An Alternating Direction Method of Multipliers for Constrained Joint Diagonalization by Congruence

(Invited Paper)

Lu Wang*†§, Laurent Albera*†‡, Lotfi Senhadji*† and Jean-Christophe Pesquet¶

*Inserm, U1099, Rennes F-35000, France.
†Université de Rennes 1, LTSI, Rennes F-35000, France.
‡Inria, Centre Inria Rennes - Bretagne Atlantique, Rennes, F-35000, France.
§Information Technology Head Office, Guotai Junan Securities Co., Ltd. 200120 Shanghai, China.
¶Université Paris-Est, Laboratoire d'Informatique Gaspard Monge, CNRS-UMR 8049, Marne-la-Vallée, France.

***Abstract*—In this paper, we address the problem of joint diagonalization by congruence (i.e. the canonical polyadic decomposition of semi-symmetric 3rd order tensors) subject to arbitrary convex constraints. Sufficient conditions for the existence of a solution are given. An efficient algorithm based on the Alternating Direction Method of Multipliers (ADMM) is then designed. ADMM provides an elegant approach for handling the additional constraint terms, while taking advantage of the structure of the objective function. Numerical tests on simulated matrices show the benefits of the proposed method for low signal to noise ratios. Simulations in the context of nuclear magnetic resonance spectroscopy are also provided.**

I. Introduction

For more than one decade, the Joint Diagonalization by Congruence (JDC) problem (i.e. the canonical polyadic decomposition of semi-symmetric 3rd order tensors) has attracted much attention in the signal processing community [2]. The problem can be simply formulated as follows:

Problem 1. ***(JDC problem)*** *Given K symmetric matrices $\boldsymbol{C}^{(k)}$ in $\mathbb{R}^{N\times N}$, find a joint transformation matrix $\boldsymbol{A}\in\mathbb{R}^{N\times P}$ and K diagonal matrices $\boldsymbol{D}^{(k)}$ in $\mathbb{R}^{P\times P}$ such that*

$$(\forall k\in\{1,\ldots,K\}) \quad \boldsymbol{C}^{(k)}=\boldsymbol{A}\boldsymbol{D}^{(k)}\boldsymbol{A}^{\mathsf{T}}. \tag{1}$$

The JDC problem has been mostly handled as an optimization problem. Depending on the optimization criterion, various strategies can be devised to solve it [2]. A popular approach is to employ a direct-fit cost function measuring the squared difference between the K given matrices $\boldsymbol{C}^{(k)}$ and their assumed model (1) as a function of the joint transformation matrix $\boldsymbol{A}$ and the K diagonal matrices $\boldsymbol{D}^{(k)}$. This criterion is expressed as:

$$\Psi_1\big(\boldsymbol{A},(\boldsymbol{D}^{(k)})_{1\leq k\leq K}\big)=\tfrac{1}{2}\textstyle\sum_{k=1}^{K}||\boldsymbol{C}^{(k)}-\boldsymbol{A}\boldsymbol{D}^{(k)}\boldsymbol{A}^{\mathsf{T}}||_{\mathrm{F}}^2. \tag{2}$$

Although efficient algorithms have been developed to solve the corresponding optimization problem [2], [21], they may fail to deliver reliable estimates especially when the K observed matrices $\boldsymbol{C}^{(k)}$ are corrupted by some noise (e.g. estimation errors, outliers, ...), the dimensionality of the problem is high (i.e. N or P are large), or K is small (which may make the problem underdetermined). In the general context of inverse problems, a classical way of circumventing this problem consists in adding constraints on the sought parameters. To be meaningful such constraints should reflect some prior information or desirable property concerning the JDC problem. It turns out that in a number of cases of practical interest such prior information is available. The objective of this paper is therefore to recast the JDC problem as a Joint Diagonalization by Congruence under Constraints (JD2C) problem which is formulated as follows:

Problem 2. ***(JD2C problem)*** *Given K matrices $\boldsymbol{C}^{(k)}$ in $\mathbb{R}^{N\times N}$, find a matrix $\boldsymbol{A}$ and K matrices $\boldsymbol{D}^{(k)}$ minimizing Ψ_1 in (2) subject to the following constraints:*

(i) $\boldsymbol{A}\in\mathcal{A}$;
(ii) $(\forall k\in\{1,\ldots,K\})\quad \boldsymbol{D}^{(k)}\in\mathcal{D}$;

where $\mathcal{A}$ is a nonempty closed subset of $\mathbb{R}^{N\times P}$ and $\mathcal{D}$ is a nonempty closed subset of the space of diagonal matrices of $\mathbb{R}^{P\times P}$.

As an example, the set $\mathcal{A}$ can be equal to $[A_{\min},A_{\max}]^{N\times P}$ with $(A_{\min},A_{\max})\in\mathbb{R}^2$ and $A_{\min}\leq A_{\max}$. Then, the nonnegativity constraint can be viewed as a limit case when $A_{\min}=0$ and $A_{\max}\to+\infty$. Its use in the context of Nonnegative JDC (NJDC) was investigated through derivative-based [4], [5], alternating least squares [5], [19] and Jacobi-like [20] optimization algorithms. Regarding the choice of $\mathcal{D}$, because of the imposed diagonal structure for the matrices $(\boldsymbol{D}^{(k)})_{1\leq k\leq K}$, this set must have the form $\mathcal{D}=\mathrm{Diag}(\mathcal{V})$, where $\mathcal{V}$ is a nonempty closed convex subset of $\mathbb{R}^P$. Various choices for $\mathcal{V}$ can be envisaged such as $\mathcal{V}=[D_{\min},D_{\max}]^P$ with $(D_{\min},D_{\max})\in\mathbb{R}^2$ and $D_{\min}\leq D_{\max}$.

The aim of this paper is twofold. First we give sufficient conditions for the existence of a solution to the JD2C problem. Secondly, we show how to compute efficiently this solution based on the Alternating Direction Method of Multipliers (ADMM) [1]. Contrarily to Problem 2, in the algorithmic part, it will be assumed that $\mathcal{A}$ and $\mathcal{D}$ are convex sets. Numerical experiments on simulated matrices show the good performance of the proposed method, especially in the case of degeneracies such as for low Signal-to-Noise Ratio (SNR) values. Eventually, the behavior of the proposed method is illustrated in the context of blind separation of Nuclear Magnetic Resonance

Spectroscopy (NMRS) signals.

II. Existence of a solution to the JD2C problem

Although our purpose in this paper is mainly to focus on algorithmic issues, an important question is to know whether the existence of a solution to Problem 2 is guaranteed. We now state sufficient conditions for the existence of such a solution:

Proposition 1. *Assume that one of the following conditions is satisfied:*

- (i) *there exist a matrix $\boldsymbol{A} \in \mathcal{A}$ and a set of K matrices $\boldsymbol{D}^{(k)} \in \mathcal{D}^K$ such that (1) is satisfied;*
- (ii) *$\mathcal{A}$ and $\mathcal{D}$ are bounded;*
- (iii) *$\mathcal{A} = [0, +\infty[^{N\times P}$ and $\mathcal{D} = \mathrm{Diag}([0, +\infty[^P)$;*
- (iv) *$\mathcal{A}$ is a subset of the cone of full column rank matrices of $\mathbb{R}^{N\times P}$ and $\mathcal{D}$ is a subset of $\mathrm{Diag}(\mathbb{R}^P)$ which contains the null matrix.*

Then, the JD2C problem admits a solution.

A proof of this proposition will be provided in an extended version of this paper. Now let us assume that at least one of the conditions (i)-(iv) is fulfilled. Our objective in the remainder of this paper is then to design an algorithm based on ADMM. As we will show, ADMM provides a suitable approach for handling the additional constraint terms, while taking advantage of the structure of the objective function.

III. The JD2C$_{\text{ADMM}}$ algorithm

A. The alternating direction method of multipliers

Lagrange multiplier techniques constitute powerful optimization tools [16]. ADMM [1] is a splitting method which combines a dual ascent step [11] with proximal steps [7] ensuring good convergence properties. By duality arguments, ADMM is also strongly related to the Douglas-Rachford method, another famous optimization algorithm [6], [10]. In its common use, ADMM aims at solving structured convex problems of the form:

$$\underset{\boldsymbol{x}\in\mathcal{H},\boldsymbol{y}\in\mathcal{G}}{\text{minimize}}\ \mathrm{f}(\boldsymbol{x}) + \mathrm{g}(\boldsymbol{y}) \quad \text{subject to } \boldsymbol{P}\boldsymbol{x} - \boldsymbol{Q}\boldsymbol{y} = \boldsymbol{c} \tag{3}$$

where $\mathrm{f}\colon \mathcal{H} \to]-\infty, +\infty[$ and $\mathrm{g}\colon \mathcal{G} \to]-\infty, +\infty[$ are two lower-semicontinuous convex functions with nonempty domains, $\boldsymbol{P}\colon \mathcal{H} \to \mathcal{K}$ and $\boldsymbol{Q}\colon \mathcal{G} \to \mathcal{K}$ are two bounded linear operators, $\boldsymbol{c} \in \mathcal{K}$, and $\mathcal{H}$, $\mathcal{G}$, and $\mathcal{K}$ are three real Hilbert spaces. Let us assume that the above minimization problem admits a solution and that $\boldsymbol{P}^*\boldsymbol{P}$ and $\boldsymbol{Q}^*\boldsymbol{Q}$ are isomorphisms, where $\boldsymbol{P}^*$ and $\boldsymbol{Q}^*$ are the adjoint operator of $\boldsymbol{P}$ and $\boldsymbol{Q}$, respectively. The augmented Lagrangian function corresponding to (3) is formed as follows, for every $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) \in \mathcal{H}\times\mathcal{G}\times\mathcal{K}$:

$$\begin{aligned}\mathrm{L}_\rho(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) = \mathrm{f}(\boldsymbol{x}) + \mathrm{g}(\boldsymbol{y}) + \langle \boldsymbol{\lambda} \mid \boldsymbol{P}\boldsymbol{x} - \boldsymbol{Q}\boldsymbol{y} - \boldsymbol{c}\rangle_{\mathcal{K}} \\ + \tfrac{\rho}{2}||\boldsymbol{P}\boldsymbol{x} - \boldsymbol{Q}\boldsymbol{y} - \boldsymbol{c}||^2_{\mathcal{K}}\end{aligned} \tag{4}$$

where $\boldsymbol{\lambda}$ is a Lagrangian multiplier vector, $\rho \in]0, +\infty[$ is a penalty parameter, $\langle\cdot \mid \cdot\rangle_{\mathcal{K}}$ is the scalar product of $\mathcal{K}$ and $\|\cdot\|_{\mathcal{K}}$ the associated norm. The ADMM algorithm includes an $\boldsymbol{x}$-minimization step, a $\boldsymbol{y}$-minimization step, and a dual variable update step. In the convex case, the convergence of ADMM is guaranteed under mild qualification conditions (see [1], [7] and the references therein). The fact that $\boldsymbol{P}^*\boldsymbol{P}$ and $\boldsymbol{Q}^*\boldsymbol{Q}$ are isomorphisms plays a prevalent role in the convergence proof by ensuring the strong convexity of $\mathrm{L}_\rho(\,\cdot, \boldsymbol{y}, \boldsymbol{\lambda}\,)$ and $\mathrm{L}_\rho(\,\boldsymbol{x}, \cdot, \boldsymbol{\lambda}\,)$, for every $\boldsymbol{x} \in \mathcal{H}$, $\boldsymbol{y} \in \mathcal{G}$, and $\boldsymbol{\lambda} \in \mathcal{K}$.

Works have been devoted to more general scenarios concerning the minimization of a sum of more than two functions, i.e.:

$$\underset{\substack{(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_J)\in\\ \mathcal{H}_1\times\ldots\times\mathcal{H}_J}}{\text{minimize}}\ \sum_{j=1}^{J} \mathrm{f}_j(\boldsymbol{x}_j) \quad \text{subject to } \sum_{i=1}^{J} \boldsymbol{Q}_j\boldsymbol{x}_j = \boldsymbol{c} \tag{5}$$

where, for every $j \in \{1, \ldots, J\}$, $\mathrm{f}_j\colon \mathcal{H}_j \to]-\infty, +\infty]$ is a lower-semicontinuous convex function with a nonempty domain, $\boldsymbol{Q}_j\colon \mathcal{H}_j \to \mathcal{K}$ is a bounded linear operator such that $\boldsymbol{Q}_j^*\boldsymbol{Q}_j$ is an isomorphism, and $\mathcal{H}_j$ is a real Hilbert space. The augmented Lagrangian function then reads, for every $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_J, \boldsymbol{\lambda}) \in \mathcal{H}_1 \times \ldots \times \mathcal{H}_J \times \mathcal{K}$:

$$\begin{aligned}\mathrm{L}_\rho(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_J, \boldsymbol{\lambda}) = \textstyle\sum_{j=1}^J \mathrm{f}_j(\boldsymbol{x}_j) \\ + \langle \boldsymbol{\lambda} \mid \textstyle\sum_{i=1}^J \boldsymbol{Q}_j\boldsymbol{x}_j - \boldsymbol{c}\rangle_{\mathcal{K}} + \tfrac{\rho}{2}||\textstyle\sum_{i=1}^J \boldsymbol{Q}_j\boldsymbol{x}_j - \boldsymbol{c}||^2_{\mathcal{K}}.\end{aligned} \tag{6}$$

A natural extension of the standard ADMM thus corresponds to the following operations performed at iteration n:

$$\begin{aligned}\boldsymbol{x}_{1,n+1} &= \underset{\boldsymbol{x}'_1\in\mathcal{H}_1}{\operatorname{argmin}}\ \mathrm{L}_\rho(\,\boldsymbol{x}'_1, \boldsymbol{x}_{n,2}, \ldots, \boldsymbol{x}_{n,J}, \boldsymbol{\lambda}_n\,) \\ &\vdots \\ \boldsymbol{x}_{J,n+1} &= \underset{\boldsymbol{x}'_J\in\mathcal{H}_J}{\operatorname{argmin}}\ \mathrm{L}_\rho(\,\boldsymbol{x}_{n+1,1}, \ldots, \boldsymbol{x}_{n+1,J-1}, \boldsymbol{x}'_J, \boldsymbol{\lambda}_n\,) \\ \boldsymbol{\lambda}_{n+1} &= \boldsymbol{\lambda}_n + \rho(\,\textstyle\sum_{i=1}^J \boldsymbol{Q}_j\boldsymbol{x}_{j,n+1} - \boldsymbol{c}\,).\end{aligned} \tag{7}$$

It was however proved recently in [3] that this algorithm may fail to converge and it was proposed in [18] a randomized version of it, which was shown to converge in expectation under some technical assumptions. Note that the use of such kind of randomized block-coordinate fixed point algorithms was investigated in [8] where almost sure convergence properties were established. Basically, the randomized form of ADMM is similar to (7) but, instead of updating the components according to their index order $1, \ldots, J$, they are updated following the order $\sigma_n(1), \ldots, \sigma_n(J)$, where $\sigma_n\colon \{1, \ldots, J\} \to \{1, \ldots, J\}$ is a permutation which is randomly chosen at each iteration n. Finally, it can be noticed that another technique for dealing with criteria split in a sum of more than two functions is to resort to parallelized versions of ADMM [12], [15].

B. Algorithm derivation

In order to facilitate the derivation of an efficient alternating scheme for the direct-fit cost function (2), we introduce auxiliary variables $\boldsymbol{A}^{(1)} \in \mathbb{R}^{N\times P}$, $\boldsymbol{A}^{(2)} \in \mathbb{R}^{N\times P}$, $\boldsymbol{U} \in \mathbb{R}^{N\times P}$, and $(\widetilde{\boldsymbol{D}}^{(k)})_{1\le k\le K} \in \big(\mathrm{Diag}(\mathbb{R}^P)\big)^K$. Then, the JD2C problem can be reformulated as the following minimization problem:

$$\begin{aligned}\underset{\substack{(\boldsymbol{D}^{(k)},\widetilde{\boldsymbol{D}}^{(k)})_{1\le k\le K},\\ \boldsymbol{A}^{(1)},\boldsymbol{A}^{(2)},\boldsymbol{U}}}{\text{minimize}}\quad & \Psi_2\big(\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{U}, (\boldsymbol{D}^{(k)}, \widetilde{\boldsymbol{D}}^{(k)})_{1\le k\le K}\big) \\ \text{subject to } & \begin{cases}\boldsymbol{A}^{(i)} = \boldsymbol{U}, & i \in \{1, 2\} \\ \boldsymbol{D}^{(k)} = \widetilde{\boldsymbol{D}}^{(k)}, & k \in \{1, \ldots, K\}\end{cases}\end{aligned} \tag{8}$$

with:

$$\begin{aligned}&\Psi_2\big(\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{U}, (\boldsymbol{D}^{(k)}, \widetilde{\boldsymbol{D}}^{(k)})_{1\le k\le K}\big) = \iota_{\mathcal{A}}(\boldsymbol{U}) \\ &+ \textstyle\sum_{k=1}^K \iota_{\mathcal{D}}(\widetilde{\boldsymbol{D}}^{(k)}) + \frac{1}{2}\sum_{k=1}^K ||\boldsymbol{C}^{(k)} - \boldsymbol{A}^{(1)}\boldsymbol{D}^{(k)}(\boldsymbol{A}^{(2)})^{\mathsf{T}}||^2_{\mathrm{F}}\end{aligned} \tag{9}$$

where $\iota_{\mathcal{A}}$ (resp. $\iota_{\mathcal{D}}$) is the indicator function of set $\mathcal{A}$ (resp. $\mathcal{D}$). The indicator of a set $\mathcal{S}$ of a Hilbert space $\mathcal{H}$ is defined as $(\forall \boldsymbol{x} \in \mathcal{H})\ \iota_{\mathcal{S}}(\boldsymbol{x}) = 0$ if $\boldsymbol{x} \in \mathcal{S}$, and $+\infty$ otherwise. It is noteworthy that the minimization problem (8) can be rewritten similarly to (5) where $J = 5$ and the matrices $(\boldsymbol{Q}_j^{\mathsf{T}}\boldsymbol{Q}_j)_{1\leq j\leq 5}$ correspond to isomorphisms. The augmented Lagrangian function associated with (8) is given by

$$\begin{aligned}\mathrm{L}_{\rho_1,\rho_2,\delta}\big((\boldsymbol{A}^{(i)},\boldsymbol{\Pi}^{(i)})_{1\leq i\leq 2},\boldsymbol{U},(\boldsymbol{D}^{(k)},\widetilde{\boldsymbol{D}}^{(k)},\boldsymbol{\Lambda}^{(k)})_{1\leq k\leq K}\big) = \\ \textstyle\sum_{k=1}^{K}\Big(\mathrm{tr}\big(\boldsymbol{\Lambda}^{(k)}(\boldsymbol{D}^{(k)}-\widetilde{\boldsymbol{D}}^{(k)})\big)+\frac{\delta}{2}||\boldsymbol{D}^{(k)}-\widetilde{\boldsymbol{D}}^{(k)}||_{\mathrm{F}}^2\Big) \\ +\textstyle\sum_{i=1}^{2}\Big(\mathrm{tr}\big(\boldsymbol{\Pi}^{(i)}(\boldsymbol{A}^{(i)}-\boldsymbol{U})^{\mathsf{T}}\big)+\frac{\rho_i}{2}||\boldsymbol{A}^{(i)}-\boldsymbol{U}||_{\mathrm{F}}^2\Big) \\ +\Psi_2\big(\boldsymbol{A}^{(1)},\boldsymbol{A}^{(2)},\boldsymbol{U},(\boldsymbol{D}^{(k)},\widetilde{\boldsymbol{D}}^{(k)})_{1\leq k\leq K}\big)\end{aligned} \tag{10}$$

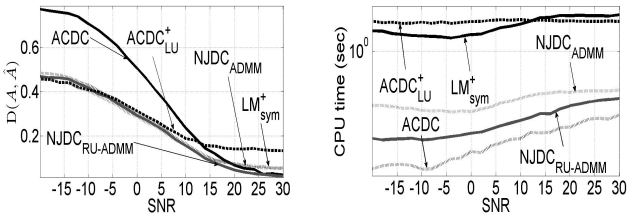
where $\boldsymbol{\Pi}^{(1)} \in \mathbb{R}^{N\times P}$, $\boldsymbol{\Pi}^{(2)} \in \mathbb{R}^{N\times P}$ and the K matrices $\boldsymbol{\Lambda}^{(k)} \in \mathrm{Diag}(\mathbb{R}^P)$ are Lagrangian multipliers, and where $\rho_1 \in]0,+\infty[$, $\rho_2 \in]0,+\infty[$, and $\delta \in]0,+\infty[$ are penalty parameters for the linear equality constraints given in (8).

An ADMM-like algorithm for solving (8) is obtained by successively minimizing the augmented Lagrangian function (10) with respect to $\boldsymbol{A}^{(1)}$, $\boldsymbol{A}^{(2)}$, $\boldsymbol{U}$, $(\boldsymbol{D}^{(k)})_{1\leq k\leq K}$, and $(\widetilde{\boldsymbol{D}}^{(k)})_{1\leq k\leq K}$, one variable at a time while setting the others to their most recent values, as described in (7). Then, the multipliers $\boldsymbol{\Pi}_n^{(1)}$, $\boldsymbol{\Pi}_n^{(2)}$, and $(\boldsymbol{\Lambda}_n^{(k)})_{1\leq k\leq K}$, where $n \in \mathbb{N}$ denotes the iteration number, are updated at each iteration. By using (9) and (10), it turns out that the alternating minimizations of (10) have the following closed form solutions:

$$\begin{aligned}\boldsymbol{A}_{n+1}^{(i)} &= \Big(\textstyle\sum_{k=1}^{K}\boldsymbol{C}^{(k)}\boldsymbol{A}_{n+i-1}^{(3-i)}\boldsymbol{D}_n^{(k)}+\rho_i\boldsymbol{U}_n-\boldsymbol{\Pi}_n^{(i)}\Big) \\ &\quad\Big(\textstyle\sum_{k=1}^{K}\boldsymbol{D}_n^{(k)}(\boldsymbol{A}_{n+i-1}^{(3-i)})^{\mathsf{T}}\boldsymbol{A}_{n+i-1}^{(3-i)}\boldsymbol{D}_n^{(k)}+\rho_i\boldsymbol{I}_P\Big)^{-1},\ i\in\{1,2\} \\ \boldsymbol{U}_{n+1} &= \mathsf{P}_{\mathcal{A}}\Big(\tfrac{1}{\rho_1+\rho_2}\Big(\rho_1\boldsymbol{A}_{n+1}^{(1)}+\rho_2\boldsymbol{A}_{n+1}^{(2)}+\boldsymbol{\Pi}_n^{(1)}+\boldsymbol{\Pi}_n^{(2)}\Big)\Big) \\ \boldsymbol{D}_{n+1}^{(k)} &= \mathrm{Diag}\Big(\Big((\boldsymbol{A}_{n+1}^{(1)}\boxdot\boldsymbol{A}_{n+1}^{(2)})^{\mathsf{T}}(\boldsymbol{A}_{n+1}^{(1)}\boxdot\boldsymbol{A}_{n+1}^{(2)})+\delta\boldsymbol{I}_{P^2}\Big)^{-1} \\ &\quad\Big((\boldsymbol{A}_{n+1}^{(1)}\boxdot\boldsymbol{A}_{n+1}^{(2)})^{\mathsf{T}}\,\mathrm{vec}(\boldsymbol{C}^{(k)})+\mathrm{vec}\,(\delta\widetilde{\boldsymbol{D}}^{(k)}{}_n-\boldsymbol{\Lambda}_n^{(k)})\Big)\Big) \\ \widetilde{\boldsymbol{D}}_{n+1}^{(k)} &= \mathsf{P}_{\mathcal{D}}(\boldsymbol{D}_{n+1}^{(k)}+\delta^{-1}\boldsymbol{\Lambda}_n^{(k)})\end{aligned} \tag{11}$$

where $\boxdot$ and $\mathsf{P}_{\mathcal{S}}$ denote the Hadamard product operator and the projection onto a convex set $\mathcal{S}$, respectively. Eventually, the resulting iterations have to be implemented using the Randomized Update Strategy (RUS) proposed in [18] in order to improve the convergence of the algorithm.

Although our JD2C$_{\mathrm{ADMM}}$ algorithm is closely related to the block-coordinate version of ADMM described in Section III-A, it does not constitute an instance of this algorithm or parallel versions of it. Indeed, the direct-fit cost term corresponds to a non-separable function of the variables $\boldsymbol{A}_n^{(1)}$, $\boldsymbol{A}_n^{(2)}$ and $(\boldsymbol{D}_n^{(k)})_{1\leq k\leq K}$. However, by updating incrementally the variables $\boldsymbol{A}_n^{(1)}$, $\boldsymbol{A}_n^{(2)}$ and $(\boldsymbol{D}_n^{(k)})_{1\leq k\leq K}$ at iteration n, the minimization of the Lagragian function is made tractable. Another difficulty is that, unlike most of the existing works concerning block-coordinate forms of ADMM, the problem we address is nonconvex. Note that a related strategy was adopted in [14] for computing the canonical polyadic decomposition of unsymmetric nonnegative multi-way arrays, but without the RUS described in [8], [18]. Besides, an interesting feature of our algorithm is that the updates of the variables $(\boldsymbol{D}_n^{(k)})_{1\leq k\leq K}$, $(\widetilde{\boldsymbol{D}}_n^{(k)})_{1\leq k\leq K}$, or $(\boldsymbol{\Lambda}_n^{(k)})_{1\leq k\leq K}$ can be performed in a parallel manner. This may be useful for providing fast implementations of JD2C$_{\mathrm{ADMM}}$ on multicore architectures.



(a) Error $\mathsf{D}(\boldsymbol{A},\hat{\boldsymbol{A}})$ evolution (b) CPU time evolution

Fig. 1. Performance evolution versus SNR.

IV. Numerical results

In this section, we restrict our attention to the nonnegative case when $\mathcal{A} = [0,+\infty[^{N\times P}$ and $\mathcal{D} = \mathrm{Diag}(\mathbb{R}^P)$. Our JD3C$_{\mathrm{ADMM}}$ algorithm, renamed NJDC$_{\mathrm{ADMM}}$ in such a context, is compared with several state-of-the-art (nonnegative) JDC methods and Blind Source Separation (BSS) algorithms. More specifically, two versions of NJDC$_{\mathrm{ADMM}}$ are studied: one based on the classical ADMM method [1] and another one using ADMM with the Randomization of the Updates described in Section III-A [8], [18], named NJDC$_{\mathrm{ADMM}}$ and NJDC$_{\mathrm{RU\text{-}ADMM}}$, respectively. The performance is measured in terms of the error between the true joint transformation $\boldsymbol{A}$ and its estimate $\hat{\boldsymbol{A}}$, as well as the source signals $\boldsymbol{S}$ and their estimate $\hat{\boldsymbol{S}}$ when the BSS context is considered. So the scale and permutation invariant distance D defined in [19, section III] is chosen as the performance measure. Moreover, 200 independent Monte Carlo trials are used to compute the average distance D. All the algorithms stop either when the relative error of the corresponding criterion between two successive iterations is less than 10^{-4} or when the number of iterations exceeds 500. In the following experiments, all the algorithms are initialized with the same random matrices drawn from a uniform distribution. All the simulations are implemented in Matlab v7.14 and run on Intel Dual-Core 2.0GHz CPU with 8Gb memory.

A. Simulated JDC model

In this first test, the behavior of NJDC$_{\mathrm{ADMM}}$ and NJDC$_{\mathrm{RU\text{-}ADMM}}$ are evaluated and compared with the classical JDC method named ACDC [21] and two NJDC algorithms, namely LM$^{+}_{\mathrm{sym}}$ [4] and ACDC$^{+}_{\mathrm{LU}}$ [19]. The synthetic matrix set $\boldsymbol{\mathcal{C}}$ is generated randomly according to (1) with $N = 8$, $P = 5$ and $K = 4$. $\boldsymbol{A}$ is drawn from the standard uniform distribution while the diagonal elements of $\boldsymbol{D}^{(k)}$ are drawn independently from the standard normal distribution. Each resulting target matrix $\boldsymbol{C}_N^{(k)}$ is perturbed by a random symmetric[1] "noise" matrix as follows:

$$\boldsymbol{C}_N^{(k)} = \boldsymbol{C}^{(k)} + \sigma_N^{(k)}\boldsymbol{N}^{(k)}\,\boldsymbol{N}^{(k)\mathsf{T}} \tag{12}$$

where σ_N is a scalar controlling the noise level and $\boldsymbol{N}^{(k)}$ is drawn from a standard normal distribution. Then the SNR is defined as $\mathrm{SNR} = -20\log_{10}(\sigma_N^{(k)}\|\boldsymbol{N}^{(k)}\boldsymbol{N}^{(k)\mathsf{T}}\|_F/\|\boldsymbol{C}^{(k)}\|_F)$.

[1] In most of the applications, the matrices to be jointly diagonalized are slices of cumulants estimated from the data, which involves a symmetric noise.

TABLE I. AVERAGE ESTIMATING ERRORS AND CPU TIME OF FIVE METHODS FOR BSS OF NMRS DATA

	$\text{NJDC}_{\text{RU-ADMM}}$	$\text{ACDC}^{+}_{\text{LU}}$	CoM2	SOBI	NNLS
$\mathsf{D}(\boldsymbol{A}, \hat{\boldsymbol{A}})$	0.0007	0.0007	0.0043	0.0480	0.0137
$\mathsf{D}(\boldsymbol{S}^{\mathsf{T}}, \hat{\boldsymbol{S}}^{\mathsf{T}})$	0.0175	0.0175	0.0304	0.0677	0.0383
CPU time (s)	0.0924	0.3251	0.0147	0.1178	5.0233

Figure 1(a) depicts the average error $\mathsf{D}(\boldsymbol{A}, \hat{\boldsymbol{A}})$ at the output of the five algorithms as a function of the SNR, which varies from -20 to 30 dB. It appears that $\text{NJDC}_{\text{ADMM}}$, $\text{NJDC}_{\text{RU-ADMM}}$ and $\text{LM}^{+}_{\text{sym}}$ give the smallest estimation errors for every SNR value. On the one hand, the use of nonnegative constraints allows us to outperform ACDC for very low SNR values. On the other hand, $\text{NJDC}_{\text{ADMM}}$, $\text{NJDC}_{\text{RU-ADMM}}$ and $\text{LM}^{+}_{\text{sym}}$ outperform $\text{ACDC}^{+}_{\text{LU}}$ for high SNR values allowing them to compete with ACDC. However, as displayed in figure 1(b), $\text{LM}^{+}_{\text{sym}}$ is more time consuming than $\text{NJDC}_{\text{ADMM}}$ and $\text{NJDC}_{\text{RU-ADMM}}$. Note that the use of the randomization step allows $\text{NJDC}_{\text{RU-ADMM}}$ to be less time consuming than $\text{NJDC}_{\text{ADMM}}$. To sum up, for most of the SNR values (especially for low values), $\text{NJDC}_{\text{RU-ADMM}}$ yields the best compromise between estimation accuracy and computation time.

B. BSS application on NMRS data

We now illustrate the ability of $\text{NJDC}_{\text{RU-ADMM}}$ to perform a Semi-Nonnegative Independent Component Analysis (SeNICA) of NMRS signals [20] through the NJDC of a set of Fourth Order (FO) cumulant matrix slices. The latter approach is compared with two ICA methods, namely CoM2 [9] and SOBI [2], [9], a nonnegative matrix factorization algorithm based on alternating NonNegativity Least Squares (NNLS) [13] and a SeNICA approach based on $\text{ACDC}^{+}_{\text{LU}}$. More specifically, two source metabolites, namely Choline (Cho) and Myo-inositol (Ins), are generated by means of Lorenzian functions with fixed parameters for a realistic representation [20]. It is assumed that the static mixing model $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S} + \boldsymbol{B}$ is valid, where $\boldsymbol{B}$ is a matrix realization of an additive zero-mean white Gaussian noise vector process such that the $\text{SNR} = -20\log_{10}(\|\boldsymbol{B}\|_F / \|\boldsymbol{A}\boldsymbol{S}\|_F)$ is equal to 10 dB. 200 matrices $\boldsymbol{X} \in \mathbb{R}^{20\times 1000}$ are created with $\boldsymbol{A}$ generated as in Section IV-A. The subset $\boldsymbol{\mathcal{C}} \subset \mathbb{R}^{20\times 20\times 20}$ is then built by computing 20 linear combinations from the 400 available FO cumulant matrix slices. Table I shows the average estimation errors $\mathsf{D}(\boldsymbol{A}, \hat{\boldsymbol{A}})$, $\mathsf{D}(\boldsymbol{S}^{\mathsf{T}}, \hat{\boldsymbol{S}}^{\mathsf{T}})$ and the CPU time at the output of the five methods. It appears that the SeNICA approach using $\text{NJDC}_{\text{RU-ADMM}}$ offers the best compromise between estimation accuracy and computation time.

V. CONCLUSION

In this paper, we gave sufficient conditions for the existence of a solution to the JDC problem subject to constraints. We also provided an ADMM-based framework allowing us to compute such a solution and to handle any kind of convex constraints. For example, we can impose lower and upper bounds of the sought values or impose that the values belong to some unit simplex, if these contraints make sense in some contexts. Numerical tests have shown the benefits of the proposed $\text{JD2C}_{\text{ADMM}}$ method in the presence of modeling errors. In our future work, we plan to investigate theoretical convergence properties of the proposed algorithm.

REFERENCES

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[2] G. Chabriel, M. Kleinsteuber, E. Moreau, H. Shen, P. Tichavsky, and A. Yeredor, "Joint matrices decompositions and blind source separation. a survey of methods, identification and applications," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 34–43, 2014.

[3] C. Chen, B. He, Y. Ye, and X. Yuan, "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent," *Mathematical Programming*, 2014.

[4] J. Coloigner, A. Karfoul, L. Albera, and P. Comon, "Line search and trust region strategies for canonical decomposition of semi-nonnegative semi-symmetric 3rd order tensors," *Linear Algebra and its Applications*, no. 450, pp. 334–374, 2014.

[5] J. Coloigner, L. Albera, A. Kachenoura, F. Noury, and L. Senhadji, "Semi-nonnegative joint diagonalization by congruence and semi-nonnegative ICA," *Signal Processing*, vol. 105, pp. 185–197, 2014.

[6] P. L. Combettes and J.-C. Pesquet, "A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery," *IEEE Journal on Selected Topics in Sig. Proc.*, vol. 1, no. 4, pp. 564–574, 2007.

[7] ——, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H. H. Bauschke, R. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, Eds. New York: Springer-Verlag, 2010, pp. 185–212.

[8] ——, "Stochastic quasi-Fejér block-coordinate fixed point iterations with random sweeping," *SIAM Journal on Optimization*, 2015, to appear. [Online]. Available: http://arxiv.org/abs/1404.7536

[9] P. Comon and C. Jutten, *Handbook of blind source separation*. Academic Press, 2010.

[10] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Programming*, vol. 55, no. 1–3, pp. 293–318, 1992.

[11] H. Everett III, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, vol. 11, no. 3, pp. 399–417, 1963.

[12] M. A. T. Figueiredo and J. M. Bioucas-Dias, "Restoration of Poissonian images using alternating direction optimization," *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3133–3145, 2010.

[13] H. Kim and H. Park, "Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method," *SIAM J. on Matrix Anal. and Appl.*, vol. 30, no. 2, pp. 713–730, 2008.

[14] A. P. Liavas and N. D. Sidiropoulos, "Parallel algorithms for constrained tensor factorization via alternating direction method of multipliers," *submitted to IEEE Transactions on Signal Processing*, 2015. [Online]. Available: http://arxiv.org/abs/1409.2383

[15] J.-C. Pesquet and N. Pustelnik, "A parallel inertial proximal optimization method," *Pacific J. of Optim.*, vol. 8, no. 2, pp. 273–305, 2012.

[16] M. J. D. Powell, "A method for nonlinear constraints in minimization problems," in *Optimization*, R. Fletcher, Ed. London: Academic Press, 1968, pp. 283–298.

[17] S. Setzer, G. Steidl, and T. Teuber, "Deblurring Poissonian images by split Bregman techniques," *Journal of Visual Communication and Image Representation*, vol. 21, no. 3, pp. 193–199, Apr. 2010.

[18] R. Sun, Z.-Q. Luo, and Y. Ye, "On the expected convergence of randomly permuted ADMM," March 2015. [Online]. Available: http://arxiv.org/abs/1503.06387

[19] L. Wang, L. Albera, A. Kachenoura, H. Z. Shu, and L. Senhadji, "Nonnegative joint diagonalization by congruence based on LU matrix factorization," *IEEE Sig. Proc. Let.*, vol. 20, no. 8, pp. 807–810, 2013.

[20] ——, "Canonical polyadic decomposition of 3rd order semi-nonnegative semi-symmetric tensors using LU and QR matrix factorizations," *Journal on Advances in Sig. Proc.*, vol. 150, 2014, special Issue on Recent Advances in Tensor Based Sig. and Image Proc.

[21] A. Yeredor, "Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation," *IEEE Transactions on Signal Processing*, vol. 50, no. 7, pp. 1545–1553, 2002.