Linear Algebra and its Applications 701 (2024) 112–137

Contents lists available at ScienceDirect

Linear Algebra and its Applications

journal homepage: www.elsevier.com/locate/laa

Complete equitable decompositions

Joseph Drapeau [a], Joseph Henderson [a], Peter Seely [a], Dallas Smith [b], Benjamin Webb [a,*]

[a] *Department of Mathematics, Brigham Young University, Provo, UT 84602, USA*
[b] *Department of Mathematics, Utah Valley University, Orem, UT 84059, USA*

A R T I C L E   I N F O

Article history:
Received 12 January 2024
Received in revised form 7 August 2024
Accepted 7 August 2024
Available online 14 August 2024
Submitted by R.A. Brualdi

MSC:
05C50
05C69
15A18
65T50

Keywords:
Equitable partition
Local equitable partition
Global eigenvectors
Local eigenvectors
Spectral graph theory
LEParD algorithm

A B S T R A C T

A classical result in spectral graph theory states that if a graph G has an equitable partition π then the eigenvalues of the divisor graph G_π are a subset of its eigenvalues, i.e. $\sigma(G_\pi) \subseteq \sigma(G)$. A natural question is whether it is possible to recover the remaining eigenvalues $\sigma(G) - \sigma(G_\pi)$ in a similar manner. Here we show that any weighted undirected graph with nontrivial equitable partition can be decomposed into a number of subgraphs whose collective spectra contain these remaining eigenvalues. Using this decomposition, which we refer to as a complete equitable decomposition, we introduce an algorithm for finding the eigenvalues of an undirected graph (symmetric matrix) with a nontrivial equitable partition. Under mild assumptions on this equitable partition we show that we can find eigenvalues of such a graph faster using this method when compared to standard methods. This is potentially useful as many real-world data sets are quite large and have a nontrivial equitable partition.

* Corresponding author.
E-mail addresses: drapeau@mathematics.byu.edu (J. Drapeau), josephforcollege1@gmail.com (J. Henderson), peter.a.seely@gmail.com (P. Seely), dallas.smith@uvu.edu (D. Smith), bwebb@mathematics.byu.edu (B. Webb).

https://doi.org/10.1016/j.laa.2024.08.008
0024-3795/

1. Introduction

Spectral graph theory studies the relationship between the graph's *spectrum*, i.e., the set of eigenvalues of an associated matrix, and the structure of the graph. This relationship between spectrum and structure is important in many real and theoretical applications. This includes understanding the interplay of structure and function in real world networks [5,15,20,25], the performance of machine learning algorithms [6,16,17,19], and the advantages of different data structures in computer science [4,11], etc.

Here the structures we are interested in are equitable partitions. Equitable partitions were originally studied due to the spectral properties they preserved [2,7,12]. More recently, equitable partitions associated with graph symmetries have gained attention due to their ubiquity in real-world networks [18]. In theoretical applications these symmetries have been used to decompose graphs [3,9,10,14] and to study the formation of synchronizing clusters and equitable partitions in dynamical network models [13,21,23,26,27].

An equitable partition $\pi = \{V_1, \ldots, V_k\}$ of a graph G is a vertex partition that, roughly speaking, partitions the vertices of the graph such that every vertex in V_i has the same number of neighbors in any V_j irrespective of which vertex is considered. This structure can be summarized by another smaller graph called the *divisor graph* G_π of G (see Section 2). In this way the divisor graph G_π gives a global summary of the graph G relative to the equitable partition π.

A well-known property of a divisor graph is that its spectrum is a subset of the graph's original spectrum, i.e., $\sigma(G_\pi) \subseteq \sigma(G)$ [2]. As one can think of $\sigma(G_\pi)$ as the *global eigenvalues* of G and it is a natural question as to whether it is possible to recover the remaining or *local eigenvalues* of G, i.e. the eigenvalues $\sigma_\pi^\ell(G) = \sigma(G) - \sigma(G_\pi)$, and whether this can be done in a similar manner using divisor graphs.

Here we show that these local eigenvalues are, in fact, eigenvalues of a collection of induced subgraphs $G_i \subseteq G$ for $i = 1, \ldots, r$ where each G_i has an equitable partition $\bar{\pi}_i \subseteq \pi$. We refer to each $\bar{\pi}_i$ as a *local equitable partition* on the local subgraph G_i and prove that the local eigenvalues of G form the set $\sigma_\pi^\ell(G) = \sigma_{\bar{\pi}_1}^\ell(G_1) \cup \cdots \cup \sigma_{\bar{\pi}_r}^\ell(G_r)$. This allows us to write the eigenvalues of G as the disjoint union

$$\sigma(G) = \sigma(G_\pi) \cup [\sigma_{\bar{\pi}_1}^\ell(G_1) \cup \cdots \cup \sigma_{\bar{\pi}_r}^\ell(G_r)],$$

which we refer to as a *complete equitable decomposition* of G with respect to π (see Theorem 2.2). A complete equitable decomposition is then a decomposition of the eigenvalues of G into its global eigenvalues and local eigenvalues of its local subgraphs (see Example 2.3).

Since a complete equitable decomposition results in a collection of smaller graphs with the same collective spectrum it is therefore possible, at least in principle, to find the eigenvalues of a graph more efficiently using this decomposition when compared to standard methods. Here we construct an algorithm which computes the eigenvalues of a graph (matrix) based on the concept of a complete equitable decomposition. We show that in the worst case, in which a graph has only the trivial equitable partition, that this algorithm has the same computational complexity as the standard algorithm. If, however, the graph has a sufficiently nontrivial equitable partition, our algorithm can compute the eigenvalues of the graph much faster (see Theorem 5.1). This we show using a family of layered graphs (see Example 5.1 in Section 5.3).

The paper structured as follows. In Section 2 we review the basic concepts and classical results related to equitable partitions. We then introduce the notion of a local equitable partition and state our main result (Theorem 2.2). In Section 3 we begin our proof of this result by describing the local and global eigenvector structure of a graph with an equitable partition. In Section 4 we complete our proof. In Section 5 we introduce our algorithm for finding the eigenvalues of a graph with respect to its coarsest equitable partition, which we refer to as the *LEParD algorithm* (Local Equitable Partition Decomposition algorithm). In Section 6 we conclude with a number of directions this research could be taken along with some open questions.

2. Equitable partitions

In this section we define the notion of an equitable partition of a graph and give some of its more well-known properties. For generality, we define a *graph* to be a weighted directed graph $G = (V, E, \omega)$ with vertex set $V = \{1, 2, \dots, n\}$, and edge set E, with weight function $\omega : E \to \mathbb{C}$. A *directed edge* from vertex i to vertex j is denoted e_{ij} where the collection of all edges, possibly including loops, is the *edge set* E. The *weight* of the edge $e_{ij} \in E$ is given by $\omega(e_{ij}) \in \mathbb{C}$.

This framework includes both unweighted and undirected graphs where an *unweighted graph* $G = (V, E, \omega)$ has the weight function

$$\omega(e_{ij}) = \begin{cases} 1 & \text{if } e_{ij} \in E \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

An *undirected graph* $G = (V, E, \omega)$ has the property that if the edge $e_{ij} \in E$ then $e_{ji} \in E$ where $\omega(e_{ij}) = \omega(e_{ji})$ and the pair of edges is thought of as a single edge between vertex i and j.

The primary way to encode the structure of a graph G including its weights is with a *weighted adjacency matrix* $A = A(G)$ where $A = [a_{ij}] \in \mathbb{C}^{n \times n}$ whose entries are given by

$$a_{ij} = \begin{cases} \omega(e_{ij}) \neq 0 & \text{if } e_{ij} \in E \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The eigenvalues $\sigma(G)$ of the graph G are the eigenvalues $\sigma(A)$ of its adjacency matrix A. We note that a graph is unweighted if its adjacency matrix $A \in \{0, 1\}^{n \times n}$ and is undirected if A is symmetric.

The focus of this paper is on equitable partitions and extending the original spectral theory associated with equitable partitions. An equitable partition is defined as follows.

Definition 2.1 *(Equitable partition).* An *equitable partition* of a graph $G = (V, E, \omega)$ is a partition $\pi = \{V_1, \ldots, V_k\}$ of V with the property that for all $s, t \in \{1, 2, \ldots, k\}$ the sum

$$\sum_{j \in V_t} a_{ij} = d_{st}$$

is constant for any $i \in V_s$. The matrix $D = [d_{st}]$, which we write as $D = A_\pi \in \mathbb{R}^{k\times k}$, is the *divisor matrix* of A associated with π. The graph G_π with adjacency matrix A_π is the *divisor graph* of G.

One can think of the divisor graph G_π as a global summary of how the vertices in the elements of π are connected to each other. The graph G_π is effectively a coarse-graining of the graph G into a directed graph in which each element of $\pi = \{V_1, V_2, \ldots, V_k\}$ is represented by a single vertex. The directed edge weight of $\omega(e_{st}) = [A_\pi]_{st}$ is the sum of weighted edges from any vertex $i \in V_s$ to vertices $j \in V_t$. If G is a simple graph, then the weight $\omega(e_{st})$ in G_π is the number of edges between any vertex in V_s and any vertex in V_t as the matrix $A \in \{0, 1\}^{n\times n}$ (see Example 2.1).

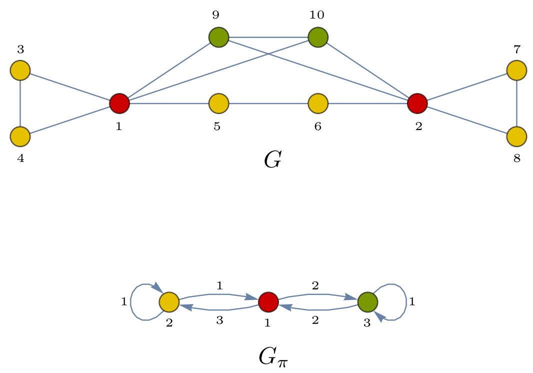
Given a vertex partition $\pi = \{V_1, \ldots, V_k\}$ of G, we can use π to partition the corresponding adjacency matrix into a block matrix of the form

$$A = \begin{bmatrix} A_{11} & \ldots & A_{1k} \\ \vdots & \ddots & \vdots \\ A_{k1} & \ldots & A_{kk} \end{bmatrix} \tag{3}$$

where A_{ij} is the $|V_i| \times |V_j|$ submatrix of A whose rows are indexed by the vertices in V_i and the columns are indexed by the vertices in V_j. For an equitable partition π we can always relabel the vertices of G such that vertices in the same element of π are labeled consecutively and vertices in V_i precede vertices in V_j for $i < j$. For simplicity, in what follows we will assume the adjacency matrix $A = A(G)$ is partitioned as in Equation (3).

For an equitable partition $\pi = \{V_1, \ldots, V_k\}$, the partitioned adjacency matrix A in Equation (3) has the property that each sub-matrix A_{st} has constant row sums. The reason is that the entries of A_{st} will be the entries of A summed in Definition 2.1 for a given V_s and V_t. Since this sum is constant for any given row i, the row sums of A_{st} are constant. This constant row sum for the submatrix A_{st} is the st-entry in the divisor matrix $D = A_\pi$.

Example 2.1. Consider the simple graph G shown in Fig. 1 (top left) and its adjacency matrix A (top right). This graph has the equitable partition $\pi = \{V_1, V_2, V_3\}$ where $V_1 = \{1, 2\}$, $V_2 = \{3, 4, 5, 6, 7, 8\}$, and $V_3 = \{9, 10\}$ represent the red, yellow, and green



$$A = \left[\begin{array}{cc|cccccc|cc} 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right]$$

$$A_\pi = \begin{bmatrix} 0 & 3 & 2 \\ 1 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix}$$

Fig. 1. Top Left: A simple graph $G = (V, E, \omega)$ with equitable partition $\pi = \{V_1, V_2, V_3\}$ is shown where $V_1 = \{1, 2\}$, $V_2 = \{3, 4, 5, 6, 7, 8\}$, and $V_3 = \{9, 10\}$ are indicated by the red, yellow, and green vertices, respectively. Bottom Left: The divisor graph G_π is shown with vertices 1 (red), 2 (yellow), and 3 (green) corresponding to the elements V_1, V_2, and V_3 of the equitable partition π, respectively. Edge weights of G_π are shown in black. Top Right: The adjacency matrix $A = A(G)$ is shown, which is partitioned with respect to π and has constant row sums in each block. Bottom Right: The divisor matrix $A_\pi = A(G_\pi)$ is shown. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

vertices, respectively. The divisor graph G_π and divisor matrix A_π are shown in Fig. 1 (bottom left and bottom right, respectively). Note each entry of A_π corresponds to the constant rows sums of each submatrix A_{st} in A.

A classical result of spectral graph theory is that the eigenvalues of G_π are a subset of the eigenvalues of G. The proof of this result can be found in [2]. (A generalization of this theorem can be found in Section 3, see Theorem 3.1.)

Theorem 2.1 *(Spectra of a divisor matrix). If π is an equitable partition of a graph G then $\sigma(G_\pi) \subseteq \sigma(G)$.*

The graph G in Example 2.1 has the eigenvalues

$$\sigma(G) = \{3.193, -2.193, 2.115, -1.861, -1, -1, -1, 1, 1, -0.254\}. \tag{4}$$

Its divisor graph has the eigenvalues

$$\sigma(G_\pi) = \{3.193, -2.193, 1\} \tag{5}$$

so that $\sigma(G_\pi) \subset \sigma(G)$ (cf. Fig. 1).

Theorem 2.1 allows us to state the following definition.

Definition 2.2 *(Global and local eigenvalues).* If π is an equitable partition of a graph G, then the eigenvalues of its divisor graph $\sigma(G_\pi)$ are the *global eigenvalues* of G associated with the equitable partition π. The other eigenvalues, denoted by $\sigma^{\ell}_{\pi}(G) = \sigma(G) - \sigma(G_\pi)$, are the *local eigenvalues* of G associated with the equitable partition π.

We note that in this definition we are considering the set of eigenvalues as a multiset. A consequence of this is that if a particular eigenvalue has an algebraic multiplicity greater than one, then it could be both a global and a local eigenvalue. For example, the eigenvalue $\lambda = 1$ from Example 2.1 is both a global and a local eigenvalue of the graph in Fig. 1 with respect to the equitable partition π (cf. Equations (4) and (5)). In particular, the global eigenvalues in this example are $\sigma(G_\pi) = \{3.193, -2.193, 1\}$ and the local eigenvalues are

$$\sigma^{\ell}_{\pi}(G) = \sigma(G) - \sigma(G_\pi) = \{2.115, -1.861, -1, -1, -1, 1, -0.254\}.$$

Given that the global eigenvalues are the eigenvalues of the divisor graph, a natural question is whether it is possible to recover the local eigenvalues in a similar manner. Here we show that the local eigenvalues of G are the eigenvalues of a collection of induced subgraphs G_i of G, where each has an equitable partition $\bar{\pi}_i$ associated with the original equitable partition π. In other words, the local eigenvalues of G are related to specific substructures of the graph G, which is the main result of this paper (see Theorem 2.2 and Example 2.3).

In order to identify these induced subgraphs we introduce the following notion of a local equitable partition.

Definition 2.3 *(Local equitable partition).* Given a graph $G = (V, E, \omega)$ with adjacency matrix A and an equitable partition $\pi = \{V_1, \dots, V_k\}$, we say that two partition elements V_s and V_t such that $s \neq t$ are *consistently connected* if $A_{st} = \alpha_{st} J_{|V_s|,|V_t|}$ for some $\alpha_{st} \in \mathbb{C}$. A subset $\bar{\pi} \subseteq \pi$ is a *local equitable partition* with respect to π if every $V_s \in \bar{\pi}$ is consistently connected to every $V_t \in \pi - \bar{\pi}$.

Example 2.2. Consider the subsets $\bar{\pi}_1 = \{V_2, V_3\}$ and $\bar{\pi}_2 = \{V_1, V_2\}$ of the equitable partition $\pi = \{V_1, V_2, V_3\}$ of the graph G in Fig. 1. In this case the subset $\bar{\pi}_1$ is not a local equitable partition with respect to π as the submatrix

$$A_{21} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}^T$$

does not have constant entries, i.e. $A_{12} \neq \alpha J_{2,6}$ for any $\alpha \in \mathbb{C}$. However, $\bar{\pi}_2$ is a local equitable partition with respect to π as

$$A_{13} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \text{and} \quad A_{23} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

have constant entries respectively.

Suppose that $G = (V, E, \omega)$ has the set of local equitable partitions $\bar{\pi}_1, \bar{\pi}_2, \dots, \bar{\pi}_r$ with respect to the same equitable partition π. If $\bar{\pi}_i = \{\bar{V}_1, \bar{V}_2, \dots, \bar{V}_\ell\}$ we let $V_{\bar{\pi}_i} = \cup_{j=1}^{\ell} \bar{V}_j$, i.e. the set of all vertices in $\bar{\pi}_i$. Using this convention, for each $i = 1, 2, \dots, r$ we let G_i be the induced subgraph of G with vertex set $V_{\bar{\pi}_i}$.

Definition 2.4 *(Complete set of local equitable partitions).* Let $G = (V, E, \omega)$ be a graph with equitable partition π. We say the set of local equitable partitions $\{\bar{\pi}_i\}_{i=1}^{r}$ is *complete* if it is both disjoint and $\pi - \cup_{i=1}^{r} \bar{\pi}_i$ is a set of singleton partition elements of π or the empty set.

Given an equitable partition $\pi = \{V_1, V_2, \dots, V_k\}$ there is a always at least one complete set of local equitable partitions associated with it. The reason is that we can always choose this complete set to be the single local equitable partition $\bar{\pi}_1 \subseteq \pi$ consisting of the nonsingle vertex elements of π. If there are more available local equitable partitions we can always create a larger set that is a complete set of local equitable partitions, i.e. a more refined complete set of local equitable partitions.

With this in place we can now give our main result, namely that the eigenvalues of a graph with an equitable partition are either global eigenvalues of the original graph, or local eigenvalues of certain subgraphs determined by the local equitable partition of π.

Theorem 2.2 *(Complete equitable decomposition). Let π be an equitable partition of the graph $G = (V, E, \omega)$ whose adjacency matrix is Hermitian. If G_i are the induced subgraphs corresponding to a complete set of local equitable partitions $\bar{\pi}_i$ of π for $i = 1, 2, \dots, r$ then*

$$\sigma(G) = \sigma(G_\pi) \cup [\sigma^{\ell}_{\bar{\pi}_1}(G_1) \cup \cdots \cup \sigma^{\ell}_{\bar{\pi}_r}(G_r)],$$

which we refer to as a complete equitable decomposition of G with respect to π.

We save the proof of Theorem 2.2 for Section 4 where we identify necessary and sufficient conditions for when G_i has the property that $\sigma^{\ell}_{\bar{\pi}_i}(G_i) \subseteq \sigma(G) - \sigma(G_\pi)$.

Example 2.3. Consider the simple graph $G = (V, E, \omega)$ in Fig. 2 with equitable partition $\pi = \{V_1, V_2, V_3, V_4, V_5, V_6\}$ colored red, yellow, green, brown, blue, orange; respectively. The equitable partition π has the four local equitable partitions $\bar{\pi}_1 = \{V_3\}$, $\bar{\pi}_2 = \{V_1, V_2\}$, $\bar{\pi}_3 = \{V_5\}$, and $\bar{\pi}_4 = \{V_6\}$ corresponding to the graphs G_1, G_2, G_3, and G_4, shown left, respectively. These collectively form a complete set of local equitable partitions, i.e. $\pi - \cup_{i=1}^{4} \bar{\pi}_i = V_4 = \{11\}$ is a singleton partition element of π. Here the divisor graph of G has the eigenvalues

$$\sigma(G_\pi) = \{3.83, -2.91, 2.82, 1.34, -1.07, 1\}.$$

The local eigenvalues of each graph G_i for $i = 1, 2, 3, 4$ are given by

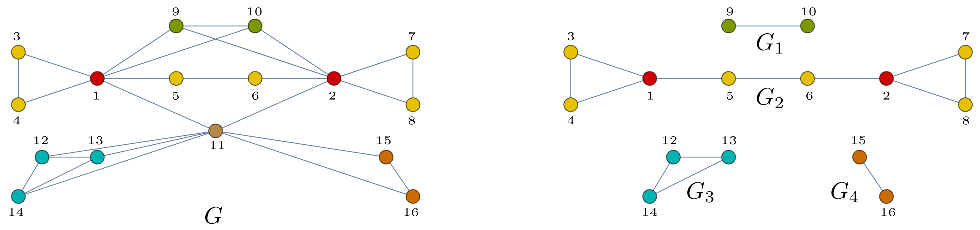


Fig. 2. A simple graph $G = (V, E, \omega)$, shown left, with equitable partition $\pi = \{V_1, V_2, V_3, V_4, V_5, V_6\}$ indicated by red, yellow, green, brown, blue, and orange; respectively. The equitable partition π has the four local equitable partitions $\bar{\pi}_1 = \{V_3\}$, $\bar{\pi}_2 = \{V_1, V_2\}$, $\bar{\pi}_3 = \{V_5\}$, and $\bar{\pi}_4 = \{V_6\}$ corresponding to the graphs G_1, G_2, G_3, and G_4, shown right, respectively. The eigenvalues of the graph G are then given by the union of its global and local eigenvalues $\sigma(G) = \sigma(G_\pi) \cup \sigma^{\ell}_{\bar{\pi}_1}(G_1) \cup \sigma^{\ell}_{\bar{\pi}_2}(G_2) \cup \sigma^{\ell}_{\bar{\pi}_3}(G_3) \cup \sigma^{\ell}_{\bar{\pi}_4}(G_4)$ (cf. Example 2.3).

$$\begin{aligned}
\sigma^{\ell}_{\bar{\pi}_1}(G_1) =& \sigma(G_1) - \sigma((G_1)_{\bar{\pi}_1}) = \{1, -1\} - \{1\} = \{-1\} \\
\sigma^{\ell}_{\bar{\pi}_2}(G_2) =& \sigma(G_2) - \sigma((G_2)_{\bar{\pi}_2}) \\
=& \{2.23, 2.11, -1.86, -1.30, -1, -1, 1, -0.25\} - \{2.23, -1.30\} \\
=& \{2.11, -1.86, -1, -1, 1, -0.25\} \\
\sigma^{\ell}_{\bar{\pi}_3}(G_3) =& \sigma(G_3) - \sigma((G_3)_{\bar{\pi}_3}) = \{2, -1, -1\} - \{2\} = \{-1, -1\} \\
\sigma^{\ell}_{\bar{\pi}_4}(G_4) =& \sigma(G_4) - \sigma((G_4)_{\bar{\pi}_4}) = \{1, -1\} - \{1\} = \{-1\}.
\end{aligned}$$

Using Theorem 2.2 the eigenvalues of the graph G are the union of its local and global eigenvalues given by $\sigma(G) = \sigma(G_\pi) \cup \sigma^{\ell}_{\bar{\pi}_1}(G_1) \cup \sigma^{\ell}_{\bar{\pi}_2}(G_2) \cup \sigma^{\ell}_{\bar{\pi}_3}(G_3) \cup \sigma^{\ell}_{\bar{\pi}_4}(G_4)$ or

$$\begin{aligned}
\sigma(G) = \{3.83, -2.91, 2.82, 1.34, -1.07, 1\} \cup \{-1\} \\
\cup \{2.11, -1.86, -1, -1, 1, -0.25\} \cup \{-1, -1\} \cup \{-1\},
\end{aligned}$$

which is a complete equitable decomposition of G with respect to π.

We note the complete set of local equitable partitions $\{\bar{\pi}^4_{i=1}\}$ is not unique. For instance, the set $\{\bar{\pi}_1, \bar{\pi}_2, \bar{\pi}_3 \cup \bar{\pi}_4\}$ is a less refined but second complete set of local equitable partitions, which can similarly be used to create a different complete equitable decomposition of the graph G.

3. Global and local eigenvectors

In order to prove Theorem 2.2, we need to develop the theory of local and global eigenvectors. In a similar manner to how we partition the adjacency matrix A with respect to π we can partition an eigenvector

$$\mathbf{v} = \begin{bmatrix} v(1)^T & \dots & v(k)^T \end{bmatrix}^T \in \mathbb{C}^n$$

of an adjacency matrix A with respect to a vertex partition $\pi = \{V_1, \dots, V_k\}$ where $v(j) \in \mathbb{C}^{|V_j|}$ denotes the entries of $\mathbf{v}$ indexed by V_j. For a collection of eigenvectors

$\mathbf{v}_1, \ldots, \mathbf{v}_n$ partitioned by π, we denote the entries of the ith eigenvector $\mathbf{v}_i$ indexed by the element V_j as $v_i(j)$.

Given a graph G on $|V| = n$ vertices with a vertex partition $\pi = \{V_1, \ldots, V_k\}$, the *characteristic matrix* of G with respect to π is the matrix $S = [s_{ij}] \in \{0,1\}^{|V| \times k}$ where each column of S represents a partition element, and each row represents a vertex. The entry s_{ij} of S is given by

$$s_{ij} = \begin{cases} 1 & \text{if } \ i \in V_j \\ 0 & \text{otherwise.} \end{cases}$$

$S^T S$ is the nonsingular diagonal matrix $diag(|V_1|, |V_2|, \ldots, |V_k|)$. This is due to $S^T S$ being a square $k \times k$ matrix and the columns of S being orthogonal. Therefore $[S^T S]_{ij} = 0$ for $i \neq j$, and $[S^T S]_{ii} = |V_i| > 0$.

The eigenvectors of the divisor graph G_π are related to the eigenvectors of G in the following way.

Theorem 3.1 *(Eigenpairs of the divisor matrix). If π is an equitable partition of a graph G with an Hermitian adjacency matrix and $(\lambda, \mathbf{v})$ is an eigenpair of the divisor matrix A_π, then $(\lambda, S\mathbf{v})$ is an eigenpair of $A = A(G)$.*

Similar to the proof of Theorem 2.1 the proof of Theorem 3.1 can be found in [2].

A useful feature of the characteristic matrix is that it connects the adjacency matrix of the graph to its divisor matrix via the following theorem which is proved in [12].

Theorem 3.2. *Let π be an equitable partition of the graph G with characteristic matrix S, A the adjacency matrix, and A_π the adjacency matrix of the divisor graph. Then*

$$AS = SA_\pi$$
$$A_\pi = (S^T S)^{-1} S^T AS.$$

A known consequence of the fact that $AS = SA_\pi$ is that the $col(S)$ is A-invariant [2,12]. Therefore there must exist a set of k orthogonal eigenvectors that span $col(S)$ and can be written as a linear combination of the columns of S.

Since $s_{ij} = 1$ if $i \in V_j$ and is 0 otherwise, then any linear combinations of the columns of S will result in a vector $\mathbf{v} = [v(1)^T \ldots v(k)^T]^T$ with constant entries in $v(j)$ for $j = 1, \ldots, k$. As each $v(j)$ has length equal to $|V_j|$ then this implies $(col(S))^\perp$ must also be A-invariant. Thus, $(col(S))^\perp$ has a basis consisting of $n - k$ eigenvectors $\mathbf{w}_{k+1}, \ldots, \mathbf{w}_n$ of A that are orthogonal to the k eigenvectors that span $col(S)$.

In order for $\mathbf{v} \cdot \mathbf{w} = 0$ where $\mathbf{v} \in col(S)$ and $\mathbf{w} \in col(S)^\perp$, the sum of the entries in $w(j)$ must be zero for a fixed $j = 1, 2, \ldots, k$. Therefore, there exists a set of eigenvectors of A such that we can divide them into two groups: those with constant entries in $v(j)$ for each j (which come from $col(S)$), and those where the entries in each $v(j)$ sum to

0 for each j (which come from $(col(S))^{\perp}$). While this is known, we formally give these eigenvectors the names *global* and *local eigenvectors*, respectively.

Definition 3.1 *(Global eigenvectors).* Let G be a graph with equitable partition $\pi = \{V_1, \ldots, V_k\}$. We say $\mathbf{v} = [\, v(1)^T \quad \ldots \quad v(k)^T \,]^T \in \mathbb{C}^n$ is a *global eigenvector* of G if $\mathbf{v}$ is an eigenvector of the graph's adjacency matrix and for all $j = 1, 2, \ldots, k$ each entry of $v(j)$ is constant.

For the graph G in Fig. 1 (top left) with equitable partition $\pi = \{V_1, V_2, V_3\}$, the global eigenvalue $\lambda = 1$ has the global eigenvector

$$\mathbf{v}_g = \begin{bmatrix} v(1)^T & v(2)^T & v(3)^T \end{bmatrix}^T = \left[\begin{array}{cc|cccccc|cc} 0 & 0 & -2 & -2 & -2 & -2 & -2 & -2 & 3 & 3 \end{array}\right]^T. \tag{6}$$

We note that as $\mathbf{v}_g$ is a global eigenvector, each of the entries in $v(j)$ for a given $j = 1, 2, 3$ is constant, where $|V_1| = 2$, $|V_2| = 6$, and $|V_3| = 2$.

Definition 3.2 *(Local eigenvectors).* Let G be a graph with equitable partition $\pi = \{V_1, \ldots, V_k\}$. We say $\mathbf{v} = [\, v(1)^T \quad \ldots \quad v(k)^T \,]^T \in \mathbb{C}^n$ is a *local eigenvector* of G if $\mathbf{v}$ is an eigenvector of the graph's adjacency matrix and for each $j = 1, 2, \ldots, k$ the sum of the entries in $v(j)$ is zero.

Again for the graph G in Fig. 1 (top left) with equitable partition $\pi = \{V_1, V_2, V_3\}$ the local eigenvalue $\lambda = 1$ has the local eigenvector

$$\mathbf{v}_\ell = \begin{bmatrix} v(1)^T & v(2)^T & v(3)^T \end{bmatrix}^T = \left[\begin{array}{cc|cccccc|cc} 0 & 0 & -2 & 1 & 1 & -2 & 1 & 1 & 0 & 0 \end{array}\right]^T. \tag{7}$$

Here the entries of $v(j)$ for a given $j = 1, 2, 3$ sum to zero where, as before, $|V_1| = 2$, $|V_2| = 6$, and $|V_3| = 2$.

In Definition 3.1 we chose the name *global eigenvectors* because the eigenvalues associated with global eigenvectors are the "global eigenvalues" defined in Definition 2.2. The converse however is not true. For instance, consider the combination of the two vectors in Equations (6) and (7) into the eigenvector

$$\begin{aligned} \mathbf{v} = \mathbf{v}_g + \mathbf{v}_\ell &= \begin{bmatrix} v(1)^T & v(2)^T & v(3)^T \end{bmatrix}^T \\ &= \left[\begin{array}{cc|cccccc|cc} 0 & 0 & -4 & -1 & -1 & -4 & -1 & -1 & 3 & 3 \end{array}\right]^T, \end{aligned} \tag{8}$$

which is neither a global eigenvector nor a local eigenvector. This is possible as the associated eigenvalue $\lambda = 1$ is both a local and a global eigenvalue of G with respect to the equitable partition.

In the following theorem we show if $\lambda \in \sigma(G_\pi)$ is a global eigenvalue, there exists a global eigenvector associated with it.

Theorem 3.3. *Let G be a graph with Hermitian adjacency matrix and equitable partition π.*
(i) If $(\lambda, \mathbf{v})$ is an eigenpair of G where $\mathbf{v}$ is a global eigenvector with respect to π then $\lambda \in \sigma(A_\pi)$; and
(ii) If $\lambda \in \sigma(A_\pi)$ then there exists a global eigenvector of A associated with λ.

Proof. If $\mathbf{v}$ is a global eigenvector, then we can write $\mathbf{v} = \begin{bmatrix} a_1\mathbf{1}_{|V_1|}^T & \dots & a_k\mathbf{1}_{|V_k|}^T \end{bmatrix}^T$ where $\mathbf{1}_{|V_j|}$ is the all ones vector of size $|V_j|$. Then

$$A\mathbf{v} = \begin{bmatrix} A_{11} & \dots & A_{1k} \\ \vdots & \ddots & \vdots \\ A_{k1} & \dots & A_{kk} \end{bmatrix} \begin{bmatrix} a_1\mathbf{1}_{|V_1|} \\ \vdots \\ a_k\mathbf{1}_{|V_k|} \end{bmatrix}$$
$$= \begin{bmatrix} a_1A_{11}\mathbf{1}_{|V_1|} + \dots + a_kA_{1k}\mathbf{1}_{|V_k|} \\ \vdots \\ a_1A_{k1}\mathbf{1}_{|V_1|} + \dots + a_kA_{kk}\mathbf{1}_{|V_k|} \end{bmatrix} = \lambda \begin{bmatrix} a_1\mathbf{1}_{|V_1|} \\ \vdots \\ a_k\mathbf{1}_{|V_k|} \end{bmatrix}.$$

Each submatrix A_{ij} will have constant row sums due to G having the equitable partition π. Therefore, we can write $A_{ij}\mathbf{1}_{|V_j|} = b_{ij}\mathbf{1}_{|V_i|}$ where the constant b_{ij} is the ijth entry in A_π. Considering an arbitrary entry in $A\mathbf{v}$, we get that the following holds for all $i = 1, \dots, k$:

$$\begin{aligned} a_1A_{i1}\mathbf{1}_{|V_1|} + \dots + a_kA_{ik}\mathbf{1}_{|V_k|} &= a_1b_{i1}\mathbf{1}_{|V_i|} + \dots + a_kb_{ik}\mathbf{1}_{|V_i|} \\ &= (a_1b_{i1} + \dots + a_kb_{ik})\mathbf{1}_{|V_i|} \\ &= \lambda a_i\mathbf{1}_{|V_i|}. \end{aligned}$$

From the final equality we can we conclude $(a_1b_{i1} + \dots + a_kb_{ik}) = \lambda a_i$. Therefore the vector $\mathbf{w} = [\, a_1 \quad \dots \quad a_k \,]^T$ satisfies $A_\pi\mathbf{w} = \lambda\mathbf{w}$ implying $\lambda \in \sigma(A_\pi)$.

If $\lambda \in \sigma(A_\pi)$, then by Theorem 3.1, which requires A to be Hermitian, there exists an eigenvector $\mathbf{v}$ of A that can be written as the matrix-vector product $S\mathbf{w} = [\, ((S\mathbf{w})(1))^T \quad \dots \quad ((S\mathbf{w})(k))^T \,]^T$ where $\mathbf{w}$ is an eigenvector of A_π associated with λ and S is the characteristic matrix for G. Recall that, for simplicity and without loss of generality, we partitioned the rows of S according to the equitable partition π. Therefore S will have $|V_i|$ consecutive, identical rows for a given $i = 1, \dots, k$ implying the entries of $(S\mathbf{w})(i)$ will be identical after multiplying S and $\mathbf{w}$. Given this is true for all i, $S\mathbf{w}$ is by definition a global eigenvector. □

For a graph G with a equitable partition π it is always possible to find a set of corresponding eigenvectors which can be partitioned into global eigenvectors and local eigenvectors. A consequence of Theorem 3.3 is that all the local eigenvalues $\sigma_\pi^\ell(G)$ must all be associated with local eigenvectors. Thus we create a simple correspondence between global and local eigenvalues as well as global and local eigenvectors, respectively.

4. Proof of the complete equitable decomposition method

In this section we prove the main result of this paper, which describes how we can recover all the eigenvalues of a graph (matrix) using its complete equitable decomposition. To do this we require the following theorem.

Theorem 4.1. *Let $\pi = \{V_1, \ldots, V_k\}$ be an equitable partition on a graph G with Hermitian adjacency matrix A, and $\bar{\pi}$ a subset of π. Let G_* be the induced subgraph of G restricted to the union of vertices contained in $\bar{\pi}$. Then $\bar{\pi}$ is a local equitable partition if and only if $\sigma^{\ell}_{\bar{\pi}}(G_*) \subseteq \sigma(G) - \sigma(G_\pi)$.*

Proof. Suppose $\bar{\pi}$ has $s < k$ partitions. Without loss of generality, suppose the first s partitions of A are the s partitions contained in $\bar{\pi}$. First we define $m = |V(G_*)|$. Also we let $T = [I_m \; 0^T_{(n-m),m}]^T$, and thus the principal submatrix $B_{11} = (T)^T AT$ is the adjacency matrix of G_*. We write A as the partitioned block matrix so that

$$A = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}.$$

Let $\mathbf{v}$ be a local eigenvector of G_* so that

$$T\mathbf{v} = \begin{bmatrix} \mathbf{v}^T & \mathbf{0}^T \end{bmatrix}^T = \begin{bmatrix} v(1)^T & \ldots & v(s)^T & \mathbf{0}^T \end{bmatrix}^T \neq \mathbf{0}.$$

In order for $T\mathbf{v}$ to be an eigenvector of A, the following equation must hold:

$$A(T\mathbf{v}) = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} B_{11}\mathbf{v} + B_{12}\mathbf{0} \\ B_{21}\mathbf{v} + B_{22}\mathbf{0} \end{bmatrix} = \begin{bmatrix} B_{11}\mathbf{v} \\ B_{21}\mathbf{v} \end{bmatrix} = \begin{bmatrix} \lambda\mathbf{v} \\ \mathbf{0} \end{bmatrix} = \lambda(T\mathbf{v})$$

which implies that $B_{21}\mathbf{v} = \mathbf{0}$. We can write B_{21} as the following block matrix

$$B_{21} = \begin{bmatrix} A_{(s+1)1} & \ldots & A_{(s+1)s} \\ \vdots & \ddots & \vdots \\ A_{k1} & \ldots & A_{ks} \end{bmatrix}$$

where A_{ij} represents connections between partitions $V_i \in \pi$ where $V_i \notin \bar{\pi}$, and $V_j \in \bar{\pi}$ for all $i \in \{s+1, \ldots, k\}$, and for all $j \in \{1, \ldots, s\}$. Hence,

$$B_{21}\mathbf{v} = \begin{bmatrix} A_{(s+1)1} & \ldots & A_{(s+1)s} \\ \vdots & \ddots & \vdots \\ A_{k1} & \ldots & A_{ks} \end{bmatrix} \begin{bmatrix} v(1) \\ \vdots \\ v(s) \end{bmatrix} = \begin{bmatrix} A_{(s+1)1}v(1) + \cdots + A_{(s+1)s}v(s) \\ \vdots \\ A_{k1}v(1) + \cdots + A_{ks}v(s) \end{bmatrix}.$$

Assume $\bar{\pi}$ is a local equitable partition. Therefore, for all i, j we have the $|V_i| \times |V_j|$ matrix $A_{ij} = \alpha_{ij} J_{|V_i|,|V_j|}$ where $\alpha \in \mathbb{C}$. Because $\mathbf{v}$ is a local eigenvector, the sum of the entries in $v(i)$ is 0. Thus $A_{ij}v(j) = \alpha_{ij} J_{|V_i|,|V_j|} v(j) = \alpha_{ij}\mathbf{0} = \mathbf{0}$ for all i and j implying $B_{21}\mathbf{v} = \mathbf{0}$. This means $(\lambda, T\mathbf{v})$ is a local eigenpair of G where $(\lambda, \mathbf{v})$ is a local eigenpair of G_*. Since $(\lambda, \mathbf{v})$ was an arbitrary local eigenpair of G_*, we have every local eigenvalue of G_* will be a local eigenvalue of G. Thus, $\sigma^{\ell}_{\bar{\pi}}(G_*) \subseteq \sigma(G) - \sigma(G_\pi)$.

For the other implication, suppose $\sigma^{\ell}_{\bar{\pi}}(G_*) \subseteq \sigma(G) - \sigma(G_\pi)$ for the subgraph G_* chosen from $\bar{\pi} \subseteq \pi$. We will show $\bar{\pi}$ must be a local equitable partition. Since we set $|V(G_*)| = m$ and assumed $\bar{\pi}$ had s partition elements, this implies $\bar{\pi}$ is an equitable partition on G_* that has s partition elements and thus s linearly independent global eigenvectors. Also there are $m - s$ linearly independent local eigenvectors of G_* because B is Hermitian due to being a principle submatrix of the Hermitian matrix A.

We are assuming that $\sigma^{\ell}_{\bar{\pi}}(G_*) \subseteq \sigma(G) - \sigma(G_\pi)$, which implies $B_{21}\mathbf{v} = \mathbf{0}$ for every local eigenvector $\mathbf{v}$ of G_*. If we consider an arbitrary row-vector $a = \begin{bmatrix} a_1 & \dots & a_m \end{bmatrix}$ of B_{21}, then we know that $\mathbf{v} \cdot a^T = 0$ for all local eigenvectors of $\mathbf{v}$ of G_*. Also, we can partition a according the vertex partitions $V_i \in \bar{\pi}$ and write $a = \begin{bmatrix} a(1)^T & \dots & a(s)^T \end{bmatrix}$. Then we can write the following system of equations in matrix form

$$La^T = \begin{bmatrix} v_1(1)^T & \dots & v_1(s)^T \\ \vdots & \ddots & \vdots \\ v_{(m-s)}(1)^T & \dots & v_{(m-s)}(s)^T \end{bmatrix} \begin{bmatrix} a(1) \\ \vdots \\ a(s) \end{bmatrix} = \mathbf{0}$$

where L is a $(m - s) \times m$ matrix that has linearly independent local eigenvector rows.

By the Rank-Nullity Theorem, we know the rank of L is $m - s$ and the nullity is s. Notice the columns of the characteristics matrix S of G_* forms a basis for the nullspace of L. Recall that the i^{th} column of S will take the form

$$s_i = \begin{bmatrix} \mathbf{0}_{|V_1|} \\ \vdots \\ \mathbf{1}_{|V_i|} \\ \vdots \\ \mathbf{0}_{|V_s|} \end{bmatrix}$$

where there is an all-ones vector corresponding to the i^{th} partition for $1 \le i \le s$ and zeros elsewhere. Since the entries in each $v_k(i)$ sum to zero for all $1 \le k \le m - s$, then $v_k(i) \cdot \mathbf{1}_{|V_i|} = 0$ for all k. Therefore, each s_i is contained in null-space of L. Clearly this set is also linearly independent. This means a^T is a linear combination of the columns of S, due to a^T being in the nullspace of L, implying that $a(i) = \alpha_i \mathbf{1}_{|V_i|}$ for all $1 \le i \le s$. Thus we can write $a = \begin{bmatrix} \alpha_1 \mathbf{1}^T_{|V_1|} & \dots & \alpha_s \mathbf{1}^T_{|V_s|} \end{bmatrix}$. Given a was an arbitrary row of B_{21}, we know this must hold for all rows of B_{21}.

The rows of the submatrix A_{ij} from B_{21} can be written as $\alpha_i \mathbf{1}^T_{|V_j|}$ where α_i could be unique to the row. However, we show this is not the case because the equitable partition π on G guarantees A_{ij} will have constant row sums. Thus α must be constant and cannot vary across the rows of A_{ij}. Hence, $A_{ij} = \alpha_{ij} J_{|V_i|,|V_j|}$. This is true for every block of B_{21} which is the definition of local equitable partition. □

Recall our main result, Theorem 2.2, that states the following. Let $\pi = \{V_1, \ldots, V_k\}$ be an equitable partition of a graph G with a Hermitian adjacency matrix A and let $\bar{\pi}_1, \ldots, \bar{\pi}_r$ be a complete set of local equitable partitions with corresponding subgraphs $G_1, \ldots, G_r$. Then

$$\sigma(G) = \sigma(G_\pi) \cup [\sigma^{\ell}_{\bar{\pi}_1}(G_1) \cup \cdots \cup \sigma^{\ell}_{\bar{\pi}_r}(G_r)].$$

The following is a proof of Theorem 2.2

Proof. Let π be an equitable partition of G with a complete set of local equitable partitions $\{\bar{\pi}_1 \ldots \bar{\pi}_r\}$. Thus $\pi - \cup_{i=1}^r \bar{\pi}_i$ is a set of singleton partition elements of π, and suppose $|\pi - \cup_{i=1}^r \bar{\pi}_i| = s$. We can partition A into $r+s$ partitions as

$$A = \begin{bmatrix} B_{11} & \ldots & B_{1r} & \ldots & B_{1(r+s)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ B_{r1} & \ldots & B_{rr} & \ldots & B_{r(r+s)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ B_{(r+s)1} & \ldots & B_{(r+s)r} & \ldots & B_{(r+s)(r+s)} \end{bmatrix}$$

where B_{ii} is the adjacency matrix for the subgraph G_i selected by the local equitable partition $\bar{\pi}_i$ for $1 \leq i \leq r$. When $r+1 \leq i, j \leq r+s$, B_{ij} represents a singleton partition element of π and is a 1×1 matrix.

Now consider a local eigenpair $(\lambda, \mathbf{v})$ of G. We partition

$$\mathbf{v} = \begin{bmatrix} v[1]^T & \ldots & v[r]^T & \ldots & v[r+s]^T \end{bmatrix}^T$$

where $v[i]$ denotes the entries of $\mathbf{v}$ indexed by $V(G_i)$, the group of partition elements that make up B_{ii}. If $r+1 \leq i \leq r+s$, then $|v[i]| = 1$ and $v[i] = 0$ because v is a local eigenvector. Thus we can write $\mathbf{v} = \begin{bmatrix} v[1]^T & \ldots & v[r]^T & \ldots & 0 \end{bmatrix}^T$. Hence, we get

$$A\mathbf{v} = \begin{bmatrix} B_{11} & \ldots & B_{1r} & \ldots & B_{1(r+s)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ B_{r1} & \ldots & B_{rr} & \ldots & B_{r(r+s)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ B_{(r+s)1} & \ldots & B_{(r+s)r} & \ldots & B_{(r+s)(r+s)} \end{bmatrix}$$

$$\times \begin{bmatrix} v[1] \\ \vdots \\ v[r] \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} B_{11}v[1] + \cdots + B_{1r}v[r] \\ \vdots \\ B_{r1}v[1] + \cdots + B_{rr}v[r] \\ \vdots \\ B_{(r+s)1}v[1] + \cdots + B_{(r+s)r}v[r] \end{bmatrix}$$

Notice for $i \neq j$ that B_{ij} is a block matrix where each block inside B_{ij} is a matrix $A_{ab} = \alpha_{ab} J_{|V_a|,|V_b|}$. Therefore $B_{ij}v[j] = \mathbf{0}$ since the sum of the entries in $v[j]$ equals zero. Now $A\mathbf{v}$ simplifies to

$$A\mathbf{v} = \begin{bmatrix} B_{11}v[1] + \cdots + B_{1r}v[r] \\ \vdots \\ B_{r1}v[1] + \cdots + B_{rr}v[r] \\ \vdots \\ B_{(r+s)1}v[1] + \cdots + B_{(r+s)r}v[r] \end{bmatrix} = \begin{bmatrix} B_{11}v[1] \\ \vdots \\ B_{rr}v[r] \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \lambda \mathbf{v}$$

By definition $\mathbf{v} \neq \mathbf{0}$, so there exists some i in which $v[i] \neq \mathbf{0}$ and $B_{ii}v[i] = \lambda v[i]$ implying $\lambda \in \sigma^{\ell}_{\bar{\pi}_i}(G_i)$. Hence $\sigma(G) - \sigma(G_\pi) \subseteq \sigma^{\ell}_{\bar{\pi}_1}(G_1) \cup \cdots \cup \sigma^{\ell}_{\bar{\pi}_r}(G_r)$.

By Theorem 4.1.1, we know $\sigma^{\ell}_{\bar{\pi}_i}(G_i) \subseteq \sigma(G) - \sigma(G_\pi)$ for all $i = 1, \ldots, r$ resulting in $\sigma^{\ell}_{\bar{\pi}_1}(G_1) \cup \cdots \cup \sigma^{\ell}_{\bar{\pi}_r}(G_r) \subseteq \sigma(G) - \sigma(G_\pi)$. Thus $\sigma(G) - \sigma(G_\pi) = \sigma^{\ell}_{\bar{\pi}_1}(G_1) \cup \cdots \cup \sigma^{\ell}_{\bar{\pi}_r}(G_r)$.

Finally, we get the result that

$$\sigma(G) = \sigma(G_\pi) \cup [\sigma(G) - \sigma(G_\pi)] = \sigma(G_\pi) \cup [\sigma^{\ell}_{\bar{\pi}_1}(G_1) \cup \cdots \cup \sigma^{\ell}_{\bar{\pi}_r}(G_r)]. \quad \square$$

Thus the eigenvalues of a graph G with an adjacency matrix A and equitable partition π can be decomposed into the global eigenvalues of G and a collection of local eigenvalues from induced subgraphs G_i of G that respect the equitable partition π. In short, the eigenvalues are the collection of the "global eigenvalues of the global equitable partition together with the local eigenvalues of the local equitable partitions."

5. Computing eigenvalues using complete equitable partitions

The standard and likely most well-known algorithm for finding the eigenvalues of a matrix $A \in \mathbb{C}^{n\times n}$ has computational complexity $\mathcal{O}(n^3)$ (see, for instance, [22]). Other algorithms have been proposed with lower computational complexity $\mathcal{O}(n^b)$ for $b < 3$ (see [8]). In this section we consider how quickly one can find all the eigenvalues of a Hermitian matrix $A \in \mathbb{C}^{n\times n}$ if it is the adjacency matrix of a graph G with a nontrivial equitable partition π.

The main result of this section is that it is, in principle, possible to find the eigenvalues of a graph (matrix) faster by finding its complete equitable decomposition rather than by using standard methods. Our strategy is to leverage our main result found in Theorem 2.2 which decomposes the eigenvalues of a graph (matrix) into the disjoint union

$$\sigma(G) = \sigma(G_\pi) \cup [\sigma^\ell_{\bar{\pi}_1}(G_1) \cup \cdots \cup \sigma^\ell_{\bar{\pi}_r}(G_r)].$$

Our strategy is to individually find the eigenvalues $\sigma(G_\pi), \sigma^\ell_{\bar{\pi}_1}(G_1), \ldots \sigma^\ell_{\bar{\pi}_r}(G_r)$. The improvement in computational complexity naturally depends on the specific details of the graph's equitable partition. Roughly speaking, the less trivial the equitable partition the faster the graph's eigenvalues can be found. (See, Theorem 5.1 and Example 5.1.)

It is worth noting that many graphs associated with real-world data are known to have nontrivial equitable partitions. For example, the authors of [18] show that many real-world networks have nontrivial symmetries. As any graph symmetry induces an equitable partition, such networks have nontrivial equitable partitions.

5.1. The LEParD algorithm

In this section we will outline our algorithm for finding the eigenvalues for a graph (matrix) and describe its computational complexity. As a complete equitable decomposition requires an equitable partition of a graph we choose the graph's *coarsest equitable partition*.

Definition 5.1 *(Coarsest equitable partition).* An equitable partition $\pi_* = \{V_1^*, \ldots, V_k^*\}$ of a graph G is the graph's *coarsest equitable partition* if for any partition element $V_i \in \pi$, where π is another equitable partition on G, the element $V_i \subseteq V_j^*$ for some $j \in \{1, \ldots, k\}$.

For a given graph, the coarsest equitable partition always exists and is unique [24]; moreover it can be found relatively quickly [1].

The algorithm we propose for finding the eigenvalues of a graph (matrix) is referred to as the **LEParD Algorithm**, or **L**ocal **E**quitable **Par**tition **D**ecomposition Algorithm, which is comprised of the following four steps.

The LEParD Algorithm. Let $G = (V, E, \omega)$ be a graph with a Hermitian adjacency matrix $A \in \mathbb{C}^{n\times n}$. To find its eigenvalues:

Step (1) Find the coarsest equitable partition $\pi = \{V_1, V_2, \ldots, V_k\}$ of G;
Step (2) Compute the global eigenvalues $\sigma(G_\pi)$ of the divisor matrix A_π;
Step (3) Find the local equitable partitions $\bar{\pi}_i$ of G to identify the subgraphs G_i; and
Step (4) For each G_i, compute its local eigenvalues $\sigma^\ell_{\pi_i}(G_i) = \sigma(G_i) - \sigma((G_i)_{\pi_i})$.

To carry out Steps (2) and (4) of the LEParD algorithm we need to compute the eigenvalues of specific graphs, i.e. $G_\pi, G_1, \ldots, G_r$. This can be done using any algorithm we like. If the particular algorithm we use has order $\mathcal{O}(a^b)$ for a graph of size $|G| = a$

we refer to this as *using an eigenvalue finder of order* $\mathcal{O}(a^b)$. With this in place we can bound the computational complexity of the LEParD algorithm.

Theorem 5.1 *(Computational complexity of the LEParD algorithm). Let* $G = (V, E, \omega)$ *be a graph with coarsest equitable partition* $\pi = \{V_1, V_2, \ldots, V_k\}$. *Using an eigenvalue finder of order* $\mathcal{O}(a^b)$ *the computational complexity of running the LEParD Algorithm on* G *has order*

$$\mathcal{O}\left(m\log(n) + k^b + n + \sum_{i=1}^{r} n_i^b\right) \tag{9}$$

where $n = |G|$, $m = |E|$, *and* $n_i = |G_i|$ *for* $i \in \{1, \ldots, r\}$.

Proof. As each piece of the temporal complexity in Equation (9) corresponds to a step in the LEParD algorithm here we prove the complexity of each piece.

1. The coarsest equitable partition π of $G = (V, E, \omega)$ is found using the algorithm in [1], which has computational complexity $\mathcal{O}(m\log(n))$.
2. As $|\pi| = k$ then $|G_\pi| = k$ and the computational complexity of finding $\sigma(G_\pi)$ is $\mathcal{O}(k^b)$.
3. We can identify the local equitable partitions $\bar{\pi}_i$ and the associated subgraphs G_i by finding the constant submatrices $A_{st} = \alpha_{st}J_{|V_s|,|V_t|}$ of the graph's adjacency matrix. In Theorem 5.2 we show that the computational complexity of this step is $\mathcal{O}(m+n)$ using Algorithm 1. Note, however, that m is omitted in the final $\mathcal{O}$ expression since $m < m\log(n)$.
4. Finding the local eigenvalues $\sigma^{\ell}_{\bar{\pi}_i}(G_i) = \sigma(G_i) - \sigma((G_i)_{\pi_i})$ has computational complexity in $\mathcal{O}\big(n_i^b\big)$ where $n_i = |G_i|$, since $|G_i| \geq |(G_i)_{\pi_i}|$. Over all local equitable partitions this has computational complexity $\mathcal{O}\big(\sum_{i=1}^{r} n_i^b\big)$.

Together, Steps (1)–(4) have computational complexity given by Equation (9). □

Since the graphs $G_1, G_2, \ldots, G_r$ are disjoint then it is possible, at least in principle, to simultaneously compute their eigenvalues, i.e., to parallelize the LEParD algorithm. Using this *Parallelized LEParD algorithm* we have the following computational complexity bound, which follows from the proof of Theorem 5.1.

Corollary 5.1 *(Computational complexity of the parallelized LEParD algorithm). Let* $G = (V, E, \omega)$ *be a graph with coarsest equitable partition* $\pi = \{V_1, V_2, \ldots, V_k\}$. *Using an eigenvalue finder of order* $\mathcal{O}(a^b)$ *the computational complexity of running the Parallelized LEParD Algorithm on* G *has order*

$$\mathcal{O}\Big(m\log(n) + k^b + n + \max\{n_i^b\}_{i=1}^{r}\Big) \tag{10}$$

where $n = |G|$, $m = |E|$, $n_i = |G_i|$ *for* $i \in \{1, \ldots, r\}$.

In the following section we show that local equitable partitions can be computed in $\mathcal{O}(m+n)$ time justifying the claim that Step (3) of our algorithm has this as its temporal complexity.

5.2. The LEP finder and monad LEP sets

To carry out Step (3) of the LEParD algorithm we will construct a specific set of local equitable partitions (LEPs) for a given graph and coarsest equitable partition π. The main idea is that $\bar{\pi} \subseteq \pi$ is a local equitable partition of π if and only if the partition elements of $\bar{\pi}$ are consistently connected to those not in $\bar{\pi}$ (see Definition 2.3). Thus, partition elements that are not consistently connected cannot be LEPs on their own. By grouping such elements together as one LEP, $\hat{\pi}$, we can create a subgraph that is consistently connected to all other $V_i \notin \hat{\pi}$, which will be an LEP.

There often exist many different sets of LEPs. However, not all LEP sets can be leveraged for efficient eigenvalue calculation. For instance, the trivial LEP $\bar{\pi}_0 = \pi$ may be formed by grouping all equitable partition elements together; however, we must then compute eigenvalues of the subgraph $G_0 = G$, providing no performance advantage over simply computing the eigenvalues of the original graph G with traditional methods. To maximize efficiency, the LEParD algorithm constructs the smallest possible LEPs from the coarsest equitable partition. We call this set of LEPs the Monad LEP Set, and define it as follows.

Definition 5.2 *(Monad LEP set).* Let $G = (V, E, \omega)$ be a graph with coarsest equitable partition $\pi = \{V_1, V_2, ..., V_k\}$. Then a complete set of LEPs $\mathscr{L} = \{\bar{\pi}_1, ..., \bar{\pi}_r\}$ with respect to π is the *Monad LEP Set* if for any other complete set of LEPs $\hat{\mathscr{L}} = \{\hat{\pi}_1, ..., \hat{\pi}_s\}$ with respect to π each

$$\hat{\pi}_i = \bigcup_{j \in J_i} \bar{\pi}_j \quad \text{for some} \quad J_i \subseteq \{1, \ldots, r\}.$$

As using the LEParD algorithm efficiently depends on finding a Monad LEP Set, it is important to understand how this set can be directly constructed for a given graph and equitable partition. A helpful observation in this regard is the following result.

Proposition 5.1. *For a graph G with coarsest equitable partition π, the associated Monad LEP Set exists and is unique.*

Proof. We prove the existence of a Monad LEP Set $\mathscr{L}$ by construction. First, given the equitable partition $\pi = \{V_1, V_2, \ldots, V_k\}$, let $V_i \sim V_j$ mean that V_i and V_j are consistently connected to each other, and $V_i \nsim V_j$ mean that they are not; it is easily shown that both $\sim$ and $\nsim$ relations are symmetric. We construct $\mathscr{L}$ as follows: for $V_i, V_j \in \pi$, we place V_i in the same element as V_j if $V_i \not\sim V_j$. More formally, V_i and V_j are in the same element $\bar{\pi}_i \in \mathscr{L}$ iff there exist $k_1, k_2, \ldots, k_t$ such that

$$V_i \not\sim V_{k_1} \not\sim V_{k_2} \not\sim \cdots \not\sim V_{k_t} \not\sim V_j.$$

Note that, since $\not\sim$ is not transitive, it may be the case that $V_i \sim V_j$; indeed, the intermediate sequence of equitable partition elements $V_{k_1}, V_{k_2}, \ldots, V_{k_t}$ is only necessary if $V_i \sim V_j$. Further, we note that $\mathscr{L} = \{\bar{\pi}_1, \ldots, \bar{\pi}_r\}$ is a partition of π, since each $V_n \in \pi$ will be contained in exactly one $\bar{\pi}_m \in \mathscr{L}$.

We will now prove that $\mathscr{L}$ is a Monad LEP Set. Given some complete set of LEPs $\hat{\mathscr{L}} = \{\hat{\pi}_1, ..., \hat{\pi}_s\}$, we wish to show that for all $\hat{\pi}_i \in \hat{\mathscr{L}}$,

$$\hat{\pi}_i = \bigcup_{j \in J_i} \bar{\pi}_j \quad \text{for some} \quad J_i \subseteq \{1, \ldots, r\}.$$

Note that each $\hat{\pi}_i$ and $\bar{\pi}_j$ is simply a set of elements V_α from π. We define $J_i = \{j \mid V_\alpha \in \bar{\pi}_j \text{ for some } V_\alpha \in \hat{\pi}_i\}$ and proceed with a proof of equality.

($\subseteq$): Given $V_\alpha \in \hat{\pi}_i$, there exists some $\bar{\pi}_k \in \mathscr{L}$ such that $V_\alpha \in \bar{\pi}_k$, since $\mathscr{L}$ is a partition of π. Since we defined J_i to include all such indices k, and since $\bar{\pi}_k \subseteq \bigcup_{j \in J_i} \bar{\pi}_j$, we have $V_\alpha \in \bigcup_{j \in J_i} \bar{\pi}_j$.

($\supseteq$): Given $V_\alpha \in \bigcup_{j \in J_i} \bar{\pi}_j$, there exists some $k \in J_i$ such that $V_\alpha \in \bar{\pi}_k$. By the definition of J_i, there also exists some $V_\beta \in \bar{\pi}_k$ such that $V_\beta \in \hat{\pi}_i$ for some $\hat{\pi}_i \in \hat{\mathscr{L}}$. Further, by our construction of $\mathscr{L}$, there exist some $k_1, k_2, \ldots, k_t$ such that

$$V_\alpha \not\sim V_{k_1} \not\sim V_{k_2} \not\sim \cdots \not\sim V_{k_t} \not\sim V_\beta.$$

Since $\hat{\pi}_i$ is an LEP, and $V_\beta \in \hat{\pi}_i$, then $V_\beta \sim V_\gamma$ for all $V_\gamma \in \pi - \hat{\pi}_i$ (see Definition 2.3). Since $V_{k_t} \not\sim V_\beta$, it follows that $V_{k_t} \in \hat{\pi}_i$. A simple induction argument is sufficient to show also that $V_\alpha \in \hat{\pi}_i$. Hence, $\hat{\pi}_i = \bigcup_{j \in J_i} \bar{\pi}_j$, so $\mathscr{L}$ is a Monad LEP Set.

Finally, we prove that the Monad LEP Set is unique. Suppose now that there exists some other Monad LEP Set $\hat{\mathscr{L}} = \{\hat{\pi}_1, \ldots, \hat{\pi}_s\}$. Since $\hat{\pi}_k = \bigcup_{j \in J_k} \bar{\pi}_j$ for some $J_k \subseteq \{1, \ldots, r\}$, it must be the case that $\hat{\pi}_k \supseteq \bar{\pi}_\ell$ for some $\ell \in J_k$. Similarly, since $\hat{\mathscr{L}}$ is also a Monad LEP Set, $\bar{\pi}_\ell = \bigcup_{i \in I_\ell} \hat{\pi}_i$ for some $I_\ell \subseteq \{1, \ldots, s\}$. Since $\hat{\pi}_k$ shares elements with $\bar{\pi}_\ell$, then $k \in I_\ell$, so $\bar{\pi}_\ell \supseteq \hat{\pi}_k$. Thus, it must be the case that $\bar{\pi}_\ell = \hat{\pi}_k$. Since $\hat{\pi}_k$ was arbitrary, we see that $\hat{\mathscr{L}} = \mathscr{L}$. Hence, the Monad LEP Set $\mathscr{L}$ is unique. □

Since a graph's coarsest equitable partition and Monad LEP Set exist and are unique, the LEParD algorithm can be used to improve the performance of standard eigenvalue finders, at least in certain cases (see Example 5.1). To compute the Monad LEP Set we use Algorithm 1 where, for a graph $G = (V, E, \omega)$, we let $N(v) = \{w \in V : e_{wv} \in$

Algorithm 1: Local Equitable Partition Finder.

Input : $G = (V, E, \omega)$
 π, the coarsest equitable partition of G
Output: The Monad LEP Set $\mathscr{L}$

```
LepGraph ← dict()
foreach V_i ∈ π do
    add (V_i, {}) to LepGraph
end
foreach V_i ∈ π do
    M ← ⋂_{v∈V_i} N(v) /* find neighbors common to all vertices in V_i */
    foreach v ∈ V_i do
        /* for each neighbor of v not shared by all vertices in V_i */
        foreach w ∈ N(v) \ M do
            /* partition elements containing v and w are not consistently connected with
               one another, so group them in the same LEP */
            if π(v) ≠ π(w) then
                LepGraph[π(v)].add(π(w))
                LepGraph[π(w)].add(π(v))
            end
        end
    end
end
LEPs ← GetConnectedComponents(LepGraph) /* see Procedure 2 */

return LEPs /* This will be the Monad LEP Set */
```

$E\}$ denote the set of *neighbors* of $v \in V$. Additionally, for an equitable partition $\pi = \{V_1, V_2, \ldots, V_k\}$ of G, we let $\pi(v) = V_i$ if $v \in V_i$.

The final step of Algorithm 1 uses Procedure 2 (`GetConnectedComponents`) to organize equitable partition elements into LEPs. This procedure has computational complexity given by the following lemma.

Lemma 5.1. *Procedure 2 has time complexity of order $\mathcal{O}(m+n)$ and spacial complexity of order $\mathcal{O}(n)$, where $m = |E|$ and $n = |V|$.*

Proof. Procedure 2 (`GetConnectedComponents`) is simply a Depth First Search (DFS) with some added constant time operations at each step (adding nodes to the *Component* or adding the *Component* to the list of *ConnectedComponents*). Since it is well known that DFS takes $\mathcal{O}(|E| + |V|)$ operations, we see that Procedure 2 likewise has runtime complexity in $\mathcal{O}(m+n)$.

ConnectedComponents and *Component* may both contain each vertex at most once, so their combined size will be at most proportional to $2n$. Since vertices in $Neighbors$ are never simultaneously in $Visited$, we have $|Neighbors| + |Visited| \leq n$, so their contribution to the algorithm's spatial complexity is at most n. Hence, the total space used is proportional to $3n \in \mathcal{O}(n)$. □

Theorem 5.2. *Algorithm 1 has both computational and spatial complexity of order $\mathcal{O}(m+n)$, where $m = |E|$ and $n = |V|$.*

Procedure 2: GetConnectedComponents(G).

Input : G, a dictionary of ($vertex$, $neighbors$) pairs, where $neighbors$ is a set of vertices connected to $vertex$

Output: $\{\pi_1, \pi_2, \ldots, \pi_k\}$, a partition of the vertices of G into connected components

```
Visited ← set() /* tracks which vertices have already been considered */
ConnectedComponents ← list()
foreach vertex i in G do
    if i is not in Visited then
        Component ← set()
        Neighbors ← list()
        Neighbors.add(i)
        repeat
            j ← Neighbors.pop()
            Component.add(j)
            Visited.add(j)
            /* add j's unvisited neighbors to Neighbors */
            Neighbors.add(G[j] \ Visited)
        until Neighbors is empty
        ConnectedComponents.add(Component)
    end
end
return ConnectedComponents
```

Proof. First, we demonstrate that $N(v)$ and $\pi(v)$ can be found in linear time and accessed thereafter in constant time. $N(v)$ may be represented by a dictionary mapping each vertex to its neighbors; it may be constructed by iterating over all vertices and edges, thus taking linear time with respect to the number of vertices and edges and being in $\mathcal{O}(n+m)$. Similarly, we may obtain $\pi(v)$ by constructing a dictionary to point vertices to their partition elements by iterating over the elements of π, using each vertex once and thereby achieving temporal complexity in $\mathcal{O}(n)$. Accessing $N(v)$ or $\pi(v)$ thereafter will be a constant-time lookup in the dictionary.

These dictionaries will have size proportional to the sizes of their keys and values. In the case of $N(v)$ we have $|keys| + |values| = |V| + |E| = n + m$. In the case of $\pi(v)$, $|keys| + |values| = |V| + |\pi| \leq 2n$, since there are at most n partition elements. Thus, their spacial complexities are in $\mathcal{O}(n+m)$ and $\mathcal{O}(n)$, respectively.

In Algorithm 1 the first loop, on lines 2-4, initializes a dictionary with an entry for each partition element in π, and therefore may take k operations and will use space proportional to k, where $k = |\pi| \leq |V|$. Next, on line 6, we compute M, the set of neighbors common to all $v \in V_i$, by iterating over each edge e_{vw} connecting some vertex $v \in V_i$ to a neighboring vertex $w \in V$. Thus, after being repeated for all $V_i \in \pi$, all edges will have been considered exactly once, so line 6 will contribute m operations to the overall time complexity. M may not contain more vertices than V, so its space is bounded by n. Similarly, the following nested loops on lines 7-8 will result in lines 9-12 being run once for each edge if M is empty (the worst case). Since adding to a set is constant in time and space, lines 9-12 may contribute up to m operations and take space proportional to m. Finally, from Lemma 5.1, `GetConnectedComponents` has time complexity in $\mathcal{O}(n+m)$ and space complexity in $\mathcal{O}(n)$, so it contributes on the order of $\mathcal{O}(n+m)$ operations and takes $\mathcal{O}(n)$ space. Thus, Algorithm 1 takes $k+3m+n \leq 3m+2n \in \mathcal{O}(m+n)$ operations and uses $k+2m+n \leq 2m+2n \in \mathcal{O}(m+n)$ space. □

In the worst case scenario, all equitable partition elements are trivial and thus the divisor matrix is identical to the adjacency matrix. In this case, the LEParD algorithm performs with the same temporal complexity as the traditional algorithm. That is, asymptotically we lose nothing by using the LEParD algorithm to find eigenvalues of a graph (matrix). Thus, though our method cannot guarantee performance improvements in all cases, it may surpass traditional methods when a graph has a nontrivial coarsest equitable partition as is often the case in real-world networks [18]. The latter is considered in the following section.

5.3. An optimized example

In practice, the complexity of the LEParD algorithm will be dominated by finding the eigenvalues of divisor matrices and LEP subgraphs (see Theorem 5.1). To illustrate the potential of the LEParD algorithm, here we consider a family of graphs that are designed to minimize this cost.

Because the LEParD algorithm can find the local eigenvalues of each of the subgraphs G_i individually, the local eigenvalues $\sigma^{\ell}_{\overline{\pi}_i}(G_i)$ can be computed simultaneously for all $i = 1, \ldots, r$. Thus, the runtime of the LEParD algorithm, which finds these eigenvalues in parallel, is determined by the size of the largest divisor graph or LEP. In the interest of minimizing this term, we examine a family of graphs $L_n = (V_n, E_n)$ for which the divisor graph $(L_n)_\pi$ and LEP subgraphs are roughly equal in size and the local divisor graphs have a single vertex.

Example 5.1 *(Layered graphs).* Let $L_n = (V_n, E_n)$ be the family of *layered graphs* with adjacency matrix

$$A(L_n) = \begin{bmatrix} I & J & & \\ J & 0 & \ddots & \\ & \ddots & \ddots & J \\ & & J & 0 \end{bmatrix}$$

where $n \in \mathbb{N}$ is a perfect square and each $J \in \{1\}^{j\times j}$ is the all ones matrix with $j = \sqrt{n}$. The graph $L_n = (V_n, E_n)$ has a coarsest equitable partition $\pi^n = \{V_1^n, V_2^n, \ldots, V_k^n\}$ where the element $V_i^n = \{(i-1)\times\sqrt{n}+1, \ldots, i\times\sqrt{n}\}$ and $k = \sqrt{n}$. Here all elements $V_i^n \in \pi^n$ are consistently connected to one another, so the Monad LEP Set of L_n is simply the set of singletons containing elements of π^n, i.e., $\{\{V_1^n\}, \{V_2^n\}, \ldots, \{V_k^n\}\}$. This equitable partition is indicated in Fig. 3 for the layered graph $L_{49} = (V_{49}, E_{49})$.

A comparison of the time needed to compute the eigenvalues of $L_n = (V_n, E_n)$ is shown in Fig. 4. The methods compared are the Multi-Shift QR algorithm (hereafter

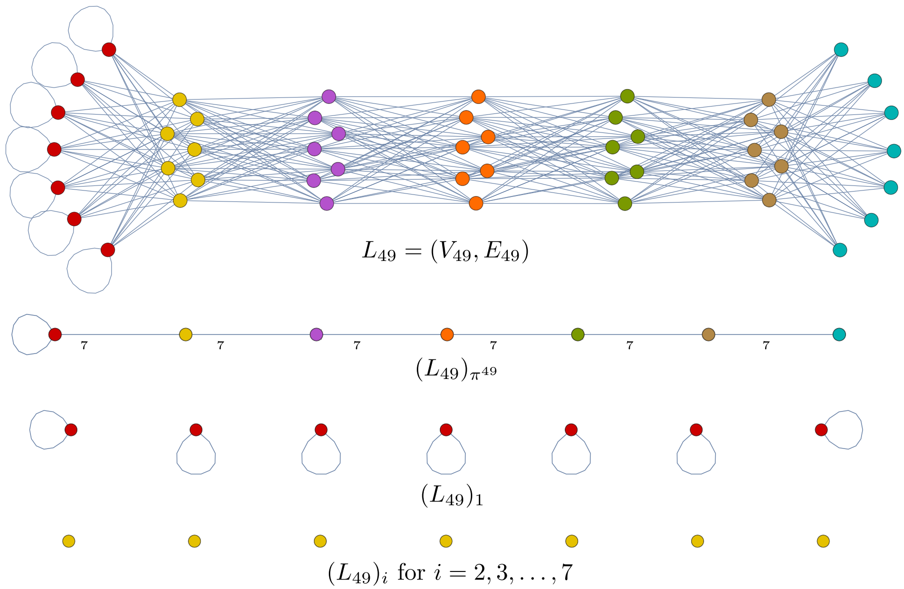


Fig. 3. Top: The layered graph $L_{49} = (V_{49}, E_{49})$ described in Example 5.1 with coarsest equitable partition $\pi^{49} = \{V_1^{49}, V_2^{49}, \ldots, V_7^{49}\}$ shown as the red, yellow, purple, orange, green, brown, and blue vertices; respectively. From left to right, each vertex in a partition element is connected to each of the vertices in the adjacent partition element(s) but to no other vertices, excepting the vertices in the red partition element, which additionally have self-loops. Top Middle: The divisor graph $(L_{49})_{\pi^{49}}$ of L_{49}. Bottom Middle and Bottom: The subgraphs $(L_{49})_1$ and $(L_{49})_i$ for $i = 2, 3, \ldots, 7$ are shown, respectively.

QR algorithm) and the LEParD algorithm using QR as its eigenvalue finder. These are shown in black and yellow, respectively.

In Fig. 4, at approximately $n = 1000$ the LEParD algorithm becomes computationally more efficient than the QR method. The slope of the red and green regression lines are approximately 2.996 and 1.896, respectively, suggesting that the approximate computational complexity of the QR method and LEParD algorithm is $\mathcal{O}(n^{2.996})$ and $\mathcal{O}(n^{1.896})$ respectively (see Proposition 5.2 and the paragraph that follows). We note that the LEParD algorithm's runtime may be further improved via multithreading in a faster language (whereas the current implementation uses multiprocessing in Python).

This example suggests that when the coarsest equitable partition of a graph is sufficiently nontrivial, the LEParD algorithm can be used to compute eigenvalues faster than can be done using standard methods.

To establish the computational complexity seen in Fig. 4 for the layer graphs $L_n = (V_n, E_n)$ we give the following proposition.

Corollary 5.2. *Using an eigenvalue finder of order $\mathcal{O}(a^b)$ the*
(i) LEParD algorithm on $L_n = (V_n, E_n)$ has order $\mathcal{O}(n^{3/2}\log(n) + n^{\frac{b+1}{2}})$; and
(ii) the Parallelized LEParD algorithm on $L_n = (V_n, E_n)$ has order $\mathcal{O}(n^{3/2}\log(n) + n^{\frac{b}{2}})$.

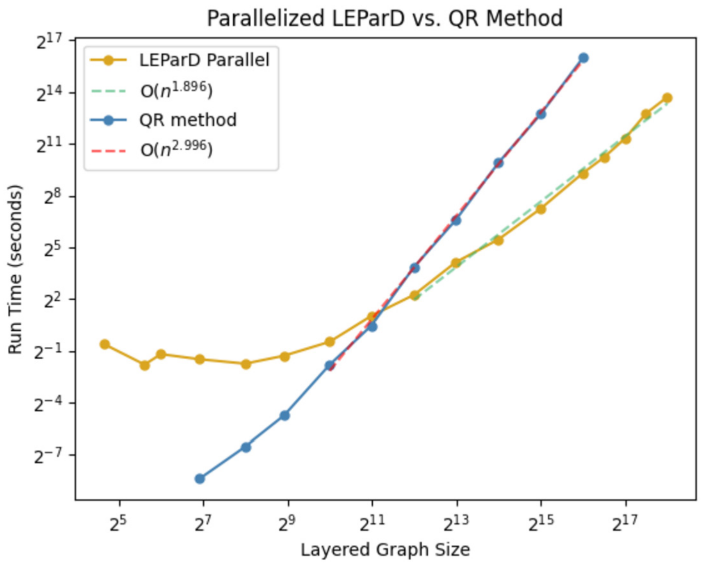


Fig. 4. The time needed to run the QR algorithm to find the eigenvalues of the layered graph $L_n = (V_n, E_n)$ is compared to the time needed to do the same using the LEParD algorithm. These are shown in blue and yellow, respectively, where the horizontal axis gives the size n of the graph and vertical axis, the time, each on a logarithmic scale.

Proof. For the graph L_n the number of edges $m = (\sqrt{n} - 1)|J| + \sqrt{n} = n^{3/2} - n + \sqrt{n}$ where $|J|$ is the number of entries in the matrix $J \in \{1\}^{\sqrt{n}\times\sqrt{n}}$, and the second $\sqrt{n}$ is the number of self-loops. The number of elements in the graph's coarsest equitable partition is $k = \sqrt{n}$ where each element contains $n_i = \sqrt{n}$ vertices. Here the Monad LEP Set has size $r = \sqrt{n}$, since each equitable partition element is preserved as a singleton LEP $\{V_i^n\}$.

From Equation (9), the LEParD algorithm has complexity

$$\mathcal{O}\Big(\left(n^{3/2} - n + \sqrt{n}\right) \log(n) + \left(\sqrt{n}\right)^b + n + \sum_{i=1}^{\sqrt{n}} \left(\sqrt{n}\right)^b \Big) = \mathcal{O}\left(n^{3/2}\log(n) + n^{\frac{b}{2}} + \sqrt{n}(n^{\frac{b}{2}})\right) = \mathcal{O}\left(n^{3/2}\log(n) + n^{\frac{b+1}{2}}\right).$$

From Equation (10), the Parallelized LEParD algorithm has complexity

$$\mathcal{O}\Big(\left(n^{3/2} - n + \sqrt{n}\right) \log(n) + \left(\sqrt{n}\right)^b + n + \left(\sqrt{n}\right)^b \Big) = \mathcal{O}\left(n^{3/2}\log(n) + 2n^{\frac{b}{2}} + n\right) = \mathcal{O}\left(n^{3/2}\log(n) + n^{\frac{b}{2}}\right). \qquad \square$$

Because $b = 3$ for the QR method, the performance of the LEParD algorithm using this as an eigenvalue finder has an approximate order of $\mathcal{O}(n^2)$. The Parallelized LEParD algorithm using the QR method has an order of $\mathcal{O}(n^{3/2}\log(n))$. The performance advantage of the Parallelized LEParD algorithm (using QR) when compared to the standard QR method is shown in Fig. 4.

6. Conclusion

In this paper we introduce the notion of a complete equitable decomposition of a graph, which allows us to decompose a graph into smaller graphs relative to an equitable partition, while maintaining its spectrum. However, to do so we assume that the adjacency matrix of the graph is Hermitian. This is necessary for the proof we give in Section 4 to hold. Currently, we are working to determine whether our result holds for the much more general class of directed graphs. If this can be shown it may be possible to modify the LEParD algorithm to much more quickly find eigenvalues of potentially any real-world data set, where nontrivial equitable partitions are expected. This will be explored in future publications.

With regard to the LEParD algorithm it is unknown to what extent this algorithm is numerically stable or how its spatial complexity scales as this method needs to store matrices in a way not needed by standard algorithms. Last, it is also possible that a complete equitable decomposition can be carried out via a similarity transform, i.e. a transform that extends the classical theory described in Theorems 3.1 and 3.2. Currently this is unknown.

Declaration of competing interest

None declared.

Data availability

No data was used for the research described in the article.

Acknowledgement

The authors thank the anonymous referee for their thorough and insightful comments and suggestions, which have very much improved this paper.

B.Z.W. was partially supported by the NSF grant #2205837 in Applied Mathematics.

References

[1] O. Bastert, Computing equitable partitions of graphs, Match 40 (1999) 265–272.
[2] A. Brouwer, W. Haemers, Spectra of Graphs, Universitext, Springer, New York, 2012.
[3] W. Barret, A. Francis, B. Webb, Equitable decompositions of graphs with symmetry, Linear Algebra Appl. 513 (2017) 409–434.
[4] K.L. Calvert, M.B. Doar, E.W. Zegura, Modeling Internet topology, IEEE Commun. Mag. 35 (1997) 160–163.

[5] P. Carrington, J. Scott, S. Wasserman, Models and Methods in Social Network Analysis, Cambridge University Press, 2005.
[6] C. Chen, W. Ye, Y. Zuo, C. Zheng, S. Ping Ong, Graph networks as a universal machine learning framework for molecules and crystals, Chem. Mater. 31 (9) (2019) 3564–3572.
[7] D. Cvetkovic, P. Rowlinson, S. Simic, An Introduction to the Theory of Graph Spectra, London Mathematical Society Student Texts, Cambridge University Press, 2009.
[8] J. Demmel, I. Dumitriu, O. Holtz, Fast linear algebra is stable, Numer. Math. 108 (1) (2007) 59–91.
[9] A. Francis, D. Smith, D. Sorensen, B. Webb, Extensions and applications for graphs with symmetries, Linear Algebra Appl. 532 (2017) 432–462.
[10] A. Francis, D. Smith, B. Webb, General equitable decompositions of graphs with symmetries, Linear Algebra Appl. 577 (2018) 287–316.
[11] G. Frederickson, Data structures for on-line updating of minimum spanning trees with applications, SIAM J. Comput. 14 (4) (1985) 781–798.
[12] C. Godsil, G.F. Royle, Algebraic Graph Theory, Graduate Texts in Mathematics, Springer, New York, 2001.
[13] E. Hannesson, J. Sellers, E. Walker, B. Webb, Network specialization: A topological mechanism for the emergence of cluster synchronization, Physica A: Statist. Mech. Appl. (ISSN 0378-4371) 600 (2022).
[14] D. Lund, J. Drapeau, B. Webb, Fourier decompositions of graphs with symmetries and equitable partitions, Linear Algebra Appl. 627 (2021) 199–226.
[15] A. Khazaee, A. Ebrahimzadeh, A. Babajani-Feremi, Identifying patients with Alzheimer's disease using resting-state fMRI and graph theory, Clin. Neurophysiol. 126 (11) (2015) 2132–2141.
[16] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, The MIT Press, Cambridge, Massachusetts, 2009.
[17] Y. Lo, S. Rensi, W. Torng, R. Altman, Machine learning in chemoinformatics and drug discovery, Drug Discov. Today 23 (2018) 1538–1546.
[18] B. MacArthur, R. Sanchez-Garcia, J. Anderson, Symmetry in complex networks, Discrete Appl. Math. 156 (18) (2008) 3525–3531.
[19] K. Murphy, Machine Learning: A Probabilistic Perspective, The MIT Press, Cambridge, Massachusetts, 2012.
[20] M. Newman, Networks: An Introduction, Oxford University Press, 2018.
[21] N. O'Clery, Y. Yuan, G. Stan, M. Barahona, Observability and coarse graining of consensus dynamics through the external equitable partition, Phys. Rev. E 88 (4) (2013).
[22] V. Pan, Z. Chen, The Complexity of the Matrix Eigenproblem, ACM Press, New York, NY, 1999.
[23] M. Schaub, N. O'Clery, Y. Billeh, J. Delvenne, R. Lambiotte, M. Barahona, Graph partitions and cluster synchronization in networks of oscillators, Chaos 26 (9) (2016).
[24] E.R. Scheinerman, in: D.H. Ullman (Ed.), Fractional Graph Theory, 1st ed., Wiley, New York, 1997.
[25] S. Shrinivas, S. Vetrivel, N. Elango, Applications of graph theory in computer science an overview, Int. J. Eng. Sci. Technol. 2 (9) (2010) 4610–4621.
[26] A. Siddique, L. Pecora, J. Hart, F. Sorrentino, Symmetry- and input-cluster synchronization in networks, Phys. Rev. E 97 (4) (2018).
[27] A. King, D. Smith, B. Webb, Minimal specialization: coevolution of network structure and dynamics, arXiv:2311.15084, 2023.