*} \otimes \boldsymbol{c}_r^{(m,n)} = [\![\boldsymbol{A}^{(m)}, \boldsymbol{A}^{(n)*}, \boldsymbol{C}^{(m,n)}]\!]_R, \tag{5}$$

where $\boldsymbol{c}_r^{(m,n)}$ denotes the *r*th column of the third factor matrix: $[\boldsymbol{c}_1^{(m,n)},...,\boldsymbol{c}_R^{(m,n)}] \triangleq \boldsymbol{C}^{(m,n)}$, which holds in its (*r*, *l*)th entry the cross-covariance coefficient between $s_r^{(m)}(t)$ and $s_r^{(n)}(t)$ at the *l*th time frame: $(\boldsymbol{C}^{(m,n)})_{r,l} = \mathrm{E}\{s_r^{(m)}(l)[s_r^{(n)}(l)]^H\}$. Therefore, with *m*, *n* varying and assuming that *R* is minimal (which is usually the case in practice), all the tensors $\boldsymbol{\mathcal{T}}^{(m,n)}$ together admit a DC-CPD.

B. Basic assumptions

Assumption 1: we assume that the DC-CPD (2) has a conjugated symmetric structure: $\boldsymbol{\mathcal{T}}^{(m,n)} = \mathrm{perm}_{(2,1,3)}(\boldsymbol{\mathcal{T}}^{(n,m)*})$.

In cases where this symmetry is not readily present, we are able to create it by tensor concatenation. The tensor concatenation scheme is explained in details in [23] and thus is not further addressed here.

Because of the conjugated symmetry in DC-CPD, it suffices to consider the following set $\Upsilon^{(m)}$ of tensors to take into account all occurrences of $\boldsymbol{A}^{(m)}$, for fixed m:

$$\Upsilon^{(m)} \triangleq \left\{ \mathcal{T}^{(m,n)} = \left[\!\left[\boldsymbol{A}^{(m)}, \boldsymbol{A}^{(n)*}, \boldsymbol{C}^{(m,n)} \right]\!\right]_R, n = 1, \ldots, M \right\}. \tag{6}$$

Note that in (6), seen as C-CPD, occurrences of $\boldsymbol{A}^{(m)*}$ in the second mode are automatically considered because of the conjugated symmetry.

Assumption 2: we assume that $\boldsymbol{A}^{(m)} \odot \boldsymbol{A}^{(n)*}$ has full column rank, for all the indices m, n. This is a necessary condition for DC-CPD to be unique. In fact, if $\boldsymbol{A}^{(m)} \odot \boldsymbol{A}^{(n)*}$ does not have full column rank, a decomposition in a smaller number of terms is possible. E.g., if $\boldsymbol{a}_R^{(m)} \otimes \boldsymbol{a}_R^{(n)*} = \sum_{r=1}^{R-1} \alpha_r^{(m,n)} \boldsymbol{a}_r^{(m)} \otimes \boldsymbol{a}_r^{(n)*}$, then $\mathcal{T}^{(m,n)} = \sum_{r=1}^{R-1} \boldsymbol{a}_r^{(m)} \otimes \boldsymbol{a}_r^{(n)*} \otimes (\boldsymbol{c}_r^{(m,n)} + \alpha_r^{(m,n)} \boldsymbol{c}_R^{(m,n)})$.

Assumption 3: we assume that $\boldsymbol{C}^{(m,n)}$ has full column rank, for all the indices m, n. Noting that the mode-3 matrix representation can be expressed as $\boldsymbol{T}_3^{(m,n)} = (\boldsymbol{A}^{(m)} \odot \boldsymbol{A}^{(n)*}) \boldsymbol{C}^{(m,n)T}$, this assumption, together with *Assumption 2*, implies that the rank of tensor $\mathcal{T}^{(m,n)}$ is equal to the rank of $\boldsymbol{T}_3^{(m,n)}$. This is useful in practice in the sense that the number of sources can be determined by checking the number of significant singular values of $\boldsymbol{T}_3^{(m,n)}$. On the other hand, if $\boldsymbol{C}^{(m,n)}$ does not have full column rank, decomposition (2) may still be unique (e.g., algebraic algorithms for CPD have been derived for cases where none of the factor matrices has full column rank [28], [29]), while the rank of tensor $\mathcal{T}^{(m,n)}$ is not equal to the rank of $\boldsymbol{T}_3^{(m,n)}$. Here we do not consider this more difficult case. Moreover, we make the notational assumption that factor matrices $\boldsymbol{C}^{(m,n)}$ in the third mode have size $R \times R$. In practice, this can always be achieved by a dimensionality reduction step: if the columns of a matrix $\boldsymbol{U}^{(m,n)}$ form an orthonormal basis of the row space of $\boldsymbol{T}_3^{(m,n)}$, then $\boldsymbol{T}_3^{(m,n)} \boldsymbol{U}^{(m,n)}$ has R columns, tensor $\mathrm{Ten}(\boldsymbol{T}_3^{(m,n)} \boldsymbol{U}^{(m,n)}, I, I, R)$ has reduced dimensionality R in the third mode, and its factor matrices are $\boldsymbol{A}^{(m)}$, $\boldsymbol{A}^{(n)*}$, and $\boldsymbol{U}^{(m,n)T} \boldsymbol{C}^{(m,n)} \in \mathbb{C}^{R \times R}$.

Remark 1: The factor matrices $\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(M)}$ need not have full column rank. Generically, this implies that $\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(M)}$ can have more columns than rows: $N < R$. DC-CPD under this assumption is called underdetermined. We note that the underdetermined DC-CPD is of particular interest in underdetermined J-BSS with more sources than observation channels, a problem that may appear in a number of practical applications.

DC-CPD with $\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(M)}$ all having full column rank is called overdetermined, and an overdetermined DC-CPD directly admits a GEVD based algebraic solution. More precisely, any of the CPDs in (2) can be computed by GEVD [31]. This yields us at least one factor matrix up to trivial scaling and permutation indeterminacies. Assume for instance that $\boldsymbol{A}^{(1)}$ has been found, then we can find all the remaining $\boldsymbol{A}^{(n)}$, $n = 2,\ldots,M$, from the CPD of $\boldsymbol{\mathcal{T}}^{(1,n)}$ for varying n, with a known factor matrix $\boldsymbol{A}^{(1)}$. Note that CPD with a known factor matrix of full column rank can be calculated via a rank-1 approximation based scheme, as explained in [32].

Note that the above are just basic assumptions in our problem definition. Under these assumptions, DC-CPD uniqueness is not yet guaranteed. In this paper, we will develop a broad framework for the algebraic computation of DC-CPD, and present new uniqueness conditions that are more relaxed than those given in [23].

III. Review of Algebraic DC-CPD in [23]

To make our derivation of the proposed framework more accessible, in this section, we briefly review the algebraic DC-CPD algorithm proposed in [23], as well as the corresponding uniqueness conditions. The algebraic DC-CPD algorithm in [23] is derived following a similar line of thought as the algebraic algorithms in [16], [17], [25]–[29], which, by the use of a rank-1 or rank-R detection mapping, convert a possibly underdetermined (coupled) CPD into an overdetermined CPD. It uses a so-called coupled rank-1 detection mapping, to convert an underdetermined DC-CPD into an overdetermined DC-CPD, which can be solved algebraically. Here we give a high-level summary of the results in [23]. For more details, the reader is referred to [23].

A. Second-order Coupled Rank-1 Detection Mapping and Uniqueness Conditions

The second-order coupled rank-1 detection mapping $\Phi_2 : (\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}) \in \mathbb{C}^{N\times P} \times \mathbb{C}^{N\times Q} \to \Phi_2(\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}) \in \mathbb{C}^{N\times N\times P\times Q}$, as proposed in [23], is defined as:

$$\left[\Phi_2(\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)})\right]_{i,j,p,q} \triangleq \begin{vmatrix} x_{i,p}^{(1)} & x_{i,q}^{(2)} \\ x_{j,p}^{(1)} & x_{j,q}^{(2)} \end{vmatrix} = x_{i,p}^{(1)} x_{j,q}^{(2)} - x_{j,p}^{(1)} x_{i,q}^{(2)}. \tag{7}$$

Two main properties of the second-order coupled rank-1 detection mapping are: (a) it is bilinear in its arguments: $\Phi_2(\sum_u \alpha_u \boldsymbol{X}_u^{(1)}, \sum_v \beta_v \boldsymbol{X}_v^{(2)}) = \sum_u \sum_v \alpha_u \beta_v \Phi_2(\boldsymbol{X}_u^{(1)}, \boldsymbol{X}_v^{(2)})$; (b) for two non-zero matrices $\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}$, $\Phi_2(\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)})$ is a zero tensor if and only if $\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}$ are two rank-1 matrices with identical column space (*Theorem 1* in [23]). The vector representation of $\Phi_2(\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)})$ is denoted as $\psi_2(\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}) \triangleq \operatorname{vec}(\Phi_2(\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}))$.

In [23], we have provided some deterministic uniqueness conditions, under which DC-CPD is unique and can be calculated algebraically. For convenience, we repeat them here as *Theorem 1*:

Theorem 1: Let $\boldsymbol{\mathcal{T}}^{(m,j)} = [\![\boldsymbol{A}^{(m)}, \boldsymbol{A}^{(j)*}, \boldsymbol{C}^{(m,j)}]\!]_R$, where $j \in \{g,h\}$. Assume that $\boldsymbol{C}^{(m,g)}, \boldsymbol{C}^{(m,h)}$ have full column rank for all $1 \leq g < h \leq M, m = 1,...,M$, and that $\psi_2(\boldsymbol{a}_t^{(m)}\boldsymbol{a}_t^{(g)H}, \boldsymbol{a}_r^{(m)}\boldsymbol{a}_r^{(h)H}) = (\boldsymbol{a}_t^{(m)} \otimes \boldsymbol{a}_r^{(m)} - \boldsymbol{a}_r^{(m)} \otimes \boldsymbol{a}_t^{(m)}) \otimes \boldsymbol{a}_t^{(g)*} \otimes \boldsymbol{a}_r^{(h)*}$ are linearly independent for $1 \leq t \neq r \leq R$. Then we have:

- The tensors $\{\mathcal{T}^{(m,n)}, m,n = 1,...,M\}$ admit a DC-CPD, i.e. they consist of the sum of R double coupled rank-1 terms, and the number of terms cannot be reduced.

- The DC-CPD is unique.

- The DC-CPD can be calculated algebraically.

We note that the conditions in *Theorem 1* are very mild, and even more relaxed than those for CPD, as shown in [23].

B. Algebraic DC-CPD algorithm in [23]

To compute the DC-CPD of tensors $\{\mathcal{T}^{(m,n)} = [\![\boldsymbol{A}^{(m)}, \boldsymbol{A}^{(n)*}, \boldsymbol{C}^{(m,n)}]\!]_R \in \mathbb{C}^{N \times N \times R}, m,n = 1,...,M\}$, where $\boldsymbol{A}^{(m)} \in \mathbb{C}^{N \times R}$, and $\boldsymbol{C}^{(m,n)} \in \mathbb{C}^{R \times R}$, the algebraic algorithm in [23] consists of the following main steps (note that only $\boldsymbol{C}^{(m,n)}$ is required to have full column rank, $\boldsymbol{A}^{(1)},...,\boldsymbol{A}^{(M)}$ can be rank-deficient, and we can allow $N < R$):

(*i*) For each pair of tensors $\mathcal{T}^{(m,g)}, \mathcal{T}^{(m,h)} \in \Upsilon^{(m)}$, we apply the second-order coupled rank-1 detection mapping (7) to the sth frontal slice of the former, and the uth frontal slice of the latter, to construct a tensor $\Phi_2(\boldsymbol{T}_{(:,:,s)}^{(m,g)}, \boldsymbol{T}_{(:,:,u)}^{(m,h)})$. The vector representation $\psi_2(\boldsymbol{T}_{(:,:,s)}^{(m,g)}, \boldsymbol{T}_{(:,:,u)}^{(m,h)})$ is of length N^4. We have R^2 such vectors for varying s and u, $1 \leq s,u \leq R$. We stack these vectors into the columns of an $N^4 \times R^2$ matrix $\boldsymbol{\Gamma}^{(m,g,h)}$.

(*ii*) Under the conditions in *Theorem 1*, i.e., the vectors $\psi_2(\boldsymbol{a}_t^{(m)}\boldsymbol{a}_t^{(g)H}, \boldsymbol{a}_r^{(m)}\boldsymbol{a}_r^{(h)H})$ are linearly independent for $1 \leq t \neq r \leq R$, we have: $\dim(\ker(\boldsymbol{\Gamma}^{(m,g,h)})) = R$, i.e. the number of sources can be found as the dimension of a matrix null space. Moreover, the R basis vectors $\boldsymbol{w}_1^{(m,p,q)},...,\boldsymbol{w}_R^{(m,p,q)}$ in $\ker(\boldsymbol{\Gamma}^{(m,g,h)})$ can be reshaped into a tensor that admits a non-symmetric overdetermined third-order CPD $\mathcal{W}^{(m,p,q)} = [\![\boldsymbol{B}^{(m,g)}, \boldsymbol{B}^{(m,h)}, \boldsymbol{F}^{(m,g,h)}]\!]_R$, where factor matrix $\boldsymbol{B}^{(m,j)} = (\tilde{\boldsymbol{C}}^{(m,j)})^{-T}$, with $\tilde{\boldsymbol{C}}^{(m,j)}$ equivalent to $\boldsymbol{C}^{(m,j)}$ up to scaling and permutation ambiguities, $j \in \{g,h\}$. For the overall DC-CPD (6), we perform the same procedure for all pairs of coupled tensors and obtain a set of overdetermined CPDs. With varying indices (m, g, h), $m = 1,\ldots,M$, $1 \leq g < h \leq M$, the tensors $\mathcal{W}^{(m,p,q)}$ together admit an overdetermined third-order DC-CPD with coupling in the first two modes.

(*iii*) As the new DC-CPD $\{\mathcal{W}^{(m,p,q)} = [\![\boldsymbol{B}^{(m,g)}, \boldsymbol{B}^{(m,h)}, \boldsymbol{F}^{(m,g,h)}]\!]_R, 1 \leq m \leq M, 1 \leq g < h \leq M\}$ is overdetermined, it admits a GEVD based algebraic solution. For the matrix pencil we may take two frontal slices of $\mathcal{W}^{(m,p,q)}$. Using all the frontal slices we can also compute the factor matrices via simultaneous diagonalization (SD). Optimization based DC-CPD algorithms such as alternating least squares (ALS), or nonlinear least squares (NLS) can be used to explicitly maximize the fit, i.e. to minimize the cost function $\sum_{m,p,q}\|\mathcal{W}^{(m,p,q)} - [\![\boldsymbol{B}^{(m,g)}, \boldsymbol{B}^{(m,h)}, \boldsymbol{F}^{(m,g,h)}]\!]_R\|$. Once $\boldsymbol{B}^{(m,n)}$ has been computed, we immediately obtain $\tilde{\boldsymbol{C}}^{(m,n)} = \boldsymbol{B}^{(m,n)-T}$ and $\tilde{\boldsymbol{A}}^{(m)} \odot \tilde{\boldsymbol{A}}^{(n)*} = \boldsymbol{T}_3^{(m,n)}\boldsymbol{B}^{(m,n)}$ for all $m,n = 1,...,M$. Therefore, the remaining factor matrices $\boldsymbol{A}^{(1)},...,\boldsymbol{A}^{(M)}$ can be recovered by exploiting the Kronecker product structure of each column of $\boldsymbol{T}_3^{(m,n)}\boldsymbol{B}^{(m,n)}$.

Note that in this algebraic DC-CPD algorithm, the second-order coupled rank-1 detection mapping operates on one pair of coupled tensors at a time. In the next section, we will generalize the second-order coupled rank-1 detection mapping (7) to arbitrary order $K \geq 2$, as will be given in (8) and (9), so that each time it can exploit the coupling structure of K tensors. Based on this new coupled rank-1 detection mapping, new uniqueness results will be presented in Section IV, and a broad framework of algebraic DC-CPD algorithms will be developed in Section V.

IV. New Uniqueness Conditions

A. *Coupled rank-1 detection mapping of order $K \geq 2$*

We give the following two definitions and a theorem.

Definition 1: Matrices $\boldsymbol{X}^{(1)} \in \mathbb{C}^{N\times J_1},...,\boldsymbol{X}^{(K)} \in \mathbb{C}^{N\times J_K}$ are said to be coupled rank-1 matrices if they are all rank-1 and have the same column space.

Definition 2: The Kth-order coupled rank-1 detection mapping $\Phi_K^{(v,w)} : (\boldsymbol{X}^{(1)},...,\boldsymbol{X}^{(K)}) \in \mathbb{C}^{N\times J_1} \times \cdots \times \mathbb{C}^{N\times J_K} \to \Phi_K^{(v,w)}(\boldsymbol{X}^{(1)},...,\boldsymbol{X}^{(K)}) \in \mathbb{C}^{N\times\cdots\times N\times J_1\times\cdots\times J_K}$ is defined as:

$$\left[\Phi_K^{(v,w)}(\boldsymbol{X}^{(1)},...,\boldsymbol{X}^{(K)})\right]_{i_1\cdots i_K, j_1\cdots j_K} \triangleq \begin{vmatrix} x_{i_v,j_v}^{(v)} & x_{i_v,j_w}^{(w)} \\ x_{i_w,j_v}^{(v)} & x_{i_w,j_w}^{(w)} \end{vmatrix} \prod_{u\notin\{v,w\}} x_{i_u,j_u}^{(u)}, \tag{8}$$

with $1 \leq v < w \leq K$. The number of arguments of $\Phi_K^{(v,w)}$, K, is called the order, and the superscript (v, w) is called the index of the coupled rank-1 detection mapping, respectively.

Theorem 2: For K nonzero matrices $\boldsymbol{X}^{(k)} \in \mathbb{C}^{N\times J_k}$, k = 1,…,K, consider the vector $\psi(\boldsymbol{X}^{(1)},...,\boldsymbol{X}^{(K)})$ of length $\mathrm{C}_K^2\, N^K \prod_{k=1}^K J_k$ obtained via the Kth-order coupled rank-1 detection mapping (8):

$$\psi(\boldsymbol{X}^{(1)},...,\boldsymbol{X}^{(K)}) \triangleq \begin{bmatrix} \mathrm{vec}\left(\Phi_K^{(1,2)}(\boldsymbol{X}^{(1)},...,\boldsymbol{X}^{(K)})\right) \\ \mathrm{vec}\left(\Phi_K^{(1,3)}(\boldsymbol{X}^{(1)},...,\boldsymbol{X}^{(K)})\right) \\ \vdots \\ \mathrm{vec}\left(\Phi_K^{(K-1,K)}(\boldsymbol{X}^{(1)},...,\boldsymbol{X}^{(K)})\right) \end{bmatrix}. \tag{9}$$

Then $\psi(\boldsymbol{X}^{(1)},...,\boldsymbol{X}^{(K)})$ is a zero vector if and only if its arguments $\boldsymbol{X}^{(1)},...,\boldsymbol{X}^{(K)}$ are coupled rank-1 matrices.

The proof of *Theorem 2* is given in the *Appendix*. We note that the above definitions and theorem generalize the second-order coupled rank-1 detection mapping (7) and relevant results.

B. Deterministic and generic uniqueness conditions

For a chosen constant $K \in [2, M]$, we formulate a theorem that provides deterministic conditions for which DC-CPD is unique, and for which it can actually be calculated algebraically.

Theorem 3: Let $\boldsymbol{\mathcal{T}}^{(m,n_k)} = [\![\boldsymbol{A}^{(m)}, \boldsymbol{A}^{(n_k)*}, \boldsymbol{C}^{(m,n_k)}]\!]_R \in \Upsilon^{(m)}$, where $m, n_k = 1,...,M$, $k = 1,...,K$, and $K \in [2, M]$ is a preset constant. We assume, for all values $1 \leq n_1 < \cdots < n_K \leq M$ and $m = 1,...,M$, that the matrices $\boldsymbol{C}^{(m,n_k)}$ and $\boldsymbol{\Phi}^{(m)}_{n_1,...,n_K}$ have full column rank. Here $\boldsymbol{\Phi}^{(m)}_{n_1,...,n_K}$ holds vectors $\psi(\boldsymbol{a}^{(m)}_{r_1}\boldsymbol{a}^{(n_1)H}_{r_1},...,\boldsymbol{a}^{(m)}_{r_K}\boldsymbol{a}^{(n_K)H}_{r_K})$ as its columns. These columns are indexed by $(r_1,...,r_K) \in \Theta^{(K)}$, where $\Theta^{(K)}$ is the set of tuples $(r_1,...,r_K)$, $1 \leq r_1 \leq \cdots \leq r_K \leq R$, in which each tuple has at least two distinct elements. Then we have:

- The tensors $\boldsymbol{\mathcal{T}}^{(m,n)}$, for varying $m, n = 1,...,M$, admit a DC-CPD, i.e. they consist of the sum of *R* double coupled rank-1 terms, and the number of terms cannot be reduced.
- The DC-CPD is unique.
- The DC-CPD can be calculated algebraically.

Note that *Theorem 3* generalizes the corresponding result in [23], as stated in *Theorem 1*, to the case $K \in [2, M]$. The derivation of the algebraic algorithm, as will be presented in *Section V*, provides a constructive proof of *Theorem 3*.

Remark 2: Theorem 3 provides *deterministic* uniqueness conditions under which DC-CPD can be calculated algebraically for $K \in [2, M]$. We can also provide the value of the upper bound of *R*, denoted as $R_{\max}$, for which the conditions in *Theorem 3* hold in the *generic* sense. We call a property generic if it holds with probability one, when the parameters it involves are drawn from continuous probability densities. We note that the generic version of the uniqueness conditions in *Theorem 3* depends both on *N* and *K*. It also depends on the third dimension of the data tensors, as the third factor matrix $\boldsymbol{C}^{(m,n)}$ is required to have full column rank, implying that $\boldsymbol{C}^{(m,n)}$ should have not fewer rows than columns, i.e., $L \geq R$ needs to hold.

Instead of deriving a general closed-form expression of $R_{\max}$, which is at the moment out of reach, we list in TABLE I the generic value of $R_{\max}$ for different values of N, for $K = 2$ and $K = 3$, respectively. For comparison, we also list the value of $R_{\max}$ for CPD of a single generic tensor of size $N \times N \times L$, $L = R$, for $K = 2$ [25] and $K = 3$ [27], respectively[1].

The numerical values of $R_{\max}$ can be easily obtained using *Theorem 3* and Fisher's lemma (Corollary in p.10 of [34]). Fisher's lemma implies that, for fixed N and R, DC-CPD is generically unique if we can find one example for which the decomposition is unique. Hence, to calculate the generic value of $R_{\max}$, one only needs to check if the matrix $\boldsymbol{\Phi}_{n_1,\ldots,n_K}^{(m)}$ in *Theorem 3* has full column rank, for a set of randomly generated factor matrices $\boldsymbol{A}^{(1)},\ldots,\boldsymbol{A}^{(M)}$.

TABLE I

GENERIC VALUE OF $R_{\max}$ OF DC-CPD AND CPD ($K = 2, 3$)

N	2	3	4	5	6	7
DC-CPD ($K = 2$)	2	5	10	16	23	32
DC-CPD ($K = 3$)	2	6	11	18	27	38
CPD ($K = 2$)	2	4	9	14	21	30
CPD ($K = 3$)	2	4	9	16	24	34

The values of $R_{\max}$ listed in TABLE I are those for which DC-CPD can generically be uniquely computed via the algebraic algorithm (for $K = 2, 3$) that will be proposed in *Section V*. They are different from the theoretical algorithm-independent upper bound of R that is inherent to the DC-CPD model, and actually offer a lower bound for the latter.

We note that the uniqueness condition depends on the preset constant K. We have generally observed that increasing K relaxes the uniqueness condition of DC-CPD. For instance, TABLE I shows that for $K = 3$, a more relaxed generic uniqueness condition is obtained than for $K = 2$. We have also observed that the uniqueness condition of DC-CPD for $K = 2$ and $K = 3$ is more relaxed than that of CPD for $K = 2$ and $K = 3$, respectively. Instead of giving a formal proof, we limit ourselves to the following explanation. First, recall that $\boldsymbol{\Phi}_{n_1,\ldots,n_K}^{(m)}$ has $\psi(\boldsymbol{a}_{r_1}^{(m)}\boldsymbol{a}_{r_1}^{(n_1)H},\ldots,\boldsymbol{a}_{r_K}^{(m)}\boldsymbol{a}_{r_K}^{(n_K)H})$ as its columns, and (9) shows that $\psi(\boldsymbol{a}_{r_1}^{(m)}\boldsymbol{a}_{r_1}^{(n_1)H},\ldots,\boldsymbol{a}_{r_K}^{(m)}\boldsymbol{a}_{r_K}^{(n_K)H})$ is obtained by concatenation of sub-vectors $\text{vec}(\Phi_K^{(v,w)}(\boldsymbol{a}_{r_1}^{(m)}\boldsymbol{a}_{r_1}^{(n_1)H},\ldots,\boldsymbol{a}_{r_K}^{(m)}\boldsymbol{a}_{r_K}^{(n_K)H}))$. The latter are obtained using the coupled rank-1 detection mapping defined in (8), $1 \le v < w \le K$. From (8) we have:

[1] The parameter K for CPD is related to parameter l in equation (2.8) of [29] by $K = l + 2$. The other parameter m in equation (2.8) of [29] is fixed to 2 under the condition that at least one of the factor matrices has full column rank. Equation (2.8) of [29], with $m = 2$ and $K = l + 2$, implies the use of a rank-1 detection mapping of order K.

$$\mathrm{vec}(\Phi_K^{(v,w)}(\boldsymbol{a}_{r_1}^{(m)}\boldsymbol{a}_{r_1}^{(n_1)H},\dots,\boldsymbol{a}_{r_K}^{(m)}\boldsymbol{a}_{r_K}^{(n_K)H}))=\left[\left(\bigotimes_{k=1}^{K}\boldsymbol{a}_{r_k}^{(m)}\right)-\left(\bigotimes_{k\in\Xi_{(v,w)}}\boldsymbol{a}_{r_k}^{(m)}\right)\right]\otimes\left(\bigotimes_{l=1}^{K}\boldsymbol{a}_{u_l}^{(n_l)*}\right),$$

where $\Xi_{(v,w)}=\{1,\dots,v-1,w,v+1,\dots,w-1,v,w+1,\dots,K\}$. This equation shows that $\mathrm{vec}(\Phi_K^{(v,w)}(\boldsymbol{a}_{r_1}^{(m)}\boldsymbol{a}_{r_1}^{(n_1)H},\dots,\boldsymbol{a}_{r_K}^{(m)}\boldsymbol{a}_{r_K}^{(n_K)H}))$ is a sum of two vectors that have a Kronecker product structure. One can expect that in matrices with such Kronecker *product* structured columns, the likelihood of *linear* dependencies between columns will decrease as the number of factors 2*K* increases. Second, we note that when *K* increases, the number of rows of $\boldsymbol{\Phi}_{n_1,\dots,n_K}^{(m)}$ increases faster than the number of columns. (Note that the size of $\boldsymbol{\Phi}_{n_1,\dots,n_K}^{(m)}$ is $\mathrm{C}_K^2\ N^{2K}\times(R^K-R)$). Hence, one can again expect a more relaxed uniqueness condition for larger *K*. In particular, the most relaxed uniqueness condition can be expected for *K* = *M*, which implies that the coupled rank-1 detection mapping in (10) involves all the tensors in set $\Upsilon^{(m)}$. On the other hand, the size of $\boldsymbol{\Gamma}^{(m,n_1,\dots,n_K)}$ is $\mathrm{C}_K^2\ N^{2K}\times R^K$, i.e., it depends exponentially on *K* such that increasing *K* increases the computational complexity significantly. For instance, for a DC-CPD with *N* = 6, *R* = 23, and *M* = 3, the matrix $\boldsymbol{\Gamma}^{(m,n_1,\dots,n_K)}$ has size 1296 × 529 for *K* = 2, and has size 139968 × 12167 for *K* = 3. As such, for efficiency of computation, in practice we may limit ourselves to the minimal value of *K* that is needed for the identification of a certain number of sources.

V. Algebraic Framework

In this section, we present a broad algebraic DC-CPD framework, which consists of a series of algebraic algorithms, each corresponding to a fixed order *K*. The algorithm associated with a particular *K* uses the *K*th-order coupled rank-1 detection mapping (8) to transform an underdetermined DC-CPD into a (*K*+1)th-order overdetermined C-CPD with coupling in the first *K* modes. A larger *K* generally relaxes the uniqueness condition, but also increases the computational complexity, as explained in *Section IV*. In practice one may start with *K* = 2, and increase its value if the number of sources that can be handled is not sufficient.

The derivation of the proposed algebraic framework is in analogy with the three steps for DC-CPD with *K* = 2, as explained in *Subsection III.B*. It provides a constructive proof of *Theorem 3* in *Section IV.*

(*i*) *Construct matrix* $\boldsymbol{\Gamma}^{(m,n_1,\dots,n_K)}$ *via Kth-order coupled rank-1 detection.*

We select *K* tensors $\boldsymbol{\mathcal{T}}^{(m,n_1)},\dots,\boldsymbol{\mathcal{T}}^{(m,n_K)}$ from $\Upsilon^{(m)}$ (6), $1\le n_1<\cdots<n_K\le M$. There are C_M^K such *K*-combinations. For each *K*-combination $\boldsymbol{\mathcal{T}}^{(m,n_1)},\dots,\boldsymbol{\mathcal{T}}^{(m,n_K)}\in\Upsilon^{(m)}$, we perform the operation (9) on all the possible combinations of the frontal slices of $\boldsymbol{\mathcal{T}}^{(m,n_1)},\dots,\boldsymbol{\mathcal{T}}^{(m,n_K)}$ to obtain a (*K*+1)th-order tensor $\boldsymbol{\mathcal{G}}^{(m,n_1,\dots,n_K)}$ of size $(\mathrm{C}_K^2\ N^{2K})\times R\times\cdots\times R$:

$$\left[\boldsymbol{\mathcal{G}}^{(m,n_1,\dots,n_K)}\right]_{(:,r_1,\dots,r_K)}\triangleq\psi\left(\boldsymbol{T}_{(:,:,r_1)}^{(m,n_1)},\dots,\boldsymbol{T}_{(:,:,r_K)}^{(m,n_K)}\right),\qquad(10)$$

where $1 \le r_k \le R$, $1 \le k \le K$. Note that operation (10) implicitly uses the Kth-order coupled rank-1 detection mapping (8). We reshape $\boldsymbol{\mathcal{G}}^{(m,n_1,\ldots,n_K)}$ into a $(\mathrm{C}_K^2\, N^{2K}) \times R^K$ matrix $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$ as:

$$\left[\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}\right]_{(l,:)} \triangleq \mathrm{vec}\left(\left[\boldsymbol{\mathcal{G}}^{(m,n_1,\ldots,n_K)}\right]_{(l,:,\ldots,:)}\right)^T, \tag{11}$$

where $1 \le l \le \mathrm{C}_K^2\, N^{2K}$. Due to the multilinearity of ψ, we have:

$$\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)} = \sum_{r_1,\ldots,r_K=1}^{R}\left[\psi(\boldsymbol{a}_{r_1}^{(m)}\boldsymbol{a}_{r_1}^{(n_1)H},\ldots,\boldsymbol{a}_{r_K}^{(m)}\boldsymbol{a}_{r_K}^{(n_K)H})\cdot(\boldsymbol{c}_{r_1}^{(m,n_1)}\otimes\cdots\otimes\boldsymbol{c}_{r_K}^{(m,n_K)})^T\right]. \tag{12}$$

According to *Theorem 2*, $\psi(\boldsymbol{a}_{r_1}^{(m)}\boldsymbol{a}_{r_1}^{(n_1)H},\ldots,\boldsymbol{a}_{r_K}^{(m)}\boldsymbol{a}_{r_K}^{(n_K)H}) = \boldsymbol{0}$ when $r_1 = \cdots = r_K$ and thus (12) can be rewritten as:

$$\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)} = \sum_{(r_1,\ldots,r_K)\in\Theta^{(K)}}\left[\psi(\boldsymbol{a}_{r_1}^{(m)}\boldsymbol{a}_{r_1}^{(n_1)H},\ldots,\boldsymbol{a}_{r_K}^{(m)}\boldsymbol{a}_{r_K}^{(n_K)H})\cdot(\boldsymbol{c}_{r_1}^{(m,n_1)}\otimes\cdots\otimes\boldsymbol{c}_{r_K}^{(m,n_K)})^T\right]. \tag{13}$$

We stack the vectors $\boldsymbol{c}_{r_1}^{(m,n_1)}\otimes\cdots\otimes\boldsymbol{c}_{r_K}^{(m,n_K)}$ into a $R^K \times (R^K - R)$ matrix $\boldsymbol{P}_{n_1,\ldots,n_K}^{(m)}$ and the vectors $\psi(\boldsymbol{a}_{r_1}^{(m)}\boldsymbol{a}_{r_1}^{(n_1)H},\ldots,\boldsymbol{a}_{r_K}^{(m)}\boldsymbol{a}_{r_K}^{(n_K)H})$ into a $\mathrm{C}_K^2\, N^{2K} \times (R^K - R)$ matrix $\boldsymbol{\Phi}_{n_1,\ldots,n_K}^{(m)}$ where these vectors are indexed by tuples $(r_1,\ldots,r_K)\in\Theta^{(K)}$. Then (13) can be rewritten as:

$$\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)} = \boldsymbol{\Phi}_{n_1,\ldots,n_K}^{(m)}\left(\boldsymbol{P}_{n_1,\ldots,n_K}^{(m)}\right)^T. \tag{14}$$

We note that $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$ is the generalization of matrix $\boldsymbol{\Gamma}^{(m,p,q)}$ in step (*i*) of *Subsection III.B* from $K = 2$ to $K \in [2, M]$.

(*ii*) *Construct tensor* $\boldsymbol{\mathcal{W}}^{(m,n_1,\ldots,n_K)}$ *in* $\ker(\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)})$. *For varying* $(m,n_1,\ldots,n_K)$, *these tensors admit a* (K+1)*th-order overdetermined C-CPD with coupling in the first K modes.*

We give a theorem implying that, under the conditions specified in *Theorem 3*, the basis vectors in the null space of $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$ can be reshaped into a tensor that admits an overdetermined (K+1)th-order CPD.

Theorem 4: Let $\boldsymbol{\mathcal{T}}^{(m,n_k)} = [\![\boldsymbol{A}^{(m)}, \boldsymbol{A}^{(n_k)*}, \boldsymbol{C}^{(m,n_k)}]\!]_R \in \Upsilon^{(m)}$, $k = 1,\ldots,K$, $1 \le n_1 < \cdots < n_K \le M$, and assume that for fixed $(m, n_1,\ldots,n_K)$ and for all values of k, $\boldsymbol{\Phi}_{n_1,\ldots,n_K}^{(m)}$ in (14) and $\boldsymbol{C}^{(m,n_k)}$ have full column rank. Then we have:

- $\ker(\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}) = \ker((\boldsymbol{P}_{n_1,\ldots,n_K}^{(m)})^T)$.
- $\dim(\ker(\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)})) = R$.
- The basis vectors $\boldsymbol{w}_r^{(m,n_1,\ldots,n_k)}$ in $\ker(\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)})$, $r = 1,\ldots,R$, can be written as linear combinations of the vectors $\boldsymbol{b}_u^{(m,n_1)}\otimes\cdots\otimes\boldsymbol{b}_u^{(m,n_K)}$, $u = 1,\ldots,R$:

$$\boldsymbol{w}_r^{(m,n_1,\ldots,n_K)} = \sum\nolimits_{u=1}^{R} f_{r,u}^{(m,n_1,\ldots,n_K)}\cdot\boldsymbol{b}_u^{(m,n_1)}\otimes\cdots\otimes\boldsymbol{b}_u^{(m,n_K)} \in \mathbb{C}^{R^K}, \tag{15}$$

where $[\boldsymbol{b}_1^{(m,n_k)},...,\boldsymbol{b}_R^{(m,n_k)}] = \boldsymbol{B}^{(m,n_k)} = \tilde{\boldsymbol{C}}^{(m,n_k)-T} \in \mathbb{C}^{R\times R}$, $\tilde{\boldsymbol{C}}^{(m,n_k)} \triangleq (\boldsymbol{C}^{(m,n_k)}\boldsymbol{D}_C^{(m,n)}\boldsymbol{\Pi})$, $\boldsymbol{D}_C^{(m,n)}$ is a diagonal matrix, representing the scaling ambiguity for $\boldsymbol{C}^{(m,n)}$, and $\boldsymbol{\Pi}$ is a common permutation matrix for all $\boldsymbol{C}^{(m,n_k)}$, which is due to the DC-CPD permutation ambiguity, $k = 1,\dots,K$.

We give the proof of this theorem in the *Appendix*. The theorem provides the following key results for the algebraic algorithm: (a) the dimension of the null space of $\boldsymbol{\Gamma}^{(m,n_1,...,n_K)}$ reveals the number of sources; (b) via (15), the basis vectors in the null space of $\boldsymbol{\Gamma}^{(m,n_1,...,n_K)}$ are explicitly linked to the inverse of $\boldsymbol{C}^{(m,n_k)T}$ up to trivial indeterminacies. Therefore, solving the problem (15) yields estimates of the factor matrices $\boldsymbol{C}^{(m,n_k)}$. Subsequently, the estimates of the factor matrices $\boldsymbol{A}^{(m)}$ can be obtained, as explained in step (*iii*). We note that *Theorem 4* extends the corresponding results in [23], from $K = 2$ to $K \in [2, M]$.

We denote $\boldsymbol{W}^{(m,n_1,...,n_K)} \triangleq [\boldsymbol{w}_1^{(m,n_1,...,n_K)},...,\boldsymbol{w}_R^{(m,n_1,...,n_K)}] \in \mathbb{C}^{R^K\times R}$, and $\boldsymbol{\mathcal{W}}^{(m,n_1,...,n_K)} \triangleq \mathrm{Ten}(\boldsymbol{W}^{(m,n_1,...,n_K)},[R,R,...,R]) \in \mathbb{C}^{R\times\cdots\times R}$. According to (15), $\boldsymbol{\mathcal{W}}^{(m,n_1,...,n_K)}$ admits the following (K+1)th-order CPD for fixed m, $n_1,\dots,n_k$:

$$\boldsymbol{\mathcal{W}}^{(m,n_1,...,n_K)} = [\![\boldsymbol{B}^{(m,n_1)},...,\boldsymbol{B}^{(m,n_K)},\boldsymbol{F}^{(m,n_1,...,n_K)}]\!]_R, \tag{16}$$

where all the factor matrices $\boldsymbol{B}^{(m,n_k)},\boldsymbol{F}^{(m,n_1,...,n_K)}$ have full column rank, $k = 1,\dots,K$. Hence, for all the possible values of $1 \le m \le M$ and $1 \le n_1 < \cdots < n_K \le M$, (16) is a ($K$+1)th-order overdetermined C-CPD with coupling in the first K modes.

(*iii*) *Solve the overdetermined (K+1)th-order C-CPD (16), and calculate factor matrices* $\boldsymbol{A}^{(1)},...,\boldsymbol{A}^{(M)}$.

When $K = 2$, (16) is an overdetermined third-order DC-CPD, and we can use the strategy explained in [23] to compute the factor matrices. When $K > 2$, the tensor $\boldsymbol{\mathcal{W}}^{(m,n_1,...,n_K)}$ in (16) has order (K+1) > 3. We now follow [16], which explains that, under conditions that are satisfied in our derivation, the CPD of a tensor of order higher than three can equivalently be expressed as a C-CPD of third-order tensors obtained by combining modes (the details of this procedure are given in the *Appendix*). This C-CPD can in turn be computed by a GEVD. Therefore, any of the (K+1)th-order CPDs can be computed by GEVD, which yields us K factor matrices $\boldsymbol{B}^{(m,n_1)},...,\boldsymbol{B}^{(m,n_K)}$, for any choice of $(m,n_1,...,n_K)$.

To obtain the full set of factor matrices $\boldsymbol{B}^{(m,n)}$, m, $n = 1,\dots,M$, we work as follows. Assuming that for instance $\boldsymbol{B}^{(m,1)}$ has been found, then for tensor $\boldsymbol{\mathcal{W}}^{(m,1,n_2,...,n_K)}$, we have:

$$\boldsymbol{W}_1^{(m,1,n_2...,n_K)}\boldsymbol{B}^{(m,1)-T} = \boldsymbol{B}^{(m,n_2)} \odot \cdots \odot \boldsymbol{B}^{(m,n_K)} \odot \boldsymbol{F}^{(m,1,n_2...,n_K)}. \tag{17}$$

We denote the rth column of $\boldsymbol{W}_1^{(m,1,n_2...,n_K)}\boldsymbol{B}^{(m,1)-T}$ by $\boldsymbol{q}_{1,r}^{(m,1,n_2,...,n_K)} \in \mathbb{C}^{R^K}$ and reshape it into a tensor $\boldsymbol{\mathcal{Q}}_{1,r}^{(m,1,n_2,...,n_K)} = \mathrm{Ten}(\boldsymbol{q}_{1,r}^{(m,1,n_2,...,n_K)},[R,...,R]) \in \mathbb{C}^{R\times\cdots\times R}$. From (17) we know that $\boldsymbol{\mathcal{Q}}_{1,r}^{(m,1,n_2,...,n_K)}$ is a Kth-order rank-1 tensor:

$$\mathcal{Q}_{1,r}^{(m,1,n_2,\ldots,n_K)} = \boldsymbol{b}_r^{(m,n_2)} \otimes \cdots \otimes \boldsymbol{b}_r^{(m,n_K)} \otimes \boldsymbol{f}_r^{(m,n_1,\ldots,n_K)}, \tag{18}$$

where $\boldsymbol{b}_r^{(m,n_u)}$ and $\boldsymbol{f}_r^{(m,1,n_2,\ldots,n_K)}$ are the rth column of $\boldsymbol{B}^{(m,n_u)}$ and $\boldsymbol{F}^{(m,1,n_2,\ldots,n_K)}$, respectively, $u = 2,\ldots,K$. In the noisy case, $\boldsymbol{b}_r^{(m,n_u)}$ and $\boldsymbol{f}_r^{(m,1,n_2,\ldots,n_K)}$ can be obtained by calculating the best rank-1 approximation of $\mathcal{Q}_{1,r}^{(m,1,n_2,\ldots,n_K)}$ [33]. Continuing this way, all factor matrices $\boldsymbol{B}^{(m,n)}$ can be determined, m, $n = 1,\ldots,M$.

Using all the frontal slices we can also compute the factor matrices via SD. Optimization based DC-CPD algorithms such as alternating least squares (ALS), or nonlinear least squares (NLS) can be used to explicitly maximize the fit, i.e. to minimize the cost function $\sum_{m,p,q}\|\mathcal{W}^{(m,p,q)} - [\![\boldsymbol{B}^{(m,g)}, \boldsymbol{B}^{(m,h)}, \boldsymbol{F}^{(m,g,h)}]\!]_R\|$. Once $\boldsymbol{B}^{(m,n)}$ has been computed, we immediately obtain $\tilde{\boldsymbol{C}}^{(m,n)} = \boldsymbol{B}^{(m,n)-T}$ and $\tilde{\boldsymbol{A}}^{(m)} \odot \tilde{\boldsymbol{A}}^{(n)*} = \boldsymbol{T}_3^{(m,n)}\boldsymbol{B}^{(m,n)}$ for all $m,n = 1,\ldots,M$.

The above GEVD or SD based C-CPD approach only exploits part of the structure, as explained in the *Appendix*. We may also compute the factor matrices $\boldsymbol{B}^{(m,n)}$ by using the full structure of the tensors $\mathcal{W}^{(m,n_1,\ldots,n_K)}$ in (16) via an optimization algorithm that maximizes the fit, i.e. minimizes $\sum_{m,n_1,\ldots,n_K}\|\mathcal{W}^{(m,n_1,\ldots,n_K)} - [\![\boldsymbol{B}^{(m,n_1)},\ldots,\boldsymbol{B}^{(m,n_K)}, \boldsymbol{F}^{(m,n_1,\ldots,n_K)}]\!]_R\|$. Some options are: (a) the framework of SDF [8] is well-suited for the task; (b) alternating least squares (ALS) is a specific type of optimization algorithm that can be used. The updating equations can be explicitly derived in analogy with the derivation in the *Supplementary materials* of [23] for DC-CPD. Note that the GEVD based approach can be used to efficiently initialize these optimization based algorithms.

Once the C-CPD (16) has been computed, we obtain $\tilde{\boldsymbol{C}}^{(m,n)} = \boldsymbol{C}^{(m,n)}\boldsymbol{D}_C^{(m,n)}\boldsymbol{\Pi} = \boldsymbol{B}^{(m,n)-T}$, and $\tilde{\boldsymbol{A}}^{(m)} \odot \tilde{\boldsymbol{A}}^{(n)*} = \boldsymbol{T}_3^{(m,n)}\boldsymbol{B}^{(m,n)}$, where $\tilde{\boldsymbol{A}}^{(m)} = \boldsymbol{A}^{(m)}\boldsymbol{D}_A^{(m)}\boldsymbol{\Pi}$ and $\tilde{\boldsymbol{A}}^{(n)} = \boldsymbol{A}^{(n)}\boldsymbol{D}_A^{(n)}\boldsymbol{\Pi}$ are estimates of $\boldsymbol{A}^{(m)}$ and $\boldsymbol{A}^{(n)}$ up to scaling and permutation ambiguities, $m,n = 1,\ldots,M$. Matrices $\boldsymbol{D}_A^{(m)}$ and $\boldsymbol{D}_A^{(n)}$ are diagonal matrices, and $\boldsymbol{D}_A^{(m)}\boldsymbol{D}_A^{(n)*}\boldsymbol{D}_C^{(m,n)} = \boldsymbol{I}_R$. We define $\boldsymbol{G}_r^{(m,n)} \triangleq \text{unvec}(\boldsymbol{T}_3^{(m,n)}\boldsymbol{B}^{(m,n)})_{(:,r)}$, and collect these matrices in a $NM \times NM$ matrix $\boldsymbol{G}_r$ as follows:

$$\boldsymbol{G}_r = \begin{bmatrix} \boldsymbol{G}_r^{(1,1)} & \cdots & \boldsymbol{G}_r^{(1,M)} \\ \vdots & \ddots & \vdots \\ \boldsymbol{G}_r^{(M,1)} & \cdots & \boldsymbol{G}_r^{(M,M)} \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{a}}_r^{(1)} \\ \vdots \\ \tilde{\boldsymbol{a}}_r^{(M)} \end{bmatrix} \cdot [\tilde{\boldsymbol{a}}_r^{(1)H},\ldots,\tilde{\boldsymbol{a}}_r^{(M)H}], \tag{19}$$

where $\tilde{\boldsymbol{a}}_r^{(m)}$ denotes the rth column of $\tilde{\boldsymbol{A}}^{(m)}$, $r = 1,\ldots,R$, $m = 1,\ldots,M$. We can calculate $[\tilde{\boldsymbol{a}}_r^{(1)T},\ldots,\tilde{\boldsymbol{a}}_r^{(M)T}]^T$ as the dominant eigenvector of $\boldsymbol{G}_r$. TABLE II summarizes the proposed algebraic DC-CPD algorithm associated with a fixed order $K \in [2, M]$.

Remark 3: We note that the main complexity of the algebraic DC-CPD algorithm, as it has been presented, is in the construction of the matrices $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$ and the calculation of the basis vectors in their null space. We have according to (8) and (9), that the complexity of the construction of each column of $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$ is $\text{O}(\text{C}_K^2 KN^{2K})$ flops, and thus the overall complexity of the construction of $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$ is $\mathrm{O}(\mathrm{C}_K^2 K N^{2K} R^K)$ flops. The basis vectors in $\ker(\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)})$ are computed as singular vectors. This step has complexity $\mathrm{O}(2\mathrm{C}_K^2 N^{2K} R^{2K} + 11R^{3K})$ flops. We construct and manipulate $M\mathrm{C}_M^K$ such matrices $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$. Therefore, the overall complexity is $\mathrm{O}(M\mathrm{C}_M^K(\mathrm{C}_K^2 K N^{2K} R^K + 2\mathrm{C}_K^2 N^{2K} R^{2K} + 11R^{3K}))$ flops. The memory requirements of the algorithm are mainly in the storage of the $M\mathrm{C}_M^K$ tensors $\boldsymbol{\mathcal{W}}^{(m,n_1,\ldots,n_K)}$ and the matrix $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$. Therefore, the memory requirements are $\mathrm{O}(M\mathrm{C}_M^K(R^{(K+1)} + \mathrm{C}_K^2 N^{2K} R^K))$ complex numbers.

However, the algebraic DC-CPD admits an efficient implementation. Instead of explicitly constructing $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$, we can calculate the $R^K \times R^K$ Hermitian matrices $\boldsymbol{\Omega}^{(m,n_1,\ldots,n_K)} \triangleq \boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)H}\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$, taking advantage of the algebraic structure of $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$. We note that $\ker(\boldsymbol{\Omega}^{(m,n_1,\ldots,n_K)}) = \ker(\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)})$, while $\boldsymbol{\Omega}^{(m,n_1,\ldots,n_K)}$ have smaller size and can be computed more efficiently than $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$. We provide the details in the *Appendix*. Compared with the original version of algebraic DC-CPD, this implementation has lower complexity and memory requirements, namely $\mathrm{O}(0.5M\mathrm{C}_M^K\mathrm{C}_K^2 N^2 R^{2K} + 2M\mathrm{C}_M^K R^{3K})$ flops and $\mathrm{O}(M^2 N^2 R^2 + M\mathrm{C}_M^K R^{2K})$ complex numbers, respectively.

TABLE II

ALGEBRAIC DC-CPD ALGORITHM WITH A FIXED K

Input: $\{\boldsymbol{\mathcal{T}}^{(m,n)} \in \mathbb{C}^{N\times N\times R} \,|\, 1 \le m,n \le M\}$ admitting DC-CPD (2), and a fixed $K \in [2, M]$.

1: Group $\boldsymbol{\mathcal{T}}^{(m,n)}$ into M sets $\Upsilon^{(m)}$ by (6).

2: For each combination of K tensors in $\Upsilon^{(m)}$, perform the operation (9) for all the frontal slices of these tensors by (10)–(14), to construct a matrix $\boldsymbol{\Gamma}(\boldsymbol{\mathcal{T}}^{(m,n_1)},\ldots,\boldsymbol{\mathcal{T}}^{(m,n_K)})$.

3: Calculate the basis vectors $\boldsymbol{w}_r^{(m,n_1,\ldots,n_K)}$, $r = 1,\ldots,R$, of the null space of $\boldsymbol{\Gamma}(\boldsymbol{\mathcal{T}}^{(m,n_1)},\ldots,\boldsymbol{\mathcal{T}}^{(m,n_K)})$. For varying $(m, n_1,\ldots,n_K)$, convert all the basis vectors $\boldsymbol{w}_r^{(m,n_1,\ldots,n_K)}$ into an overdetermined C-CPD (16).

4: Compute the nonsingular matrices $\boldsymbol{B}^{(m,n)}$ by solving the C-CPD (16), then we have $\tilde{\boldsymbol{C}}^{(m,n)} = \boldsymbol{B}^{(m,n)-T}$.

5: Compute $\boldsymbol{G}_r^{(m,n)} = \mathrm{unvec}(\boldsymbol{T}_3^{(m,n)}\boldsymbol{B}^{(m,n)})_{(:,r)}$, and construct matrix $\boldsymbol{G}_r$ by (19). Then the rth column of the factor matrices $\tilde{\boldsymbol{A}}^{(m)}$ can be computed as the dominant eigenvector of $\boldsymbol{G}_r$, $m = 1,\ldots,M$, $r = 1,\ldots,R$.

Output: Estimates of the factor matrices $\tilde{\boldsymbol{A}}^{(m)}$ and $\tilde{\boldsymbol{C}}^{(m,n)}$, $1 \le m,n \le M$.

VI. SIMULATION RESULTS

In this section we present simulation results to demonstrate the performance of the proposed algebraic DC-CPD framework, in comparison with the algebraic DC-CPD algorithm proposed in [23], an iterative DC-CPD algorithm based on nonlinear least squares, implemented via SDF in Tensorlab 3.0 [9], and standard CPD. In *Simulation A*, we discuss high-accuracy computation of exact decompositions with high rank. In *Simulation B*, we examine the performance of the compared DC-CPD and CPD algorithms in approximate decompositions, where we add noise term into the data and vary the level of it, mainly to demonstrate the sensitivity of different algorithms to the level of noise in several different settings. In *Simulation C*, we demonstrate the performance of the compared DC-CPD and CPD algorithms in underdetermined J-BSS. In *Simulation D*, we demonstrate the performance of the compared DC-CPD and CPD algorithms in the application of extremely underdetermined wideband DOA estimation.

We use the following abbreviations:

- DC-CPD-ALG: the proposed algebraic DC-CPD algorithms, based on coupled rank-1 detection mapping of order $K \geq 2$. In the simulations, we will mainly consider algorithms associated with $K = 2$, $K = 3$ and $K = 4$, and we will specify the value of K in the abbreviation, e.g. DC-CPD-ALG($K = 3$). We note that DC-CPD-ALG($K = 2$) is the algebraic DC-CPD algorithm proposed in [23].
- DC-CPD-NLS: DC-CPD based on nonlinear least squares, implemented by SDF.

DC-CPD-NLS is implemented using the 'sdf_nls.m' function in Tensorlab 3.0. In the implementation of DC-CPD-ALG with $K \geq 2$, we use the SD based scheme to solve the overdetermined C-CPD in (16). In the implementation of DC-CPD-NLS, we initialize either randomly or by the results of DC-CPD-ALG, depending on the application. The initialization details will be given in each of the following examples. We use two different implementations for CPD. In *Simulation A*, we use the algebraic CPD algorithm proposed in [25], as it can return exact results when the CPD uniqueness conditions hold. In *Simulation B, C, D*, we use the 'cpd.m' function with default setting in Tensorlab 3.0, which is an optimization algorithm based on nonlinear least squares[2]. The reason is that an optimization algorithm that directly maximizes the fit to the data tensors provides the optimal results in the LS sense, if it converges to the global minimum.

All the simulations are performed on a workstation with two CPUs: Intel (R) Xeon (R) CPU E5-2640 v4 @ 2.40GHz, and 128GB memory. The operating system is 64bit Windows10, and we use MATLAB R2016b to run the simulations.

A. Results for exact decompositions

The data tensors are directly generated by (2), where both the real and imaginary parts of each entry of the factor matrices $\boldsymbol{A}^{(m)} \in \mathbb{C}^{N \times R}$ and $\boldsymbol{C}^{(m,n)} \in \mathbb{C}^{R \times R}$ are randomly drawn from a standard normal distribution, and no noise is added.

[2] The default setting of 'cpd.m' in Tensorlab 3.0 is: (i) If the rank of a tensor is smaller than its first two dimensions, then a GEVD based initialization is used, otherwise a random initialization is used with entries drawn from the normal distribution; (ii) A Gauss-Newton based optimization algorithm is used to compute the CPD. This algorithm is implemented in 'cpd_nls.m'. The tolerance on the relative function value and the tolerance on the relative step size in the stopping criteria are set to TolFun = 10^{-12} and TolX = 10^{-8}, respectively.

We fix the third dimension of all the data tensors to R and let the number of mixing matrices $M = 4$. The mean relative error of the estimates of all the loading matrices is defined as:

$$\varepsilon = M^{-1}\sum\nolimits_{m=1}^{M} \min_{\boldsymbol{\Pi}^{(m)},\boldsymbol{D}^{(m)}} \left(\left\| \boldsymbol{A}^{(m)} - \boldsymbol{\Pi}^{(m)}\boldsymbol{D}^{(m)}\tilde{\boldsymbol{A}}^{(m)} \right\|_F^2 \Big/ \left\| \boldsymbol{A}^{(m)} \right\|_F^2 \right), \tag{20}$$

where $\tilde{\boldsymbol{A}}^{(m)}$ is the estimate of the true mixing matrix $\boldsymbol{A}^{(m)}$, $\boldsymbol{\Pi}^{(m)}$ is a permutation matrix and $\boldsymbol{D}^{(m)}$ is a diagonal matrix. We consider three DC-CPD-ALG algorithms associated with $K = 2, 3, 4$, respectively. For DC-CPD-NLS, we try ten random initial values and select the one that gives the best fit after the first ten iterations to effectively initialize the algorithm. As there is no noise, we want to estimate the factor matrices with high precision. Therefore, we set the termination parameters 'TolFun' and 'TolX' in 'sdf_nls.m' to 10^{-12} and 10^{-6}, respectively. In addition, we set the maximal number of iterations to 1000 for DC-CPD-NLS. The mean relative error and CPU time (denoted as 't') of DC-CPD-ALG($K = 2$), DC-CPD-ALG($K = 3$), DC-CPD-ALG($K = 4$), DC-CPD-NLS, and CPD for different values of N and R are reported in TABLE III. The average number of iterations of DC-CPD-NLS (denoted as 'it') is also reported in TABLE III. All the results in TABLE III are averaged over 100 independent runs. In addition, we draw in Fig. 2, for each compared algorithm, the percentage of successful independent runs in which the exact solution is found. Here we call a run of an algorithm successful if the mean relative error of this algorithm is less than 10^{-5}.

TABLE III

PERFORMANCE OF DC-CPD-ALG($K = 2$), DC-CPD-ALG($K = 3$), DC-CPD-ALG($K = 4$), DC-CPD-NLS, AND CPD IN EXACT DECOMPOSITIONS, 'N' IS THE NUMBER OF OBSERVATION CHANNELS, 'R' IS THE NUMBER OF SOURCES, 'ε' IS THE MEAN RELATIVE ERROR, 't' IS THE MEAN CPU TIME, AND 'it' IS THE AVERAGE NUMBER OF ITERATIONS FOR DC-CPD-NLS.

	$N = 3, R = 4$	$N = 3, R = 5$	$N = 3, R = 6$	$N = 4, R = 9$	$N = 4, R = 10$	$N = 4, R = 11$	$N = 4, R = 12$
DC-CPD-ALG(K = 2)	**$\varepsilon = 7.08\times10^{-16}$** $t = 0.077$ sec.	**$\varepsilon = 3.74\times10^{-15}$** $t = 0.095$ sec.	$\varepsilon = 0.555$ $t = 0.837$ sec.	**$\varepsilon = 1.39\times10^{-14}$** $t = 0.212$ sec.	**$\varepsilon = 1.59\times10^{-13}$** $t = 0.255$ sec.	$\varepsilon = 0.719$ $t = 2.498$ sec.	$\varepsilon = 0.758$ $t = 1.931$ sec.
DC-CPD-ALG(K = 3)	**$\varepsilon = 6.77\times10^{-15}$** $t = 0.149$ sec.	**$\varepsilon = 3.62\times10^{-14}$** $t = 0.311$ sec.	**$\varepsilon = 2.34\times10^{-12}$** $t = 0.655$ sec.	**$\varepsilon = 3.93\times10^{-14}$** $t = 6.871$ sec.	**$\varepsilon = 9.57\times10^{-14}$** $t = 13.54$ sec.	**$\varepsilon = 6.97\times10^{-13}$** $t = 25.78$ sec.	$\varepsilon = 0.746$ $t = 50.21$ sec.
DC-CPD-ALG(K = 4)	**$\varepsilon = 1.77\times10^{-13}$** $t = 0.286$ sec.	**$\varepsilon = 9.23\times10^{-13}$** $t = 1.521$ sec.	**$\varepsilon = 8.47\times10^{-12}$** $t = 6.914$ sec.	**$\varepsilon = 6.09\times10^{-13}$** $t = 786.14$ sec.	**$\varepsilon = 1.28\times10^{-12}$** $t = 1.382\times10^{3}$ sec.	**$\varepsilon = 8.46\times10^{-12}$** $t = 4.334\times10^{3}$ sec.	**$\varepsilon = 6.32\times10^{-9}$** $t = 1.765 \times10^{4}$ sec.
DC-CPD-NLS	$\varepsilon = 1.17 \times10^{-4}$ $t = 39.29$ sec $it = 209$.	$\varepsilon = 0.0159$ $t = 83.32$ sec $it = 368$.	$\varepsilon = 0.1607$ $t = 197.55$ sec $it = 897$.	$\varepsilon = 3.5 \times10^{-3}$ $t = 128.41$ sec $it = 552$.	$\varepsilon = 0.199$ $t = 205.20$ sec $it = 908$.	$\varepsilon = 0.488$ $t = 231.86$ sec $it = 1000$.	$\varepsilon = 0.643$ $t = 279.62$ sec $it = 1000$.
CPD	**$\varepsilon = 6.77\times10^{-15}$** $t = 0.074$ sec.	$\varepsilon = 0.1875$ $t = 1.441$ sec.	$\varepsilon = 1.436$ $t = 2.403$ sec.	**$\varepsilon = 1.426\times10^{-8}$** $t = 2.02$ sec.	$\varepsilon = 1.547$ $t = 12.20$ sec.	$\varepsilon = 2.204$ $t = 17.532$ sec.	$\varepsilon = 2.735$ $t = 6.76$ sec.

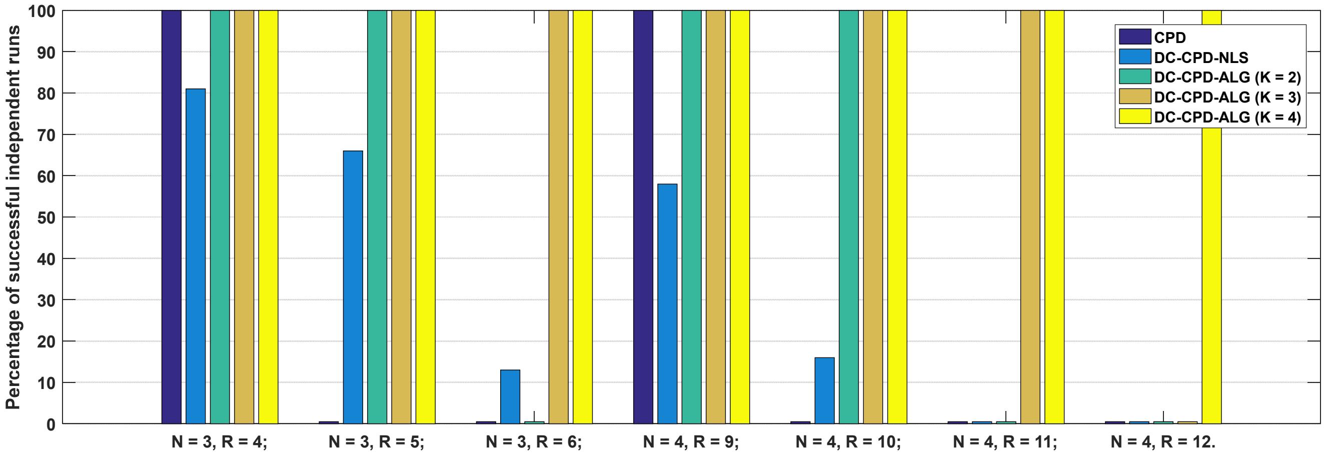


Fig. 2. Percentage of independent runs in which exact solution is found for the compared algorithms in seven settings. We observe that DC-CPD-ALG($K = 4$) finds the exact solution in all independent runs in all the settings. DC-CPD-ALG($K = 3$) finds the exact solution in the first six settings.

The observations in TABLE III and Fig. 2 generally suggest the following results:

(*i*) DC-CPD-ALG may be unique even when none of its constituting CPDs is unique. More precisely, in TABLE III and Fig. 2 we observe that CPD finds the exact solution only in cases $N = 3$, $R = 4$ and $N = 4$, $R = 9$. On the other hand, DC-CPD-ALG, with a properly chosen K, generates the correct results in all the considered cases.

(*ii*) DC-CPD-ALG with a larger K has more relaxed uniqueness conditions, i.e. $R_{\max}$ increases as K increases. For example, when $N = 4$, we observe that DC-CPD-ALG($K = 2$) only finds the exact solution when $R \leq 10$. DC-CPD-ALG($K = 3$), on the other hand, can return the correct results when $R \leq 11$. DC-CPD-ALG($K = 4$) finds the exact solution when $R \leq 12$. Similar results can be observed for $N = 3$.

(*iii*) DC-CPD-NLS with random initialization does not always find the correct solution for underdetermined DC-CPD problems, even in the exact case. As observed in Fig. 2, when $N = 3$, the percentage of successful independent runs for DC-CPD-NLS is 81% for $R = 4$, 66% for $R = 5$, and 0% for $R = 6$, respectively. When $N = 4$, the percentage of successful independent runs for DC-CPD-NLS is 58% for $R = 9$, 16% for $R = 10$, and 0% for $R = 11, 12$, respectively. These observations suggest that DC-CPD-NLS is less likely to find the correct solution when the problem gets more difficult. We note that in cases $N = 3$, $R = 6$ and $N = 4$, $R = 11, 12$, DC-CPD-NLS does not generate the correct result in any of the independent runs. This observation suggests that for DC-CPD with very large R, the exact solution can be found by algebraic DC-CPD algorithms while the optimization based DC-CPD-NLS algorithm does not work anymore.

(*iv*) As K increases, the complexity of DC-CPD-ALG increases significantly, as shown in TABLE III. In particular, when $K = 4$, $N = 4$, $R = 9, 10, 11, 12$, DC-CPD-ALG becomes computationally expensive. However, when $K = 2, 3$, the DC-CPD-ALG algorithm is in general faster than DC-CPD-NLS. In practice, noise is usually present, and DC-CPD-ALG will be less accurate but can still be used as a low-cost initialization for DC-CPD-NLS, which directly maximizes the fit of the data tensors in the LS sense. This will be shown in the next three simulations.

B. Results for approximate decompositions

In this simulation, we illustrate the performance of DC-CPD-ALG($K = 2$), DC-CPD-ALG($K = 3$), DC-CPD-ALG($K = 4$), DC-CPD-NLS and CPD for approximate DC-CPD. We consider DC-CPD-NLS with two different initialization schemes: (*i*) DC-CPD-NLS with random initialization (we try ten random initial values and select the one that gives the best fit after the first ten iterations to effectively initialize the algorithm); (*ii*) DC-CPD-NLS initialized with the results of DC-CPD-ALG($K = 3$). We denote these two implementations of DC-CPD-NLS as DC-CPD-NLS(Rand) and DC-CPD-NLS(ALG), respectively. The termination parameters 'TolFun' and 'TolX' of the 'sdf_nls.m' function used in DC-CPD-NLS are set to 10^{-6} and 10^{-3}, respectively.

The data tensors are generated as follows:

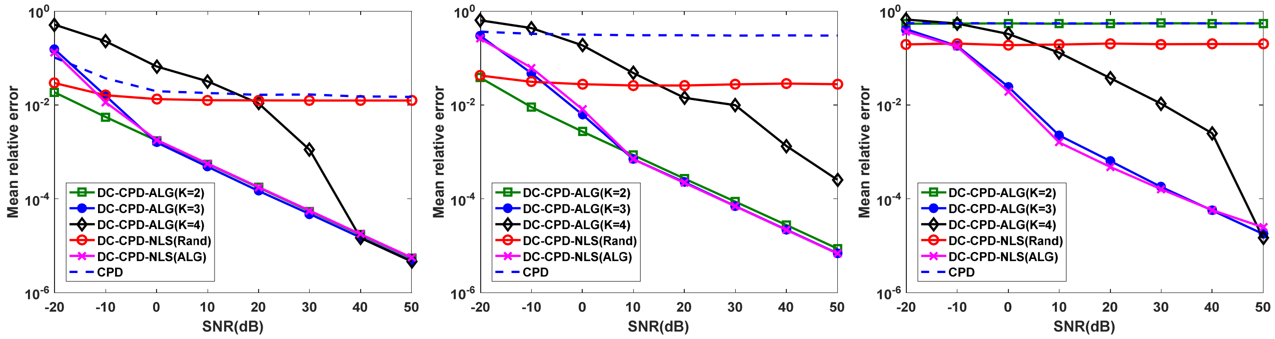
$$\mathcal{T}^{(m,n)} = \sigma_s \frac{\mathcal{T}'^{(m,n)}}{\left\|\mathcal{T}'^{(m,n)}\right\|_F} + \sigma_n \frac{\mathcal{N}^{(m,n)}}{\left\|\mathcal{N}^{(m,n)}\right\|_F}, \tag{21}$$

where $\mathcal{T}'^{(m,n)} \in \mathbb{C}^{N\times N\times R}$ is generated by (2) and $\mathcal{N}^{(m,n)} \in \mathbb{C}^{N\times N\times R}$ is the noise term. In the generation of $\mathcal{T}'^{(m,n)}$ and $\mathcal{N}^{(m,n)}$, both the real and imaginary parts of each entry of the factor matrices $\boldsymbol{A}^{(m)} \in \mathbb{C}^{N\times R}$, $\boldsymbol{C}^{(m,n)} \in \mathbb{C}^{R\times R}$, and tensor $\mathcal{N}^{(m,n)}$ are randomly drawn from a standard normal distribution. The parameters σ_s and σ_n denote the signal level and noise level, respectively. The signal-to-noise ratio (SNR) is defined as:

$$\text{SNR} = 20\log_{10}(\sigma_s / \sigma_n). \tag{22}$$

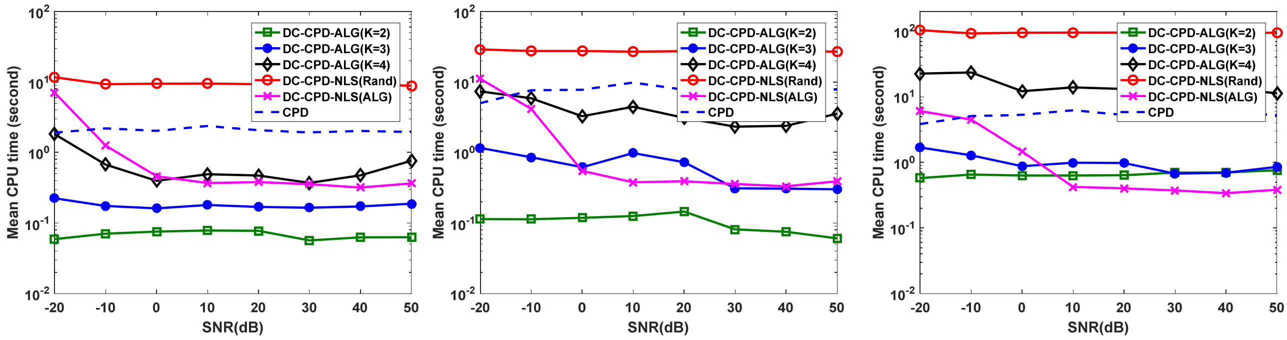
We fix the third dimension of all the data tensors to R and the number of mixing matrices to $M = 4$. We let SNR vary from –20dB to 50dB. The mean relative error ε versus the SNR is plotted in Fig. 3 to illustrate the performance of the compared algorithms in the following three settings: (a) a slightly underdetermined case $N = 3$, $R = 4$, $M = 4$, (b) a highly underdetermined case $N = 3$, $R = 5$, $M = 4$, and (c) an extremely underdetermined case $N = 3$, $R = 6$, $M = 4$. We perform 200 Monte Carlo runs for each SNR to calculate each point in the curves. In addition, we plot the mean CPU time of the compared algorithms versus the SNR in Fig. 4.

In Fig. 3 (a), we see that in the slightly underdetermined case, all the algorithms can accurately identify the mixing matrices when the SNR is sufficiently high. Comparing DC-CPD-ALG with different K, we note that as K increases, the accuracy of DC-CPD-ALG decreases. We see from Fig. 4 (a) that an increase in K also results in a larger mean CPU time for DC-CPD-ALG. This observation suggests that DC-CPD-ALG with smaller K is more robust to noise and is computationally more efficient, as long as the uniqueness conditions hold.



(a) $N = 3$, $R = 4$, $M = 4$ (b) $N = 3$, $R = 5$, $M = 4$ (c) $N = 3$, $R = 6$, $M = 4$

Fig. 3. Mean relative error of DC-CPD-ALG(K = 2), DC-CPD-ALG(K = 3), DC-CPD-ALG(K = 4), DC-CPD-NLS(Rand), DC-CPD-NLS(ALG) and CPD vs. SNR in *Simulation B*. The plots illustrate the performance in (a) a slightly underdetermined setting: $N = 3$, $R = 4$, $M = 4$, (b) a highly underdetermined setting: $N = 3$, $R = 5$, $M = 4$, and (c) an extremely underdetermined setting: $N = 3$, $R = 6$, $M = 4$.



(a) $N = 3$, $R = 4$, $M = 4$ (b) $N = 3$, $R = 5$, $M = 4$ (c) $N = 3$, $R = 6$, $M = 4$

Fig. 4. Mean CPU time of DC-CPD-ALG($K = 2$), DC-CPD-ALG($K = 3$), DC-CPD-ALG($K = 4$), DC-CPD-NLS(Rand), DC-CPD-NLS(ALG) and CPD vs. SNR in *Simulation B*. The plots illustrate the performance in (a) a slightly underdetermined setting: $N = 3$, $R = 4$, $M = 4$, (b) a highly underdetermined setting: $N = 3$, $R = 5$, $M = 4$, and (c) an extremely underdetermined setting: $N = 3$, $R = 6$, $M = 4$.

We observe in Fig. 3 (a) that DC-CPD-ALG(K = 2), DC-CPD-ALG(K = 3) and DC-CPD-NLS(ALG) are more accurate than CPD, as they make use of the coupling between the data tensors. In addition, we see that DC-CPD-NLS(ALG) is more accurate than DC-CPD-NLS(Rand). This indicates that the accuracy of DC-CPD-NLS can be improved by using the results of DC-CPD-ALG as initialization. In fact, this type of initialization for DC-CPD-NLS not only improves its accuracy, but also reduces its mean CPU time, as can be seen in Fig. 4 (a).

In Fig. 3 (b), we see that CPD does not generate correct results in the highly underdetermined case. The accuracy of other algorithms also deteriorates in comparison with the results in Fig. 3 (a), because the problem is becoming more difficult. Comparing DC-CPD-ALG with different *K*, we again observe in Fig. 3 (b) and Fig. 4 (b), that DC-CPD-ALG with smaller *K* is more accurate and faster. In addition, by comparing DC-CPD-NLS(Rand) and DC-CPD-NLS(ALG), we again observe that an initialization by the results of DC-CPD-ALG not only improves the accuracy, but also reduces the mean CPU time of DC-CPD-NLS.

In Fig. 3 (c), we see that DC-CPD-ALG(K = 2), DC-CPD-NLS(Rand), and CPD do not generate correct results in this extremely underdetermined case. DC-CPD-ALG(K = 3) is more accurate and faster than DC-CPD-ALG(K = 4). We also observe that DC-CPD-NLS(ALG) can identify the mixing matrices with considerable accuracy, thanks to the effective initialization by the results of DC-CPD-ALG(K = 3).

We summarize the above observations in approximate DC-CPD decompositions and make the following statements:

(*i*) One may increase *K* for DC-CPD-ALG, if the number of double coupled rank-1 terms that can be handled is not sufficient. This can be seen in Fig. 3 (c), where DC-CPD-ALG(K = 2) does not obtain correct results while DC-CPD-ALG with K = 3, 4 provides accurate estimates.

(*ii*) Once the uniqueness conditions hold for DC-CPD-ALG for certain *N* and *R* and multiple *K*, the one with smaller *K* is more accurate and faster. This is consistently observed in all the plots in Fig. 3 and Fig. 4.

(*iii*) DC-CPD-NLS(Rand) is usually inaccurate and slow in underdetermined problems. However, an effective initialization with the results of DC-CPD-ALG can significantly improve its accuracy and computational efficiency. On the other hand, DC-CPD-NLS(ALG) does not improve the accuracy of DC-CPD-ALG(K = 3) in this simulation. This observation shows that DC-CPD-ALG(K = 3) has already found an accurate solution.

C. Underdetermined J-BSS

The multi-set signals are generated as a finitely sampled version of (3) in the same manner as in *Experiment B* of [23]:

$$\boldsymbol{X}^{(m)} = \sigma_s \frac{\boldsymbol{A}^{(m)}\boldsymbol{S}^{(m)T}}{\| \boldsymbol{A}^{(m)}\boldsymbol{S}^{(m)T} \|_F} + \sigma_n \frac{\boldsymbol{N}^{(m)}}{\| \boldsymbol{N}^{(m)} \|_F}, \quad m = 1,...,M, \tag{23}$$

where $\boldsymbol{X}^{(m)} \in \mathbb{C}^{N \times Q}$ denotes the observed signal in the *m*th dataset, $\boldsymbol{A}^{(m)} \in \mathbb{C}^{N \times R}$ denotes the mixing matrix in the *m*th dataset, of which the real and imaginary part of each entry are drawn from a standard normal distribution. The matrix

$\boldsymbol{N}^{(m)}$ denotes white Gaussian noise added in the mth dataset, and σ_s, σ_n denote the signal level and noise level, respectively. SNR is calculated with σ_s, σ_n by (22). Parameters N, R, Q denote the number of observation channels, the number of source signals, and the number of data samples, respectively.

The source matrix $\boldsymbol{S}^{(m)} \in \mathbb{C}^{Q\times R}$ is generated from an auxiliary matrix $\boldsymbol{S}_r'$ of size $Q \times M$ as follows:

$$\boldsymbol{S}_r^T = \boldsymbol{Q}_r \boldsymbol{S}_r'^T, \tag{24}$$

where $\boldsymbol{S}_r \triangleq [(\boldsymbol{S}^{(1)})_{:,r},...,(\boldsymbol{S}^{(M)})_{:,r}] \in \mathbb{C}^{Q\times M}$, $(\boldsymbol{S}^{(m)})_{:,r}$ denotes the rth source vector in the mth dataset and $\boldsymbol{Q}_r \in \mathbb{C}^{M\times M}$ is a full rank matrix used to introduce inter-set dependence between the corresponding source signals in different datasets. Both the real and imaginary part of each entry of $\boldsymbol{Q}_r$ are drawn from a standard normal distribution. The underlying generating source matrix $\boldsymbol{S}_r'(t) \triangleq [\boldsymbol{s}_r'^{(1)},...,\boldsymbol{s}_r'^{(M)}]$ consists of complex binary phase shift keying (BPSK) signals that are amplitude modulated across P time slots of length L':

$$\boldsymbol{s}_r'^{(m)} = [\eta_1 \cdot \boldsymbol{s}_{r,1}'^{(m)T},...,\eta_P \cdot \boldsymbol{s}_{r,P}'^{(m)T}]^T \in \mathbb{C}^Q, \tag{25}$$

where $\eta_1,...,\eta_P$ are amplitude modulation coefficients that are randomly drawn from a uniform distribution over [0, 1], and where $Q = PL$. The sub-vector $\boldsymbol{s}_{r,p}'^{(m)}$ of length L is a complex BPSK sequence with entries chosen from symbols $[1,-1]$ with equal probability. By definition, $\boldsymbol{s}_r'^{(m)}$ is the concatenation of P BPSK sequences, the amplitudes of which have been modulated by coefficients $\eta_1,...,\eta_P$. In this simulation, P is fixed to 20 and $L = Q/20$.

We construct the data tensor $\tilde{\mathcal{T}}^{(m,n)}$ as the finite-sample version of the cross-covariance tensor $\mathcal{T}^{(m,n)}$ between the mth and the nth datasets (5):

$$\left(\tilde{\mathcal{T}}^{(m,n)}\right)_{(:,:,k)} = L'^{-1} \cdot \left[\sum\nolimits_{l=1}^{L'} \left(\boldsymbol{X}^{(m,k)}\right)_{(:,l)} \left(\boldsymbol{X}^{(n,k)}\right)_{(:,l)}^H\right], \tag{26}$$

where $\boldsymbol{X}^{(m,k)} \in \mathbb{C}^{N\times L}$ denotes the kth temporal frame of $\boldsymbol{X}^{(m)}$ in (23) with frame length L, overlapping with adjacent frames with overlap ratio $\alpha \in [0,1]$.

We use DC-CPD-ALG(K = 2), DC-CPD-ALG(K = 3), DC-CPD-NLS and CPD to decompose the data tensors $\tilde{\mathcal{T}}^{(m,n)}$. DC-CPD-NLS is initialized by the results of DC-CPD-ALG with the minimal value of K that is needed for identifying the given number of sources, i.e. we choose the minimal K for which the generic $R_{\max}$ is not less than the given R. For instance, for N = 3, R = 4, 5, we initialize DC-CPD-NLS by the results of DC-CPD-ALG(K = 2), while for N = 3, R = 6, we initialize DC-CPD-NLS by the results of DC-CPD-ALG(K = 3). As in *Simulation B*, DC-CPD-NLS initialized by the results of DC-CPD-ALG is denoted as DC-CPD-NLS(ALG). Note that in this simulation we do not include DC-CPD-NLS(Rand), as it turned out that most often random initialization was not good enough to find the correct solution.

For CPD, we consider two different implementations: (*i*) CPD of each auto-covariance tensor $\tilde{\mathcal{T}}^{(m,m)}$, $m = 1,\ldots,M$; (*ii*) CPD of the cross-covariance tensor $\tilde{\mathcal{T}}^{(m,n)}$, $1 \leq m \neq n \leq M$. We call the former the second-order blind identification of underdetermined mixtures [35] (SOBIUM), and the latter CPD of covariance tensors (CPD-C). The termination parameters 'TolFun' and 'TolX' of the 'sdf_nls.m' function used in DC-CPD-NLS are set to 10^{-6} and 10^{-3}, respectively.

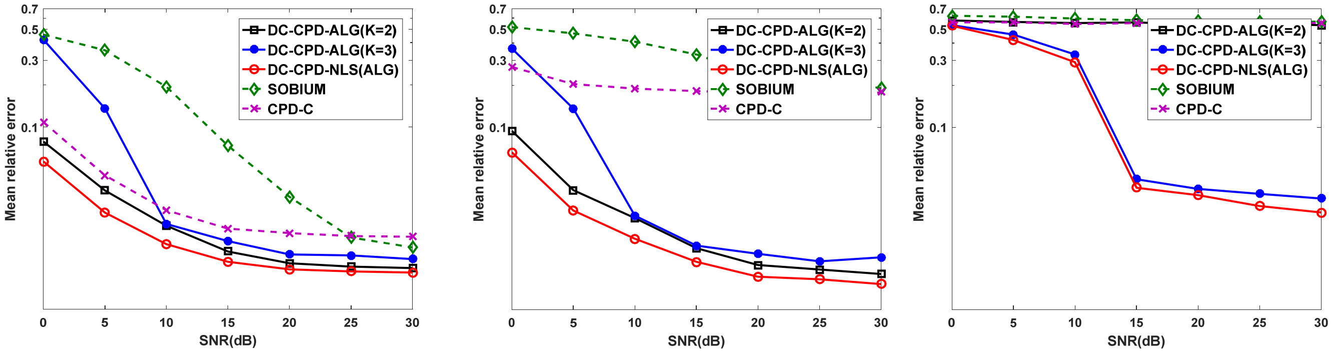
We let SNR vary from 0dB to 30dB. The mean relative error ε versus the SNR is plotted in Fig. 5 to illustrate the performance of the compared algorithms in the following three settings: (a) a slightly underdetermined case $N = 3$, $R = 4$, $M = 4$, (b) a highly underdetermined case $N = 3$, $R = 5$, $M = 4$, and (c) an extremely underdetermined case $N = 3$, $R = 6$, $M = 4$. In setting (a) we fix the framelength $L = 5000$, in setting (b) we fix $L = 20000$, and in setting (c) we fix $L = 50000$. In all the settings we let the overlap ratio $\alpha = 0.5$ and the number of frames $T = 19$. We perform 200 Monte Carlo runs for each SNR to calculate each point in the curves. In addition, we plot the mean CPU time of the compared algorithms versus the SNR in Fig. 6.

In Fig. 5 (a) and Fig. 5 (b) we see that in the slightly and highly underdetermined cases, DC-CPD-ALG($K = 2$) is more accurate than DC-CPD-ALG($K = 3$). This observation again suggests that DC-CPD-ALG with smaller K is more accurate provided that the uniqueness conditions hold. In Fig. 5 (a) we observe that both SOBIUM and CPD-C are less accurate than the DC-CPD algorithms. In Fig. 5 (b), we see that SOBIUM and CPD-C do not generate correct results. This is because the CPD based algorithms do not exploit as much information as DC-CPD algorithms. In particular, they do not exploit the coupling between covariance tensors.

In Fig. 5 (c), we see that DC-CPD-ALG($K = 3$) generates accurate results when the SNR is higher than 10dB. The other algorithms, DC-CPD-ALG($K = 2$), SOBIUM, CPD-C, do not generate correct results in this extremely underdetermined case. This observation suggests that J-BSS via DC-CPD-ALG with a larger K is able to identify more source signals. We also observe in Fig. 5 that DC-CPD-NLS(ALG) slightly improves the accuracy of DC-CPD-ALG. This observation suggests that in the presence of noise and errors caused by finite sample effects, the optimization based DC-CPD-NLS provides more accurate results than DC-CPD-ALG, given that it is well initialized by the results of the latter. As mentioned above, DC-CPD-NLS generally does not find an accurate solution if it is initialized randomly.

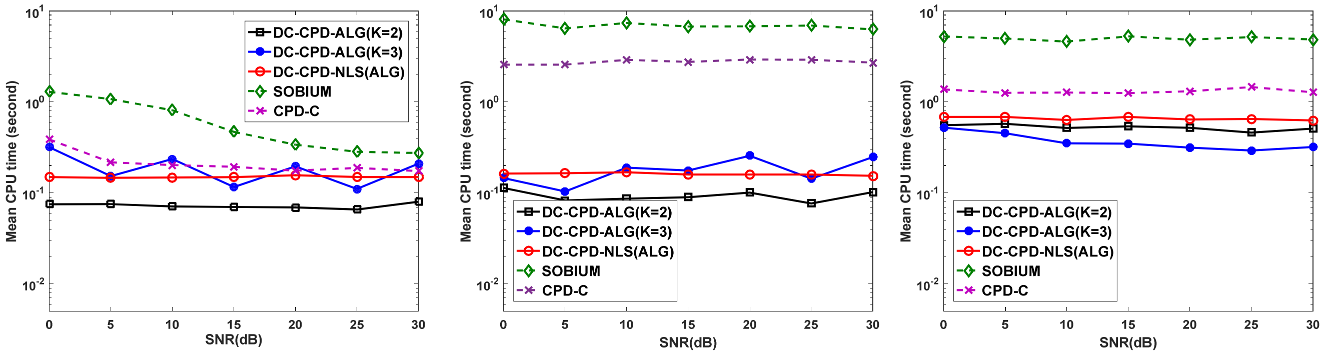
In Fig. 6, we observe that DC-CPD-ALG($K = 2$) and DC-CPD-ALG($K = 3$) are computationally very efficient. DC-CPD-NLS(ALG) is also efficient when it is well initialized, due to the low per-iteration cost and close to quadratic convergence near (local) optimum. The CPD based methods, SOBIUM and CPD-C, are slower than the other algorithms in our simulations.

The above observations generally illustrate the advantage of using DC-CPD over CPD in J-BSS applications. In particular, DC-CPD-ALG($K = 3$) is of interest for extremely underdetermined J-BSS when the other algorithms fail to generate accurate results.



(a) $N = 3$, $R = 4$, $M = 4$ (b) $N = 3$, $R = 5$, $M = 4$ (c) $N = 3$, $R = 6$, $M = 4$

Fig. 5. Mean relative error of DC-CPD-ALG($K = 2$), DC-CPD-ALG($K = 3$), DC-CPD-NLS(ALG), SOBIUM and CPD-C vs. SNR in *Simulation C*. The plots illustrate the performance in (a) a slightly underdetermined setting: $N = 3$, $R = 4$, $M = 4$, (b) a highly underdetermined setting: $N = 3$, $R = 5$, $M = 4$, and (c) an extremely underdetermined setting: $N = 3$, $R = 6$, $M = 4$.



(a) $N = 3$, $R = 4$, $M = 4$ (b) $N = 3$, $R = 5$, $M = 4$ (c) $N = 3$, $R = 6$, $M = 4$

Fig. 6. Mean CPU time of DC-CPD-ALG ($K = 2$), DC-CPD-ALG ($K = 3$), DC-CPD-NLS, SOBIUM and CPD-C vs. SNR in *Simulation B*. The plots illustrate the performance in (a) a slightly underdetermined setting: $N = 3$, $R = 4$, $M = 4$, (b) a highly underdetermined setting: $N = 3$, $R = 5$, $M = 4$, and (c) an extremely underdetermined setting: $N = 3$, $R = 6$, $M = 4$.

D. Extremely underdetermined wideband DOA estimation

Wideband DOA estimation is a challenging problem in array processing and also an important application of J-BSS. In an experiment of a preceding paper [23], we have demonstrated the performance of DC-CPD methods in a general wideband DOA estimation application, including an overdetermined setting and an underdetermined setting. In this simulation, we consider the extremely underdetermined case: $N = 3$, $R = 6$, to emphasize the power of the proposed algebraic DC-CPD algorithm in a difficult problem.

We assume that R = 6 wideband source signals impinge on an L-shaped array of N = 3 antennas. The DOA of the rth source signal is represented by the direction vector, defined as $\boldsymbol{p}_{(\theta_r,\varphi_r)} \triangleq [\cos\theta_r \sin\varphi_r, \sin\theta_r \sin\varphi_r, \cos\varphi_r]^T \in \mathbb{R}^3$, where θ_r and φ_r denote the azimuth and elevation of the rth source signal, respectively. We refer to Fig. 4 (a) of [23] for the angle definition and an illustration of the adopted array. In addition, we define the direction matrix as $\boldsymbol{P} \triangleq [\boldsymbol{p}_{(\theta_1,\varphi_1)},...,\boldsymbol{p}_{(\theta_R,\varphi_R)}] \in \mathbb{R}^{3\times R}$. In the simulation, we fix the DOAs of the source signals as: $(\theta_1,\varphi_1) = (28.5^\circ, 7.5^\circ)$, $(\theta_2,\varphi_2) = (85^\circ, 22.5^\circ)$, $(\theta_3,\varphi_3) = (141.5^\circ, 37.5^\circ)$, $(\theta_4,\varphi_4) = (198^\circ, 52.5^\circ)$, $(\theta_5,\varphi_5) = (255^\circ, 67.5^\circ)$, $(\theta_6,\varphi_6) = (311.5^\circ, 82.5^\circ)$.

We consider the practical setting in which each antenna is connected to a filterbank consisting of M filters. The central frequencies of the M filters are $f_1,..., f_M$, respectively. If the bandwidth of each filter is much smaller than its central frequency, then the array outputs associated with different antennas and filters of identical central frequency f_m approximately satisfy the classical narrowband array model (we refer to [36] for details of this setting):

$$\boldsymbol{X}^{(m)} = \boldsymbol{A}^{(m)}\boldsymbol{S}^{(m)T} + \boldsymbol{N}^{(m)}, \quad m = 1,...,M, \tag{27}$$

where $\boldsymbol{X}^{(m)} \in \mathbb{C}^{N\times Q}$ and $\boldsymbol{N}^{(m)} \in \mathbb{C}^{N\times Q}$ denote the array output signal associated with the mth filter and the corresponding noise term, respectively. The mixing matrix $\boldsymbol{A}^{(m)} \in \mathbb{C}^{N\times R}$ holds the steering vectors as columns, defined as:

$$\boldsymbol{A}^{(m)} \triangleq \exp\left(-\mathrm{i}2\pi c^{-1} f_m\right) \cdot \boldsymbol{K}^T \boldsymbol{P}, \tag{28}$$

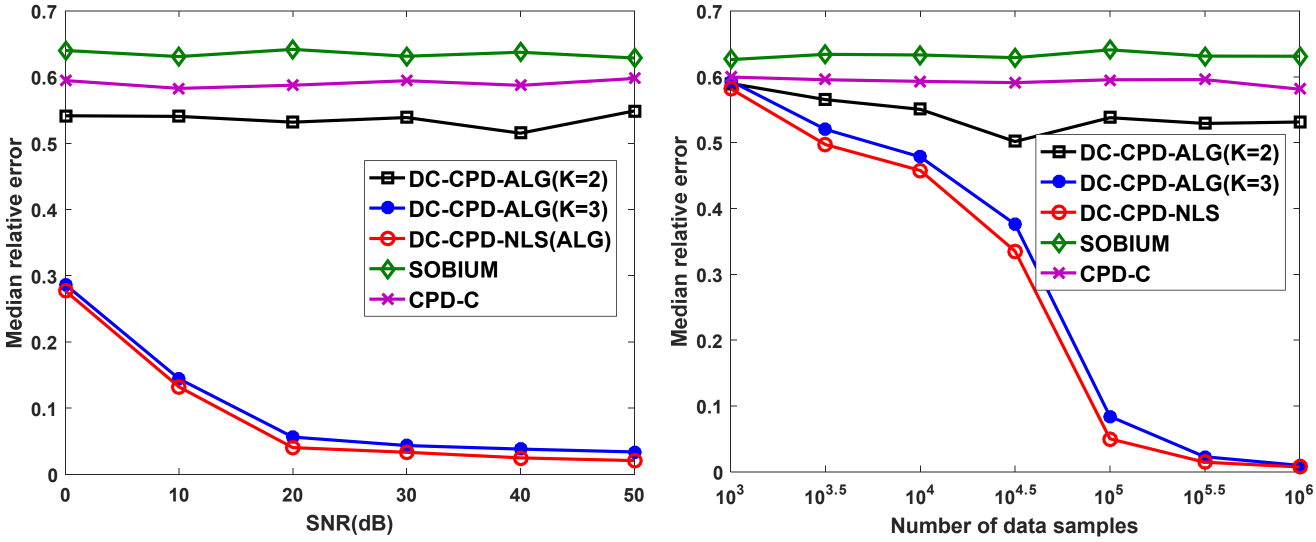
where $\boldsymbol{K} \triangleq [\boldsymbol{k}_1,...,\boldsymbol{k}_N] \in \mathbb{R}^{3\times N}$, with each column vector $\boldsymbol{k}_n$ holding the Cartesian coordinates of the nth sensor, and c denotes the wave propagation speed. For the adopted L-shaped array of N = 3 antennas, $\boldsymbol{K}$ is of size 3 × 3 and has the following form: $\boldsymbol{K} = \left[[d,0,0]^T, [0,0,0]^T, [0,d,0]^T\right]^T$, where d denotes the inter-sensor spacing.

The source matrix is defined as $\boldsymbol{S}^{(m)} \triangleq [\boldsymbol{s}_1^{(m)},...,\boldsymbol{s}_R^{(m)}] \in \mathbb{C}^{Q\times R}$, where $\boldsymbol{s}_r^{(m)}$ denotes the narrowband source component with frequency f_m, associated with the rth impinging source signal. In this simulation, we generate all the narrowband source components $\{\boldsymbol{s}_r^{(m)}, r = 1,...,R, m = 1,...,M\}$ in a similar way as how the multi-set source signals are generated in *Simulation C*. The SNR is defined as in (22), where σ_s and σ_n denote the level of the signal and noise part of the array output signal associated with each filter, respectively.

We first compute the estimates of the M mixing matrices $\{\boldsymbol{A}^{(m)}, m = 1,...,M\}$ via DC-CPD based J-BSS, or CPD based BSS applied to each dataset separately. Then we further obtain M estimates of the direction matrix $\boldsymbol{P}$ by solving the equation (28) for each $\boldsymbol{A}^{(m)}$. We calculate the relative error between the true and the estimate of the direction matrix, obtained for a particular m. To evaluate the overall accuracy, we adopt as a metric the median relative error over all the M estimates. In this simulation, we set $M = 3$, and $f_1 = 0.2\,\text{MHz}$, $f_2 = 2f_1$, $f_3 = 3f_1$. The inter-sensor spacing d is set to half the smallest wavelength among all the narrowband filter outputs.

The compared algorithms are as follows: DC-CPD-ALG($K = 2$), DC-CPD-ALG($K = 3$), and DC-CPD-NLS(ALG), in which the results of DC-CPD-ALG($K = 3$) are used for the initialization. We do not include DC-CPD-NLS(Rand) in the comparison, as it turned out that most often random initialization was not good enough to find the DOAs in this simulation. As in *Simulation C*, we include SOBIUM and CPD-C in the comparison.

We first fix the number of data samples as $Q = 200000$, and let SNR vary from 0dB to 50dB. The median relative error versus SNR is plotted in Fig. 7 (a). Then we fix SNR = 20dB and let Q vary from 10^3 to 10^6. The median relative error versus Q is plotted in Fig. 7 (b). Each point of the curve is calculated as the median over 200 Monte Carlo runs.



(a) Median relative error vs. SNR

(b) Median relative error vs. number of data samples

Fig. 7. Performance of DC-CPD-ALG($K = 2$), DC-CPD-ALG($K = 3$), DC-CPD-NLS(ALG), SOBIUM and CPD-C in the extremely underdetermined wideband DOA estimation: $N = 3$, $R = 6$. The plots show (a) the median relative error vs. SNR with fixed number of data samples $Q = 200000$, and (b) the median relative error vs. the number of data samples with fixed SNR = 20dB.

From Fig. 7 we see that DC-CPD-ALG($K = 2$), SOBIUM, and CPD-C do not find the DOAs in the extremely underdetermined case. On the other hand, DC-CPD-ALG($K = 3$) does yield accurate DOA estimates for sufficiently high SNR and number of data samples Q. DC-CPD-NLS(ALG) can also find the DOAs, thanks to the effective initialization by DC-CPD-ALG($K = 3$). The optimization refines the estimates obtained with the algebraic procedure, showing the interest in using algebraic DC-CPD algorithm to initialize the optimization-based algorithm in this application. DC-CPD-NLS(Rand) does not find the correct solution in this simulation, further demonstrating the importance of the algebraic initialization. The above observations illustrate the interest in DC-CPD-ALG($K = 3$) for extremely underdetermined wideband DOA estimation.

Conclusion

DC-CPD has attracted increasing interest in J-BSS applications recently. In this study, we consider the problem of algebraic computation of DC-CPD and its uniqueness conditions. We generalize the second-order coupled rank-1 detection mapping to arbitrary order $K \geq 2$. Based on the generalized coupled rank-1 detection mapping, we propose a broad framework for the algebraic computation of DC-CPD, and present new results on deterministic and generic uniqueness conditions of DC-CPD, which are shown to be more relaxed than existing results for DC-CPD and CPD. Simulation results are provided to demonstrate the performance of the proposed DC-CPD algorithms, in comparison with existing DC-CPD and CPD algorithms.

Appendix

A. Proofs

Theorem 2:

Assuming that $\boldsymbol{X}^{(1)},...,\boldsymbol{X}^{(K)}$ are nonzero matrices, by definition we have that $\mathrm{vec}(\Phi_K^{(v,w)}(\boldsymbol{X}^{(1)},...,\boldsymbol{X}^{(K)}))$ in (9) is an all-zero vector if and only if $\boldsymbol{X}^{(v)}$ and $\boldsymbol{X}^{(w)}$ are both rank-1 and have identical column spaces. Then *Theorem 2* holds as $\psi(\boldsymbol{X}^{(1)},...,\boldsymbol{X}^{(K)})$ is defined as the concatenation of $\mathrm{vec}(\Phi_K^{(v,w)}(\boldsymbol{X}^{(1)},...,\boldsymbol{X}^{(K)}))$ for all possible values of (v,w). ■

Theorem 4:

We prove the following partial statements:

(i) $\ker(\boldsymbol{\Gamma}^{(m,n_1,...,n_K)}) = \ker((\boldsymbol{P}_{n_1,...,n_K}^{(m)})^T)$.

(ii) $\dim(\ker(\boldsymbol{\Gamma}^{(m,n_1,...,n_K)})) = R$.

(iii) The basis vectors $\boldsymbol{w}_r^{(m,n_1,...,n_k)}$ in $\ker(\boldsymbol{\Gamma}^{(m,n_1,...,n_K)})$, $r = 1,...,R$, can be written as a linear combination of vectors $\boldsymbol{b}_u^{(m,n_1)} \otimes \cdots \otimes \boldsymbol{b}_u^{(m,n_K)}$, $u = 1,\ldots,R$, where $[\boldsymbol{b}_1^{(m,n_k)},...,\boldsymbol{b}_R^{(m,n_k)}] = \boldsymbol{B}^{(m,n_k)} \triangleq (\boldsymbol{C}^{(m,n_k)}\boldsymbol{D}_C^{(m,n_k)}\boldsymbol{\Pi})^{-T} \in \mathbb{C}^{R\times R}$, $\boldsymbol{D}_C^{(m,n_k)}$ is a diagonal matrix, and $\boldsymbol{\Pi}$ is a permutation matrix that is common to all the matrices $\boldsymbol{C}^{(m,n_k)}$, $k = 1,\ldots,K$.

Note that $m, n_1,..., n_K$ are fixed in the above statements.

First, as $\boldsymbol{\Phi}_{n_1,...,n_K}^{(m)}$ is assumed to have full column rank, and $\boldsymbol{\Gamma}^{(m,n_1,...,n_K)} = \boldsymbol{\Phi}_{n_1,...,n_K}^{(m)}(\boldsymbol{P}_{n_1,...,n_K}^{(m)})^T$, we have $\ker(\boldsymbol{\Gamma}^{(m,n_1,...,n_K)}) = \ker((\boldsymbol{P}_{n_1,...,n_K}^{(m)})^T)$.

Second, as $\boldsymbol{C}^{(m,n_1)},...,\boldsymbol{C}^{(m,n_K)}$ have full column rank, the Kronecker product $\boldsymbol{C}^{(m,n_1)} \otimes \cdots \otimes \boldsymbol{C}^{(m,n_K)}$ also has full column rank. The $R^K \times (R^K - R)$ matrix $\boldsymbol{P}^{(m)}_{n_1,\ldots,n_K}$ has full column rank as well, since its columns constitute a subset of the columns of $\boldsymbol{C}^{(m,n_1)} \otimes \cdots \otimes \boldsymbol{C}^{(m,n_K)}$. By the rank nullity theorem, which states that $\dim[\ker(\boldsymbol{A})] + \operatorname{rank}(\boldsymbol{A}) = J$ for any matrix $\boldsymbol{A}$ of size $I \times J$, we have:

$$\dim[\ker((\boldsymbol{P}^{(m)}_{n_1,\ldots,n_K})^T)] = R^K - \operatorname{rank}((\boldsymbol{P}^{(m)}_{n_1,\ldots,n_K})^T) = R.$$

Third, we prove that $\boldsymbol{b}_r^{(m,n_1)} \otimes \cdots \otimes \boldsymbol{b}_r^{(m,n_K)}$ is in $\ker((\boldsymbol{P}^{(m)}_{n_1,\ldots,n_K})^T)$, $r = 1,\ldots,R$. This is equivalent to proving the following for all the values $1 \le r_1,\ldots,r_K \le R$:

$$\left(\boldsymbol{c}_{r_1}^{(m,n_1)} \otimes \cdots \otimes \boldsymbol{c}_{r_K}^{(m,n_K)}\right)^T \left(\boldsymbol{b}_r^{(m,n_1)} \otimes \cdots \otimes \boldsymbol{b}_r^{(m,n_K)}\right) = \left(\boldsymbol{c}_{r_1}^{(m,n_1)T} \boldsymbol{b}_r^{(m,n_1)}\right) \otimes \cdots \otimes \left(\boldsymbol{c}_{r_K}^{(m,n_K)T} \boldsymbol{b}_r^{(m,n_K)}\right) = 0. \tag{A1}$$

Recall that $\boldsymbol{b}_r^{(m,n_k)}$ is the rth column of $(\tilde{\boldsymbol{C}}^{(m,n_k)})^{-T}$, where $\tilde{\boldsymbol{C}}^{(m,n_k)} = \boldsymbol{C}^{(m,n_k)} \boldsymbol{D}_C^{(m,n_k)} \boldsymbol{\Pi}$. We assume the matrix $\boldsymbol{\Pi}$ permutes the columns of $\boldsymbol{C}^{(m,n_k)}$ in such a way that $(\boldsymbol{C}^{(m,n_k)}\boldsymbol{\Pi})_{:,r} = \boldsymbol{c}_u^{(m,n_1)}$, where $r = \pi(u)$, $1 \le u \le R$, and $(\pi(1),\ldots,\pi(R))$ is a permutation of $(1,\ldots,R)$. We denote $\tilde{\boldsymbol{c}}_r^{(m,n_k)}$ as the rth column of $\tilde{\boldsymbol{C}}^{(m,n_k)}$. By definition we have $\tilde{\boldsymbol{c}}_r^{(m,n_k)} = d_r^{(m,n_k)} \boldsymbol{c}_u^{(m,n_k)}$, where $d_r^{(m,n_k)}$ is the rth entry on the diagonal of $\boldsymbol{D}_C^{(m,n_k)}$.

Therefore, we have $\boldsymbol{c}_{r_k}^{(m,n_k)T} \boldsymbol{b}_r^{(m,n_k)} = 0$ if $r = \pi(u)$ and $u \ne r_k$. Since at least two of the indices $r_1,\ldots,r_K$ are distinct according to the definition of $\boldsymbol{P}^{(m)}_{n_1,\ldots,n_K}$, at least one term in $\boldsymbol{c}_{r_1}^{(m,n_1)T} \boldsymbol{b}_r^{(m,n_1)},\ldots,\boldsymbol{c}_{r_K}^{(m,n_K)T} \boldsymbol{b}_r^{(m,n_K)}$ is equal to zero, which proves (A1). Moreover, as $\boldsymbol{b}_1^{(m,n_k)},\ldots,\boldsymbol{b}_R^{(m,n_k)}$ are linearly independent, the set $\{\boldsymbol{b}_r^{(m,n_1)} \otimes \cdots \otimes \boldsymbol{b}_r^{(m,n_K)}, r = 1,\ldots,R\}$ contains R linearly independent vectors, and thus is a maximal set of linearly independent vectors of $\ker(\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)})$. Hence, the basis vectors $\boldsymbol{w}_r^{(m,n_1,\ldots,n_k)}$ of $\ker(\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)})$ can be expressed as a linear combination of $\boldsymbol{b}_r^{(m,n_1)} \otimes \cdots \otimes \boldsymbol{b}_r^{(m,n_K)}$, $r = 1,\ldots,R$. ■

B. Algebraic computation of overdetermined C-CPD (16) for K > 2

We copy (16) here:

$$\mathcal{W}^{(m,n_1,\ldots,n_K)} = \left[\!\left[\boldsymbol{B}^{(m,n_1)},\ldots,\boldsymbol{B}^{(m,n_K)}, \boldsymbol{F}^{(m,n_1,\ldots,n_K)} \right]\!\right]_R, \tag{A2}$$

where $\boldsymbol{B}^{(m,n_k)} = \tilde{\boldsymbol{C}}^{(m,n_k)-T} \in \mathbb{C}^{R \times R}$, $\tilde{\boldsymbol{C}}^{(m,n_k)} \triangleq \boldsymbol{C}^{(m,n_k)} \boldsymbol{D}_C^{(m,n)} \boldsymbol{\Pi}$, $\boldsymbol{D}_C^{(m,n)}$ is a diagonal matrix, and $\boldsymbol{\Pi}$ is a common permutation matrix for all $\boldsymbol{C}^{(m,n_k)}$, $k = 1,\ldots,K$. For all the possible values of $1 \le m \le M$ and $1 \le n_1 < \cdots < n_K \le M$, (A2) is a ($K$+1)th-order overdetermined C-CPD with coupling in the first K modes.

When $K > 2$, the tensor $\mathcal{W}^{(m,n_1,\ldots,n_K)}$ in (A2) has order (K+1) > 3. We now follow [16], in which it is explained that, under conditions that are satisfied in our derivation, the CPD of a tensor of order higher than three can equivalently be expressed as a C-CPD of third-order tensors obtained by combining modes. Briefly, for each tensor $\mathcal{W}^{(m,n_1,\ldots,n_K)}$ in (A2), we consider:

$$\boldsymbol{\mathcal{V}}_{w}^{(m,n_1,\ldots,n_K)} = [\![\boldsymbol{B}_{w}^{(m,n_1,\ldots,n_K)}, \boldsymbol{B}_{w}^{\prime(m,n_1,\ldots,n_K)}, \boldsymbol{F}^{(m,n_1,\ldots,n_K)}]\!]_R, \tag{A3}$$

where:

$$\begin{cases} \boldsymbol{B}_{w}^{(m,n_1,\ldots,n_K)} \triangleq \bigodot_{u\in\Omega_{w}^{(m,n1,\ldots,nK)}} \boldsymbol{B}^{(m,u)} \in \mathbb{C}^{\mathrm{card}(\Omega_{w}^{(m,n1,\ldots,nK)})\cdot R\times R} \\ \boldsymbol{B}_{w}^{\prime(m,n_1,\ldots,n_K)} \triangleq \bigodot_{v\in\Omega_{w}^{\prime(m,n1,\ldots,nK)}} \boldsymbol{B}^{(m,v)} \in \mathbb{C}^{\mathrm{card}(\Omega_{w}^{\prime(m,n1,\ldots,nK)})\cdot R\times R} \\ \Omega_{w}^{(m,n_1,\ldots,n_K)} \bigcup \Omega_{w}^{\prime(m,n_1,\ldots,n_K)} = \{n_1,\ldots,n_K\} \\ \Omega_{w}^{(m,n_1,\ldots,n_K)} \bigcap \Omega_{w}^{\prime(m,n_1,\ldots,n_K)} = \varnothing, \end{cases}$$

where $\Omega_{w}^{(m,n_1,\ldots,n_K)}$ and $\Omega_{w}^{\prime(m,n_1,\ldots,n_K)}$ are two complementary sets of numbers taken from $\{n_1,\ldots,n_K\}$, $w=1,\ldots,W^{(m,n_1,\ldots,n_K)}$, and $W^{(m,n_1,\ldots,n_K)}$ is the number of all the possible sets $\Omega_{w}^{(m,n_1,\ldots,n_K)}$ (or $\Omega_{w}^{\prime(m,n_1,\ldots,n_K)}$) for fixed $(m,n_1,\ldots,n_K)$, $1 \le n_1 < \ldots < n_K \le M$, $m=1,\ldots,M$. The above operations combine the modes, whose indices belong to the set $\Omega_{w}^{(m,n_1,\ldots,n_K)}$, into the first mode of $\boldsymbol{\mathcal{V}}_{w}^{(m,n_1,\ldots,n_K)}$, and combine the remaining modes into the second mode of $\boldsymbol{\mathcal{V}}_{w}^{(m,n_1,\ldots,n_K)}$. All the tensors $\boldsymbol{\mathcal{V}}_{w}^{(m,n_1,\ldots,n_K)}$ with fixed $(m,n_1,\ldots,n_K)$ and different w are coupled in the third mode, and this C-CPD can be computed by a GEVD, as explained in [16]. Therefore, any of the (K+1)th-order CPDs can be computed by GEVD, which yields us the factor matrices $\boldsymbol{B}^{(m,n_1)},\ldots,\boldsymbol{B}^{(m,n_K)}$, for any choice of $(m,n_1,\ldots,n_K)$.

C. An efficient implementation of the proposed algorithm

We note that the main complexity of the algebraic DC-CPD algorithm, as it has been presented in the manuscript, is in the construction of the $(\mathrm{C}_K^2\, N^{2K})\times R^K$ matrices $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$ and the calculation of the basis vectors in their null space. Here we present an efficient way to calculate the basis vectors. In this implementation, we do not explicitly construct $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$. Instead, we calculate the $R^K \times R^K$ matrices $\boldsymbol{\Omega}^{(m,n_1,\ldots,n_K)} \triangleq \boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)H} \boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$. We note that $\ker(\boldsymbol{\Omega}^{(m,n_1,\ldots,n_K)}) = \ker(\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)})$, while $\boldsymbol{\Omega}^{(m,n_1,\ldots,n_K)}$ have smaller size and can be computed in a more efficient way than $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$, as will be explained later. Hence, in an efficient implementation we use $\boldsymbol{\Omega}^{(m,n_1,\ldots,n_K)}$ instead of $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$.

More precisely, for each $m,n_1,\ldots,n_K$, we denote $\boldsymbol{\Omega}^{(m,n_1,\ldots,n_K)} = \boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)H} \boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$, where $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$ holds $\psi(\boldsymbol{T}_{(:,:,r_1)}^{(m,n_1)},\ldots,\boldsymbol{T}_{(:,:,r_K)}^{(m,n_K)})$ as its columns. We can express $\boldsymbol{\Omega}^{(m,n_1,\ldots,n_K)}$ as:

$$\boldsymbol{\Omega}^{(m,n_1,\ldots,n_K)} = \sum_{v,w} \boldsymbol{\Omega}_{(v,w)}^{(m,n_1,\ldots,n_K)}, \tag{A4}$$

where

$$\left(\boldsymbol{\Omega}_{(v,w)}^{(m,n_1,\ldots,n_K)}\right)_{\tilde{s},\tilde{u}} = \psi_K^{(v,w)H}(\boldsymbol{T}_{(:,:,s_1)}^{(m,n_1)},\ldots,\boldsymbol{T}_{(:,:,s_K)}^{(m,n_K)})\psi_K^{(v,w)}(\boldsymbol{T}_{(:,:,u_1)}^{(m,n_1)},\ldots,\boldsymbol{T}_{(:,:,u_K)}^{(m,n_K)}), \tag{A5}$$

where $\tilde{s}=\sum_{k=1}^{K-1}(s_k-1)R^{K-k-1}+1$, $\tilde{u}=\sum_{k=1}^{K-1}(u_k-1)R^{K-k-1}+1$, and $\psi_K^{(v,w)}(\cdot)$ is the vector representation of $\Phi_K^{(v,w)}(\cdot)$ defined in (8), $1\le v<w\le K$. According to (8) and (9), we have:

$$\left(\boldsymbol{\Omega}_{(v,w)}^{(m,n_1,\ldots,n_K)}\right)_{\tilde{s},\tilde{u}}=2\prod_{l=1}^{K}\boldsymbol{t}_{s_l}^{(m,n_l)H}\boldsymbol{t}_{u_l}^{(m,n_l)}-\left(\prod_{l'\notin\{v,w\}}\boldsymbol{t}_{s_{l'}}^{(m,n_{l'})H}\boldsymbol{t}_{u_{l'}}^{(m,n_{l'})}\right)$$
$$\cdot\sum_{g,h}\left[\left(\boldsymbol{\tau}_{g,s_v}^{(m,n_v)H}\boldsymbol{\tau}_{h,u_v}^{(m,n_v)}\right)\left(\boldsymbol{\tau}_{h,s_w}^{(m,n_w)H}\boldsymbol{\tau}_{g,u_w}^{(m,n_w)}\right)+\left(\boldsymbol{\tau}_{h,s_v}^{(m,n_v)H}\boldsymbol{\tau}_{g,u_v}^{(m,n_v)}\right)\left(\boldsymbol{\tau}_{g,s_w}^{(m,n_w)H}\boldsymbol{\tau}_{h,u_w}^{(m,n_w)}\right)\right], \tag{A6}$$

where vectors $\boldsymbol{t}_r^{(m,n_l)}=\mathrm{vec}(\boldsymbol{T}_{(:,:,r)}^{(m,n_l)})$ have length N^2, and vectors $\boldsymbol{\tau}_{k,r}^{(m,n_l)}=\boldsymbol{T}_{(k,:,r)}^{(m,n_l)}$ have length N, l = 1,…,K, $k\in\{g,h\}$, $r\in\{s_1,\ldots,s_K,u_1,\ldots,u_K\}$. We construct $R\times R$ Hermitian matrices $\boldsymbol{Z}^{(m,n_l)}\triangleq\boldsymbol{T}_3^{(m,n_l)H}\boldsymbol{T}_3^{(m,n_l)}$, and $NR\times NR$ Hermitian matrices $\boldsymbol{Y}^{(m,n_i)}=\boldsymbol{T}_2^{(m,n_i)}\boldsymbol{T}_2^{(m,n_i)H}$, where $i\in\{v,w\}$. Index permutation yields the following matrices:

$$\left[\boldsymbol{Y}_1^{(m,n_i)}\right]_{(s_i-1)R+u_i,(g-1)N+h}=\left[\boldsymbol{Y}^{(m,n_i)}\right]_{(h-1)R+u_i,(g-1)R+s_i},\left[\boldsymbol{Y}_2^{(m,n_i)}\right]_{(s_i-1)R+u_i,(h-1)N+g}=\left[\boldsymbol{Y}^{(m,n_i)}\right]_{(h-1)R+u_i,(g-1)R+s_i}. \tag{A7}$$

Then we have:

$$\left(\boldsymbol{\Omega}_{(v,w)}^{(m,n_1,\ldots,n_K)}\right)_{\tilde{s},\tilde{u}}=2\prod_{l=1}^{K}\left(\boldsymbol{Z}^{(m,n_l)}\right)_{s_l,u_l}-\prod_{l'\notin\{v,w\}}\left(\boldsymbol{Z}^{(m,n_l)}\right)_{s_{l'},u_{l'}}\cdot\left(\boldsymbol{Y}_1^{(m,n_v)}\boldsymbol{Y}_2^{(m,n_w)T}+\boldsymbol{Y}_2^{(m,n_v)}\boldsymbol{Y}_1^{(m,n_w)T}\right)_{(s_v-1)R+u_v,(s_w-1)R+u_w}. \tag{A8}$$

Therefore, we could precompute $\boldsymbol{Z}^{(m,n_l)}$ and $\boldsymbol{Y}^{(m,n_i)}$, where $l=1,\ldots,K$, $i\in\{v,w\}$, and construct Hermitian $R^K\times R^K$ matrices $\boldsymbol{\Omega}_{(v,w)}^{(m,n_1,\ldots,n_K)}$ according to (A8).

D. Analysis of complexity and memory requirements

We first analyze the complexity and memory requirements of the original version of the algebraic DC-CPD algorithm, and then provide results related to its efficient implementation.

We recall that the main complexity of the original algebraic DC-CPD algorithm, as it has been presented, is in the construction of matrices $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$ and the calculation of the basis vectors in their null space. We have according to (8) and (9), that the complexity of the construction of each column of $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$ is $\mathrm{O}(\mathrm{C}_K^2KN^{2K})$ flops, and thus the overall complexity of the construction of $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$ is $\mathrm{O}(\mathrm{C}_K^2KN^{2K}R^K)$ flops. The basis vectors in ker($\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$) are computed as singular vectors. This step has complexity $\mathrm{O}(2\mathrm{C}_K^2N^{2K}R^{2K}+11R^{3K})$ flops. We construct and manipulate $M\mathrm{C}_M^K$ such matrices $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$. Therefore, the overall complexity is $\mathrm{O}(M\mathrm{C}_M^K(\mathrm{C}_K^2KN^{2K}R^K+2\mathrm{C}_K^2N^{2K}R^{2K}+11R^{3K}))$ flops. The memory requirements of the algorithm are mainly in the storage of the $M\mathrm{C}_M^K$ tensors $\boldsymbol{\mathcal{W}}^{(m,n_1,\ldots,n_K)}$ and the matrix $\boldsymbol{\Gamma}^{(m,n_1,\ldots,n_K)}$. Therefore, the memory requirements are $\mathrm{O}(M\mathrm{C}_M^K(R^{(K+1)}+\mathrm{C}_K^2N^{2K}R^K))$ complex numbers.

In the efficient implementation, the complexity of computing a single Hermitian matrix $\boldsymbol{Z}^{(m,n_l)}$ is $\mathrm{O}(0.5N^2R^2)$ flops, and the complexity of computing a single Hermitian matrix $\boldsymbol{Y}^{(m,n_l)}$, is $\mathrm{O}(0.5N^3R^2)$ flops. The additional complexity of computing a single Hermitian matrix $\boldsymbol{\Omega}_{(v,w)}^{(m,n_1,\ldots,n_K)}$ by (A8) is $\mathrm{O}(0.5R^{2K}+0.5N^2R^{2K})$ flops. To compute $\boldsymbol{\Omega}^{(m,n_1,\ldots,n_K)}$, we need to construct C_K^2 matrices $\boldsymbol{\Omega}_{(v,w)}^{(m,n_1,\ldots,n_K)}$, and thus the complexity of computing a single Hermitian matrix $\boldsymbol{\Omega}^{(m,n_1,\ldots,n_K)}$ is $\mathrm{O}(0.5\mathrm{C}_K^2R^{2K}+0.5\mathrm{C}_K^2N^2R^{2K})$. The complexity of finding the basis vectors of $\ker(\boldsymbol{\Omega}^{(m,n_1,\ldots,n_K)})$ is $\mathrm{O}(2R^{3K})$ if they are computed by modified Gram-Schmidt orthogonalization. If we use EVD, then the computational cost is $\mathrm{O}(6R^{3K})$ flops.

For the indices $m=1,\ldots,M, 1\le n_1<\cdots<n_K\le M$, the algebraic DC-CPD overall requires the construction of M^2 matrices $\boldsymbol{Z}^{(m,l)}$ of size $R\times R$, M^2 matrices $\boldsymbol{Y}^{(m,l)}$ of size $NR\times NR$, $M\mathrm{C}_M^K$ matrices $\boldsymbol{\Omega}^{(m,n_1,\ldots,n_K)}$ of size $R^K\times R^K$, and the calculation of the R basis vectors of $\ker(\boldsymbol{\Omega}^{(m,n_1,\ldots,n_K)})$. The overall complexity is $\mathrm{O}(0.5M^2N^2R^2+0.5M^2N^3R^2+0.5M\mathrm{C}_M^K\mathrm{C}_K^2R^{2K}+0.5M\mathrm{C}_M^K\mathrm{C}_K^2N^2R^{2K}+2M\mathrm{C}_M^KR^{3K})$ flops. For the memory requirements we consider the storage of the matrices $\boldsymbol{Z}^{(m,l)},\boldsymbol{Y}^{(m,l)},\boldsymbol{\Omega}^{(m,n_1,\ldots,n_K)}$, and the tensors $\boldsymbol{\mathcal{W}}^{(m,n_1,\ldots,n_K)}$ of size $R\times\cdots\times R$ (16). Hence, we need to store $\mathrm{O}(M^2R^2+M^2N^2R^2+M\mathrm{C}_M^KR^{2K}+M\mathrm{C}_M^KR^{K+1})$ complex numbers. The above expressions for complexity and memory requirements can be simplified to $\mathrm{O}(0.5M\mathrm{C}_M^K\mathrm{C}_K^2N^2R^{2K}+2M\mathrm{C}_M^KR^{3K})$ flops and $\mathrm{O}(M^2N^2R^2+M\mathrm{C}_M^KR^{2K})$ complex numbers, respectively. Comparing with the original algebraic DC-CPD algorithm, we note that the efficient implementation indeed has much reduced complexity and memory requirements.

Acknowledgment

The authors would like to express their sincere gratitude to Dr. Ignat Domanov and Dr. Mikael Sorensen for their insightful comments on the algebraic DC-CPD algorithm, and to Prof. Dr. You-Gen Xu for his constructive suggestions on the extremely underdetermined wideband DOA estimation application.

References

[1] Y. -O. Li, T. Adalı, W. Wang, V. Calhoun, "Joint blind source separation by multiset canonical correlation analysis," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3918–3929, Oct. 2009.

[2] X. -L. Li, T. Adalı, M. Anderson, "Joint blind source separation by generalized joint diagonalization of cumulant matrices," *Signal Process.*, vol. 91, no. 10, pp. 2314–2322, Oct. 2011.

[3] M. Congedo, R. Phlypo. J. Chatel-Goldman, "Orthogonal and non-orthogonal joint blind source separation in the least-squares sense," in *Proc. EUSIPCO'2012*, Bucharest, Romania, Aug. 27–31, 2012.

[4] X. -F. Gong, X. -L. Wang, Q. -H. Lin, "Generalized non-orthogonal joint diagonalization with LU decomposition and successive rotations," *IEEE Trans. Signal Process.*, vol. 63, no. 5, pp. 1322–1334, Mar. 2015.

[5] X. -F. Gong, L. Mao, Y. -L. Liu, Q. -H. Lin, "A Jacobi generalized orthogonal joint diagonalization algorithm for joint blind source separation," *IEEE Access*, vol. 6, pp. 38464–38474, Jun. 2018.

[6] D. Lahat, C. Jutten, "A new link between joint blind source separation using second order statistics and the canonical polyadic decomposition," in *Proc. LVA/ICA'2018*, Guildford, England, Jul. 2–6, 2018, pp. 171–180.

[7] L. Zou, X. Chen, X. Ji, Z. Wang, "Underdetermined joint blind source separation of multiple datasets," *IEEE Access*, vol. 5, pp. 7474–7487, Apr. 2017.

[8] L. Sorber, M. Van Barel, L. De Lathauwer, "Structured data fusion," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 586–600, Jun. 2015.

[9] N. Vervliet. O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, *Tensorlab 3.0*, Mar. 2016. [Online], Available: http://www.tensorlab.net/.

[10] E. Acar, R. Bro, A. K. Smilde, "Data fusion in metabolomics using coupled matrix and tensor factorizations," *Proc. IEEE*, vol. 103, no. 9, pp. 1602–1620, Sep. 2015.

[11] M. Sorensen, L. De Lathauwer, "Multiple invariance ESPRIT for nonuniform linear arrays: a coupled canonical polyadic decomposition approach," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3693–3704, Jul. 2016.

[12] Y.-N. Yu, A. Petropulu, "PARAFAC-based blind estimation of possibly underdetermined convolutive MIMO systems," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 111–124, Jan. 2008.

[13] H. Becker, P. Comon, L. Albera, "Tensor-based preprocessing of combined EEG/MEG data," in *Proc. EUSIPCO'2012*, Bucharest, Romania, Aug. 27–31, 2012, pp. 275–279.

[14] X.-F. Gong, Y. -N. Hao, Q. -H. Lin, "Joint canonical polyadic decomposition of two tensors with one shared loading matrix" in *Proc. MLSP'2013*, Southampton, U. K., Sep. 22–25, 2013.

[15] T. Yokota, A. Cichocki, Y. Yamashita, "Linked PARAFAC / CP tensor decomposition and its fast implementation for multi-block tensor analysis," in *Proc. ICONIP'2012*, Doha, Qatar, Nov. 12–15, 2012, pp. 84–92.

[16] M. Sorensen, L. De Lathauwer, "Coupled canonical polyadic decompositions and (coupled) decompositions in multilinear rank-($L_{r,n}$, $L_{r,n}$, 1) terms—Part I: uniqueness," *SIAM J. Matrix Anal. Appl.*, vol. 36, no. 2, pp. 496–522, May 2015.

[17] M. Sorensen, I. Domanov, L. De Lathauwer, "Coupled canonical polyadic decompositions and (coupled) decompositions in multilinear rank-($L_{r,n}$, $L_{r,n}$, 1) terms—Part II: algorithms," *SIAM J. Matrix Anal. Appl.*, vol. 36, no. 3, pp. 1015–1045, Jul. 2015.

[18] M. Sorensen, L. De Lathauwer, "Multidimensional harmonic retrieval via coupled canonical polyadic decompositions—Part I: model and identifiability," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 517–527, Jan. 2017.

[19] M. Sorensen, L. De Lathauwer, “Multidimensional harmonic retrieval via coupled canonical polyadic decompositions—Part II: algorithm and multirate sampling,” *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 528–539, Jan. 2017.

[20] M. Sorensen, L. De Lathauwer, “Coupled tensor decompositions in array processing,” ESAT-STADIUS, KU Leuven, Belgium, Tech. Rep. 13–241, 2014.

[21] B. Rivet, M. Duda, A. Guérin-Dugué, C. Jutten, P. Comon, “Multimodal approach to estimate the ocular movements during EEG recordings: a coupled tensor factorization method,” in *Proc. EMBC'2015*, Milan, Italy, Aug. 25–29, 2015.

[22] R. Cabral Farias, J. E. Cohen, P. Comon, “Exploring multimodal data fusion through joint decompositions with flexible couplings,” *IEEE Trans. Signal Process.,* vol. 64, no. 18, pp. 4830–4844, Sept. 2016.

[23] X.-F. Gong, Q. -H. Lin, C.-F. Yu, L. De Lathauwer, “Double coupled canonical polyadic decomposition for joint blind source separation,” *IEEE Trans. Signal Process.,* vol. 66, no. 13, pp. 3475–3490, Apr. 2018.

[24] J.-F. Cardoso, “Super-symmetric decomposition of the fourth-order cumulant tensor. Blind identification of more sources than sensors,” in *Proc. ICASSP'1991*, Toronto, Canada, May. 14–17, 1991, pp. 3109– 3112.

[25] L. De Lathauwer, “A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization,” *SIAM J. Matrix Anal. Appl*., vol. 28, no. 3, pp. 642–666, Sep. 2006.

[26] I. Domanov, L. De Lathauwer, “On the uniqueness of the canonical polyadic decomposition of third-order tensors—Part I: basic results and uniqueness of one factor matrix,” *SIAM J. Matrix Anal. Appl*., vol. 34, no. 3, pp. 855–875, Jul. 2013.

[27] I. Domanov, L. De Lathauwer, “On the uniqueness of the canonical polyadic decomposition of third-order tensors—Part II: uniqueness of the overall decomposition,” *SIAM J. Matrix Anal. Appl*., vol. 34, no. 3, pp. 876–903, Jul. 2013.

[28] I. Domanov, L. De Lathauwer, “Canonical polyadic decomposition of third-order tensors: reduction to generalized eigenvalue decomposition,” *SIAM J. Matrix Anal. Appl*., vol. 35, no. 2, pp. 636–660, May 2014.

[29] I. Domanov, L. De Lathauwer, “Canonical polyadic decomposition of third-order tensors: relaxed uniqueness conditions and algebraic algorithm,” *Linear Algebra Appl*., vol. 513, pp. 342–375, Jan. 2017.

[30] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, C. Faloutsos, “Tensor decomposition for signal processing and machine learning,” *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp.3551–3582, Apr. 2017.

[31] E. Sanchez, L. S. Ramos, B. R. Kowalski, “Generalized rank annihilation method : I. Application to liquid chromatography—diode array ultraviolet detection data,” *J. Chromatogr. A*, vol. 385, no. 1, pp. 151–164, Jan. 1987.

[32] M. Sorensen, L. De Lathauwer, “New uniqueness conditions for the canonical polyadic decomposition of third-order tensors,” *SIAM J. Matrix Anal. Appl*., vol. 36, no. 4, pp. 1381–1403, Oct. 2015.

[33] L. De Lathauwer, B. De Moor, J. Vandewalle, “On the best rank-1 and rank-(R_1, R_2,…, R_N) approximation of higher-order

tensors," *SIAM J. Matrix Anal. Appl*., vol. 21, no. 4, pp. 1324–1342, May 2000.

[34] R. Gunning, H. Rossi, "Holomorphic functions" in *Analytic functions of several complex variables.* U.S.: Prentice-Hall, Inc, Englewood Cliffs, N.J., 1965, pp. 1–63.

[35] L. De Lathauwer, J. Castaing, "Blind identification of underdetermined mixtures by simultaneous matrix diagonalization," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp.1096–1105, Mar. 2008.

[36] E. D. Claudio, R. Parisi, "WAVES: weighted average of signal subspaces for robust wideband direction finding," *IEEE Trans. Signal Process*., vol. 49, no. 10, pp. 2179–2191, Oct. 2001.

[37] X. -F. Gong, J. -C. Jiang, H. Li, Y. -G. Xu, Z. -W. Liu, "Spatially spread dipole/loop quint for vector-cross-product-based direction finding and polarisation estimation," *IET Signal Process*., vol. 12, no. 5, pp. 636 – 642, June 2018.