Received: 3 March 2018 | Revised: 4 February 2019 | Accepted: 5 March 2019

DOI: 10.1002/nla.2238

RESEARCH ARTICLE

WILEY

Solving the general joint block diagonalization problem via linearly independent eigenvectors of a matrix polynomial

Yunfeng Cai[1] | Guanghui Cheng[2] | Decai Shi[3]

[1]Cognitive Computing Lab, Baidu Research, Beijing, China

[2]School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, China

[3]School of Mathematics and Statistics, Xidian University, Xi'an, China

Correspondence
Yunfeng Cai, Cognitive Computing Lab, Baidu Research, East Xibeiwang Road, Baidu Technology Park, Beijing 100193, China.
Email: yfcai1116@gmail.com

Funding information
National Natural Science Foundation of China, Grant/Award Number: 11671023, 11421101, and 11301013

Summary

In this paper, we consider the exact/approximate general joint block diagonalization (GJBD) problem of a matrix set $\{A_i\}_{i=0}^{p}$ ($p \geq 1$), where a nonsingular matrix W (often referred to as a diagonalizer) needs to be found such that the matrices $W^{\mathrm{H}}A_iW$'s are all exactly/approximately block-diagonal matrices with as many diagonal blocks as possible. We show that the diagonalizer of the exact GJBD problem can be given by $W = [x_1, x_2, \ldots, x_n]\Pi$, where Π is a permutation matrix and x_i's are eigenvectors of the matrix polynomial $P(\lambda) = \sum_{i=0}^{p} \lambda^i A_i$, satisfying that $[x_1, x_2, \ldots, x_n]$ is nonsingular and where the geometric multiplicity of each λ_i corresponding with x_i is equal to 1. In addition, the equivalence of all solutions to the exact GJBD problem is established. Moreover, a theoretical proof is given to show why the approximate GJBD problem can be solved similarly to the exact GJBD problem. Based on the theoretical results, a three-stage method is proposed, and numerical results show the merits of the method.

KEYWORDS

general joint block diagonalization, matrix polynomial, tensor decomposition

1 | INTRODUCTION

The problem of the joint block diagonalization (JBD) of matrices (also called simultaneously the block diagonalization problem) is a particular decomposition of a third-order tensor in block terms.[1–4] Over the past two decades, such a decomposition has found many applications in independent component analysis[5–8] and semidefinite programming.[9–12] For example, in blind source separation (BSS), people aim to recover source signals from the observed mixtures, without knowing either the distribution of the sources or the mixing process.[13–15] Different assumptions on the source signals lead to different models and methods. Typically, there are three cases: First, the source signals are mutually statistically independent, and the mixing system can be estimated by joint diagonalization (JD), for example, JADE,[16] eJADE,[17] SOBI,[18] and Hessian ICA[19,20]; second, there are several groups of signals, in which components from different groups are mutually statistically independent and statistical dependence occurs between components in the same group (known as multidimensional BSS or group BSS), and the mixing system can be estimated by JBD[5,7]; third, the number of groups and the size of each group are unknown, and the mixing system can be estimated by general joint block diagonalization (GJBD).

To proceed, in what follows, we reuse some definitions and notations and reformulate the JBD problem and the GJBD problem mathematically as in the works of Cai and Liu[21] and Cai et al.[22]

Definition 1. We call $\tau_n = (n_1, n_2, \ldots, n_t)$ a *partition* of positive integer n if $n_1, n_2, \ldots, n_t$ are all positive integers and their sum is n, that is, $\sum_{i=1}^{t} n_i = n$. The integer t is called the *cardinality* of the partition τ_n, denoted by $\mathrm{card}(\tau_n)$. The set of all partitions of n is denoted by $\mathbb{T}_n$.

Definition 2. Given a partition $\tau_n = (n_1, n_2, \ldots, n_t) \in \mathbb{T}_n$, for any $n \times n$ matrix A, define its block-diagonal part and off–block-diagonal part associated with τ_n as

$$\mathrm{Bdiag}_{\tau_n}(A) = \mathrm{diag}(A_{11}, A_{22}, \ldots, A_{tt}), \quad \mathrm{OffBdiag}_{\tau_n}(A) = A - \mathrm{Bdiag}_{\tau_n}(A),$$

respectively, where A_{ii} is $n_i \times n_i$ for $i = 1, 2, \ldots, t$. A matrix A is referred to as a τ_n block-diagonal matrix if $\mathrm{OffBdiag}_{\tau_n}(A) = 0$. The set of all τ_n block-diagonal matrices is denoted by $\mathbb{D}_{\tau_n}$.

Let $\mathbb{S}_n, \mathbb{H}_n, \mathbb{O}_n, \mathbb{U}_n, \mathbb{GL}(n, \mathbb{R})$, and $\mathbb{GL}(n, \mathbb{C})$ denote the $n \times n$ matrix set of a real symmetric matrix, a complex Hermitian matrix, a real orthogonal matrix, a complex unitary matrix, a real nonsingular matrix, and a complex nonsingular matrix, respectively. Let $\mathbb{A}_n = \mathbb{S}_n, \mathbb{H}_n, \mathbb{R}^{n\times n}$, or $\mathbb{C}^{n\times n}$, $\mathbb{W}_n = \mathbb{O}_n, \mathbb{U}_n, \mathbb{GL}(n, \mathbb{R})$, or $\mathbb{GL}(n, \mathbb{C})$. Then, the JBD problem and the GJBD problem can be formulated as follows.

The JBD problem. Given a matrix set $\mathcal{A} = \{A_i\}_{i=0}^p$ with $A_i \in \mathbb{A}_n$ and a partition $\tau_n = (n_1, n_2, \ldots, n_t) \in \mathbb{T}_n$, find a matrix $W = W(\tau_n) \in \mathbb{W}_n$ such that $W^\star A_i W \in \mathbb{D}_{\tau_n}$ for $i = 0, 1, \ldots, p$, that is,

$$W^\star A_i W = \mathrm{diag}\left(A_i^{(11)}, A_i^{(22)}, \ldots, A_i^{(tt)}\right), \quad \text{for} \quad i = 0, 1, \ldots, p, \tag{1}$$

where $A_i^{(jj)}$ is $n_j \times n_j$ for $j = 1, 2, \ldots, t$, and the symbol $(\cdot)^\star$ stands for the transpose of a real matrix or the conjugate transpose of a complex matrix.

The GJBD problem. Given a matrix set $\mathcal{A} = \{A_i\}_{i=0}^p$ with $A_i \in \mathbb{A}_n$, find a partition $\tau_n' = (n_1', n_2', \ldots, n_t') \in \mathbb{T}_n$ and a matrix $W = W(\tau_n') \in \mathbb{W}_n$ such that

$$\mathrm{card}\left(\tau_n'\right) = \max\{\mathrm{card}(\tau_n) \,|\, \text{there exists a } W = W(\tau_n) \text{ that solves JBD}\}.$$

The transformation matrix W is often referred to as a diagonalizer. The (G)JBD problem is called symmetric (or Hermitian) (G)JBD if $\mathbb{A}_n = \mathbb{S}_n$ (or $\mathbb{H}_n$), exact (or approximate) (G)JBD if (1) is satisfied exactly (or approximately), orthogonal (or nonorthogonal) (G)JBD if $\mathbb{A}_n = \mathbb{R}^{n\times n}$ and $\mathbb{W}_n = \mathbb{O}_n$ (or $\mathbb{GL}(n, \mathbb{R})$), and, similarly, unitary (or nonunitary) (G)JBD if $\mathbb{A}_n = \mathbb{C}^{n\times n}$ and $\mathbb{W}_n = \mathbb{U}_n$ (or $\mathbb{GL}(n, \mathbb{C})$).

In practical applications, the matrices A_i's are usually constructed from empirical data; as a result, the exact JBD problem has no solutions, and the exact GJBD problem has only the trivial solution $((n), I_n)$. Consequently, the approximate (G)JBD problem is considered instead. For the JBD problem, it is natural to formulate it as a constrained optimization problem $C(W) = \min$, where $C(\cdot)$ is a cost function used to measure the off–block-diagonal parts of A_i's, and W is a diagonalizer in certain feasible sets. Different cost functions and feasible sets, together with various optimization methods, lead to many numerical methods. Since this paper mainly concentrates on algebraic methods, we will not list the detailed literature on the optimization methods; we refer the readers to the works of Chabriel et al.,[23] De Lathauwer,[24] and Tichavsky et al.[25] as well as the references therein. For the GJBD problem, one needs to minimize the off–block-diagonal parts of A_i's and maximize the number of diagonal blocks simultaneously; it is difficult to formulate it as a simple optimization problem that can be easily solved. By assuming that the GJBD problem shares the same local minima with the JD problem, the GJBD problem is simply solved with a JD algorithm, followed by a permutation, which is used to reveal the block structure.[8,26]

Without good initial guesses, optimization methods may suffer from slow convergence or converge to degenerate solutions. Algebraic methods, on the other hand, are able to find a solution in finite steps with predictable computational costs. Even if the solutions returned by algebraic methods are of "low quality," they are usually good initial guesses for optimization methods. In the current literature, the algebraic methods for the GJBD problem fall into two categories: One is based on matrix $*$-algebra (see, for example, the works of de Klerk et al.,[27] Maehara and Murota,[28,29] and Murota et al.[30] for the orthogonal GJBD problem and a recent generation by Cai and Liu[21] for the nonorthogonal GJBD problem), and the other is based on a matrix polynomial (see the work of Cai et al.[22] for the Hermitian GJBD problem). In the former category, the null space of a linear operator needs to be computed, which requires $\mathcal{O}(n^6)$ flops; thus, for problems with large n values, such an approach will be quite expensive for computation. In this paper, we will focus on the latter category, which we will show later that it only requires $\mathcal{O}(n^3)$ flops.

As the results in this paper are an extension of those in the work of Cai et al.,[22] in what follows, we summarize some related results therein. For a Hermitian matrix set $\{A_i\}_{i=0}^p$, the corresponding matrix polynomial is constructed as $P_{\mathcal{A}}(\lambda) = \sum_{i=0}^p \lambda^i A_i$. Assuming that $P_{\mathcal{A}}(\lambda)$ is regular and has only simple eigenvalues and using the spectral decomposition of the Hermitian matrix polynomial, theoretically, it is shown that the column vectors of the diagonalizer of the exact Hermitian GJBD problem of $\{A_i\}_{i=0}^p$ can be given by n linearly independent eigenvectors (in a certain order)

of $P_{\mathcal{A}}(\lambda)$ (see corollary 3.5 in the work of Cai et al.[22]); all solutions to the Hermitian GJBD problem are equivalent, that is, all solutions are unique up to block permutations and block-diagonal transformations (see theorem 3.8 in the work of Cai et al.[22]). Therefore, one can solve the Hermitian GJBD problem by finding n linearly independent eigenvectors $x_1, x_2, \ldots, x_n$ of $P_{\mathcal{A}}(\lambda)$, followed by determining a permutation Π via revealing the block-diagonal structure of $\Pi^{\mathrm{T}}[x_1, \ldots, x_n]^{\mathrm{H}} A_i [x_1, \ldots, x_n]\Pi$ (MPSA-II). Numerically, it is shown that MPSA-II, although designed to solve the exact Hermitian GJBD problem, is able to deal with the approximate Hermitian GJBD problem to some extent. However, the approach in the work of Cai et al.[22] suffers from the following three disadvantages: First, the proofs are difficult to follow if the readers are unfamiliar with the spectral decomposition of a matrix polynomial; second, there is no theoretical proof to show why the approach is applicable for the approximate Hermitian GJBD problem; and third, the approach cannot be used to solve a non-Hermitian GJBD problem directly.* In this paper, we try to give remedies. For a matrix set $\mathcal{A} = \{A_i\}_{i=0}^{p}$ (A_i's are not necessarily Hermitian), we still construct the matrix polynomial as $P_{\mathcal{A}}(\lambda) = \sum_{i=0}^{p} \lambda^i A_i$. Let $P_{\mathcal{A}}(\lambda)$ be regular and $x_1, x_2, \ldots, x_n$ be n linearly independent eigenvectors of $P_{\mathcal{A}}(\lambda)$. Under the assumption that the geometric multiplicities of the corresponding eigenvalues are equal to 1 (weaker than that "all eigenvalues are simple"), we show that the diagonalizer of the exact GJBD problem can be written as $X\Pi = [x_1, x_2, \ldots, x_n]\Pi$, where Π is a permutation matrix; all solutions to the exact GJBD problem are equivalent. The proofs of these results are easy to follow, without using the spectral decomposition of a matrix polynomial. Furthermore, using the perturbation theory, we give a theoretical proof for using $X\Pi$ as the diagonalizer for the approximate GJBD problem. Lastly, a three-stage method, which is modified from MPSA-II in the work of Cai et al.,[22] is proposed. Numerical examples show that the proposed method is effective and efficient. It is worth mentioning here that, although all results are established for a general complex matrix set, the proposed method is also applicable to a real matrix set, and a real diagonalizer can be found (see Section 4.4).

The rest of this paper is organized as follows. In Section 2, we give some preliminary results on matrix polynomials and motivations for using a matrix polynomial to solve the GJBD problem. In Section 3, the main results are presented. A numerical method and numerical examples are given in Sections 4 and 5, respectively. Finally, we present some concluding remarks in Section 6.

Notation. The imaginary unit $\sqrt{-1}$ is denoted by $\imath$. For a matrix $A = [a_{ij}]$, A^{T} and A^{H} stand for the transpose and complex transpose of A, respectively, and $|A|$, $\|A\|_2$, and $\|A\|_F$ denote $[|a_{ij}|]$, the 2-norm, and the Frobenius norm, respectively. The eigenvalue sets of a square matrix A and a matrix polynomial $P(\lambda)$ are denoted by $\lambda(A)$ and $\lambda(P)$, respectively. The MATLAB convention is adopted to access the entries of vectors and matrices. The set of integers from i to j inclusive is $i : j$. For a matrix A, its submatrices $A(k : \ell, i : j)$, $A(k : \ell, :)$, and $A(:, i : j)$ consist of intersections of row k to row ℓ and column i to column j, row k to row ℓ and all columns, and all rows and column i to column j, respectively.

2 | PRELIMINARY AND MOTIVATION

A matrix polynomial of degree p is defined as

$$P(\lambda) \overset{\text{def}}{=} \lambda^p A_p + \lambda^{p-1} A_{p-1} + \cdots + A_0, \tag{2}$$

where the coefficient matrices A_i's are all $n \times n$ matrices and $A_p \neq 0$. $P(\lambda)$ is called regular if $\det(P(\lambda))$ is not identically zero. *Throughout the rest of this paper, we will always assume that $P(\lambda)$ is regular.* A scalar λ is called an eigenvalue of $P(\lambda)$ if $\det(P(\lambda)) = 0$. A nonzero vector is called the corresponding eigenvector if $P(\lambda)x = 0$. Such λ together with x are called an eigenpair of $P(\lambda)$, denoted by (λ, x).

The polynomial eigenvalue problem (PEP) $P(\lambda)x = 0$ can be transformed into a generalized eigenvalue problem (GEP) $(\lambda M + N)u = 0$, where

$$u = u(x, \lambda) \overset{\text{def}}{=} [x^{\mathrm{T}}, \lambda x^{\mathrm{T}}, \ldots, \lambda^{p-1} x^{\mathrm{T}}]^{\mathrm{T}}, \tag{3}$$

*By constructing a Hermitian matrix polynomial $\widehat{P}_{\mathcal{A}}(\lambda) \overset{\text{def}}{=} \lambda^{2p+1}(A_p + A_p^{\mathrm{H}}) + \lambda^{2p}\imath(A_p - A_p^{\mathrm{H}}) + \cdots + \lambda(A_0 + A_0^{\mathrm{H}}) + \imath(A_0 - A_0^{\mathrm{H}})$, one can still follow the approach in the work of Cai et al.[22] to solve the general GJBD problem of $\{A_i\}_{i=0}^{p}$, but the degree of $\widehat{P}_{\mathcal{A}}(\lambda)$ is $2p + 1$, almost twice as many as the degree of $P_{\mathcal{A}}(\lambda)$.

M and N are some $np \times np$ matrices. Such a transformation is called linearization. Linearizations are not unique, among which a commonly used one can be given by

$$M = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ 0 & I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & I & 0 \\ 0 & \cdots & 0 & 0 & A_p \end{bmatrix}, \quad N = \begin{bmatrix} 0 & -I & 0 & \cdots & 0 \\ 0 & 0 & -I & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & -I \\ A_0 & A_1 & \cdots & A_{p-2} & A_{p-1} \end{bmatrix}. \tag{4}$$

For more linearizations and some structure-preserving linearizations for structured matrix polynomials, we refer the readers to the works of Higham et al.,[31] Mackey et al.,[32] and Mehrmann and Watkins.[33] The eigenvalues and eigenvectors of the PEP can be obtained via those of the GEP, and vice versa.

At first glance, it seems that the matrix polynomial is not related to the GJBD problem at all. However, in fact, they are closely related. Let us consider the GJBD problem of three matrices $\{A_0, A_1, A_2\}$ with

$$A_0 = \begin{bmatrix} 7 & 8 & 9 \\ 4 & -12 & -8 \\ 5 & -4 & 7 \end{bmatrix}, \quad A_1 = \begin{bmatrix} -8 & 8 & 8 \\ -4 & 4 & 0 \\ -4 & 12 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 5 & 0 & 3 \\ -8 & 4 & -4 \\ -5 & 4 & 1 \end{bmatrix}.$$

The eigenvector matrix X and the eigenvalue matrix T of the PEP $(\lambda^2 A_2 + \lambda A_1 + A_0)x = 0$ can be given by

$$X = [x_1, x_2, \ldots, x_6] = \begin{bmatrix} -0.1690 & -0.5774 & 0.3981 + 0.4094\imath & 0.3981 - 0.4094\imath & -0.5774 & 0.4904 \\ -0.9710 & -0.5774 & 0.1108 + 0.5792\imath & 0.1108 - 0.5792\imath & -0.5774 & -0.7205 \\ -0.1690 & 0.5774 & 0.3981 + 0.4094\imath & 0.3981 - 0.4094\imath & 0.5774 & 0.4904 \end{bmatrix},$$

$$T = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_6) = \operatorname{diag}(-3.5830, 3.0000, -0.5283 + 1.3793\imath, -0.5283 - 1.3793\imath, 1.0000, 0.6396).$$

Let $W = X(:, [2, 1, 6])$, then

$$W^T A_0 W = \begin{bmatrix} 4.0000 & -0.0000 & 0.0000 \\ -0.0000 & -10.5142 & -8.3243 \\ 0.0000 & -13.1077 & 0.5034 \end{bmatrix}, \quad W^T A_1 W = \begin{bmatrix} -5.3333 & -0.0000 & -0.0000 \\ 0.0000 & 6.2833 & 7.4702 \\ -0.0000 & -6.8800 & -4.5381 \end{bmatrix},$$

$$W^T A_2 W = \begin{bmatrix} 1.3333 & -0.0000 & 0.0000 \\ 0.0000 & 2.5726 & 8.6677 \\ 0.0000 & -0.8992 & 5.8646 \end{bmatrix}.$$

By calculations, we can show that the 2×2 blocks in $W^{\mathrm{T}} A_0 W$, $W^{\mathrm{T}} A_1 W$, and $W^{\mathrm{T}} A_2 W$ cannot be simultaneously diagonalized. Therefore, $((1, 2), W)$ is a solution to the GJBD problem of $\{A_0, A_1, A_2\}$. This example shows that the GJBD problem can be indeed solved via linearly independent eigenvectors of a matrix polynomial; in the subsequent sections, we will present the theoretical proofs.

Given a matrix set $\mathcal{A} = \{A_i\}_{i=0}^{p}$, the matrix polynomial $P_{\mathcal{A}}(\lambda)$ is constructed in a particular order of the matrices in $\mathcal{A}$. However, the matrices are not ordered in any way for the GJBD problem of $\mathcal{A}$. Later, we will see that the results in this paper do not depend on such an order. Hence, it suffices to show the results for only one matrix polynomial, for example, $P_{\mathcal{A}}(\lambda)$.

3 | MAIN RESULTS

In this section, we give our main results for the exact and approximate GJBD problems in Sections 3.1 and 3.2, respectively.

3.1 | On exact GJBD problem

In this subsection, we first characterize the diagonalizer W and then show the equivalence of the solutions.

Theorem 1. *Given a matrix set $\mathcal{A} = \{A_i\}_{i=0}^p$. Let $P_{\mathcal{A}}(\lambda) = \sum_{i=0}^p \lambda^i A_i$, $X = [x_1, x_2, \dots, x_n]$, $T = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$, where (λ_j, x_j), for $j = 1, 2, \dots, n$, are n eigenpairs of $P_{\mathcal{A}}(\lambda)$. Assume that X is nonsingular and that the geometric multiplicities of $\lambda_1, \lambda_2, \dots, \lambda_n$ are all equal to 1. If the GJBD problem has a solution (τ_n, W), then there exist a permutation matrix Π and a nonsingular matrix $D \in \mathbb{D}_{\tau_n}$ such that $WD = X\Pi$, that is, $(\tau_n, X\Pi)$ is also a solution to the GJBD problem.*

Proof. As (τ_n, W) is a solution to the GJBD problem, we have

$$D_i \overset{\text{def}}{=} W^{\mathrm{H}} A_i W = \mathrm{diag}\left(A_i^{(11)}, \dots, A_i^{(tt)}\right) \in \mathbb{D}_{\tau_n} \text{ for } i = 0, 1, \dots, p. \tag{5}$$

Since (λ_j, x_j), for $j = 1, 2, \dots, n$, are eigenpairs of $P_{\mathcal{A}}(\lambda)$, we also have

$$A_p X T^p + A_{p-1} X T^{p-1} + \cdots + A_0 X = 0. \tag{6}$$

Premultiplying (6) by W^{H} and using (5), we get

$$D_p W^{-1} X T^p + D_{p-1} W^{-1} X T^{p-1} + \cdots + D_0 W^{-1} X = 0. \tag{7}$$

Let $Y = W^{-1}X = [y_1, y_2, \dots, y_n]$, then (λ_j, y_j), for $j = 1, 2, \dots, n$, are eigenpairs of $P_D(\lambda) = \sum_{i=0}^p \lambda^i D_i$, and the geometric multiplicities of λ_j's, as eigenvalues of $P_D(\lambda)$, are all equal to 1. Denoting $\mathcal{A}_j = \{A_i^{(jj)}\}_{i=0}^p$, we know that, for each λ_j, it belongs to a unique $\lambda(P_{\mathcal{A}_k})$ since $\lambda(P_D) = \cup_{j=1}^t \lambda(P_{\mathcal{A}_j})$, and the geometric multiplicities of λ_j's are all equal to 1. Let $n_1', n_2', \dots, n_t'$ be the numbers of λ_j's in $\lambda(P_{\mathcal{A}_1})$, $\lambda(P_{\mathcal{A}_2})$, $\dots$, $\lambda(P_{\mathcal{A}_t})$, respectively. Then, there exists a permutation matrix Π such that

$$W^{-1}X\Pi = \begin{bmatrix} Y_{11} & \cdots & Y_{1t} \\ \vdots & \ddots & \vdots \\ Y_{t1} & \cdots & Y_{tt} \end{bmatrix}, \quad \Pi^T T \Pi = \mathrm{diag}(T_1, \dots, T_t),$$

where $Y_{jk} \in \mathbb{C}^{n_j \times n_k'}$ for $1 \le j, k \le t$, $T_j \in \mathbb{C}^{n_j' \times n_j'}$, $\lambda(T_j) \subset \lambda(P_{\mathcal{A}_j})$ for $j = 1, 2, \dots, t$. The assumption that the geometric multiplicities of $\lambda_1, \lambda_2, \dots, \lambda_n$ are all equal to 1 implies that $\lambda(T_j) \cap \lambda(P_{\mathcal{A}_k}) = \emptyset$ for $j \ne k$; therefore, by (7), we have $Y_{jk} = 0$ for $j \ne k$. Thus, for any $1 \le k \le t$, it holds that

$$\begin{aligned} n &= \mathrm{rank}(W^{-1}X\Pi) = \mathrm{rank}(Y_{kk}) + \mathrm{rank}(\mathrm{diag}(Y_{11}, \dots, Y_{k-1,k-1}, Y_{k+1,k+1}, \dots, Y_{tt})) \\ &\le \min\{n_k, n_k'\} + \min\{n - n_k, n - n_k'\} = \min\{n_k, n_k'\} + n - \max\{n_k, n_k'\} \le n. \end{aligned}$$

Then, it follows that $\min\{n_k, n_k'\} = \max\{n_k, n_k'\}$, and hence, $n_k = n_k'$. Thus, $D \overset{\text{def}}{=} W^{-1}X\Pi \in \mathbb{D}_{\tau_n}$ is a nonsingular τ_n block-diagonal matrix. The conclusion follows. □

Based on Theorem 1, we can solve the exact GJBD problem by finding n linearly independent eigenvectors X of $P_{\mathcal{A}}(\lambda)$ and then determining a permutation Π by revealing the block structure of $\Pi^{\mathrm{T}} X^{\mathrm{H}} A_i X \Pi$. The clustering methods[25,34] can find such a permutation; we will discuss the details in Section 4.

Let $\tau_n = (n_1, n_2, \dots, n_t) \in \mathbb{T}_n$, (τ_n, W) be a solution to the GJBD problem, then $(\tau_n \Pi_t, WD\Pi)$ also solves the GJBD problem, where $\Pi_t \in \mathbb{R}^{t \times t}$ is a permutation matrix; $\Pi \in \mathbb{R}^{n \times n}$ is a block permutation matrix corresponding with Π_t, which can be obtained by replacing the "1" and "0" elements in the ith row of Π_t by a permutation matrix of order n_i and zero matrices of right sizes, respectively; and $D \in \mathbb{D}_{\tau_n}$ is nonsingular. For two solutions (τ_n, W) and $(\hat{\tau}_n, \widehat{W})$ to the GJBD problem, we say that $(\hat{\tau}_n, \widehat{W})$ is equivalent to (τ_n, W) if there exist a permutation matrix Π_t and a nonsingular matrix $D \in \mathbb{D}_{\tau_n}$ such that $(\hat{\tau}_n, \widehat{W}) = (\tau_n \Pi_t, WD\Pi)$, where Π is the block permutation matrix corresponding with Π_t.

Next, we show that all solutions to the GJBD problem are equivalent, under mild conditions.

Theorem 2. *Given a matrix set $\mathcal{A} = \{A_i\}_{i=0}^p$, if there exists a matrix set $\mathcal{B} = \{B_i\}_{i=0}^q$ such that $\mathrm{span}\{B_0, B_1, \dots, B_q\} = \mathrm{span}\{A_0, A_1, \dots, A_p\}^\dagger$ and $P_{\mathcal{B}}(\lambda) = \sum_{i=0}^q \lambda^i B_i$ has n eigenpairs (λ_j, x_j) for $j = 1, 2, \dots, n$ satisfying that $X = [x_1, x_2, \dots, x_n]$ is nonsingular and the geometric multiplicities of $\lambda_1, \lambda_2, \dots, \lambda_n$ are all equal to 1, then all solutions to the GJBD problem of $\mathcal{A}$ are equivalent.*

†The space spanned by several matrices is defined as $\mathrm{span}\{A_0, A_1, \dots, A_p\} = \{\sum_{i=0}^p \alpha_i A_i \mid [\alpha_0, \alpha_1, \dots, \alpha_p]^{\mathrm{T}} \in \mathbb{C}^{p+1}\}$.

Proof. First, by Theorem 1, for any two solutions (τ_n, W) and $(\hat{\tau}_n, \widehat{W})$ to the GJBD problem of $\mathcal{B}$, there exist two permutation matrices Π and $\widehat{\Pi}$ and two nonsingular matrices $D \in \mathbb{D}_{\tau_n}$ and $\widehat{D} \in \mathbb{D}_{\hat{\tau}_n}$ such that $WD = X\Pi$ and $\widehat{W}\widehat{D} = X\widehat{\Pi}$. Let $H = \sum_{i=0}^{q} |X^H B_i X|$, $\hat{H} = [\hat{h}_{ij}]$ with

$$\hat{h}_{ij} = \begin{cases} 1, & \text{if } i \neq j \text{ and } h_{ij} \neq 0, \\ 0, & \text{otherwise,} \end{cases}$$

where h_{ij} is the (i,j) entry of H. On the one hand, using the fact that (τ_n, W) and $(\hat{\tau}_n, \widehat{W})$ are both solutions to the GJBD problem, we know that $\Pi^T \widehat{H} \Pi \in \mathbb{D}_{\tau_n}$ and $\widehat{\Pi}^T \widehat{H} \widehat{\Pi} \in \mathbb{D}_{\hat{\tau}_n}$, and $\mathrm{card}(\tau_n) = \mathrm{card}(\hat{\tau}_n)$. On the other hand, notice that $\widehat{H}$ is a vertex-adjacency matrix of a graph G, with G having t components $G_1, G_2, \ldots, G_t$, and G_i is connected with n_i vertices for $i = 1, 2, \ldots, t$, where n_i is the ith entry of τ_n. No matter how we relabel the vertices of G, the collection of the numbers of vertices of all connected components of G remains invariant. Therefore, there exists a permutation matrix Π_t of order t such that $\hat{\tau}_n = \tau_n \Pi_t$ and $\widetilde{\Pi} = \Pi^T \widehat{\Pi}$ is the block permutation matrix corresponding with Π_t. Then, it follows that

$$\widehat{W} = X\widehat{\Pi}\widehat{D}^{-1} = (X\Pi D^{-1})D(\Pi^T \widehat{\Pi})\widehat{D}^{-1} = WD\widetilde{\Pi}\widehat{D}^{-1} = WD(\widetilde{\Pi}\widehat{D}^{-1}\widetilde{\Pi}^T)\widetilde{\Pi}.$$

In other words, all solutions to the GJBD problem of $\mathcal{B}$ are equivalent. Second, notice that, for a partition τ_n, W is a diagonalizer of the JBD problem of $\mathcal{A}$ if and only if W is a diagonalizer of the JBD problem of $\mathcal{B}$, since $\mathrm{span}\{B_0, B_1, \ldots, B_q\} = \mathrm{span}\{A_0, A_1, \ldots, A_p\}$. The conclusion follows immediately. □

Theorem 2 implies that the solution to the GJBD problem does not depend on the choices of the matrix polynomials (the coefficient matrices of the matrix polynomial can be in any order or even linear combinations of A_i's) or on the choices of n linearly independent eigenvectors.

Remark 1. The conclusions of Theorems 1 and 2 are similar to that of corollary 3.5 and theorem 3.8 in the work of Cai et al.,[22] respectively. However, the latter two are established under stronger assumptions. To be specific, the latter requires that the matrices A_i's are all Hermitian and that all eigenvalues of $P_{\mathcal{A}}(\lambda)$ are simple, whereas the former only requires that the geometric multiplicities of the eigenvalues corresponding with the n linearly independent eigenvectors are equal to 1 and that A_i's are not necessarily Hermitian.

3.2 | On approximate GJBD problem

In this subsection, we show that the solution to the approximate GJBD problem can also be written in the form of $(\tau_n, X\Pi)$, where τ_n is some partition of n, X is a nonsingular matrix whose columns are eigenvectors of $P_{\mathcal{A}}(\lambda)$, and Π is some permutation matrix. The following two lemmas are needed for the proof.

Lemma 1. *Let $x, y \in \mathbb{C}^n$ be two nonzero vectors, then $\min_{t\in\mathbb{C}} \|x - ty\|_2 = \|x\|_2 \sin\angle(x, y)$, where $\angle(x, y)$ is the angle between the subspaces spanned by x and y, respectively.*

The proof of Lemma 1 is simple, and we will omit it here. The following lemma is rewritten from theorem 4.1 in the work of Nakatsukasa and Tisseur.[35]

Lemma 2. *Let $(\hat{\lambda}, \hat{x})$ be an approximate eigenpair of $P(\lambda) = \sum_{i=0}^{p} \lambda^i A_i$ with residual $P(\hat{\lambda})\hat{x} \neq 0$. Suppose $L(\lambda) = \lambda M + N$ is a linearization of $P(\lambda)$ given by (4) and $L(\lambda)$ has a generalized Schur form*

$$Q^H M Z = \begin{bmatrix} \alpha_1 & * \\ 0 & M_1 \end{bmatrix}, \quad Q^H N Z = \begin{bmatrix} \beta_1 & * \\ 0 & N_1 \end{bmatrix}, \lambda_1 = -\frac{\beta_1}{\alpha_1},$$

in which M_1 and N_1 are both upper triangular. Then, the eigenvector x_1 of $P(\lambda)$ corresponding with λ_1 satisfies

$$\sin\angle(\hat{x}, x_1) \leq \frac{g\|P(\hat{\lambda})\hat{x}\|_2}{\sqrt{\sum_{i=0}^{p-1} |\hat{\lambda}|^{2i}}\|\hat{x}\|_2}, \tag{8}$$

where

$$g = g(\lambda_1, \hat{\lambda}; P(\lambda)) \overset{\text{def}}{=} \frac{1}{\|(\hat{\lambda}M_1 + N_1)^{-1}\|_2}. \tag{9}$$

Define a matrix set $\mathcal{W}$ as

$$\mathcal{W} \overset{\text{def}}{=} \{W = [w_1, w_2, \ldots, w_n] \mid \det(W) \neq 0, \|w_i\|_2 = 1 \text{ for } i = 1, 2, \ldots, n\}. \tag{10}$$

Now, we are ready to present the third main theorem.

Theorem 3. *Given a matrix set $\mathcal{A} = \{A_i\}_{i=0}^p$, for a partition τ_n, assume that there exists a matrix $W \in \mathcal{W}$ such that*

$$\|E_i\|_F \leq \mu \|W^{\mathrm{H}} A_i W\|_F, \tag{11}$$

where $E_i = \text{OffBdiag}_{\tau_n}(W^{\mathrm{H}} A_i W)$, and μ is the smallest real number such that (11) holds for all $i = 0, 1, \ldots, p$. Let (λ_j, x_j) for $j = 1, 2, \ldots, n$ be n eigenpairs of $P_{\mathcal{A}}(\lambda) = \sum_{i=0}^p \lambda^i A_i$ and $X = [x_1, x_2, \ldots, x_n] \in \mathcal{W}$. Denote $D_i = \text{Bdiag}_{\tau_n}(W^{\mathrm{H}} A_i W)$, $\mathcal{D} = \{D_i\}_{i=0}^p$, $P_{\mathcal{D}}(\lambda) = \sum_{i=0}^p \lambda^i D_i$. For $j = 1, 2, \ldots, n$, let $\mu_j = \text{argmin}_{\mu \in \lambda(P_{\mathcal{D}})} g(\mu, \lambda_j; P_{\mathcal{D}}(\lambda))$, y_j be the eigenvector of $P_{\mathcal{D}}(\lambda)$ corresponding with μ_j. Further assume that the geometric multiplicities of μ_j's are all equal to 1, $Y = [y_1, y_2, \ldots, y_n]$ is nonsingular, and (τ_n, I_n) is a solution to the exact GJBD problem of $\mathcal{D}$, that is, $\mathcal{D}$ cannot be further block-diagonalized.

Denote

$$g_j = g(\mu_j, \lambda_j; P_{\mathcal{D}}(\lambda)), \quad \text{for } j = 1, 2, \ldots, n, \tag{12a}$$

$$\eta = \max_{1 \leq j \leq n} \frac{g_j \sum_{i=0}^p |\lambda_j|^i \|W^{\mathrm{H}} A_i W\|_{\mathrm{F}}}{\sqrt{\sum_{i=0}^{p-1} |\lambda_j|^{2i}}}. \tag{12b}$$

Then, there exists a permutation matrix Π such that

$$\| \text{OffBdiag}_{\tau_n} \left(\Pi^{\mathrm{T}} X^{\mathrm{H}} A_i X \Pi \right) \|_{\mathrm{F}} \leq C\mu \|W^{\mathrm{H}} A_i W\|_{\mathrm{F}}, \tag{13}$$

where $C = (1 + 2\eta + 3\mu\eta^2)\|W^{-1}X\|_{\mathrm{F}}^2$.

Proof. First, using the assumption that (τ_n, I_n) is a solution to the exact GJBD problem of $\mathcal{D}$, together with Theorem 1, we know that there exists a permutation matrix Π such that $Y\Pi \in \mathbb{D}_{\tau_n}$.

Second, for each $1 \leq j \leq n$, take $(\lambda_j, W^{-1}x_j)$ as an approximate eigenpair of $P_{\mathcal{D}}(\lambda)$. Using Lemma 2, we have

$$\sin \angle(W^{-1}x_j, y_j) \leq \frac{g_j \|P_{\mathcal{D}}(\lambda_j) W^{-1} x_j\|_2}{\sqrt{\sum_{i=0}^{p-1} |\lambda_j|^{2i}} \|W^{-1}x_j\|_2}. \tag{14}$$

Let $\mathcal{E} = \{E_i\}_{i=0}^p$, $P_{\mathcal{E}}(\lambda) = \sum_{i=0}^p \lambda^i E_i$, it holds that $P_{\mathcal{D}}(\lambda) + P_{\mathcal{E}}(\lambda) = W^{\mathrm{H}} P_{\mathcal{A}}(\lambda) W$. Then, it follows from (11) and (14) that

$$\begin{aligned}
\sin \angle(W^{-1}x_j, y_j) &\leq \frac{g_j(\|W^{\mathrm{H}} P_{\mathcal{A}}(\lambda_j) x_j\|_2 + \|P_{\mathcal{E}}(\lambda_j) W^{-1} x_j\|_2)}{\sqrt{\sum_{i=0}^{p-1} |\lambda_j|^{2i}} \|W^{-1}x_j\|_2} \\
&\leq \frac{g_j \|P_{\mathcal{E}}(\lambda_j)\|_2}{\sqrt{\sum_{i=0}^{p-1} |\lambda_j|^{2i}}} \leq \frac{g_j \sum_{i=0}^p |\lambda_j|^i \|E_i\|_2}{\sqrt{\sum_{i=0}^{p-1} |\lambda_j|^{2i}}} \leq \frac{\mu g_j \sum_{i=0}^p |\lambda_j|^i \|W^{\mathrm{H}} A_i W\|_F}{\sqrt{\sum_{i=0}^{p-1} |\lambda_j|^{2i}}} \leq \mu\eta.
\end{aligned}$$

By Lemma 1, there exists a $t_j \in \mathbb{C}$ such that

$$\|W^{-1}x_j - t_j y_j\|_2 = \|W^{-1}x_j\|_2 \sin \angle(W^{-1}x_j, y_j).$$

Let $f_j = W^{-1}x_j - t_j y_j$, then it holds that $\|f_j\|_2 \le \mu\eta\|W^{-1}x_j\|_2$.

Now, denote $\widehat{Y} = [t_1 y_1, \ldots, t_n y_n]$ and $F = [f_1, \ldots, f_n]$, then we have $\|F\|_{\mathrm{F}} \le \mu\eta\|W^{-1}X\|_{\mathrm{F}}$, $X = W(\widehat{Y} + F)$, and

$$\|\widehat{Y}\|_2 = \|W^{-1}X - F\|_2 \le \|W^{-1}X\|_2 + \|F\|_2 \le (1 + \mu\eta)\|W^{-1}X\|_F. \tag{15}$$

Finally, direct calculations give rise to

$$\begin{aligned}
&\| \operatorname{OffBdiag}_{\tau_n}(\Pi^{\mathrm{T}}X^{\mathrm{H}}A_iX\Pi)\|_{\mathrm{F}} \\
&= \|\operatorname{OffBdiag}_{\tau_n}(\Pi^{\mathrm{T}}(\widehat{Y} + F)^{\mathrm{H}}W^{\mathrm{H}}A_iW(\widehat{Y} + F)\Pi - \Pi^{\mathrm{T}}\widehat{Y}^{\mathrm{H}}D_i\widehat{Y}\Pi)\|_F && \text{(16a)} \\
&= \| \operatorname{OffBdiag}_{\tau_n}(\Pi^{\mathrm{T}}\widehat{Y}^{\mathrm{H}}D_iF\Pi + \Pi^{\mathrm{T}}F^{\mathrm{H}}D_i\widehat{Y}\Pi + \Pi^{\mathrm{T}}F^{\mathrm{H}}D_iF\Pi + \Pi^{\mathrm{T}}(\widehat{Y} + F)^{\mathrm{H}}E_i(\widehat{Y} + F)\Pi)\|_{\mathrm{F}} && \text{(16b)} \\
&\le \left(2\|\widehat{Y}\|_2\|F\|_2 + \|F\|_2^2\right)\|D_i\|_{\mathrm{F}} + \|W^{-1}X\|_2^2\|E_i\|_{\mathrm{F}} \\
&\le \left(2\|\widehat{Y}\|_2\|F\|_2 + \|F\|_2^2 + \|W^{-1}X\|_2^2\mu\right)\|W^{\mathrm{H}}A_iW\|_{\mathrm{F}} \\
&\le \left(2(1 + \mu\eta)\mu\eta + \mu^2\eta^2 + \mu\right)\|W^{-1}X\|_{\mathrm{F}}^2\|W^{\mathrm{H}}A_iW\|_{\mathrm{F}} && \text{(16c)} \\
&= C\mu\|W^{\mathrm{H}}A_iW\|_{\mathrm{F}}, && \text{(16d)}
\end{aligned}$$

where (16a) uses $X = W(\widehat{Y} + F)$ and $\operatorname{OffBdiag}_{\tau_n}(\Pi^{\mathrm{T}}\widehat{Y}^{\mathrm{H}}D_i\widehat{Y}\Pi) = 0$, (16b) uses $W^{\mathrm{H}}A_iW = D_i + E_i$, and (17) uses $\|W^{-1}X\|_2 \le \|W^{-1}X\|_{\mathrm{F}}$, $\|F\|_2 \le \|F\|_{\mathrm{F}} \le \mu\eta\|W^{-1}X\|_{\mathrm{F}}$, and (15). This completes the proof. □

Several remarks follow.

Remark 2. If $\mu = 0$, the approximate GJBD problem becomes the exact GJBD problem. The conclusion in Theorem 3 agrees with that in Theorem 1.

Remark 3. Inequality (13) implies that $X\Pi$ is a "suboptimal" diagonalizer. The constant C plays a crucial role in bounding the off–block-diagonal part of $\Pi^{\mathrm{T}}X^{\mathrm{H}}A_iX\Pi$. Notice that $\|W^{-1}X\|_{\mathrm{F}}^2$ dominates the value of C. When W is good-conditioned, $\|W^{-1}X\|_{\mathrm{F}}$ is small. However, if W is ill-conditioned, $\|W^{-1}X\|_{\mathrm{F}}$ can be quite large, which means that $X\Pi$ can be of "low quality." In fact, from the perturbation theory of the JBD problem[36] (see also the perturbation theory of the JD problem in the works of Shi et al.[37] and Afsari[38]), the diagonalizer is sensitive to the perturbation when it is ill-conditioned. Therefore, it is not surprising to conclude that C can be large when W is ill-conditioned.

Remark 4. Due to rounding errors, a defective eigenvalue of a matrix polynomial would become distinct but very close eigenvalues, and the corresponding eigenvectors would be almost linearly dependent. As a result, if we use part of those eigenvectors to form the eigenvector matrix X, X will be ill-conditioned. Therefore, if we can find a good-conditioned X, then the column vectors of X will not be eigenvectors corresponding with a defective eigenvalue.

4 | NUMERICAL METHOD

According to Theorems 1 and 3, solutions to the exact/approximate GJBD problem can be obtained by the following three-stage procedure.

Stage 1—eigenproblem solving stage. Compute n linearly independent eigenvectors $x_1, \ldots, x_n$ of the matrix polynomial $P_A(\lambda)$, and let $X = [x_1, \ldots, x_n]$.

Stage 2—block structure revealing stage. Determine a partition τ_n and a permutation matrix Π such that $\Pi^{\mathrm{T}}X^{\mathrm{H}}A_iX\Pi$ for $i = 0, 1, \ldots, p$ are all approximately τ_n block diagonal, and $\operatorname{card}(\tau_n)$ is maximized.

Stage 3—refinement stage. Refine $X\Pi$ to improve the quality of the diagonalizer.

We call the above three-stage method a partial eigenvector approach with refinement (PEAR) method for the GJBD problem. Next, we will discuss the implementation details of each stage of the above procedure.

4.1 | Stage 1—eigenproblem solving stage

In this stage, we compute n linearly independent eigenvectors of the PEP $P_{\mathcal{A}}(\lambda)$. By section 3 in the work of Pereira,[39] the number of choices for n linearly independent eigenvectors is no less than p. How shall we make such a choice? To answer this question, let us consider the following GJBD problem first.

Example 1. Let $A_i = VD_iV^{\mathrm{H}}$ for $i = 0, 1, 2$, where

$$V = \begin{bmatrix} 0.5377 + 2.7694\imath & 0.8622 + 0.7254\imath & -0.4336 - 0.2050\imath \\ 1.8339 - 1.3499\imath & 0.3188 - 0.0631\imath & 0.3426 - 0.1241\imath \\ -2.2588 + 3.0349\imath & -1.3077 + 0.7147\imath & 3.5784 + 1.4897\imath \end{bmatrix},$$

$$D_0 = \begin{bmatrix} 0.2939 - 0.7873\imath & 0.0137 + 0.0044\imath & -0.0171 + 0.0038\imath \\ 0.0032 - 0.0086\imath & 0.8884 - 2.9443\imath & -1.0689 + 0.3252\imath \\ 0.0031 - 0.0003\imath & -1.1471 + 1.4384\imath & -0.8095 - 0.7549\imath \end{bmatrix},$$

$$D_1 = \begin{bmatrix} -0.1649 + 0.6277\imath & -0.0077 + 0.0022\imath & 0.0037 + 0.0075\imath \\ -0.0109 + 0.0055\imath & 1.0933 - 1.2141\imath & -0.8637 - 0.0068\imath \\ 0.0003 + 0.0110\imath & 1.1093 - 1.1135\imath & 0.0774 + 1.5326\imath \end{bmatrix},$$

$$D_2 = \begin{bmatrix} 1.5442 + 0.0859\imath & -0.0076 + 0.0025\imath & -0.0140 + 0.0062\imath \\ -0.0018 + 0.0142\imath & -1.4916 - 0.6156\imath & -1.0616 - 0.1924\imath \\ -0.0020 + 0.0029\imath & -0.7423 + 0.7481\imath & 2.3505 + 0.8886\imath \end{bmatrix}.$$

Let $W = V^{-\mathrm{H}}$, then we know that (τ_n, W) is a solution to the approximate GJBD problem of $\{A_0, A_1, A_2\}$, where $\tau_n = (1, 2)$.

By calculations, we know that all eigenvalues of the PEP $P_{\mathcal{A}}(\lambda) = \lambda^2 A_2 + \lambda A_1 + A_0$ are simple, and the eigenvector matrix X, whose columns are all of unit length, can be given by

$$X(:, 1:3) = \begin{bmatrix} -0.5269 - 0.3384\imath & -0.3566 + 0.4066\imath & 0.0646 - 0.5182\imath \\ -0.1210 - 0.7587\imath & -0.8248 + 0.0209\imath & 0.0578 - 0.7205\imath \\ -0.1247 + 0.0436\imath & -0.0681 + 0.1491\imath & -0.4029 - 0.2062\imath \end{bmatrix},$$

$$X(:, 4:6) = \begin{bmatrix} 0.1636 - 0.3165\imath & -0.2841 + 0.2244\imath & 0.4363 - 0.4082\imath \\ 0.7059 + 0.6000\imath & -0.4017 - 0.8314\imath & 0.7443 - 0.1520\imath \\ 0.0065 - 0.1214\imath & -0.0599 + 0.1128\imath & -0.0381 - 0.2541\imath \end{bmatrix}.$$

Let $\widehat{W}$ be a 3×3 matrix whose columns are selected from the columns of X. Define

$$f(\tau_n, \widehat{W}) = \sum_{i=0}^{2} \|\mathrm{OffBdiag}_{\tau_n}(\widehat{W}^{\mathrm{H}} A_i \widehat{W})\|_{\mathrm{F}}^2,$$

$$\theta(W, \widehat{W}) = \max\{\angle(W(:,1), \widehat{W}(:,1)), \angle(W(:,[2,3]), \widehat{W}(:,[2,3]))\}.$$

For four different choices of $\widehat{W}$, we compute the condition number of $\widehat{W}$, $f(\tau_n, \widehat{W})$, and $\theta(W, \widehat{W})$, and the results are listed in Table 1.

TABLE 1 Condition number, cost function $f(\tau_n, \widehat{W})$, and angle $\theta(W, \widehat{W})$

Case	$\widehat{W}$	cond($\widehat{W}$)	$f(\tau_n, \widehat{W})$	$\theta(W, \widehat{W})$
1	$X(:, [4, 1, 2])$	2.3e1	0.0048	0.0066
2	$X(:, [4, 2, 3])$	9.2e0	0.1061	0.0154
3	$X(:, [6, 4, 5])$	4.9e2	0.0247	0.8855
4	$X(:, [1, 2, 3])$	1.2e3	9.5550	0.5005

From Example 1, we can see that the qualities of the four approximate diagonalizers $\widehat{W}$ are quite different, although they are all consisted of linearly independent eigenvectors: $\widehat{W}$ in Cases 1 and 2 are good in the sense that $f(\tau_n, \widehat{W})$ and $\theta(W, \widehat{W})$ are small, and $\widehat{W}$ in Case 1 is better; $\widehat{W}$ in Case 3 is not good although $f(\tau_n, \widehat{W})$ is small; $\widehat{W}$ in Case 4 is not even an approximate diagonalizer.

Based on the above observations and Remarks 3 and 4, it is reasonable to choose n linearly independent eigenvectors $x_1, x_2, \ldots, x_n$ such that the condition number of $[x_1, x_2, \ldots, x_n]$ is as small as possible. Then, the task of Stage 1 is reduced to find such n linearly independent eigenvectors. Classic eigensolvers for the PEP concentrate on computing extreme eigenvalues or eigenvalues close to a prescribed number (and their corresponding eigenvectors), which are not suitable for the task. In this paper, we use the following two steps to accomplish the task.

Step 1. Compute k unit length eigenvectors $x_1, \ldots, x_k$ of the PEP by certain eigensolvers, where k is larger than n, for example, a multiple of n, $k = 2n, 4n$. In our numerical tests, we first transform the PEP into the GEP $(\lambda M + N)u = 0$, where u and M and N are given by (3) and (4), respectively. Then, when p is small, we use the QZ method to find all eigenvectors of the GEP; when p is large, we use the Arnoldi method to compute the k largest magnitude eigenvalues and the corresponding eigenvectors of the GEP. Finally, the eigenvectors of the PEP can be obtained via those of the GEP.
Step 2. Compute the QR decomposition of $[x_1, \ldots, x_k]$ with column pivoting, that is, $[x_1, \ldots, x_k]P = QR$, where P is a permutation matrix of order k, Q is unitary, and R is upper triangular with main diagonal entries in a decreasing order. If the (n, n) entry of R is small,[‡] return to Step 1 to find more eigenvectors; else, set X as the first n columns of $[x_1, \ldots, x_k]P$.

The above two-step procedure is perhaps the simplest way to accomplish the task of Stage 1, but it may still be worth developing some particular eigensolvers for it.

4.2 | Stage 2—block structure revealing stage

In this stage, we need to determine a partition τ_n and a permutation matrix Π such that $\Pi^{\mathrm{T}} X^{\mathrm{H}} A_i X \Pi$'s are all approximately τ_n block diagonal, and $\mathrm{card}(\tau_n)$ is maximized. This stage is of great importance for determining the solutions to the GJBD problem; however, without knowing the number of the diagonal blocks, determining a correct τ_n can be very tricky, especially when the noise is high and the block-diagonal structure is fuzzy. From our numerical experience, the clustering method described in the work of Tichavsky et al.[25] is powerful and efficient for finding τ_n and Π. Let

$$H = [h_{ij}] = \sum_{i=0}^{p} \left(|X^{\mathrm{H}} A_i X| + |X^{\mathrm{H}} A_i^{\mathrm{H}} X| \right). \tag{17}$$

H can be taken as a weighted adjacency matrix of a similarity graph, which contains n vertices $v_1, \ldots, v_n$ (data points), and each edge between two vertices v_i and v_j carries a nonnegative weight $h_{ij} \geq 0$ (the similarity between v_i and v_j). Then, revealing the block structure of H amounts to finding a partition of the graph such that the edges between different groups have low weights and the edges within a group have high weights. Simply speaking, this method begins with a trivial clustering that consists of n singletons, and in each subsequent step, it merges two clusters that have the maximum average similarity between their members. We refer the readers to the work of Tichavsky et al.[25] for more details.

4.3 | Stage 3—refinement stage

Theorem 3 only ensures that $\widehat{W} = X\Pi$ is a "suboptimal" diagonalizer. To improve the quality of the diagonalizer, we propose the following refinement procedure.

Let $(\tau_n, \widehat{W})$ be an approximate solution to the GJBD problem produced by the first two stages of PEAR. Suppose $\tau_n = (n_1, \ldots, n_t)$, $\widehat{W} = [\widehat{W}_1, \ldots, \widehat{W}_t]$ with $\widehat{W}_j \in \mathbb{C}^{n \times n_j}$ for $j = 1, \ldots, t$. Denote $\widehat{W}_{-j} = [\widehat{W}_1, \ldots, \widehat{W}_{j-1}, \widehat{W}_{j+1}, \ldots, \widehat{W}_t]$ and

$$\mathscr{B}_j = \begin{cases} \left[A_0 \widehat{W}_{-j}, A_1 \widehat{W}_{-j}, \ldots, A_p \widehat{W}_{-j} \right]^{\mathrm{H}}, & \text{if } A_j's \text{ are Hermitian;} \\ \left[A_0 \widehat{W}_{-j}, A_0^{\mathrm{H}} \widehat{W}_{-j}, \ldots, A_p \widehat{W}_{-j}, A_p^{\mathrm{H}} \widehat{W}_{-j} \right]^{\mathrm{H}}, & \text{otherwise.} \end{cases} \tag{18}$$

Fixing $\widehat{W}_{-j}$, we can minimize $f(\tau_n, \widehat{W}) = \sum_{i=0}^{p} \|\mathrm{OffBdiag}_{\tau_n}(\widehat{W}^{\mathrm{H}} A_i \widehat{W})\|_F^2$ by updating $\widehat{W}_j$ as V_j, where the column vectors of V_j are the n_j right singular vectors of $\mathscr{B}_j$ corresponding with the n_j smallest singular values. For $j = 1, 2, \ldots, t$, we

[‡]This indicates that $x_1, \ldots, x_k$ are almost linearly dependent, and it is generally impossible to find a good diagonalizer via those eigenvectors.

update $\widehat{W}_j$ as above; we call it a refinement loop. We can repeat the refinement loop until the diagonalizer is sufficiently good. In our numerical test, three refinement loops are sufficient.

Note that the effectiveness of this refinement procedure is built on the assumption that τ_n obtained in Stage 2 is correct. Without such an assumption, the refinement procedure may make the diagonalizer even worse.

It is also worth mentioning here that the above refinement procedure can be used to update any approximate diagonalizer, not necessarily the one produced by Stage 2. As a matter of fact, the procedure itself can be used to find a diagonalizer; however, without a good initial guess, the convergence can be quite slow.

Remark 5. In Stage 1, assuming each eigenpair can be found in $\mathcal{O}(1)$ steps, then $k = \mathcal{O}(n)$ eigenpairs can be obtained in $\mathcal{O}(pn^3)$ flops since one matrix vector multiplication $M^{-1}Nv$ requires $\mathcal{O}(pn^2)$ flops. Both Stages 2 and 3 require $\mathcal{O}(n^3)$ flops. Hence, the overall computational cost of PEAR is $\mathcal{O}(pn^3)$ flops.

When the matrices A_i's are all real, a real diagonalizer is required. However, the diagonalizer returned by PEAR is, in general, complex. Can we make some simple modifications to PEAR to get a real diagonalizer? The answer is positive. In fact, we can add the following "Stage 2.5" to get a real diagonalizer from PEAR.

4.4 | Stage 2.5

Let $\widehat{W} = [\widehat{W}_1, \ldots, \widehat{W}_t]$ be the approximate diagonalizer at the beginning of Stage 3, where $\widehat{W}_j \in \mathbb{C}^{n\times n_j}$ for $j = 1, \ldots, t$. For each j, denote $\widehat{W}_j = \widehat{W}_{jR} + \imath\widehat{W}_{jI}$, where $\widehat{W}_{jR}, \widehat{W}_{jI} \in \mathbb{R}^{n\times n_j}$ are the real and imaginary parts of $\widehat{W}_j$, respectively. Let the singular value decomposition of $[\widehat{W}_{jR}, \widehat{W}_{jI}]$ be $[\widehat{W}_{jR}, \widehat{W}_{jI}] = U_j\Sigma_j V_j^{\mathrm{T}}$, where $U_j \in \mathbb{R}^{n\times n}, V_j \in \mathbb{R}^{2n_j\times 2n_j}$ are orthogonal and $\Sigma_j \in \mathbb{R}^{n\times 2n_j}$ is diagonal, with its main diagonal nonnegative and in a decreasing order. Then, we update $\widehat{W}_j$ as $\widehat{W}_j = U_j(:, 1 : n_j)$. As a consequence, $\widehat{W} = [\widehat{W}_1, \ldots, \widehat{W}_t]$ will be a real diagonalizer.

The mechanic behind the above procedure is the following critical assumption.

(A) All solutions to the GJBD problem of $\mathcal{A}$ are equivalent, and the intersection of all diagonalizers and $\mathbb{R}^{n\times n}$ is nonempty.

Based on the above assumption, for the diagonalizer $\widehat{W} \in \mathbb{C}^{n\times n}$ at the beginning of Stage 3, there exists a nonsingular τ_n block-diagonal matrix $D = \mathrm{diag}(D_{11}, \ldots, D_{tt})$ such that $\widetilde{W} := \widehat{W}D^{-1}$ is a real diagonalizer. Partition $\widetilde{W}$ as $= [\widetilde{W}_1, \ldots, \widetilde{W}_t]$ with $\widetilde{W}_j \in \mathbb{C}^{n\times n_j}$, and let $D_{jj} = D_{jR} + \imath D_{jI}$, where $D_{jR}, D_{jI} \in \mathbb{R}^{n_j\times n_j}$. Then, by $\widetilde{W}_j = \widehat{W}_j D_{jj}^{-1}$, we have

$$\widetilde{W}_j[D_{jR}, D_{jI}] = [\widehat{W}_{jR}, \widehat{W}_{jI}].$$

Then, it follows that

$$\mathrm{span}([\widehat{W}_{jR}, \widehat{W}_{jI}]) = \mathrm{span}(\widetilde{W}_j[D_{jR}, D_{jI}]) = \mathrm{span}(\widetilde{W}_j),$$

since $[D_{jR}, D_{jI}]$ is of full row rank. Therefore, updating $\widehat{W}_j$ as $U_j(:, 1 : n_j)$, we know that $\widehat{W} = [U_1(:, 1 : n_1), \ldots, U_t(:, 1 : n_t)]$ is a real diagonalizer.

Assumption (A), in general, holds, but not always. For example, consider the GJBD problem of $\{A_0, A_1\}$, where

$$A_0 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -3 \\ -3 & 1 & 1 \end{bmatrix}, \qquad A_1 = \begin{bmatrix} 3 & -1 & 1 \\ -1 & 3 & -3 \\ -3 & 1 & 3 \end{bmatrix}.$$

Then, by calculations, we know that $((1,2), \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix})$ and $((1,1,1), \begin{bmatrix} 1 & 1-\imath & 1+\imath \\ 1 & 1+\imath & 1-\imath \\ 0 & 2 & 2 \end{bmatrix})$ are two inequivalent solutions to the GJBD problem with the diagonalizers in $\mathbb{R}^{n\times n}$ and $\mathbb{C}^{n\times n}$, respectively. How to find a real diagonalizer when assumption (A) does not hold is difficult and needs further investigations. In our numerical tests (Section 5.2), Stage 2.5 works perfectly for finding a real diagonalizer.

5 | NUMERICAL EXAMPLES

In this section, we present several examples to illustrate the performance of PEAR. All the numerical examples were carried out on a quad-core Intel Core i5-6300HQ running at 2.30 GHz with 3.87-GB RAM, using MATLAB R2014a with machine $\epsilon = 2.2 \times 10^{-16}$.

We compare the performance of PEAR (with and without refinement) with the second GJBD algorithm in the work of Cai and Liu,[21] namely, ⋆-commuting–based method with a conservative strategy, SCMC for short, and two algorithms for the JBD problem, namely, JBD-LM[40] and JBD-NCG.[4] For PEAR, three refinement loops are used to improve the quality of the diagonalizer. For SCMC, the tolerance is set as $3n^2 10^{-\mathrm{SNR}/20}$, where SNR is the signal-to-noise ratio defined below. For the JBD-LM method, the stopping criteria are $\|W_{k+1} - W_k\|_F < 10^{-12}$, or $|\frac{\phi_k - \phi_{k+1}}{\phi_k}| < 10^{-8}$ for successive five steps, or the maximum number of iterations, which is set as 200, exceeded. For the JBD-NCG method, the stopping criteria are $|\phi_k - \phi_{k+1}| < 10^{-8}$, or $|\frac{\phi_k - \phi_{k+1}}{\phi_k}| < 10^{-8}$ for successive five steps, or the maximum number of iterations, which is set as 2,000, exceeded. Here, W_k and ϕ_k are the W matrix and the value of the cost function in the kth step, respectively. In JBD-LM and JBD-NCG algorithms, 20 initial values (19 random initial values and an EVD-based initial value[4]) are used to iterate 20 steps first, and then, the iteration that produces the smallest value of the cost function proceeds until one of the stopping criteria is satisfied.

5.1 | Random data

Let $\tau_n = (n_1, n_2, \ldots, n_t)$ be a partition of n, we will generate the matrix set $\mathcal{A} = \{A_i\}_{i=0}^{p}$ by the following model:

$$A_i = V^{\mathrm{H}} D_i V, \quad i = 0, 1, \ldots, p, \tag{19}$$

where V and D_i are, respectively, the mixing matrix and the approximate τ_n block-diagonal matrices. The elements in V and $\mathrm{Bdiag}_{\tau_n}(D_i)$ are all complex numbers whose real and imaginary parts are drawn from a standard normal distribution, whereas the elements in $\mathrm{OffBdiag}_{\tau_n}(D_i)$ are all complex numbers whose real and imaginary parts are drawn from a normal distribution with mean zero and variance σ^2. The SNR is defined as $\mathrm{SNR} = 10\log(1/\sigma^2)$.

For model (19), we define the following performance index to measure the quality of the computed diagonalizer W, which is used in the works of Cai and Liu[21] and Cai et al.[22]:

$$\mathrm{PI}(V^{-1}, W) = \min_{\pi} \max_{1 \le i \le t} \mathrm{subspace}(V_i, W_{\pi(i)}),$$

where $V^{-1} = [V_1, V_2, \ldots, V_t]$, $W = [W_1, W_2, \ldots, W_t]$, $V_i, W_{\pi(i)} \in \mathbb{C}^{n \times n_i}$ for $i = 1, 2, \ldots, t$, the vector $(\pi(1), \pi(2), \ldots, \pi(t))$ is a permutation of $\{1, 2, \ldots, t\}$ satisfying $(n_{\pi(1)}, n_{\pi(2)}, \ldots, n_{\pi(t)}) = \tau_n$, and the expression $\mathrm{subspace}(E, F)$ denotes the angle between two subspaces specified by the columns of E and F, which can be computed by the MATLAB function "subspace."

In what follows, we generate the matrix set by model (19) with the following parameters:

P1. $n = 9, \tau_n = (3, 3, 3), p + 1 = 25$;
P2. $n = 9, \tau_n = (2, 3, 4), p + 1 = 25$;
P3. $n = 9, \tau_n = (2, 3, 4), p + 1 = 20, 40, \ldots, 200$, SNR = 80;
P4. $n = 9m, \tau_n = (2m, 3m, 4m)$ for $m = 1, 2, \ldots, 6, p + 1 = 10$, SNR = 80.

Experiment 1. For different SNRs, we generate the data with parameters P1 and P2, respectively. Then, for each matrix set generated by those parameters, we perform PEAR for 1,000 independent runs. It is known that when the SNR is small, PEAR may fail, specifically, the block-diagonal structure in Stage 2 of PEAR is fuzzy, and the computed $\hat{\tau}_n$ may not be consistent with the true τ_n, namely, there is no $(0, 1)$ matrix N such that $\tau_n = \hat{\tau}_n N$. In Table 2, we list the percentages of

TABLE 2 The percentages of successful runs of the partial eigenvector approach with refinement over 1,000 independent runs

SNR	30	40	50	60	70	80	90	100
P1.	77.4%	97.5%	99.7%	100%	100%	100%	100%	100%
P2.	75.3%	97.1%	99.6%	100%	100%	100%	100%	100%

Note. SNR = signal-to-noise ratio.

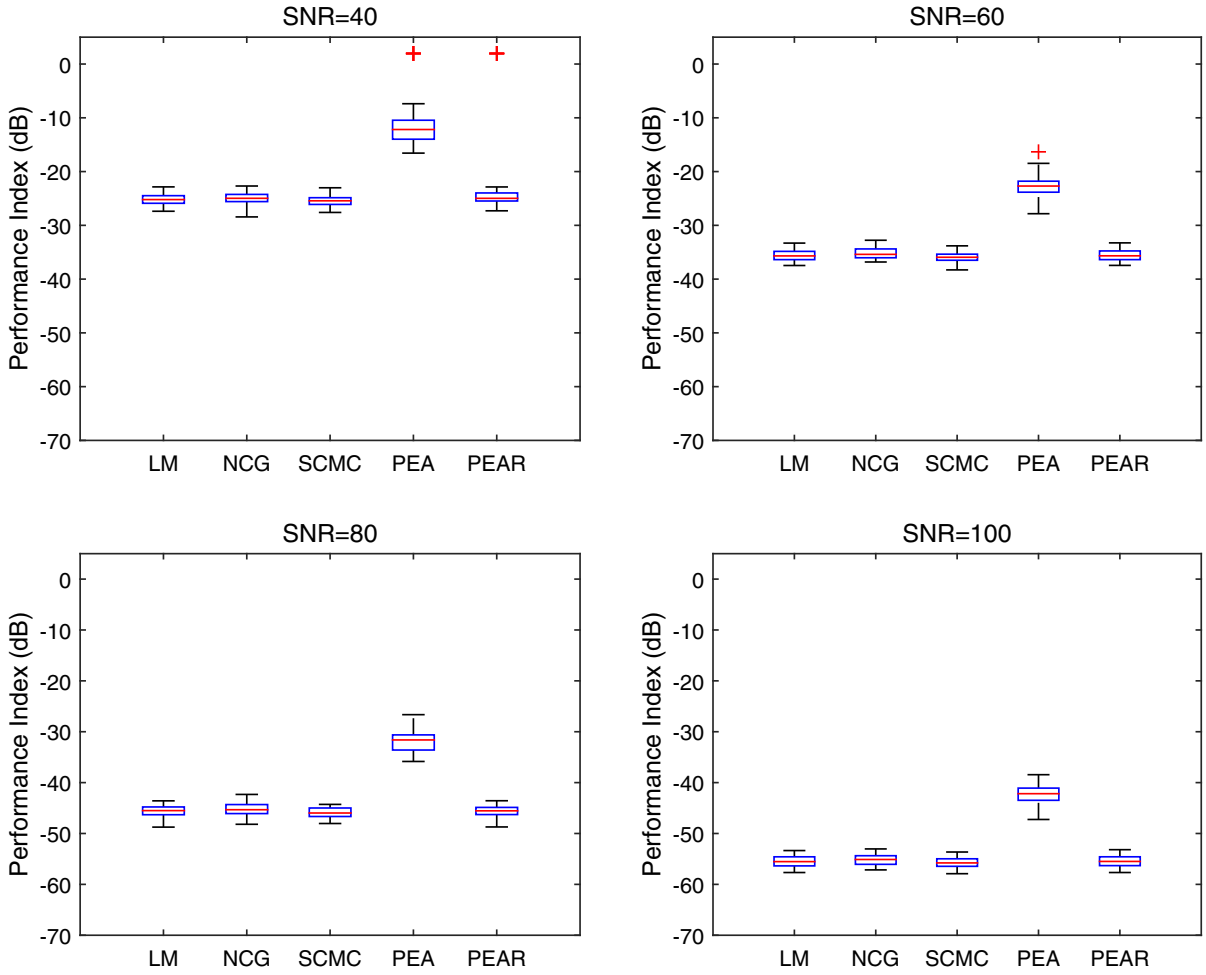

FIGURE 1 Performance indices of five methods with different signal-to-noise ratios (SNRs) for P1

successful runs of PEAR. From the Table, we can see that the smaller the SNR is, the more likely PEAR may fail. In our tests, when SNR is no less than 60, PEAR did not fail.

Experiment 2. For SNR = 40, 60, 80, 100, we generate the data with parameters P1 and P2, respectively. For each matrix set generated by those parameters, we perform JBD-LM, JBD-NCG, SCMC, PEA (PEAR without refinement), and PEAR for 50 independent runs and then compare their performance indices. The box plot (generated by the MATLAB function "boxplot") of the results for P1 is displayed in Figure 1, and the results for P2 are similar to those for P1.

We can see from Figure 1 that, when the SNR is equal to 40, 60, 80, or 100, the performance indices produced by JBD-LM, JBD-NCG, SCMC, and PEAR are almost the same; the performance indices produced by PEA are larger than those of the other four methods, which indicates that the diagonalizers produced by the first two stages of PEAR indeed suffer from low quality, and the refinement stage of PEAR is effective. For all methods, the performance indices decrease as the SNR increases.

Experiment 3. We generate the data with the parameters in P3 and P4, respectively. Fifty independent trials are performed for each matrix set. The average performance index and the average CPU time of SCMC and PEAR are displayed in Figures 2 (for P3) and Figure 3 (for P4). From Figure 2, we can see that as the number of matrices increases, the performance indices of two methods decrease, and the CPU times of the two methods increase almost linearly. From Figure 3, we can see that as matrix size increases, the performance indices of both methods increase; the performance index for SCMC increases slower, the CPU time of SCMC increases dramatically, and the CPU time of PEAR increases much slower.

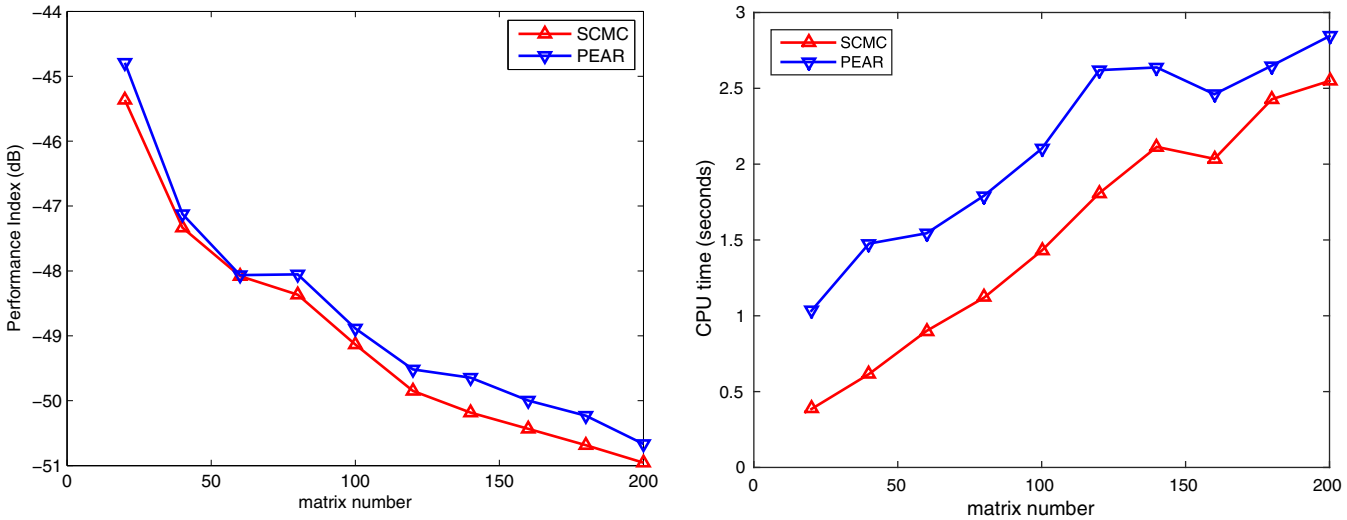

FIGURE 2 (Left) Performance indices of two methods with different matrix numbers; (right) CPU times of two methods with different matrix numbers

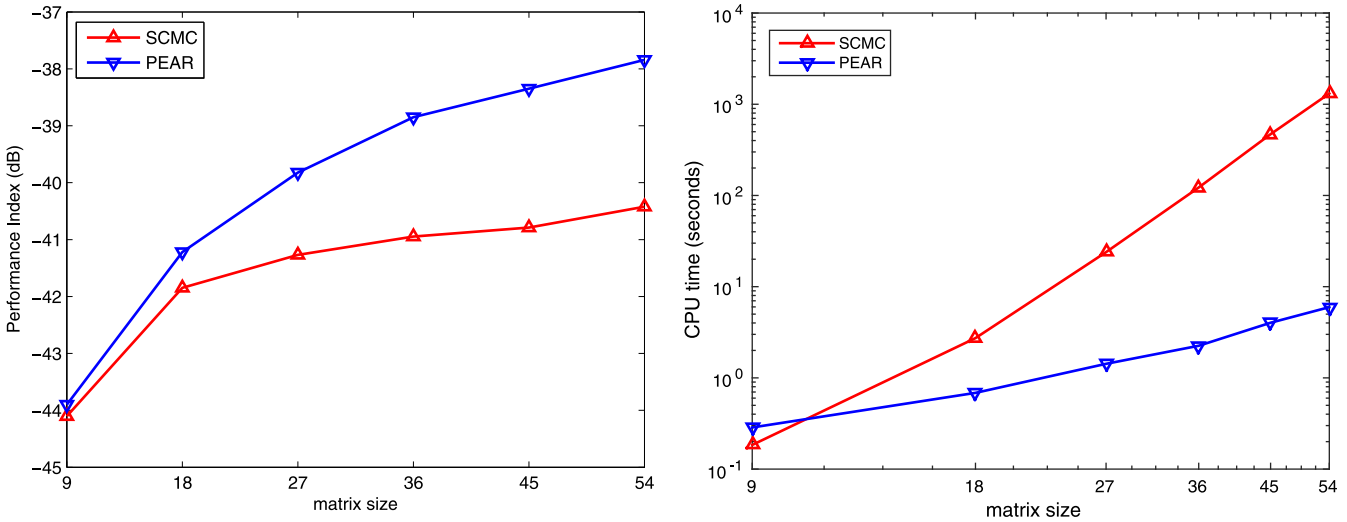

FIGURE 3 (Left) Performance indices of two methods with different matrix sizes; (right) CPU times of two methods with different matrix sizes

5.2 | Separation of convolutive mixtures of source

We consider example 4.2 in the work of Cai and Liu,[21] where a real diagonalizer is required. All settings are kept the same. In Figure 4, for different SNRs, we plot the correlations between the source signals and the extracted signals obtained from computed solutions by JBD-LM, JBD-NCG, SCMC, and PEAR, respectively. All displayed results have been averaged over 50 independent trials.

We can see from Figure 4 that when the SNR is larger than 60, the recovered signals obtained from all four methods are all good approximations of the source signals; when the SNR is less than 60, SCMC is the best, and PEAR is the second best. The reason why SCMC is better than PEAR in this example is that PEAR fails to find the correct partition in Stage 2 when the SNR is small; meanwhile, with a large tolerance for SCMC, SCMC is able to find a consistent partition with the correct one. In Stage 2 of PEAR, if we use some normalized Laplacian to find the partition, the numerical results of PEAR can be improved.

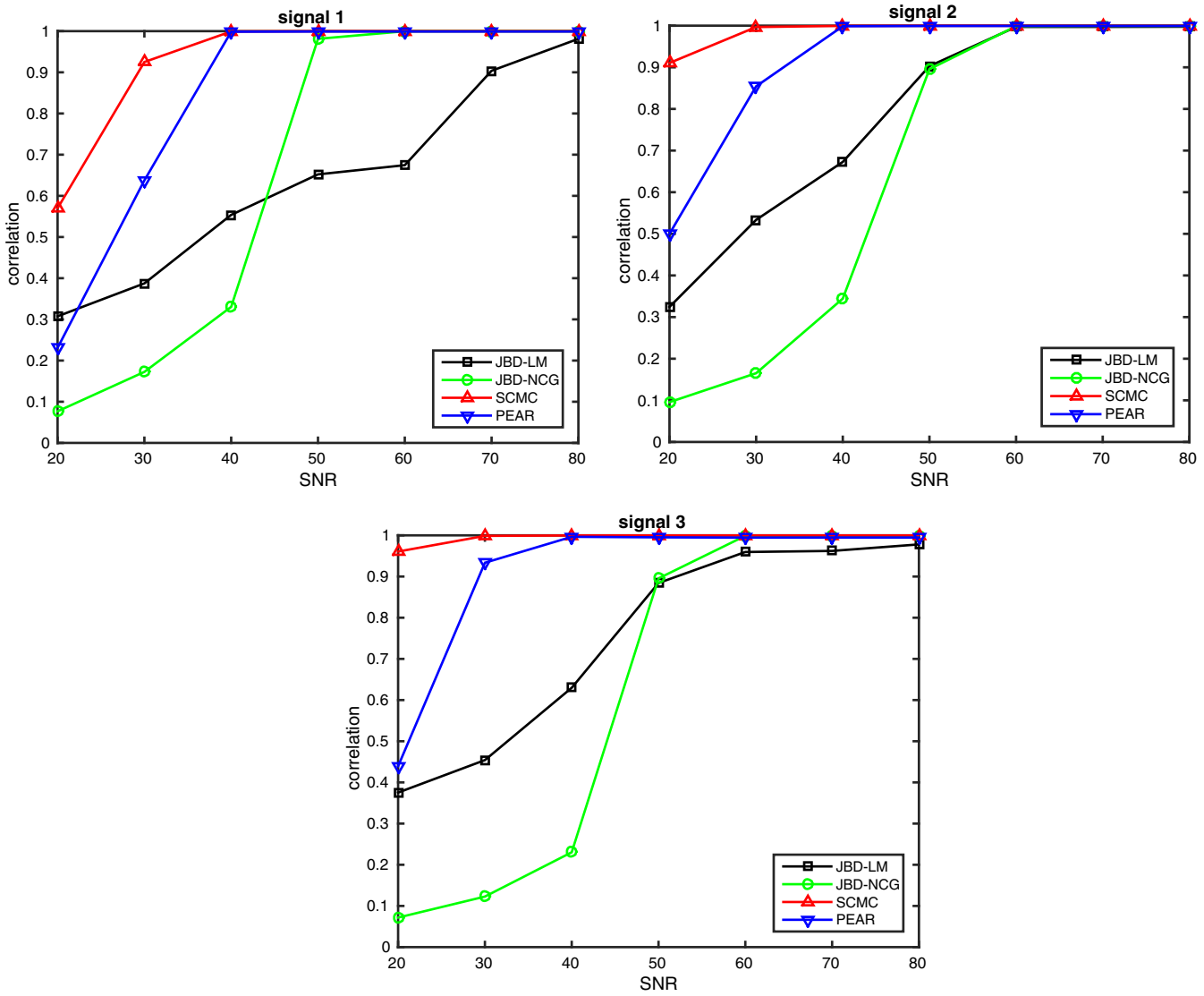

FIGURE 4 Correlation between recovered signals and source signals

6 | CONCLUSION

In this paper, we have shown how the GJBD problem of a matrix set is related to a matrix PEP. Theoretically, under mild conditions, we show that (a) the GJBD problem of $\{A_i\}_{i=0}^{p}$ can be solved by n linearly independent eigenvectors of the matrix polynomial $P(\lambda) = \sum_{i=0}^{p} \lambda^i A_i$, (b) all solutions to the GJBD problem are equivalent, and (c) a suboptimal solution of the approximate GJBD problem can also be given by n linearly independent eigenvectors. Algorithmically, we proposed a three-stage method, that is, PEAR, to solve the GJBD problem. Numerical experiments show the merits of PEAR.

Finally, it is worth mentioning here that the GJBD problem discussed in the paper, compared with the BTD of tensors, is limited in several aspects[21]: The matrices are square rather than general nonsquare ones, the matrices are factorized via a congruence transformation rather than a general one, etc. Is it possible to use the matrix polynomial approach in this paper to compute a blind BTD (BTD without knowing the number of terms and the size of each term) of tensors? We will try to answer this question in our further work.

ACKNOWLEDGEMENTS

We are grateful for the reviewers' comments, which helped improve this paper. There are no conflicts of interest to this work. This research was supported by National Natural Science Foundation of China under grants 11671023, 11421101, and 11301013.

ORCID

Yunfeng Cai https://orcid.org/0000-0003-2387-191X
Guanghui Cheng https://orcid.org/0000-0002-7325-8892

REFERENCES

1. De Lathauwer L. Decompositions of a higher-order tensor in block terms–part I: lemmas for partitioned matrices. SIAM J Matrix Anal Appl. 2008;30(3):1022–1032.
2. De Lathauwer L. Decompositions of a higher-order tensor in block terms–part II: definitions and uniqueness. SIAM J Matrix Anal Appl. 2008;30(3):1033–1066.
3. De Lathauwer L, Nion D. Decompositions of a higher-order tensor in block terms–part III: alternating least squares algorithms. SIAM J Matrix Anal Appl. 2008;30(3):1067–1083.
4. Nion D. A tensor framework for nonunitary joint block diagonalization. IEEE Trans Signal Process. 2011;59(10):4585–4594.
5. Cardoso J-F. Multidimensional independent component analysis. Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing; 1998 May 12–15; Seattle, WA. Piscataway, NJ: IEEE; 1998; p. 1941–1944.
6. De Lathauwer L, De Moor B, Vandewalle J. Fetal electrocardiogram extraction by blind source subspace separation. IEEE Trans Biomed Eng. 2000;47(5):567–572.
7. Theis FJ. Blind signal separation into groups of dependent signals using joint block diagonalization. Proceedings of the 2005 IEEE International Symposium on Circuits and Systems; 2005 May 23–26; Kobe, Japan. Piscataway, NJ: IEEE; 2005; p. 5878–5881.
8. Theis FJ. Towards a general independent subspace analysis. Proceedings of the 19th International Conference on Neural Information Processing Systems; 2006 Dec 4–7; Vancouver, Canada. Cambridge, MA: MIT Press; 2006; p. 1361–1368.
9. Bai Y, de Klerk E, Pasechnik D, Sotirov R. Exploiting group symmetry in truss topology optimization. Optim Eng. 2009;10(3):331–349.
10. de Klerk E, Pasechnik DV, Schrijver A. Reduction of symmetric semidefinite programs using the regular $*$-representation. Math Program. 2007;109(2–3):613–624.
11. de Klerk E, Sotirov R. Exploiting group symmetry in semidefinite programming relaxations of the quadratic assignment problem. Math Program. 2010;122:225.
12. Gatermann K, Parrilo PA. Symmetry groups, semidefinite programs, and sums of squares. J Pure Appl Algebra. 2004;192(1–3):95–128.
13. Choi S, Cichocki A, Park H-M, Lee S-Y. Blind source separation and independent component analysis: a review. Neural Inf Process Lett Rev. 2005;6(1):1–57.
14. Comon P, Jutten C. Handbook of blind source separation: Independent component analysis and applications. Cambridge, MA: Academic Press; 2010.
15. Hyvärinen A, Karhunen J, Oja E. Independent component analysis, vol. 46. Hoboken, NJ: John Wiley & Sons; 2004.
16. Cardoso J-F, Souloumiac A. Blind beamforming for non-Gaussian signals. IEE Proc F Radar Signal Process. 1993;140(6):362–370.
17. Moreau E. A generalization of joint-diagonalization criteria for source separation. IEEE Trans Signal Process. 2001;49(3):530–541.
18. Belouchrani A, Abed-Meraim K, Cardoso J-F, Moulines E. A blind source separation technique using second-order statistics. IEEE Trans Signal Process. 1997;45(2):434–444.
19. Theis FJ. A new concept for separability problems in blind source separation. Neural Comput. 2004;16(9):1827–1850.
20. Yeredor A. Blind source separation via the second characteristic function. Signal Process. 2000;80(5):897–902.
21. Cai Y, Liu C. An algebraic approach to nonorthogonal general joint block diagonalization. SIAM J Matrix Anal Appl. 2017;38(1):50–71.
22. Cai Y, Shi D, Xu S. A matrix polynomial spectral approach for general joint block diagonalization. SIAM J Matrix Anal Appl. 2015;36(2):839–863.
23. Chabriel G, Kleinsteuber M, Moreau E, Shen H, Tichavsky P, Yeredor A. Joint matrices decompositions and blind source separation: a survey of methods, identification, and applications. IEEE Signal Process Mag. 2014;31(3):34–43.
24. De Lathauwer L. A survey of tensor methods. Proceedings of the 2009 IEEE International Symposium on Circuits and Systems; 2009 May 24–27; Taipei, Taiwan. Piscataway, NJ: IEEE; 2009; p. 2773–2776.
25. Tichavsky P, Phan A-H, Cichocki A. Non-orthogonal tensor diagonalization. Signal Process. 2017;138:313–320.
26. Abed-Meraim K, Belouchrani A. Algorithms for joint block diagonalization. Paper presented at: 12th European Signal Processing Conference 2004; 2004 Sep 6–10; Vienna, Austria. Piscataway, NJ: IEEE; 2004; p. 209–212.
27. de Klerk E, Dobre C, Pasechnik DV. Numerical block diagonalization of matrix $*$-algebras with application to semidefinite programming. Math Program. 2011;129(1):91–111.
28. Maehara T, Murota K. A numerical algorithm for block-diagonal decomposition of matrix $*$-algebras with general irreducible components. Japan J Indust Appl Math. 2010;27(2):263–293.
29. Maehara T, Murota K. Algorithm for error-controlled simultaneous block-diagonalization of matrices. SIAM J Matrix Anal Appl. 2011;32(2):605–620.
30. Murota K, Kanno Y, Kojima M, Kojima S. A numerical algorithm for block-diagonal decomposition of matrix $*$-algebras with application to semidefinite programming. Japan J Indust Appl Math. 2010;27(1):125–160.
31. Higham NJ, Mackey DS, Mackey N, Tisseur F. Symmetric linearizations for matrix polynomials. SIAM J Matrix Anal Appl. 2006;29(1):143–159.

32. Mackey DS, Mackey N, Mehl C, Mehrmann V. Vector spaces of linearizations for matrix polynomials. SIAM J Matrix Anal Appl. 2006;28(4):971–1004.
33. Mehrmann V, Watkins D. Polynomial eigenvalue problems with Hamiltonian structure. Electron Trans Numer Anal. 2002;13:106–118.
34. von Luxburg U. A tutorial on spectral clustering. Stat Comput. 2007;17(4):395–416.
35. Nakatsukasa Y, Tisseur F. Eigenvector error bound and perturbation for polynomial and rational eigenvalue problems. Tokyo, Japan: Department of Mathematical Informatics, The University of Tokyo; 2016. Mathematical Engineering Technical Reports METR 2016-04.
36. Cai Y, Li R. Perturbation analysis for matrix joint block diagonalization. Numer Anal. 2017. arXiv:1703.00591
37. Shi D, Cai Y, Xu S. Some perturbation results for a normalized non-orthogonal joint diagonalization problem. Linear Algebra Appl. 2015;484:457–476.
38. Afsari B. Sensitivity analysis for the problem of matrix joint diagonalization. SIAM J Matrix Anal Appl. 2008;30(3):1148–1171.
39. Pereira E. On solvents of matrix polynomials. Appl Numer Math. 2003;47(2):197–208.
40. Cherrak O, Ghennioui H, Abarkan E-H, Thirion-Moreau N. Non-unitary joint block diagonalization of matrices using a Levenberg-Marquardt algorithm. Proceedings of the 21st European Signal Processing Conference (EUSIPCO 2013); 2013 Sep 9–13; Marrakech, Morocco. Piscataway, NJ: IEEE; 2013; p. 1–5.

How to cite this article: Cai Y, Cheng G, Shi D. Solving the general joint block diagonalization problem via linearly independent eigenvectors of a matrix polynomial. *Numer Linear Algebra Appl*. 2019;e2238. https://doi.org/10.1002/nla.2238