

Algorithme de diagonalisation conjointe par similitude avec procédure de balayage et paramétrisation non-arbitraires

Rémi ANDRÉ¹ Xavier LUCIANI²

¹Aix-Marseille Univ, CNRS, Centrale Marseille, Institut Fresnel, Marseille, France

²Université de Toulon, Aix Marseille Université, CNRS, Seatech, LIS, UMR 7020, La Garde, France

Résumé – La diagonalisation conjointe par similitude consiste à estimer une base de vecteurs propres commune à un ensemble de matrices. Les algorithmes de diagonalisation conjointe estiment séquentiellement deux variables à la fois à travers une stratégie de balayage par paires. Ces algorithmes diffèrent en fonction de la factorisation matricielle utilisée pour estimer ces deux variables. Dans cet article, nous proposons un nouvel algorithme permettant de sélectionner de manière non-arbitraire la factorisation matricielle et l'ordre de balayage des variables à estimer .

Abstract – Joint eigenvalue decomposition consists in estimating a common basis of eigenvectors of a set of square matrices. Most joint eigenvalue decomposition algorithms are based on the sequential optimization of small subsets of two variables through a sweeping procedure. The estimation of both variables differs from an algorithm to another according to the matrix factorization used to update the matrices to diagonalize. In this paper, we propose a novel algorithm for the non-arbitrary selection of the couple of parameters to estimate and for the factorization of the updating matrix.

1 Introduction

Au cours des deux dernières décennies, la Diagonalisation Conjointe par Similitude (DCS) a montré son efficacité dans de nombreuses applications de traitement du signal telles que la séparation aveugle de sources [3, 7] ou encore la décomposition canonique polyadique de tenseurs [10, 6]. La DCS consiste à déterminer une matrice de vecteurs propres $\mathbf{A} \in \mathbb{C}^{N \times N}$ commune à un ensemble de K matrices. En pratique, nous disposons d'un ensemble de K matrices $\mathbf{M}^{(k)} \in \mathbb{C}^{N \times N}$ définies comme

$$\mathbf{M}^{(k)} = \mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{-1} + \mathbf{E}^{(k)} \quad \forall k = 1, \dots, K \quad (1)$$

où les matrices $\mathbf{D}^{(k)} \in \mathbb{C}^{N \times N}$ sont diagonales et les matrices $\mathbf{E}^{(k)} \in \mathbb{C}^{N \times N}$ représentent l'erreur de modélisation.

Le principe des algorithmes de DCS [7, 6, 5, 1, 2] consiste alors à estimer une matrice diagonalisante $\mathbf{B} \in \mathbb{C}^{N \times N}$ de manière à rendre les matrices

$$\mathbf{N}^{(k)} = \mathbf{B}\mathbf{M}^{(k)}\mathbf{B}^{-1} \quad (2)$$

aussi diagonales que possible. La matrice \mathbf{B} permet donc d'approximer \mathbf{A}^{-1} à la permutation et l'échelle des colonnes près [4].

Une stratégie classique et efficace consiste à optimiser séquentiellement les paramètres de mise à jour de la matrice \mathbf{B} à travers une stratégie de balayage par paires. Cela signifie, que lors d'une itération, la matrice \mathbf{B} est mise à jour plusieurs fois à l'aide de matrices dépendant de deux paramètres. Dans le présent article, nous nous intéressons à la classe d'algorithmes proposée dans [2] appelée JAPAM (Joint eigenvalue decomposition using a Parametrized Matrix). Ceux-ci permettent d'estimer simultanément le couple de paramètres de chaque mise à jour. Ils diffèrent entre eux par la factorisation matricielle utilisée pour construire la matrice de mise à jour. L'une des versions de JAPAM permet de sélectionner de manière

non-arbitraire cette factorisation matricielle à chaque mise à jour. Dans cet article, nous améliorons cette approche en sélectionnant de manière non-arbitraire à la fois l'ordre de balayage des couples de variables à estimer et la factorisation de la matrice de mise à jour.

Le présent document est organisé comme suit. La section suivante explique la structure générale des algorithmes à balayage par paires. Dans la section 3, nous présentons la classe d'algorithme JAPAM. La section 4 est dédiée à la nouvelle stratégie de sélection automatique du sous-ensemble de variables. Enfin, les performances de l'algorithme proposé sont évaluées dans la section 5 au moyen de simulations numériques.

Notations. Les scalaires sont désignés par une minuscule (a), les vecteurs par une minuscule en gras (\mathbf{a}) et les matrices par une majuscule en gras (\mathbf{A}). $[1, P]_{\mathbb{N}}$ est l'ensemble des P premiers entiers naturels.

2 Algorithmes à balayage par paires

Les algorithmes de DCS à balayage par paires calculent, à chaque itération, une matrices de mise à jour $\mathbf{X} \in \mathbb{C}^{N \times N}$ pouvant être décomposée de la manière suivante :

$$\mathbf{X} = \prod_{i=1}^{N-1} \prod_{j=i+1}^N \mathbf{X}^{(i,j)} \quad (3)$$

où chacune des matrices $\mathbf{X}^{(i,j)}$ est égale à la matrice identité à l'exception des éléments $X_{i,i}^{(i,j)}$, $X_{i,j}^{(i,j)}$, $X_{j,i}^{(i,j)}$ et $X_{j,j}^{(i,j)}$. Ainsi, à chaque itération, la matrice \mathbf{B} et l'ensemble de matrices $\mathbf{N}^{(k)}$ sont mis à jour $N(N-1)/2$ fois comme

$$\begin{cases} \mathbf{B} \leftarrow \mathbf{X}^{(i,j)}\mathbf{B} \\ \mathbf{N}^{(k)} \leftarrow \mathbf{X}^{(i,j)}\mathbf{N}^{(k)}\mathbf{X}^{(i,j)-1} \quad \forall k = 1, \dots, K. \end{cases} \quad (4)$$

Comme le montre la figure 1, les termes $N_{i,i}^{(k)}$, $N_{i,j}^{(k)}$, $N_{j,i}^{(k)}$ et $N_{j,j}^{(k)}$ sont affectés deux fois par la mise à jour (4). Ainsi, de

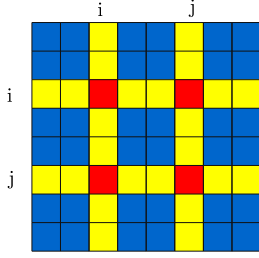


FIGURE 1 : impact de la mise à jour (4) par la matrice $\mathbf{X}^{(i,j)}$ sur une matrice $\mathbf{N}^{(k)}$ (les cases bleues sont les éléments non affectés par la transformation, les jaunes sont les éléments affectés une seule fois et les rouges sont les éléments affectés deux fois).

nombreux algorithmes de DCS estiment les paramètres des matrices $\mathbf{X}^{(i,j)}$ en minimisant le critère de diagonalisation suivant :

$$C^{(i,j)}(\mathbf{X}^{(i,j)}) = \sum_{k=1}^K |N'_{i,j}{}^{(k)}|^2 + |N'_{j,i}{}^{(k)}|^2 \quad (5)$$

avec $N'{}^{(k)} = \mathbf{X}^{(i,j)}\mathbf{N}^{(k)}\mathbf{X}^{(i,j)-1}$.

Minimiser le critère précédent n'est pas équivalent à minimiser l'ensemble des termes hors-diagonaux des matrices $\mathbf{N}^{(k)}$. Cependant, cette optimisation locale s'avère suffisante en pratique [8] et permet de réécrire le problème initial en une suite de sous problèmes de dimension 2. Ainsi dans la suite de cette section, nous utiliserons pour la matrice $\mathbf{X}^{(i,j)}$ la notation simplifiée

$$\tilde{\mathbf{X}}^{(i,j)} = \begin{pmatrix} X_{i,i}^{(i,j)} & X_{i,j}^{(i,j)} \\ X_{j,i}^{(i,j)} & X_{j,j}^{(i,j)} \end{pmatrix}. \quad (6)$$

Du fait de l'indétermination d'échelle sur les colonnes de la matrice \mathbf{A} , il est important de noter que la matrice \mathbf{X} admet N degrés de liberté et donc il est possible de paramétrer les matrices $\mathbf{X}^{(i,j)}$ à l'aide de seulement deux paramètres. Les algorithmes à balayage par paires utilisent généralement des factorisations matricielles pour paramétrer les matrices $\mathbf{X}^{(i,j)}$ telles que

- la factorisation LU :

$$\tilde{\mathbf{X}}^{(i,j)} = \begin{pmatrix} 1 & 0 \\ t^{(i,j)} & 1 \end{pmatrix} \begin{pmatrix} 1 & u^{(i,j)} \\ 0 & 1 \end{pmatrix} \quad (7)$$

- le produit d'une matrice orthogonale complexe et d'une matrice triangulaire supérieure que nous avons appelé factorisation QR algébrique :

$$\tilde{\mathbf{X}}^{(i,j)} = \frac{1}{\sqrt{1+(t^{(i,j)})^2}} \begin{pmatrix} 1 & -t^{(i,j)} \\ t^{(i,j)} & 1 \end{pmatrix} \begin{pmatrix} 1 & r^{(i,j)} \\ 0 & 1 \end{pmatrix} \quad (8)$$

avec $(t^{(i,j)})^2 \neq -1$.

- le produit d'une matrice orthogonale complexe et d'une matrice symétrique complexe, appelé décomposition polaire algébrique :

$$\tilde{\mathbf{X}}^{(i,j)} = \frac{\begin{pmatrix} 1 & -t^{(i,j)} \\ t^{(i,j)} & 1 \end{pmatrix} \begin{pmatrix} 1 & h^{(i,j)} \\ h^{(i,j)} & 1 \end{pmatrix}}{\sqrt{(1+(t^{(i,j)})^2)(1-(h^{(i,j)})^2)}} \quad (9)$$

avec $(t^{(i,j)})^2 \neq -1$ et $(h^{(i,j)})^2 \neq 1$.

L'intérêt des paramétrisations précédentes est que l'existence de $\mathbf{X}^{(i,j)-1}$ est assurée puisque le déterminant de $\mathbf{X}^{(i,j)}$ est égal à 1 par construction.

3 Estimation simultanée des paramètres de mise à jour

Les algorithmes de type JAPAM estiment simultanément les deux paramètres des matrices $\mathbf{X}^{(i,j)}$.

Les deux termes hors diagonaux de chacune des matrices $\mathbf{N}^{(k)}$ peuvent être réécrit $\forall k \in [1, K]_{\mathbb{N}}$ comme

$$\begin{cases} N'_{i,j}{}^{(k)} = X_{i,i}^{(i,j)} X_{i,j}^{(i,j)} (N_{j,j}^{(k)} - N_{i,i}^{(k)}) + X_{i,i}^{(i,j)2} \tilde{N}_{i,j}^{(k)} - X_{i,i}^{(i,j)2} N_{j,i}^{(k)}, \\ N'_{j,i}{}^{(k)} = X_{j,i}^{(i,j)} X_{j,j}^{(i,j)} (N_{i,i}^{(k)} - N_{j,j}^{(k)}) + X_{j,j}^{(i,j)2} N_{j,i}^{(k)} - X_{j,i}^{(i,j)2} N_{i,j}^{(k)}. \end{cases} \quad (10)$$

Dans le but d'avoir une expression analytique des paramètres à estimer, nous faisons les hypothèses suivantes :

1. La matrice $\mathbf{X}^{(i,j)}$ est proche d'un point stationnaire de $C^{(i,j)}(\mathbf{X}^{(i,j)})$.
2. La matrice \mathbf{B} est proche de la solution diagonalisante.

La première hypothèse implique que $\mathbf{X}^{(i,j)}$ est proche de la matrice identité et donc que $|X_{i,i}^{(i,j)}|$ et $|X_{j,i}^{(i,j)}|$ sont proches de zéro. La seconde, quant à elle, implique que les matrices $\mathbf{N}^{(k)}$ sont proches de matrices diagonales et donc que $|N_{j,i}^{(k)}|$ et $|N_{i,j}^{(k)}|$ sont proches de 0 $\forall k$. Ainsi le critère (5) peut être approché par

$$C_a^{(i,j)}(\mathbf{X}^{(i,j)}) = \mathbf{v}^H \mathbf{F}_{ij}^H \mathbf{F}_{ij} \mathbf{v} + \mathbf{w}^H \mathbf{F}_{ji}^H \mathbf{F}_{ji} \mathbf{w} \quad (11)$$

où

$$\mathbf{v} = \begin{pmatrix} X_{i,i}^{(i,j)} X_{i,j}^{(i,j)} \\ X_{i,i}^{(i,j)2} \end{pmatrix}; \quad \mathbf{w} = \begin{pmatrix} X_{j,i}^{(i,j)} X_{j,j}^{(i,j)} \\ X_{j,i}^{(i,j)2} \end{pmatrix}; \quad (12)$$

$$\mathbf{F}_{ij} = \begin{pmatrix} N_{j,j}^{(1)} - N_{i,i}^{(1)} & N_{i,j}^{(1)} \\ \vdots & \vdots \\ N_{j,j}^{(K)} - N_{i,i}^{(K)} & N_{i,j}^{(K)} \end{pmatrix} \text{ et } \mathbf{F}_{ji} = \begin{pmatrix} N_{i,i}^{(1)} - N_{j,j}^{(1)} & N_{j,i}^{(1)} \\ \vdots & \vdots \\ N_{i,i}^{(K)} - N_{j,j}^{(K)} & N_{j,i}^{(K)} \end{pmatrix}. \quad (13)$$

Comme les deux termes du critère $C_a^{(i,j)}$ de l'équation (11) ne partagent aucune variable à optimiser, ils peuvent être minimisés séparément par rapport à \mathbf{v} et \mathbf{w} . Les vecteurs de norme 1 minimisant $C_a^{(i,j)}$ sont alors les vecteurs propres normés associés à la plus petite valeur propre des matrices $\mathbf{F}_{ij}^H \mathbf{F}_{ij}$ et $\mathbf{F}_{ji}^H \mathbf{F}_{ji}$.

Nous notons respectivement \mathbf{e} et \mathbf{f} ces deux vecteurs et nous avons les relations suivantes :

$$\begin{cases} \mathbf{e} = \alpha \mathbf{v}, \\ \mathbf{f} = \beta \mathbf{w}, \end{cases} \quad (14)$$

où α et β sont des paramètres scalaires inconnus.

Nous pouvons directement déduire du système (14) et de (12) :

$$\begin{cases} \frac{X_{i,j}^{(i,j)}}{X_{i,i}^{(i,j)}} = \frac{e_1}{e_2}, \\ \frac{X_{j,i}^{(i,j)}}{X_{j,j}^{(i,j)}} = \frac{f_1}{f_2}. \end{cases} \quad (15)$$

Solution dans le cas d'une factorisation LU. Étant donné la paramétrisation de l'équation (7), nous pouvons directement déduire

$$\begin{cases} u^{(i,j)} = \frac{e_1}{e_2}, \\ t^{(i,j)} = \frac{f_1 e_1}{e_2 f_2 - e_1 f_1}. \end{cases} \quad (16)$$

Solution dans le cas d'un factorisation QR algébrique. Étant donné la paramétrisation de l'équation (8), nous pouvons déduire

$$\begin{cases} r^{(i,j)} = \frac{e_1}{e_2} + t^{(i,j)}, \\ t^{(i,j)^2} e_2 f_1 + t^{(i,j)} (e_1 f_1 - e_2 f_2) + e_1 f_2 = 0. \end{cases} \quad (17)$$

Solution dans le cas d'un décomposition polaire algébrique. Étant donné la paramétrisation de l'équation (9), nous pouvons déduire

$$\begin{cases} h^{(i,j)} = \frac{e_1 + e_2 t^{(i,j)}}{e_2 - e_1 t^{(i,j)}}, \\ t^{(i,j)^2} (e_2 f_1 - e_1 f_2) + 2t^{(i,j)} (e_1 f_1 + e_2 f_2) + (e_1 f_2 - e_2 f_1) = 0. \end{cases} \quad (18)$$

Pour les équations (17) et (18), nous choisissons comme racines des polynômes celles de plus petits modules dans le but de respecter au mieux notre première hypothèse.

L'utilisation de ces différentes factorisations a permis de développer plusieurs algorithmes plus ou moins sensibles aux erreurs d'approximation. Ces erreurs d'approximation empêchent le fonctionnement des algorithmes à partir de certaines valeurs de N . Les factorisations LU, QR algébrique et polaire algébrique ne fonctionnent plus correctement pour $N > 6$, $N > 9$ et $N > 12$ respectivement. Une dernière version, appelée JAPAM-M, sélectionnant automatiquement la meilleure factorisation à chaque mise à jour permet d'élargir la zone de fonctionnement à $N < 15$.

Complexité numérique La complexité numérique est définie comme le nombre de multiplications que l'algorithme effectue à chaque itération. La complexité numérique des algorithmes de DCS est largement dominée par la mise à jour de l'ensemble de matrices $\mathbf{N}^{(k)}$ (équation (4)). Considérant qu'une multiplication complexe équivaut à quatre multiplications réelles, le terme prépondérant de la complexité numérique des algorithmes de la classe JAPAM est approximativement $16KN^3$.

4 Stratégie de balayage non-arbitraire

Dans cette section, nous présentons une nouvelle stratégie permettant de sélectionner automatiquement l'ordre des paramètres à optimiser dans la procédure de balayage par paires. En effet, la décomposition de la matrice \mathbf{X} à l'équation (3) impose de manière arbitraire l'ordre des paramètres à optimiser. Nous proposons alors, à chaque itération, la décomposition suivante :

$$\mathbf{X} = \prod_{p=1}^{\frac{N(N-1)}{2}} \mathbf{X}^{(i_p, j_p)} \quad (19)$$

telle que $(i_p, j_p) \in [1, N-1]_{\mathbb{N}} \times [2, N]_{\mathbb{N}}$, $i_p < j_p$ et $(i_{p1}, j_{p1}) \neq (i_{p2}, j_{p2}) \forall (p1, p2)$. Dans le but de sélectionner le couple d'indices (i_p, j_p) de manière pertinente, définissons tout d'abord l'ensemble des indices qui n'ont pas encore été sélectionnés à la mise à jour courante p :

$$\mathcal{I}_p = \{(i, j) \in [1, N-1]_{\mathbb{N}} \times [2, N]_{\mathbb{N}} | i < j\} \setminus \cup_{q=1}^{p-1} (i_q, j_q). \quad (20)$$

Avant chaque mise à jour, tout comme pour JAPAM-M nous sélectionnons parmi les trois factorisations précédentes celle qui conduit à la plus petite valeur de $C_a^{(i,j)}$ pour tout $(i, j) \in \mathcal{I}_p$. Cette étape augmente de manière négligeable la complexité numérique.

Le couple (i_p, j_p) est alors choisi comme celui faisant le plus décroître le critère $C_a^{(i,j)}$, $\forall (i, j) \in \mathcal{I}_p$. Ainsi,

$$(i_p, j_p) = \underset{(i,j)}{\operatorname{argmax}} \Phi(i, j) \quad (21)$$

où la fonction Φ est définie par

$$\begin{aligned} \Phi: \mathcal{I}_p &\rightarrow \mathbb{R} \\ (i, j) &\mapsto \left(\sum_{k=1}^K |N_{i,j}^{(k)}|^2 + |N_{j,i}^{(k)}|^2 \right) - C_a^{(i,j)}(\mathbf{X}^{(i,j)}). \end{aligned} \quad (22)$$

Cet algorithme est décrit dans la table algorithmique 1, nous l'appelons VS-JAPAM-M (Variable-Selection JAPAM-M).

Algorithme 1 : VS-JAPAM-M

- 1 Soit \mathcal{S}_C un critère d'arrêt et It_{max} le nombre maximal d'itérations ;
 - 2 Initialisation : \mathbf{B} avec l'identité et $it \leftarrow 1$;
 - 3 **tant que** \mathcal{S}_C n'est pas atteint ou $it \leq It_{max}$ **faire**
 - 4 **pour** $p = 1 \dots N(N-1)/2$ **faire**
 - 5 Construire l'ensemble \mathcal{I}_p (équation (20)) ;
 - 6 Construire la matrice $\mathbf{X}^{(i,j)}$ en utilisant la factorisation donnant la plus petite valeur de $C_a^{(i,j)} \forall (i, j) \in \mathcal{I}_p$;
 - 7 Calculer $\Phi(i, j) \forall (i, j) \in \mathcal{I}_p$ (équation (22)) ;
 - 8 Déterminer (i_p, j_p) (équation (21)) ;
 - 9 Mettre jour $\mathbf{B} \leftarrow \mathbf{X}^{(i_p, j_p)} \mathbf{B}$;
 - 10 Mettre jour $\mathbf{N}^{(k)} \leftarrow \mathbf{X}^{(i_p, j_p)} \mathbf{N}^{(k)} \mathbf{X}^{(i_p, j_p)-1} \forall k$;
 - 11 **fin**
 - 12 $it \leftarrow it + 1$;
 - 13 Calculer la nouvelle valeur du critère d'arrêt \mathcal{S}_C ;
 - 14 **fin**
-

Puisque la mise à jour de l'ensemble de matrices $\mathbf{N}^{(k)}$ par la matrice $\mathbf{X}^{(i_p, j_p)}$ ne modifie que les colonnes et les lignes i_p et j_p (cases jaunes et rouges de la figure 1), la fonction Φ doit être recalculée uniquement pour les couples (i_p, a) , (b, i_p) , (c, j_p) et (j_p, d) appartenant à \mathcal{I}_{p+1} pour tout $a > i_p$, $b < i_p$, $c < j_p$ et $d > j_p$. Cela signifie qu'à la fin de la procédure de balayage par paires, $\frac{N(N-1)(N-2)}{2}$ valeurs de Φ ont été recalculées. Sachant que l'équation (22) implique $40K$ multiplications, le terme prépondérant de la complexité numérique de VS-JAPAM-M est d'environ $36KN^3$.

5 Simulations numériques

Les performances de VS-JAPAM-M sont maintenant évaluées à l'aide de simulations numériques en considérant un ensemble de 20 matrices complexes de taille 13. Nous le comparons avec les algorithmes JAPAM-M et SJDTE [1]. Dans notre étude,

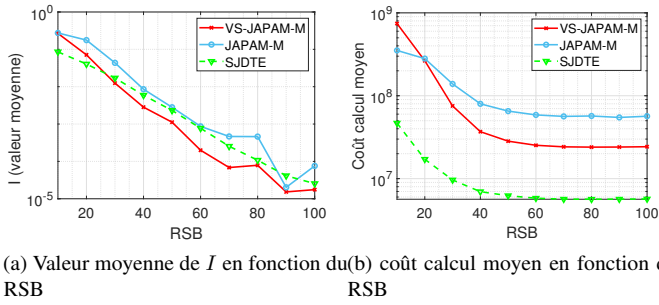


FIGURE 2 : DCS de 20 matrices de taille 13

les algorithmes sont arrêtés lorsque nombre d'itérations devient supérieur à 500 ou que le critère $\mathcal{S}_C = \frac{|C(\mathbf{B}_{it+1}) - C(\mathbf{B}_{it})|}{C(\mathbf{B}_{it})}$ devient inférieur ou égal à 10^{-6} , avec

$$C(\mathbf{B}_{it}) = \sum_{\substack{i,j=1 \\ i \neq j}}^N \left(\mathbf{B}_{it} \mathbf{M}^{(k)} \mathbf{B}_{it}^{-1} \right)_{i,j} \quad (23)$$

Les matrices $\mathbf{M}^{(k)}$ à diagonaliser sont construites en utilisant l'équation (1) et en tirant aléatoirement les éléments des matrices \mathbf{A} , $\mathbf{D}^{(k)}$ et $\mathbf{E}^{(k)}$ à partir d'une loi normale centrée et réduite. La norme de Frobenius est ensuite fixée à 1 pour toutes les matrices $\mathbf{A} \mathbf{D}^{(k)} \mathbf{A}^{-1}$ et à σ pour toutes les matrices $\mathbf{E}^{(k)}$. Dans la présente étude les performances des algorithmes sont étudiées en fonction du Rapport Signal sur Bruit (RSB) défini comme $RSB = -20 \log_{10}(\sigma)$. Pour évaluer les performances des différents algorithmes, nous utilisons le critère de performances présenté dans [9] défini comme $I(\mathbf{G}) = \frac{\Gamma(\mathbf{G})}{2N(N-1)}$ où

$$\Gamma(\mathbf{G}) = \sum_{i=1}^N \left(\sum_{j=1}^N \frac{|G_{i,j}|^2}{\max_l |G_{i,l}|^2} - 1 \right) + \sum_{j=1}^N \left(\sum_{i=1}^N \frac{|G_{i,j}|^2}{\max_l |G_{l,j}|^2} - 1 \right) \quad (24)$$

et $\mathbf{G} = \mathbf{B}\mathbf{A}$. En d'autres termes, $I(\mathbf{G})$ mesure l'écart entre \mathbf{G} et le produit d'une matrice diagonale avec une matrice de permutation puisque \mathbf{B}^{-1} est une estimation de \mathbf{A} , aux permutations et à l'échelle des colonnes près. Les algorithmes sont également évalués en prenant en compte leur coût calcul, défini comme le produit entre la complexité numérique et le nombre d'itérations nécessaire pour atteindre le critère d'arrêt. Nous faisons varier le RSB de 10 dB à 100 dB par pas de 10. Pour chaque valeur du RSB, la valeur moyenne de $I(\mathbf{G})$ (figure 2a) et le coût calcul moyen (figure 2b) sont calculés sur 4000 réalisations de l'ensemble de matrices.

Il est intéressant de noter que, pour le scénario considéré, le coût calcul moyen de VS-JAPAM-M est inférieur à celui de JAPAM-M pour toutes les valeurs de RSB sauf 10 et 20 dB. Pour ces valeurs, VS-JAPAM-M a donc besoin de moins d'itérations pour atteindre le critère d'arrêt. SJDTE, quant à lui, est le moins coûteux des trois algorithmes. Concernant l'estimation de la matrice \mathbf{A} , VS-JAPAM-M procure les meilleurs résultats pour les moyennes et hautes valeurs de RSB et améliore systématiquement les résultats de JAPAM-M. Une étude plus poussée montre que VS-JAPAM-M permet de diagonaliser des matrices jusqu'à $N = 17$.

6 Conclusion

Nous avons présenté dans cet article un nouvel algorithme de diagonalisation conjointe par similitude basé sur la même struc-

ture que les algorithmes de la classe JAPAM [2]. La méthode proposée sélectionne de manière non-arbitraire les paramètres à optimiser à chaque mise à jour de l'ensemble de matrices à diagonaliser. Les simulations numériques montrent que la stratégie proposée permet réduire le coût calcul moyen en comparaison à un algorithme sélectionnant de manière arbitraire les paramètres à optimiser. De plus, l'algorithme développée améliore l'estimation moyenne de la matrice de vecteur propres. Il serait maintenant intéressant de proposer une stratégie de sélection de paramètres ayant une complexité numérique plus faible.

Références

- [1] Rémi ANDRÉ, Xavier LUCIANI et Eric MOREAU : Joint eigenvalue decomposition algorithms based on first-order Taylor expansion. *IEEE Transactions on Signal Processing*, 68:1716–1727, 2020.
- [2] R. ANDRÉ, Xavier LUCIANI et Eric MOREAU : A new class of block coordinate algorithms for the joint eigenvalue decomposition of complex matrices. *Signal Processing*, 145:78 – 90, 2018.
- [3] A. BOUDJELLAL, A. MESLOUB, K. ABED-MERAIM et A. BELOUHRANI : Separation of dependent autoregressive sources using joint matrix diagonalization. *IEEE Signal Processing Letters*, 22(8):1180–1183, Aug 2015.
- [4] L. DE LATHAUWER, B. DE MOOR et J. VANDEWALLE : Computation of the canonical decomposition by means of a simultaneous Schur decomposition. *SIAM Journal on Matrix Analysis and Applications*, 26(2):295–327, 2004.
- [5] R. IFERROUDJENE, K. ABED-MERAIM et A. BELOUHRANI : A new Jacobi-like method for joint diagonalization of arbitrary non-defective matrices. *Applied Mathematics and Computation*, 211(2):363–373, 2009.
- [6] X. LUCIANI et L. ALBERA : Canonical polyadic decomposition based on joint eigenvalue decomposition. *Chemometrics and Intelligent Laboratory Systems*, 132(0): 152 – 167, 2014.
- [7] X. LUCIANI et L. ALBERA : Joint eigenvalue decomposition of non-defective matrices based on the LU factorization with application to ICA. *IEEE Transactions on Signal Processing*, 63(17):4594–4608, Sept 2015.
- [8] A. MESLOUB, A. BELOUHRANI et K. ABED-MERAIM : Efficient and stable joint eigenvalue decomposition based on generalized Givens rotations. *In 2018 26th European Signal Processing Conference (EUSIPCO)*, pages 1247–1251, Sep. 2018.
- [9] E. MOREAU et O. MACCHI : High-order contrasts for self-adaptive source separation. *International Journal of Adaptive Control and Signal Processing*, 10:19 – 46, 01 1996.
- [10] F. ROEMER et M. HAARDT : A semi-algebraic framework for approximate CP decompositions via simultaneous matrix diagonalizations (SECSI). *Signal Processing*, 93(9):2722 – 2738, 2013.