THE SERIATION PROBLEM IN THE PRESENCE OF A DOUBLE FIEDLER VALUE

A. CONCAS*, C. FENU*, G. RODRIGUEZ*, AND R. VANDEBRIL†

Abstract. Seriation is a problem consisting of seeking the best enumeration order of a set of units whose interrelationship is described by a bipartite graph. An algorithm for spectral seriation based on the use of the Fiedler vector of the Laplacian matrix associated to the problem was developed by Atkins et al., under the assumption that the Fiedler value is simple. In this paper, we analyze the case in which the Fiedler value of the Laplacian is not simple, discuss its effect on the set of the admissible solutions, and study possible approaches to actually perform the computation. Examples and numerical experiments illustrate the effectiveness of the proposed methods.

1. Introduction. The *seriation* problem consists of finding the best enumeration order of a set of units according to a particular similarity rule. Such order can be chronological, or any sequential structure which characterizes the data. When the ordering is chronological, seriation is employed when absolute dating methods cannot be used. Indeed, it concerns relative dating of objects or events. This means that it determines a sequence for the units which can be read in both directions. The problem here considered is also denoted by *linear* seriation, in contrast to *circular* seriation [23, 1, 38, 9, 16, 7], which aims at finding a circular order for a set of units, reflecting their dissimilarity.

Seriation arose for the first time in the context of archaeological studies, where it is typically formulated as the problem of dating excavation sites on the basis of the findings discovered inside them, and of determining their relative chronology, i.e., a dating which indicates if a given site is chronologically preceding or subsequent to another. The first systematic formalization of the seriation problem was made by the English Egyptologist Petrie in 1899 [35].

The idea of seriation successively appeared in various other applicative contexts [6, 14, 22, 32]; see Concas et al. [10] for an overview. In genomics, seriation finds application in the *de novo* genome sequencing. In this case, from a randomly oversampled DNA strand (the so-called *reads*) the whole sequence is reconstructed. Oversampling is necessary to increase the probability of all parts being covered. The reads which overlap are then considered as similar and their ordering is obtained by placing similar reads close to each other. In mathematics, the seriation problem is strictly connected to the reordering of sparse matrices in order to reduce their so-called envelope size [3], and to the combinatorial 2-SUM problem, a quadratic minimization problem over permutations [20]. The interested reader can refer to [39, 15, 31, 21] for a recent discussion.

Seriation data are usually given in terms of a matrix of size $n \times m$, called the *data matrix*, whose row and/or column indices represent the elements to be ordered. In archaeology, the rows of the data matrix correspond to the units (e.g., the sites) and the columns represent the types of the archaeological findings detected in the units. The seriation data matrix has been referred in [36] either as incidence or abundance matrix. Incidence matrices are associated to binary data, i.e., the (i, j) entry of the

[0]Version November 16, 2022

*Dipartimento di Matematica e Informatica, Università di Cagliari, viale Merello 92, 09123 Cagliari, Italy. E-mail: anna.concas@unica.it, kate.fenu@unica.it, rodriguez@unica.it. Research supported in part by INdAM-GNCS.

†Department of Computer Science, KU Leuven, Celestijnenlaan 200A, 3001 Heverlee, Leuven, Belgium. E-mail: raf.vandebril@cs.kuleuven.be

matrix is 1 if the type j is associated in the unit i, and 0 otherwise. In abundance matrices, each element represents the number of objects corresponding to a certain type, or its percentage. Following the usual terminology of complex networks theory, we will use the term *adjacency matrix* to denote the binary representation of the seriation data. More details can be found in [10]. We mention the fact that in the quadratic assignment problem literature (see, e.g., [34]) the adjacency matrix is often associated to a weighted graph. In the following, we will use the term mainly for unweighted graphs.

Solving the seriation problem amounts to ordering the rows and columns of the data matrix so that the largest nonzero entries are close to the diagonal. Various algorithms for handling seriation data and solving the problem have been implemented in software libraries; see [10] for an overview. A spectral algorithm for the solution of the seriation problem was considered by Atkins et al. [2], and an optimized Matlab implementation has recently been proposed by Concas et al. [10]. Each solution is obtained by solving a particular optimization problem. The spectral algorithm is based on the use of the Fiedler vector of the Laplacian matrix associated with the problem, and describes the set of solutions in terms of a data structure known as a PQ-tree.

In this paper, we discuss the implications of the presence of a multiple Fiedler value on the number of admissible permutations, which have been disregarded up to now. This issue occurs, for example, in the case of circular seriation [23, 1, 38]. Our interest is mainly for the case of multiplicity two, for which we illustrate the effects on the set of solutions. When there is no uniqueness in the choice of the Fiedler vector, sorting the entries of all the admissible vectors does not necessarily lead to all possible index permutations, which equals the factorial of the number of units. We experimentally observed that, in these "noisy" situations, there are constraints in the solution space that limit the number of permutations produced from sorting a Fiedler vector. This number appears to be related to the structure of the eigenspace, not simply to the multiplicity of the Fiedler value.

The impact of a Fiedler value having multiplicity larger than one on applications is that the spectral method from [2], probably one of the best available algorithms for seriation, is not able to produce the whole set of solutions. In the software described in [10] this situation is recognized, to warn the user that the computed solution will not be exhaustive. Here, after recalling some basic mathematical notions, we study three particular test networks that exhibit a double Fiedler value, and propose two algorithms to determine the full set of permutations that solve the problem. Numerical experiments will show that only one of the algorithms is trustable. The handling of a Fiedler value with a multiplicity larger than two is still an open problem and will be the object of future research.

The plan of the paper is the following. Section 2 reviews the necessary mathematical background and sets up the terminology to be used in the rest of the paper. The special case of a multiple Fiedler value is discussed in Section 3. In Section 4, we extensively analyze three example networks whose Laplacian admits a double Fiedler value, showing the consequences on the set of solutions of the seriation problem. Section 5 describes two practical algorithms for computing the admissible solutions, and Section 6 reports some numerical results. Finally, Section 7 contains concluding remarks.

2. Mathematical preliminaries. We recall some basic results from graph theory and seriation. In the following, we will denote vectors by lower case bold letters

and matrices by upper case roman letters; their elements will be represented by lower case letters with one or two indices, respectively.

Let $\mathcal{G}$ be a simple graph with n nodes and let $G \in \mathbb{R}^{n\times n}$ denote its adjacency matrix. If $\mathcal{G}$ is weighted, the entry $g_{i,j}$ of G contains the weight of the edge connecting node i to node j; when two nodes are not connected $g_{i,j} = 0$. A graph is said to be unweighted if the weight is either 0 or 1. The adjacency matrix is irreducible if the graph is connected, it is symmetric if the graph is undirected.

Let G be the adjacency matrix of an undirected connected graph $\mathcal{G}$. The *degree matrix* is the diagonal matrix $D = \mathrm{diag}(d_1, \ldots, d_n)$, such that $d_i = \sum_{j=1}^{n} g_{i,j}$, that is, the ith diagonal element is the sum of the weights of all the edges starting from node i. In the case of an unweighted graph, d_i is simply the number of nodes connected to node i. The (unnormalized) *graph Laplacian* associated to $\mathcal{G}$ is the symmetric positive semidefinite matrix $L_G = D - G$. A well-known property of the graph Laplacian is that it always admits $\lambda_1 = 0$ as an eigenvalue, with associated eigenvector $\mathbf{1} = (1, \ldots, 1)^T \in \mathbb{R}^n$, and all the remaining eigenvalues are non-negative, that is,

$$\lambda_1 = 0 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$

The *Fiedler value*, or the *algebraic connectivity*, of the graph $\mathcal{G}$ is the second smallest eigenvalue of L_G; its eigenvector, which is orthogonal to $\mathbf{1}$, is called the *Fiedler vector* [17, 18, 19]. The use of the Fiedler vector is a well-known method for partitioning a graph. The interested reader can refer to the review [13] and references therein. The Fiedler vector is also used in the computation of the graph robustness [24, 25].

By the Courant-Fisher Theorem, the Fiedler vector may also be defined as

$$\min_{\mathbf{x}^T\mathbf{1}=0,\ \mathbf{x}^T\mathbf{x}=1} \mathbf{x}^T L_G \mathbf{x}. \tag{2.1}$$

The seriation problem is generally formulated in terms of a *bipartite graph* $\mathcal{B}$, whose vertices can be divided into two disjoint sets U and V, containing n and m nodes, such that every edge connects a node in U to one in V. In the archaeological setting, U and V represent the excavations sites (units) and the findings (types), respectively. The adjacency matrix $B \in \mathbb{R}^{n\times m}$ associated to the seriation problem describes the connections in the bipartite graph $\mathcal{B}$: $b_{i,j} = 1$ if unit i contains type j, and $b_{i,j} = 0$ otherwise. A matrix whose entries are either 0 or 1, such that the 1's in each column occur consecutively is often called a Petrie matrix in the literature; see [27, 28].

The approach described in [5, 40] for solving the seriation problem is based on the construction of a symmetric *similarity matrix* S, whose element $s_{i,j}$ expresses the resemblance of the units $i, j \in U$ in terms of the types they contain. The similarity matrix is often defined as $S = BB^T$, where B is the adjacency matrix of the bipartite graph introduced above, so that $s_{i,j}$ denotes the number of types shared between units i and j. A slightly different definition of the similarity matrix was given by Robinson in [40]. A permutation for the rows and columns of S that clusters the largest entries close to the main diagonal, when applied to the elements of U, brings the units which are similar according to the chosen rule in close positions. Such permutation constitutes a solution of the seriation problem and it is not uniquely determined. The solution is generally identified with a permutation matrix P.

When the similarity matrix is correctly permuted, so that larger entries are close to the main diagonal and off-diagonal entries are nonincreasingly ordered moving away

from the diagonal, it is said to be in *Robinson's form* or in R-form; it is also called an R-matrix. A similarity matrix is pre-R if and only if there exists a simultaneous permutation of its rows and columns which takes it to Robinson's form [8, 29, 30, 37, 41]. This case corresponds to a well-posed seriation problem, but for some *non-perfect* data sets such permutation does not exist. When this happens, one may be interested in obtaining an approximate solution even for a non pre-R matrix.

We now briefly review the spectral algorithm for the seriation problem introduced in [2]. Starting from a pre-R matrix, it constructs a PQ-tree [4] describing the set of all the row and column permutations that lead to an R-matrix. The algorithm is also able to produce an approximate solution when a non pre-R matrix is given as input.

Let a similarity matrix S for a graph $\mathcal{G}$ be given. Its entries $s_{i,j}$ express the desire that the units i and j are close to each other in the sought sequence. Introducing the notation $i \preccurlyeq j$ to indicate that unit i precedes j in an ordering, the aim of the algorithm is to find all index permutations such that

$$i \preccurlyeq j \preccurlyeq k \quad \Longleftrightarrow \quad s_{i,j} \geq s_{i,k} \quad \text{and} \quad s_{j,k} \geq s_{i,k}. \tag{2.2}$$

We set $S = BB^T$, where B is the adjacency matrix of the bipartite graph associated to the seriation problem.

The algorithm proposed in [2] solves the constrained optimization problem

$$\begin{aligned} &\text{minimize} \quad h(\mathbf{x}) = \frac{1}{2}\sum_{i,j=1}^{n} s_{i,j}(x_i - x_j)^2, \\ &\text{subject to} \quad \sum_i x_i = 0 \quad \text{and} \quad \sum_i x_i^2 = 1, \end{aligned} \tag{2.3}$$

instead of (2.1). The computed minimizing vector $\mathbf{x}_{\min}$ is then sorted, either nonincreasingly or nondecreasingly, and the corresponding sorting permutation is applied to the units in U. Such permutation realizes (2.2) and solves the problem. It may be non unique in the presence of repeated entries in the vector $\mathbf{x}_{\min}$. An optimized version of the algorithm is available in [10].

In [2], the authors prove that when S is pre-R the algorithm produces all the index permutations that solve the seriation problem. If a real data set is inconsistent, i.e., it does not produce a pre-R similarity matrix, sorting the entries of the Fiedler vector generates orderings that try to bring highly correlated elements close to each other. Thus, the algorithm furnishes approximate solutions to the seriation problem. We refer to such orderings as *admissible permutations*.

In the well-posed case, the above algorithm produces the whole set of solutions of the seriation problem under the assumption that the Fiedler value is a simple eigenvalue of the Laplacian L_G. When the Fiedler value has multiplicity q larger than one, solving the optimization problem (2.3) only yields one vector $\mathbf{x}_{\min}$, while the eigenspace containing the Fiedler vectors has dimension q and its elements may lead to a number of admissible permutations much larger than those corresponding to $\mathbf{x}_{\min}$. Therefore, the spectral algorithm is not able to reproduce all the solutions in the presence of a multiple Fiedler value.

3. Multiple Fiedler values in seriation. In this section we analyze the case of the presence of a multiple Fiedler value and its effect on the spectral algorithm discussed above.

Let us assume that the Fiedler value has multiplicity k, and let $\mathbf{q}_1, \ldots, \mathbf{q}_k$ be an orthonormal basis of the corresponding eigenspace $\mathcal{F}$. For each $\mathbf{x} \in \mathcal{F}$, there is a

vector $\tilde{\mathbf{y}} = (y_1, \ldots, y_k)^T$ such that

$$\mathbf{x} = Q_k \tilde{\mathbf{y}},$$

where $Q_k = [\mathbf{q}_1, \ldots, \mathbf{q}_k]$. We remind the reader that a solution to the seriation problem is determined by sorting the vector $\mathbf{x}$ either nonincreasingly or nondecreasingly.

When $k = 1$ there is in general only one permutation which solves the problem, together with its reverse. There are multiple solutions if the eigenvector $\mathbf{x}$ has ℓ equal components. In this case, there will be $\ell!$ solutions.

When $k > 1$, after extending Q_k to a square orthogonal matrix Q, we can write $\mathbf{x} = Q\mathbf{y}$, with

$$\mathbf{y} = \begin{bmatrix} \tilde{\mathbf{y}} \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^n.$$

Although it is clear that only the first k entries are relevant in determining $\mathbf{x}$, it is not trivial to understand how many permutations are allowed to sort $\mathbf{x}$ when the components of $\tilde{\mathbf{y}}$ vary.

Let us analyze the situation where $\mathbf{q}_i = \mathbf{e}_i$, the vectors of the canonical basis in $\mathbb{R}^n$, $i = 1, \ldots, k$, so that we may set $Q = I$, Even in the case $\mathbf{x} = \mathbf{y}$, the conclusion is not trivial. If the first k components of $\mathbf{y}$ are different from zero and distinct, then the indices associated to the last $n - k$ zero components admit $(n - k)!$ equivalent permutations. We can consider such indices in the whole vector $\mathbf{y}$ as grouped in a unique "vector" index, as the corresponding components all share the same position in each possible sorting. Under this assumption, the number of different orderings for $\mathbf{y}$ is $(k + 1)!$. Substituting to the vector index all its possible permutations, the number of admissible solutions grows to

$$(k + 1)!(n - k)!. \tag{3.1}$$

If there are groups of equal components in $\tilde{\mathbf{y}}$, this number is going to increase accordingly. In the general case, that is when $Q \neq I$, the number of admissible permutations depends upon the structure of the Fiedler vectors.

Concas et al. [10] pointed out that non pre-R matrices can lead to Laplacian matrices whose Fiedler value is not simple and conjectured, through the following simple example, that the number of permutations (3.1) may be incorrect.

Let us consider the seriation problem described by the bipartite graph depicted in Figure 4.3 (left). The relationship between nodes on the left (units) and nodes on the right (types) is represented by edges. The adjacency matrix of this graph and the resulting similarity matrix are, respectively

$$B = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{and} \qquad S = BB^T = \begin{bmatrix} 2 & 1 & 0 & 0 & 1 \\ 1 & 2 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 2 & 1 \\ 1 & 0 & 0 & 1 & 2 \end{bmatrix}.$$

Note that S can be seen as the adjacency matrix of the graph shown in Figure 4.3 (right).

A solution to the seriation problem does not exist in this case, since the associated graph describes a *cycle*: each unit is similar to surrounding units and the two extremal units are similar to each other. This leads to a non pre-R similarity matrix. As shown

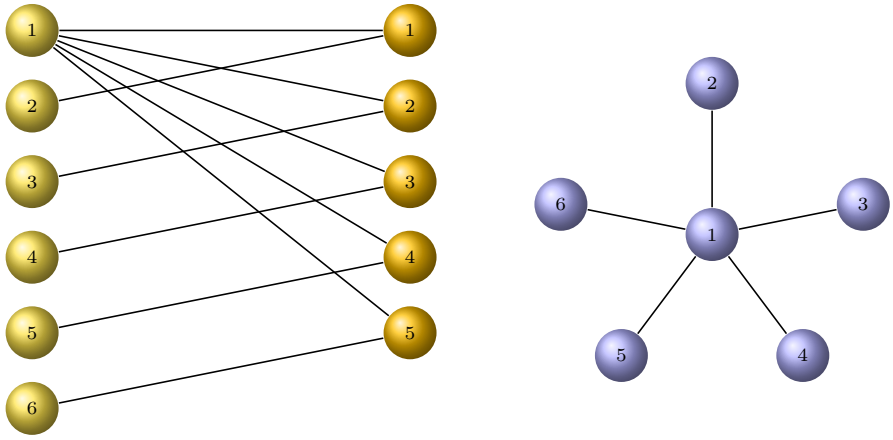

Fig. 4.1: The bipartite graph associated with the data matrix B (4.1) with $n = 6$ (left) which leads to the star graph $\mathcal{S}_6$ (right).

in [10], the Fiedler value of the Laplacian $L_S = D - S$ has multiplicity 2, so each vector belonging to the *Fiedler plane* can be sorted to obtain the admissible permutations of the units. In the same paper, the authors considered a randomized approximated approach, which will be discussed in Section 5, to determine such permutations. They found only 5 admissible permutations, much less than the number $(k+1)!(n-k)! = (2+1)!(5-2)! = 36$ determined by (3.1).

In the following, we will show that this estimate for the number of admissible permutations is wrong. Nevertheless, we will confirm the fact that when a Fiedler value is multiple some constraints are imposed on the admissible permutations of the units. In particular, we will show that their number does not only depend on the multiplicity of the Fielder value, but also on the structure of the underlying bipartite graph.

In the following, we often focus on the number of permutations found. Referring to such a number is significant only to show that, in the cases analyzed, the number of admissible solutions is always smaller than the forecast given by (3.1). We stress the fact that solving the seriation problem consists of listing all the admissible permutations of the nodes. Any theoretical analysis or numerical algorithm must be able to produce such result.

4. Three case studies. In this section, to gain insight into the behavior of other similar examples, we analyze three different graphs whose Laplacian exhibits a double Fiedler value: the modified star graph, the cycle graph, and the generalized Petersen graph.

4.1. The modified star graph. Consider the bipartite graph represented in Figure 4.1 (left) whose associated data matrix is

$$B = \left[\begin{array}{c} \mathbf{1}_{n-1}^T \\ \hline I_{n-1} \end{array}\right] \in \mathbb{R}^{n\times(n-1)}, \tag{4.1}$$

where $\mathbf{1}_k = (1,\dots,1)^T \in \mathbb{R}^k$, and I_k denotes the identity matrix of size k. As already stated, $b_{i,j} = 1$ indicates that unit i contains objects of type j.

The resulting similarity and Laplacian matrices are given by

$$S = BB^T = \left[\begin{array}{c|c} n-1 & \mathbf{1}_{n-1}^T \\ \hline \mathbf{1}_{n-1} & I_{n-1} \end{array}\right], \qquad L_S = D - S = \left[\begin{array}{c|c} n-1 & -\mathbf{1}_{n-1}^T \\ \hline -\mathbf{1}_{n-1} & I_{n-1} \end{array}\right], \tag{4.2}$$

where $D = \mathrm{diag}(d_1, \ldots, d_n)$, $d_i = \sum_{j=1}^n s_{i,j}$, is the degree matrix associated to S.

The matrix S can be interpreted as the adjacency matrix of a star graph; see Figure 4.1 (right). A star graph $\mathcal{S}_n$ is a connected graph with n vertices and $n-1$ edges, where one vertex, the *center* of the star, has degree $n-1$ and the other $n-1$ vertices have degree 1. It is a special case of a complete bipartite graph in which one set has one vertex and the other set contains the remaining $n-1$ vertices.

Both the Laplacian and the similarity matrix (4.2) are arrowhead matrices, that is, real symmetric matrices of the form

$$\begin{bmatrix} \alpha & \mathbf{z}^T \\ \mathbf{z} & \Delta \end{bmatrix} \tag{4.3}$$

where α is a scalar, $\mathbf{z} \in \mathbb{R}^{n-1}$, and $\Delta = \mathrm{diag}(\delta_1, \ldots, \delta_{n-1})$. From the Cauchy interlacing theorem [43] for the eigenvalues of Hermitian matrices, it follows that the sorted eigenvalues $\lambda_1, \ldots, \lambda_n$ of (4.3) interlace the sorted elements δ_i of the diagonal matrix Δ. If $\delta_1 \geq \delta_2 \geq \cdots \geq \delta_{n-1}$ and if the eigenvalues λ_i, $i = 1, \ldots, n$, are sorted accordingly, then the following inequality holds

$$\lambda_1 \geq \delta_1 \geq \lambda_2 \geq \delta_2 \geq \cdots \geq \lambda_{n-1} \geq \delta_{n-1} \geq \lambda_n. \tag{4.4}$$

If $\delta_i = \delta_{i-1}$ for some i, the above inequality implies that δ_i is an eigenvalue of the arrowhead matrix (4.3) considered.

The following theorem identifies the eigenvalues of the Laplacian matrix in the case of a star graph.

THEOREM 4.1. *Let S be the adjacency matrix of a star graph $\mathcal{S}_n$. Then, the spectrum of the Laplacian matrix L_S consists of the three eigenvalues 0, 1, and n, with the second having multiplicity $n-2$.*

Proof. A well known result states that the smallest eigenvalue of the Laplacian is $\lambda_n = 0$. From the Cauchy interlacing theorem applied to the matrix L_S in (4.2), it follows (see (4.4)) that 1 is an eigenvalue with multiplicity $n-2$. Setting $\mathbf{v} = (-(n-1), 1, \ldots, 1)^T \in \mathbb{R}^n$, we see that $L_S\mathbf{v} = n\mathbf{v}$, so that $\lambda_1 = n$. ∎

COROLLARY 4.2. *Let S be an adjacency matrix of a star graph. Then, the Fiedler value has multiplicity $n-2$ and the $n-2$ Fiedler vectors have a null component in the position corresponding to the center node.*

Proof. Without loss of generality we can assume that the first node is the center, which has degree $n-1$. To determine the Fiedler vectors one has to solve the homogeneous linear system $(L_S - I_n)\mathbf{v} = 0$, whose coefficient matrix is

$$L_S - I_n = \begin{bmatrix} n-2 & -\mathbf{1}_{n-1}^T \\ -\mathbf{1}_{n-1} & 0 \end{bmatrix}.$$

The last $n-1$ equations of the system show that the first component of the Fiedler vectors is always 0, while the first equation implies that the sum of the components is 0. ∎

Since we are focusing on the case of a double Fiedler value, let us consider the *modified star graph* obtained by adding some nodes and edges to the data matrix

corresponding to the star graph. We start by considering the bipartite graph on the left of Figure 4.1. We add $n-4$ nodes to the set of the types (nodes 6 and 7, in this case). Then we connect the first of the new nodes to units 2 and 3, the second one to units 3 and 4, and so on. We obtain the bipartite graph in Figure 4.2 (left). The seriation data matrix associated to this graph is

$$B = \left[\begin{array}{c|c} \mathbf{1}_{n-1}^T & \mathbf{0}_{n-4}^T \\ \hline I_{n-1} & B_{n-1,3} \end{array}\right] \in \mathbb{R}^{n\times(2n-5)}, \tag{4.5}$$

where $\mathbf{0}_k \in \mathbb{R}^k$ is a null vector, and $B_{k,\ell} \in \mathbb{R}^{k\times(k-\ell)}$ is the lower bidiagonal matrix whose elements are 1 on the main diagonal and on the sub-diagonal, and zero otherwise.

The resulting similarity matrix is

$$S = \left[\begin{array}{c|cc} n-1 & \multicolumn{2}{c}{\mathbf{1}_{n-1}^T} \\ \hline \mathbf{1}_{n-1} & T_{n-3} & O \\ & O & I_2 \end{array}\right], \tag{4.6}$$

where O denotes a null matrix of suitable size and T_{n-3} is the tridiagonal matrix

$$\begin{bmatrix} 2 & 1 & & & \\ 1 & 3 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 3 & 1 \\ & & & 1 & 2 \end{bmatrix}. \tag{4.7}$$

The similarity matrix S can be seen as the adjacency matrix of the modified star graph in Figure 4.2 (right), which we denote by $\widehat{\mathcal{S}}_6$.

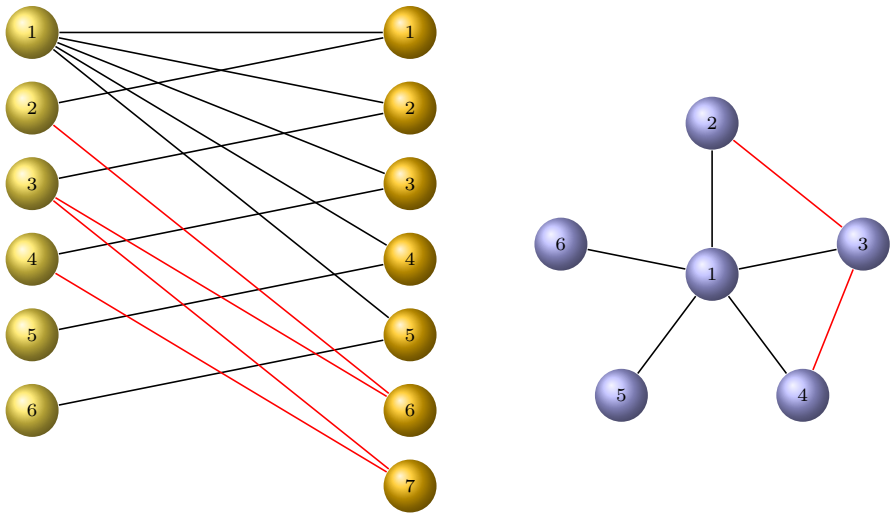

Fig. 4.2: Bipartite graph represented by matrix (4.5) (left) and resulting graph $\widehat{\mathcal{S}}_6$ (right). The edges in red are the added ones.

The Laplacian matrix of S is given by

$$L_S = D - S = \left[\begin{array}{c|cc} n-1 & -\mathbf{1}_{n-1}^T & \\ \hline -\mathbf{1}_{n-1} & \widetilde{T}_{n-3} & O \\ \hline & O & I_2 \end{array}\right], \tag{4.8}$$

where O denotes a null matrix of suitable size and $\widetilde{T}_{n-3}$ is like (4.7), but with the elements in the sub- and in the super-diagonal of opposite sign.

The following theorem explains the behavior of the Fiedler value of the Laplacian matrix in the case of the modified star graph $\widehat{\mathcal{S}}_n$.

THEOREM 4.3. *Let S be the adjacency matrix of a modified star graph $\widehat{\mathcal{S}}_n$. Then, the spectrum of the Laplacian matrix L_S (4.8) contains the three eigenvalues 0, 1, and n, with the second having multiplicity 2, while the remaining $n-4$ eigenvalues are in the interval $(1, 5)$.*

Proof. A direct computation shows that $\lambda_1 = 0$, $\lambda_2 = \lambda_3 = 1$, and $\lambda_n = n$, are eigenvalues of L_S with associated eigenvectors

$$\mathbf{v}_1 = \mathbf{1}_n, \qquad \mathbf{v}_2 = \begin{bmatrix} 0 \\ -\mathbf{1}_{n-2} \\ n-2 \end{bmatrix}, \qquad \mathbf{v}_3 = \begin{bmatrix} 0 \\ -\mathbf{1}_{n-3} \\ n-3 \\ 0 \end{bmatrix}, \qquad \mathbf{v}_n = \begin{bmatrix} 1-n \\ \mathbf{1}_{n-1} \end{bmatrix}.$$

By a simple application of the Gram-Schmidt process, we see that any vector orthogonal to $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_n$ has a null first and last component, like $\mathbf{v}_3$. So, the remaining $n-4$ eigenvectors take the form

$$\mathbf{v}_i = \begin{bmatrix} 0 \\ \tilde{\mathbf{v}} \\ 0 \end{bmatrix}, \qquad i = 4, \ldots, n-1,$$

with $\tilde{\mathbf{v}} \in \mathbb{R}^{n-2}$. Given the expression (4.8) of matrix L_S, any such vector $\tilde{\mathbf{v}}$ is an eigenvector of the principal submatrix

$$\widetilde{L}_S = \left[\begin{array}{c|c} \widetilde{T}_{n-3} & \mathbf{0}_{n-3} \\ \hline \mathbf{0}_{n-3}^T & 1 \end{array}\right].$$

Besides the eigenvalue $\lambda_2 = 1$, the remaining eigenvalues of $\widetilde{L}_S$ are those of $\widetilde{T}_{n-3}$.

Since the eigenvalues of $\widetilde{T}_{n-3}$ are a subset of those of L_S, we will denote them by $(\lambda_3, \ldots, \lambda_{n-1})$. The Gershgorin circle theorems applied to $\widetilde{T}_{n-3}$ yields $1 \leq \lambda_i < 5$, $i = 3, \ldots, n-1$. It is immediate to observe that $\lambda_3 = 1$ with associated eigenvector $\mathbf{1}_{n-3}$. It is a simple eigenvalue because a symmetric tridiagonal matrix with nonzero subdiagonal elements has distinct eigenvalues [33]. This completes the proof. ∎

COROLLARY 4.4. *Let S be the adjacency matrix of a modified star graph $\widehat{\mathcal{S}}_n$. Then, its Fiedler value is equal to 1 and has multiplicity 2.*

In the case of the modified star graph $\widehat{\mathcal{S}}_n$, an orthogonal basis for the eigenspace $\mathcal{F}$ corresponding to the Fiedler value is given by

$$Q_2 = \begin{bmatrix} \mathbf{q_1} & \mathbf{q_2} \end{bmatrix},$$

where $\mathbf{q_1} = \mathbf{v}_3$ and $\mathbf{q_2} = \mathbf{v}_2$. For the sake of simplicity, we do not normalize the two eigenvectors. Letting $\widetilde{\mathbf{y}} = (\alpha, \beta)^T \in \mathbb{R}^2 \setminus \{(0,0)\}$, every $\mathbf{x} \in \mathcal{F}$ can be expressed as

$$\mathbf{x} = Q_2\widetilde{\mathbf{y}} = \begin{bmatrix} 0 & 0 \\ -1 & -1 \\ \vdots & \vdots \\ -1 & -1 \\ n-3 & -1 \\ 0 & n-2 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 0 \\ -\alpha-\beta \\ \vdots \\ -\alpha-\beta \\ (n-3)\alpha-\beta \\ (n-2)\beta \end{bmatrix}. \tag{4.9}$$

The admissible permutations are then related to the possible reorderings of the entries of $\mathbf{x} \in \mathcal{F}$, and these sortings depend on the values of the coefficients α and β. We remark that they cannot be both zero, as $\mathbf{x}$ is an eigenvector.

We let $x_1 = 0$, $x_2 = -\alpha - \beta$, $x_{n-1} = (n-3)\alpha - \beta$, and $x_n = (n-2)\beta$. The relative position of such components is governed by the following inequalities, where we initially consider only strict inequality

$$\begin{cases} x_2 > x_1, & \text{for } \alpha < -\beta, \\ x_{n-1} > x_1, & \text{for } \alpha > \frac{1}{n-3}\beta, \\ x_n > x_1, & \text{for } \beta > 0, \\ x_{n-1} > x_2, & \text{for } \alpha > 0, \\ x_n > x_2, & \text{for } \alpha > -(n-1)\beta, \\ x_n > x_{n-1}, & \text{for } \alpha < \frac{n-1}{n-3}\beta. \end{cases} \tag{4.10}$$

Let us denote by $\mathbf{x}_2 = (x_2, \ldots, x_2)^T \in \mathbb{R}^{n-3}$, the subvector of (4.9) containing the equal components in $\mathbf{x}$. Each particular ordering of the vector $\mathbf{x}$ stays unchanged when the components of $\mathbf{x}_2$ are permuted, so there are $(n-3)!$ index permutations corresponding to it. To identify such permutations we consider the following cases:

1. $\alpha, \beta > 0$: in correspondence to the three inequalities

$$0 < \alpha < \frac{1}{n-3}\beta, \qquad \frac{1}{n-3}\beta < \alpha < \frac{n-1}{n-3}\beta, \qquad \alpha > \frac{n-1}{n-3}\beta, \tag{4.11}$$

we find the following increasingly ordered vectors $\mathbf{x}$,

$$\begin{bmatrix} \mathbf{x}_2 \\ x_{n-1} \\ x_1 \\ x_n \end{bmatrix}, \qquad \begin{bmatrix} \mathbf{x}_2 \\ x_1 \\ x_{n-1} \\ x_n \end{bmatrix}, \qquad \begin{bmatrix} \mathbf{x}_2 \\ x_1 \\ x_n \\ x_{n-1} \end{bmatrix}, \tag{4.12}$$

respectively.
By permuting the components of $\mathbf{x}_2$, we obtain $(n-3)!$ index permutations for each of the three vectors, that is, $3(n-3)!$ admissible permutations.
For example, for $n = 5$ we obtain the 6 permutations contained in the columns of the following matrix

$$\begin{bmatrix} 2 & 3 & 2 & 3 & 2 & 3 \\ 3 & 2 & 3 & 2 & 3 & 2 \\ 4 & 4 & 1 & 1 & 1 & 1 \\ 1 & 1 & 4 & 4 & 5 & 5 \\ 5 & 5 & 5 & 5 & 4 & 4 \end{bmatrix}.$$

2. $\alpha > 0 > \beta$: the three inequalities

$$0 < \alpha < -\beta, \qquad -\beta < \alpha < -(n-1)\beta, \qquad \alpha > -(n-1)\beta, \tag{4.13}$$

correspond to the sorted vectors

$$\begin{bmatrix} x_n \\ x_1 \\ \mathbf{x}_2 \\ x_{n-1} \end{bmatrix}, \qquad \begin{bmatrix} x_n \\ \mathbf{x}_2 \\ x_1 \\ x_{n-1} \end{bmatrix}, \qquad \begin{bmatrix} \mathbf{x}_2 \\ x_n \\ x_1 \\ x_{n-1} \end{bmatrix}, \tag{4.14}$$

which originate $3(n-3)!$ further index permutations for $\mathbf{x}$.
For $n = 5$, we obtain

$$\begin{bmatrix} 5 & 5 & 5 & 5 & 2 & 3 \\ 1 & 1 & 2 & 3 & 3 & 2 \\ 2 & 3 & 3 & 2 & 5 & 5 \\ 3 & 2 & 1 & 1 & 1 & 1 \\ 4 & 4 & 4 & 4 & 4 & 4 \end{bmatrix}.$$

The above cases are exhaustive. Indeed, the inequalities $\alpha, \beta < 0$ and $\alpha < 0 < \beta$ produce permutations which are the reverse of the ones already considered in 1 and 2, respectively. The total number of permutations accounted for so far is

$$N_1 = 6(n-3)!.$$

We now consider equalities in (4.10), that is, we seek the values of the parameters α and β for which some components of the vector $\mathbf{x}$ in (4.9) become equal, besides those of $\mathbf{x}_2$.

It is important to remark that if two scalar components are equal, no new permutations are introduced. For example, $(n-3)\alpha = \beta$ makes $x_1 = x_{n-1}$, but the vector orderings deriving from the permutation of these two components have already been considered in the first two vectors of (4.12).

On the contrary, when x_2 is equal to any of the three other different components, then new index permutations are generated by permuting that component with the entries of the vector $\mathbf{x}_2$. When $\alpha \geq 0 > \beta$, the special cases where $x_2 = x_1$, $x_2 = x_{n-1}$, and $x_2 = x_n$, correspond to the conditions

$$\alpha = -\beta, \qquad \alpha = 0, \qquad \alpha = -(n-1)\beta, \tag{4.15}$$

respectively, and lead to the sorted vectors

$$\begin{bmatrix} x_n \\ \widetilde{\mathbf{x}}_{2,1} \\ x_{n-1} \end{bmatrix}, \qquad \begin{bmatrix} x_n \\ x_1 \\ \widetilde{\mathbf{x}}_{2,n-1} \end{bmatrix}, \qquad \begin{bmatrix} \widetilde{\mathbf{x}}_{2,n} \\ x_1 \\ x_{n-1} \end{bmatrix}, \tag{4.16}$$

where

$$\widetilde{\mathbf{x}}_{2,k} = \begin{bmatrix} \mathbf{x}_2 \\ x_k \end{bmatrix} = (x_2, \ldots, x_2, x_k)^T \in \mathbb{R}^{n-2}, \qquad k = 1, n-1, n.$$

Each vector in (4.16) produces $(n-2)!$ index permutations, from which one must subtract those already considered in (4.12) and (4.14). For example, for the first

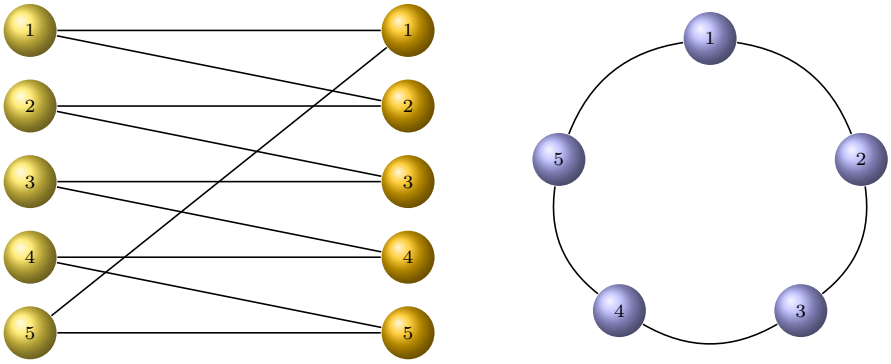

Fig. 4.3: The bipartite graph associated with the data matrix B in (4.18) for $n = 5$ (left), which leads to the cycle graph $\mathcal{C}_5$ (right).

vector of (4.16) the permutations

$$\begin{bmatrix} x_n \\ \mathbf{x}_2 \\ x_1 \\ x_{n-1} \end{bmatrix}, \quad \begin{bmatrix} x_n \\ x_1 \\ \mathbf{x}_2 \\ x_{n-1} \end{bmatrix},$$

have already been considered in the first two vectors of (4.14). This leads to

$$N_2 = 3\big((n-2)! - 2(n-3)!\big) = 3(n-4)(n-3)!$$

permutations. For $n = 5$ we obtain

$$\begin{bmatrix} 5 & 5 & 5 & 5 & 2 & 3 \\ 2 & 3 & 1 & 1 & 5 & 5 \\ 1 & 1 & 2 & 3 & 3 & 2 \\ 3 & 2 & 4 & 4 & 1 & 1 \\ 4 & 4 & 3 & 2 & 4 & 4 \end{bmatrix}.$$

To conclude with, the vector $\mathbf{x}$ defined in (4.9) possesses

$$N = N_1 + N_2 = 3(n-2)! \tag{4.17}$$

admissible permutations for $\alpha, \beta \in \mathbb{R} \setminus \{(0,0)\}$. Such permutations are one half of those foreseen by formula (3.1), that is, $3!(n-2)!$, confirming the conjecture that the structure of the problem introduces some constraints on the number of admissible solutions for the seriation problem.

4.2. The cycle graph. The second example of a graph whose Laplacian has a multiple Fiedler value is the cycle, or circular graph, $\mathcal{C}_n$, whose vertices are connected in a closed chain. The number of edges in $\mathcal{C}_n$ equals the number of vertices and, since every node has exactly two edges incident to it, every vertex has degree 2. Hence a cycle is a regular graph, i.e., a graph in which each vertex has the same degree k.

We will apply linear seriation to the cycle graph, but for this kind of networks it is sometimes relevant to consider circular seriation; see [23, 1, 38].

Consider the bipartite graph represented in Figure 4.3 (left) with associated data matrix

$$B = \left[\begin{array}{c} B_{n,1}^T \\ \hline \mathbf{b}_n^T \end{array}\right] \in \mathbb{R}^{n\times n}, \tag{4.18}$$

where $B_{n,1} \in \mathbb{R}^{n\times(n-1)}$ is the lower bidiagonal matrix defined in (4.5), $\mathbf{b}_n = (1, \mathbf{0}_{n-2}^T, 1)^T$, and $\mathbf{0}_k$ is the null vector of length k. As $B_{n,1}^T\mathbf{b}_n = \mathbf{b}_{n-1}^T$, its similarity matrix and Laplacian are, respectively,

$$S = BB^T = \left[\begin{array}{c|c} C_{n-1} & \mathbf{b}_{n-1} \\ \hline \mathbf{b}_{n-1}^T & 2 \end{array}\right], \qquad L_S = D - S = \left[\begin{array}{c|c} \widetilde{C}_{n-1} & -\mathbf{b}_{n-1} \\ \hline -\mathbf{b}_{n-1}^T & 2 \end{array}\right], \tag{4.19}$$

where

$$C_{n-1} = \begin{bmatrix} 2 & 1 & & & \\ 1 & 2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 2 & 1 \\ & & & 1 & 2 \end{bmatrix} \in \mathbb{R}^{(n-1)\times(n-1)},$$

and $\widetilde{C}_{n-1}$ is the tridiagonal matrix like C_{n-1}, with the elements in the sub- and super-diagonal of opposite sign. The matrix S can be seen as the adjacency matrix of a cycle graph C_n; see Figure 4.3.

The matrix L_S is circulant, that is, it is fully specified by its first column, while the other columns are cyclic permutations of the first one with an offset equal to the column index [12]. A basic property of a circulant matrix C is that its spectrum is analytically known. It is given by

$$\sigma(C) = \{\widehat{C}(1), \widehat{C}(\omega), \ldots, \widehat{C}(\omega^{n-1})\}, \tag{4.20}$$

where

$$\widehat{C}(\zeta) = \sum_{k=0}^{n-1} c_k \zeta^{-k} \tag{4.21}$$

is the discrete Fourier transform of the first column $(c_0, c_1, \ldots, c_{n-1})^T$ of C, $\omega = \mathrm{e}^{\frac{2\pi\mathbf{i}}{n}}$ is the minimal phase nth root of unity, and $\mathbf{i}$ the imaginary unit.

The next theorem states the behavior of the eigenvalues of the Laplacian matrix in the special case of a circular graph.

THEOREM 4.5. *Let B be the similarity matrix of a cycle graph with at least $n \geq 3$ vertices. Then, the eigenvalues of the Laplacian matrix $L_B = D - B$ are coupled as follows*

$$\lambda_j = \lambda_{n-j+2}, \qquad j = 2, \ldots, \left\lfloor \frac{n}{2} \right\rfloor + 1,$$

where $\lfloor m \rfloor$ denotes the minimal integer part of m. In particular, if n is odd $\lambda_1 = 0$ is the only simple eigenvalue. If n is even, the eigenvalues $\lambda_1 = 0$ and $\lambda_{n/2}$, of smallest and largest modulus, respectively, are the only simple ones.

The property trivially results from L_B being a symmetric circulant matrix. For the sake of clarity, we give a simple proof.

Proof. First, we recover a well known result in graph theory which states that the eigenvalue of smallest modulus of the Laplacian is $\lambda_1 = 0$. Indeed, from (4.20) and (4.21), it follows that the discrete Fourier transform of the first column of L_B is

$$\widehat{L}_B(\zeta) = 2 - \zeta^{-1} - \zeta^{-(n-1)},$$

and that $\lambda_1 = \widehat{L}_B(1) = 0$. Next, let $k = 1, \ldots, n-1$. From (4.20) and (4.21) we obtain

$$\lambda_{k+1} = \widehat{L}_B(\omega^k) = 2 - e^{-\frac{2\pi i}{n}k} - e^{\frac{2\pi i}{n}k} = 2 - 2\cos(\theta_k),$$

where $\theta_k = -\frac{2\pi i}{n}k$. The thesis follows from the property $\omega^k = \overline{\omega^{n-k}}$. ∎

The theorem immediately implies the following.

COROLLARY 4.6. *Let a graph satisfy the assumptions of Theorem 4.5. Then, its Fiedler value has multiplicity* 2.

The normalized eigenvectors of an $n \times n$ circulant matrix are the columns of the normalized Fourier matrix, that is,

$$\mathbf{v}_j = \frac{1}{\sqrt{n}} \left(1, \omega^{(j-1)}, \omega^{2(j-1)}, \ldots, \omega^{(n-1)(j-1)}\right)^T, \qquad j = 1, \ldots, n. \tag{4.22}$$

A basis for the eigenspace corresponding to the Fiedler value is given by $\{\mathbf{v}_2, \mathbf{v}_n\}$, where the entries of $\mathbf{v}_n$ are the conjugates of those of $\mathbf{v}_2$. To obtain eigenvectors with real entries we consider the vectors

$$\mathbf{w}_1 = \frac{(\mathbf{v}_2 + \mathbf{v}_n)}{2}, \qquad \mathbf{w}_2 = \frac{(\mathbf{v}_2 - \mathbf{v}_n)}{2\mathbf{i}}, \tag{4.23}$$

with components

$$(\mathbf{w}_1)_j = \cos\frac{2(j-1)\pi}{n}, \qquad (\mathbf{w}_2)_j = \sin\frac{2(j-1)\pi}{n}, \qquad j = 1, \ldots, n.$$

These vectors are, in fact, related to the discrete cosine transform (DCT) and the discrete sine transform (DST), respectively. They have many symmetries,

$$(\mathbf{w}_1)_j = (\mathbf{w}_1)_{n-j+2}, \qquad (\mathbf{w}_2)_j = -(\mathbf{w}_2)_{n-j+2}, \qquad j = 2, \ldots, \left\lfloor \frac{n}{2} \right\rfloor + 1,$$

and more relations are valid for n either odd or even.

Every Fiedler vector $\mathbf{x}$ lies in the eigenspace generated by $\mathbf{w}_1$ and $\mathbf{w}_2$, so that it can be expressed as

$$\mathbf{x} = \alpha\mathbf{w}_1 + \beta\mathbf{w}_2, \tag{4.24}$$

for α and $\beta \in \mathbb{R}$.

The presence of many symmetries in the vectors $\mathbf{w}_1$ and $\mathbf{w}_2$, makes it difficult to find a general rule to obtain the number of admissible permutations, i.e., of all the possible reorderings of the components of $\mathbf{x}$ for any n. The task is made harder by the fact that for specific values of the coefficients α and β, groups of components of the Fiedler vector $\mathbf{x}$ take the same value, generating bunches of admissible permutations.

We analyzed in detail the situation for $n = 4, 5, 6, 7$, determining 8, 15, 30, and 49 permutations, respectively. These results will be confirmed numerically in Section 6. We report here the permutations obtained for $n = 4$

$$P_{x_{(n=4)}} = \begin{bmatrix} 2 & 2 & 3 & 3 & 3 & 3 & 4 & 4 \\ 3 & 3 & 2 & 2 & 4 & 4 & 3 & 3 \\ 1 & 4 & 1 & 4 & 1 & 2 & 1 & 2 \\ 4 & 1 & 4 & 1 & 2 & 1 & 2 & 1 \end{bmatrix}.$$

We remark that, according to formula (3.1), the number of admissible solutions for $n = 4, 5, 6, 7$ should be 12, 36, 144, and 720, respectively, much larger than the actual values.

4.3. The generalized Petersen graph. The generalized Petersen graph is another graph whose Fiedler value has multiplicity 2. It was introduced by Coxeter [11] and it was given its name later, in 1969, by Watkins [42]. We denote it by $GPG(n, k)$. It has $2n$ vertices and $3n$ edges given, respectively, by

$$V(GPG(n,k)) = \{u_i, v_i | 1 \le i \le n\},$$
$$E(GPG(n,k)) = \{u_i u_{i+1}, u_i v_i, v_i v_{i+k} | 1 \le i \le n\},$$

where the subscripts are expressed as integers modulo n ($n \ge 5$) and k is the so called "skip". Let $\mathcal{U}(n, k)$ (respectively, $\mathcal{V}(n, k)$) be the subgraph of $GPG(n, k)$ consisting of the vertices $\{u_i | 1 \le i \le n\}$ (respectively, $\{v_i | 1 \le i \le n\}$) and edges $\{u_i u_{i+1} | 1 \le i \le n\}$ (respectively, $\{v_i v_{i+k} | 1 \le i \le n\}$). We will call $\mathcal{U}(n, k)$ (respectively, $\mathcal{V}(n, k)$) the outer (respectively, inner) subgraph of $GPG(n, k)$.

The $2n \times 2n$ data matrix of the graph $GPG(n, k)$ has the block structure

$$B = \begin{bmatrix} U & I_n \\ I_n & V_k \end{bmatrix} \tag{4.25}$$

where I_n is the $n \times n$ identity matrix, the block U is the adjacency matrix of the outer subgraph $\mathcal{U}(n, k)$, it coincides with the adjacency matrix (4.19) of a cycle graph, with the diagonal elements equal to 3. The block V_k is the adjacency matrix for the inner graph $\mathcal{V}(n, k)$, whose structure is determined by the skip k. The matrices U and V_k are circulant. They are specified by their first column given, respectively, by

$$\mathbf{c} = (0, 1, \underbrace{0, \ldots, 0}_{n-3}, 1)^T, \qquad \mathbf{c}^{(k)} = (\mathbf{0}_k, 1, \underbrace{0, \ldots, 0}_{n-2k-1}, 1, \mathbf{0}_{k-1})^T,$$

where $\mathbf{0}_j$ denotes the null vector of length j, or the empty vector when $j = 0$. We will write $U = \text{circ}(\mathbf{c})$ and $V_k = \text{circ}(\mathbf{c}^{(k)})$.

We consider the data matrix represented by the graph in Figure 4.4 (left) for $n = 5$ whose similarity matrix can be seen as the adjacency matrix of the generalized Petersen graph $GPG(n, k)$ with a skip $k = 1$; see Figure 4.4 (right). In this particular case, also the inner subgraph is a cycle graph and the incidence matrix has the block structure

$$\widetilde{B} = \begin{bmatrix} B^T & I_n & \mathbf{0}_n \\ \mathbf{0}_n & I_n & B^T \end{bmatrix} \in \mathbb{R}^{2n \times 3n}, \tag{4.26}$$

where $B \in \mathbb{R}^{n \times n}$ is the incidence matrix of the cycle defined in (4.18). Its similarity matrix and Laplacian are, respectively

$$S = \begin{bmatrix} F & I_n \\ I_n & F \end{bmatrix} \qquad L_S = \begin{bmatrix} \widetilde{F} & -I_n \\ -I_n & \widetilde{F} \end{bmatrix} \tag{4.27}$$

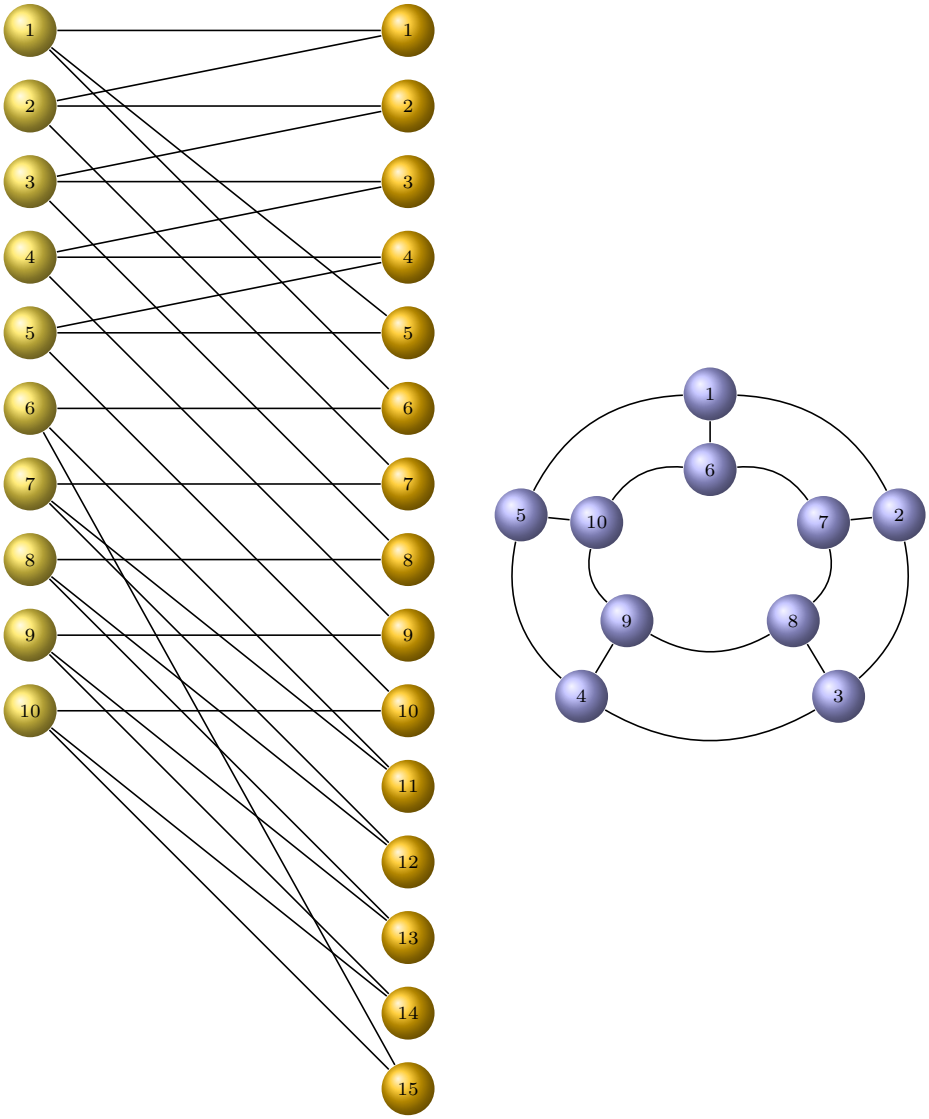

Fig. 4.4: The bipartite graph associated with the data matrix B in (4.25) for $n = 5$ and $k = 1$ (left), which leads to the generalized Petersen graph $GPG(5, 1)$ (right).

where F and $\widetilde{F}$ are $n \times n$ circulant matrices given respectively by

$$F = \text{circ}(3, 1, \underbrace{0, \ldots, 0}_{n-3}, 1) \qquad \widetilde{F} = \text{circ}(3, -1, \underbrace{0, \ldots, 0}_{n-3}, -1). \tag{4.28}$$

THEOREM 4.7. *Let $\widetilde{B}$ be the $2n \times 3n$ data matrix* (4.26)*. Then, the Fiedler value of the Laplacian matrix L_S has multiplicity 2.*

Proof. L_S is a block circulant matrix with circulant blocks $\widetilde{F}$ and $-I_n$. A block circulant matrix can be expressed as the sum of Kronecker products. In our case, we

have

$$L_S = P_1 \otimes \widetilde{F} + P_2 \otimes (-I_n),$$

where $P_1 = I_2$ and $P_2 = \mathrm{circ}(0,1)$. More in general, one has $P_i = \mathrm{circ}(\mathbf{e}_i)$, with $\mathbf{e}_i$ the ith canonical basis vector.

If we define the matrix-valued function

$$H(x) = x^0 \otimes \widetilde{F} + x^1 \otimes (-I_n),$$

so that $H(P_2) = L_S$, it can be shown (see [26]) that the spectrum of L_S is the union of the spectra of $H(\lambda_1)$ and $H(\lambda_2)$, where λ_1 and λ_2 are the eigenvalues of P_2. Moreover, the eigenvectors of L_S are given by the Kronecker products $\mathbf{v}_i \otimes \mathbf{u}_j$, $i,j = 1,2$, where v_i are the eigenvectors of P_2 and u_i are the eigenvectors of both $H(\lambda_1)$ and $H(\lambda_2)$.

In our case, $\lambda_1 = 1$ and $\lambda_2 = -1$, so that $H(\lambda_1) = \widetilde{F} - I_n$ and $H(\lambda_2) = \widetilde{F} + I_n$. An immediate result is that the eigenvalues of L_B are given by

$$\mu_i = \begin{cases} \sigma_i - 1 & \text{if} \quad i = 1, \ldots, n \\ \sigma_{i-n} + 1 & \text{if} \quad i = n+1, \ldots, 2n \end{cases},$$

where σ_i, $i = 1, \ldots, n$, are the eigenvalues of the matrix $\widetilde{F}$. Since $\widetilde{F}$ is symmetric circulant, its eigenvalues are coupled (see Theorem 4.5) and this completes the proof. □

COROLLARY 4.8. *Let σ be the second smallest eigenvalue of the matrix F (4.28) and $\{\mathbf{w}_1, \mathbf{w}_2\}$ be a basis for the eigenspace corresponding to σ. Then, $\sigma - 1$ is the Fiedler value of the Laplacian matrix L_S given in (4.27) and $\{\mathbf{v}_1, \mathbf{v}_2\}$ is a basis for the associated eigenspace, where*

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes \mathbf{w}_1 = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_1 \end{bmatrix} \quad \textit{and} \quad \mathbf{v}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes \mathbf{w}_2 = \begin{bmatrix} \mathbf{w}_2 \\ \mathbf{w}_2 \end{bmatrix}. \tag{4.29}$$

Proof. The proof follows from Theorem 4.7, noting that $(1,1)^T$ is the eigenvector of P_2 associated to the eigenvalue $\lambda_1 = 1$. □

Since the eigenvectors of the matrix $\widetilde{F}$ are the columns of the normalized Fourier matrix, we can obtain the set of admissible permutations from the results obtained for the cycle graph. Indeed, the vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ defined in (4.29) have the same entries as the vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ in (4.23), but each entry is doubled. This means that the components of a vector $\mathbf{x}$ in the *Fiedler plane* come in pairs. Consequently, the number of the admissible permutations for a generalized Petersen graph $GPG(n,1)$ is 2^n times the admissible permutations obtained for a cycle graph.

For $n = 4, 5, 6, 7$, we expect at least 128, 480, 1920, and 6272 permutations, respectively. Other admissible permutations may appear in case other equalities occur between the entries of $\mathbf{v}_1$ and those of $\mathbf{v}_2$. Since the graph has $2n$ nodes, formula (3.1) in this case produces the much larger estimates 4320, 241920, $2.18 \cdot 10^7$, and $2.87 \cdot 10^9$ respectively, for the number of solutions.

5. Two numerical methods to determine admissible permutations. Let the Laplacian matrix L of a graph with n nodes have a double Fiedler value λ_2, and let

$$\mathbf{v} = (v_1, v_2, \ldots, v_n)^T \quad \text{and} \quad \mathbf{w} = (w_1, w_2, \ldots, w_n)^T$$

be an orthogonal basis for the corresponding eigenspace $\mathcal{F}$ of dimension 2. In principle, to determine the admissible permutations one should consider all the linear combinations

$$\mathbf{x} = \alpha\mathbf{v} + \beta\mathbf{w}, \qquad \alpha, \beta \in \mathbb{R}, \tag{5.1}$$

and list all the different permutations corresponding to sorting the vector $\mathbf{x}$ as the coefficients α and β vary, removing the reversed permutations.

To reduce the number of free parameters, we consider vectors of the type

$$\mathbf{x} = \mathbf{v} + \gamma\mathbf{w}, \qquad \gamma \in \mathbb{R}. \tag{5.2}$$

This representation excludes only the vector $\mathbf{x} = \mathbf{w}$, which has to be considered separately. Then, the problem leads to considering all the possible sortings of the numbers $x_i = v_i + \gamma w_i$, $i = 1, \ldots, n$, for $\gamma \in \mathbb{R}$.

This formulation admits an interesting geometric representation. Let

$$P_i = (v_i, w_i), \qquad i = 1, \ldots, n,$$

be points on the Euclidean plane and consider a straight line passing through the origin, parallel to the normalized vector $\mathbf{d} = (1, \gamma)/\sqrt{1+\gamma^2}$. The orthogonal projections of the points P_i on the line are proportional to the numbers x_i, so the admissible permutations can be found by letting the straight line rotate on the plane, and sorting the projections of the points for each fixed value of γ. The permutation corresponding to the vector $\mathbf{w}$ is obtained as γ approaches $\pm\infty$.

A first possibility to realize this procedure is to resort to a randomized algorithm. To this end, we developed a simple Monte Carlo approach. In the case of a double Fiedler value, we consider the Fiedler vectors (5.2) generated by N random values of γ in $\mathbb{R}$. Each vector is then sorted, and the corresponding permutations of indices are stored in the columns of a matrix. After removing all the repeated permutations and the swapped ones, we obtain a set of allowed permutations of the n nodes in the considered graph.

This approach is easy implemented and it can be generalized immediately to the case of a Fiedler value with multiplicity larger than 2. The high computational cost, deriving from determining whether a particular permutation has already been identified, might be mitigated by optimized hashing and sorting algorithms. Admissible permutations might be stored in a PQ-tree [4], a very efficient data structure for storing index permutations which has been used in the spectral algorithm [2] and is implemented, e.g., in [10]. Nevertheless, the main drawback of this method is that it is not able to identify index permutations corresponding to particular values of the coefficients of the linear combination (5.1). For example, the Fiedler vectors (4.16) for the modified star graph are produced when the parameter α takes exactly the values given in equation (4.15), condition which has zero probability in the randomized process considered. This aspect will be investigated in the numerical examples of Section 6, where we will apply this numerical method and the following one to the case studies considered in Section 4.

To efficiently compute all the admissible permutations in the particular case of a Fielder value with multiplicity 2, we developed a *graphical method* which uses a deterministic approach to implement the point projection method described above.

The idea behind the method, described in Algorithm 1, is considering the vector (5.2) as a function of the parameter $\gamma \in \mathbb{R}$

$$\mathbf{f}(\gamma) = \mathbf{v} + \gamma\mathbf{w} = (f_1(\gamma), \ldots, f_n(\gamma))^T.$$

Then, its components $f_i(\gamma) = v_i + \gamma w_i$, $i = 1, \ldots, n$, can be interpreted as straight lines in the Euclidean plane. Letting γ vary, one obtains all the possible Fiedler vectors, whose components are given by the intersections of a vertical line with the graphs of the linear functions $f_i(\gamma)$; see, e.g., Figure 5.1.

Computing the intersections of these lines (line 9 of Algorithm 1) identifies intervals characterized by a specific ordering of the components of $\mathbf{f}(\gamma)$. The position of the lines before the first intersection point (line 20) gives the reordering of the Fiedler vector $\mathbf{x} = \mathbf{f}(\lambda)$ which corresponds to the first admissible permutation of the nodes. New permutations are obtained by reordering the values of $\mathbf{f}(\gamma)$ at each intersection point. Indeed, an intersection point introduces one or more swaps in the components in the Fiedler vector, as γ increases, and so new admissible permutations.

The performance of the two procedures is analyzed and compared in the numerical examples illustrated in the following section.

Algorithm 1 Graphic method for determining the admissible reorderings of the nodes in a graph with a double Fiedler value

1: **Requires:** Fiedler vectors $\mathbf{v}, \mathbf{w} \in \mathbb{R}^n$ and tolerance τ
2: **Ensure:** matrix P containing admissible node reorderings
3: $f(\gamma) = \mathbf{v} + \gamma\mathbf{w}$
4: Φ (2 columns matrix, initially empty, for intersections and their multiplicity)
5: $m = 0$ (number of intersections found)
6: **for** $i = 1, \ldots, n-1$
7: **for** $j = i+1, \ldots, n$
8: **if** $|w_i - w_j| > \tau$
9: $\gamma_{\text{int}} = (v_i - v_j)/(w_j - w_i)$ (new intersection abscissa)
10: let $r \in \{1, \ldots, m\}$ such that $|\gamma_{\text{int}} - \Phi_{r,1}| < \tau$, otherwise $r = 0$
11: **if** $r = 0$ (γ_{int} is not in Φ)
12: $m = m + 1$, $\Phi_{m,1} = \gamma_{\text{int}}$, $\Phi_{m,2} = 1$ (add new intersection)
13: **else** $\Phi_{r,2} = \Phi_{r,2} + 1$ (increment multiplicity)
14: **end if**
15: **end if**
16: **end for**
17: **end for**
18: sort rows of Φ so that intersections are in increasing order
19: store in P the permutations corresponding to the possible orderings of $\mathbf{w}$
20: $\mathbf{y}_1 = f(\Phi_{1,1} - 1)$ (values of the lines in the first interval)
21: add to P the permutations corresponding to the possible orderings of $\mathbf{y}_1$
22: **for** $i = 1, \ldots, m-1$
23: $\mathbf{y}_1 = f(\Phi_{i,1})$ (left endpoint of ith interval)
24: $\mathbf{y}_2 = f((\Phi_{i,1} + \Phi_{i+1,1})/2)$ (center point of ith interval)
25: add to P the permutations corresponding to the orderings of $\mathbf{y}_1$ and $\mathbf{y}_2$
26: **end for**
27: $\mathbf{y}_1 = f(\Phi_{m,1})$ (last intersection)
28: $\mathbf{y}_2 = f(\Phi_{m,1} + 1)$ (last interval)
29: add to P the permutations corresponding to the orderings of $\mathbf{y}_1$ and $\mathbf{y}_2$
30: remove from P repeated or reversed permutations

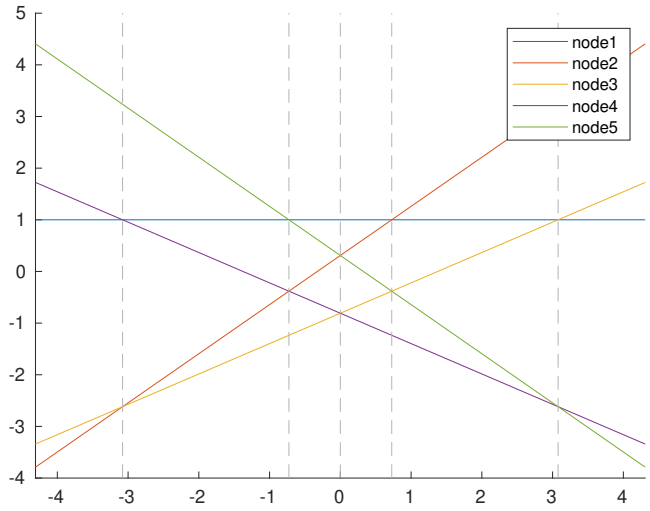

Fig. 5.1: Lines corresponding to the nodes in the cycle graph C_n with $n = 5$ nodes.

To illustrate the functioning of the graphical method, we consider the cycle graph with $n = 5$ nodes, depicted in Figure 4.3. As pointed out in Section 4.2, the number of admissible permutations is 15. They can be obtained through the graphical method by considering the swap of the indices corresponding to the lines which intersect. More precisely, in Figure 5.1 we display the graph of the functions $f_i(x) = v_i + xw_i$, for $i = 1, \ldots, 5$, each one representing the behaviour of the $i-$th entry of the Fiedler vector (5.2). Intersection points are highlighted in the figure by vertical dashed lines. As explained above, the first admissible permutation is obtained by considering the position of the lines before the first intersection points and therefore it is given by (`5 4 1 3 2`). The first vertical dashed line points out that there are two pairs of lines that intersect. Consequently, new permutations are obtained from the first one by swapping the indices corresponding to nodes (`1,4`) and (`2,3`). Hence, the additional permutations are

(`5 1 4 3 2`), (`5 1 4 2 3`), (`5 4 1 2 3`).

After the first intersection, the position of the lines produces the permutation of the nodes (`5 1 4 2 3`), which has already been considered. The second vertical dashed line, corresponding to the second intersection point, reveals that two pairs of lines intersect, i.e., we need to swap the indices of nodes (`1,5`) and (`2,4`). The new admissible permutations are then

(`5 1 2 4 3`), (`1 5 2 4 3`), (`1 5 4 2 3`).

After the second intersection, the lines follow the order (`1 5 2 4 3`), that is contained in the previous set. In correspondence to the third intersection two further pairs of lines intersect, i.e., the indices of nodes (`2,5`) and (`3,4`) must be swapped. In this case, the new permutations are

(`1 5 2 3 4`), (`1 2 5 4 3`), (`1 2 5 3 4`).

After this intersection the permutation is (1 2 5 3 4), which has been already taken into account. Considering the fourth vertical dashed line, which highlights that lines 1-2 and 3-5 intersect, one obtains the admissible permutations

$$(1\ 2\ 3\ 5\ 4), \qquad (2\ 1\ 5\ 3\ 4), \qquad (2\ 1\ 3\ 5\ 4).$$

After the fourth intersection point, the position of the lines gives the permutation (2 1 3 5 4), already present in our set of permutations. The last intersection yields that lines 1-3 and 4-5 intersect, leading to the permutations

$$(2\ 3\ 1\ 5\ 4), \qquad (2\ 1\ 3\ 4\ 5), \qquad (2\ 3\ 1\ 4\ 5).$$

The last permutation (2 3 1 4 5), coincides with the reverse of the first one. Removing it leaves 15 admissible permutations of the indices, reported in the following matrix

$$\begin{bmatrix} 5 & 5 & 5 & 5 & 5 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 4 & 1 & 1 & 4 & 1 & 5 & 5 & 5 & 2 & 2 & 2 & 1 & 1 & 3 & 1 \\ 1 & 4 & 4 & 1 & 2 & 2 & 4 & 2 & 5 & 5 & 3 & 5 & 3 & 1 & 3 \\ 3 & 3 & 2 & 2 & 4 & 4 & 2 & 3 & 4 & 3 & 5 & 3 & 5 & 5 & 4 \\ 2 & 2 & 3 & 3 & 3 & 3 & 3 & 4 & 3 & 4 & 4 & 4 & 4 & 4 & 5 \end{bmatrix}.$$

6. Numerical experiments. In this section we report the results produced by the two methods introduced in Section 5 for determining the admissible permutations of a set of units, in the case the Fiedler value of the associated graph has multiplicity 2. To verify the performance of the methods, the graphical (see Algorithm 1) and the Monte Carlo methods have been implemented in Matlab R2021a and applied to the three case studies described in Section 4. The numerical experiments were performed on an Intel Xeon Gold 6136 computer (16 cores, 32 threads) equipped with 128 GB RAM, running the Linux operating system.

The first computed example consists of finding the admissible permutations of the nodes of a modified star graph $\widehat{\mathcal{S}}_n$ with data matrix (4.5). As stated in Corollary 4.4, the Laplacian of the similarity matrix associated to the graph has a double Fiedler value equal to 1. Since an orthogonal basis for the eigenspace $\mathcal{F}$ corresponding to the Fiedler value is known, every $\mathbf{x} \in \mathcal{F}$ can be expressed by $\mathbf{x} = Q_2\mathbf{y}$, with $\mathbf{y} = [\alpha, \beta]^T$, as in (4.9). As explained in detail in Section 4.1, the permutations of the nodes that yield a solution to the seriation problem are given by all the possible reorderings of the entries of $\mathbf{x}$.

The results of the experiments concerning the application of the graphical and the Monte Carlo methods to a graph $\widehat{\mathcal{S}}_n$ with a number of nodes n ranging from 5 to 10 are displayed in Table 6.1. In particular, the second column contains the number $3(n-2)!$ of admissible permutations for a modified star graph stated in (4.17). It coincides with the number of admissible permutations found by the graphical method, reported in the third column of the table. We note that such number is one half of the estimate furnished by Equation (3.1), for $k = 2$. For the following examples, the reduction with respect to this estimate is even larger.

As the fifth column shows, the Monte Carlo method fails to identify all the permutations, after considering $N = 1000$ random linear combinations of the orthonormal basis for the eigenspace $\mathcal{F}$. We verified that increasing the value of N up to 5000 the performance of the method does not improve. In this test, the graphical algorithm is, for every n, much faster than the Monte Carlo method, as one can observe comparing the computing time in seconds reported in the fourth and sixth columns of Table 6.1.

		Graphical method		Monte Carlo method	
n	$3(n-2)!$	found perms	time	found perms	time
5	18	18	1.17e-01	14	1.27e-01
6	72	72	1.57e-02	48	8.19e-02
7	360	360	1.63e-02	216	2.58e-01
8	2160	2160	9.02e-02	1200	5.22e+00
9	15120	15120	1.11e+00	7920	8.80e+01
10	120960	120960	2.21e+01	60480	1.76e+03

Table 6.1: Results obtained by applying the graphical and the Monte Carlo methods to the modified star graph with data matrix B (4.5).

	Graphical method		Monte Carlo method	
n	found perms	time	found perms	time
4	8	1.53e-01	4	1.61e-01
5	15	1.57e-01	7	4.87e-02
6	30	1.48e-02	14	6.77e-02
7	49	4.03e-03	13	6.13e-02
8	88	4.90e-03	20	7.52e-02
9	135	1.33e-02	23	7.68e-02
10	230	5.25e-03	54	8.10e-02

Table 6.2: Results obtained by applying the graphical and the Monte Carlo methods to the cycle graph with data matrix B (4.18).

We remark that the failure of the Monte Carlo approach is due to the fact that many admissible permutations result from specific values of the coefficients α and β in the linear combination (4.9); see, e.g., (4.16). Assuming such values is an event with zero probability in a random draw of real numbers, so it is very unlikely to occur in the algorithm. On the contrary, the graphical method explicitly considers equal components in the Fiedler vectors when it processes intersections between the lines; see lines 23 and 27 of Algorithm 1.

A similar comparison between the two methods has also been considered for the cycle graph $\mathcal{C}_n$ analyzed in Section 4.2. The results are displayed in Table 6.2. In this case, every vector $\mathbf{x}$ in the eigenspace associated with the double Fiedler value of $\mathcal{C}_n$ can be represented as in Equation (4.24). In Section 4.2, we have not been able to foresee the number of admissible permutations for this graph, but the result we found for $n = 4, 5, 6, 7$ are confirmed by the outcome of the graphical method; see the second column in Table 6.2. Again, the graphical method proves to be the fastest one and the Monte Carlo method fails in recovering all the admissible permutations. The reason for this failure is the same as discussed above.

The results displayed in Table 6.3 are obtained by applying the two methods to the generalized Petersen graph $GPG(n, 1)$. As discussed in Section 4.3, both the outer and the inner subgraphs in $GPG(n, 1)$ are cycle graphs and the total number of nodes is $2n$. By following the discussion regarding the cycle graph and the results contained in Theorem 4.7 and Corollary 4.8 it follows that each vector $\mathbf{x}$ in the eigenspace corresponding to the Fiedler value has n pairs of coinciding entries. Then, keeping

		Graphical method		Monte Carlo method	
n	$2^n n$	found perms	time	found perms	time
5	160	5600	2.57e-01	160	1.61e+00
6	384	48000	7.44e-01	384	1.38e+01
7	896	192640	1.83e+01	896	3.99e+01
8	2048	1546240	4.17e+02	2048	9.77e+01
9	4608	5967360	3.10e+04	4608	2.38e+02

Table 6.3: Results obtained by applying the graphical and the Monte Carlo methods to the Generalized Petersen graph with data matrix B (4.25).

into account the number of permutations for a cycle, the admissible permutations of the nodes in $GPG(n, 1)$ are at least $2^n n$.

The second column of Table 6.3 reports this minimum value for the admissible permutations. It is remarkable to observe that this is exactly the number of permutations recovered by the Monte Carlo method. Anyway, the real number of admissible permutations is much larger than that, as testified by the results of the graphical method in the third column of the table. This huge number of permutations requires a large computing time, making the graphical method extremely slower than in the other examples. Nevertheless, it is effective when computing the complete solution of the problem, while the randomized approach it is not, even though in this case $N = 5000$ random Fiedler vectors have been used.

We analyzed the performance of both methods by means of the "profiler" available in Matlab. It turns out that the bottleneck for the execution time of the algorithms are the tests for verifying that a new permutation does not appear in the list of those already computed either in direct or reverse ordering. When the number of admissible permutations is not too large, this does not significantly affect the complexity of the graphical method, while it does in the case of the generalized Petersen graph.

As a last numerical experiment, we illustrate the effect of the admissible permutations corresponding to a double Fiedler value on the similarity matrix S given in (4.6), associated to the modified star graph, for $n = 5$. In the left pane of Figure 6.1 we report a spy plot of the matrix S: it reproduces the pattern of nonzero entries of the matrix. On the right, in the same figure, we display the PQ-tree produced by the package PQser [10], which implements the spectral algorithm from [2], when the matrix S is given as input. In this case, the package inserts a conventional "M-node", which outlines the presence of a multiple Fiedler value and the impossibility to determine a complete set of solutions.

As already observed, in this case there are 18 admissible permutations. They have been explicitly constructed in Section 4.1, and they are correctly reproduced by our graphical algorithm; see Table 6.1. By applying these permutations to the rows and columns of the similarity matrix S, we obtain 6 different matrices. We display the spy plots of 3 of them in Figure 6.2. It can be observed that all of them have a smaller bandwidth than the original matrix, and that none of them is an R-matrix, as the data set is *non-perfect*.

7. Conclusions. In this paper, we study the possible orderings of the Fiedler vector of a graph, under the assumption that the Fiedler value has multiplicity larger than one. Determining such orderings is related to the solution of the seriation problem and to many other graph and matrix problems; see [10, 13] for a review.

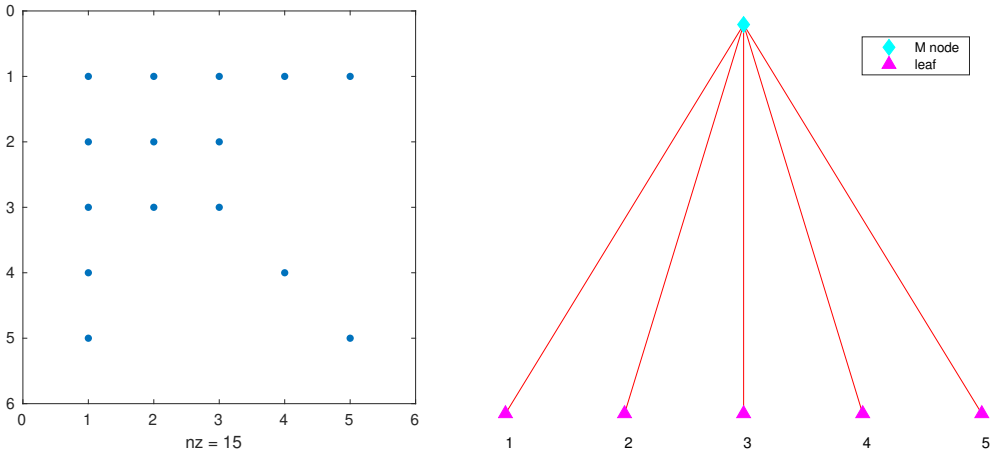

Fig. 6.1: Modified star graph with $n = 5$ nodes: on the left, spy plot of the similarity matrix S; on the left, PQ-tree given as output by the package PQser [10], which implements the spectral algorithm from [2].

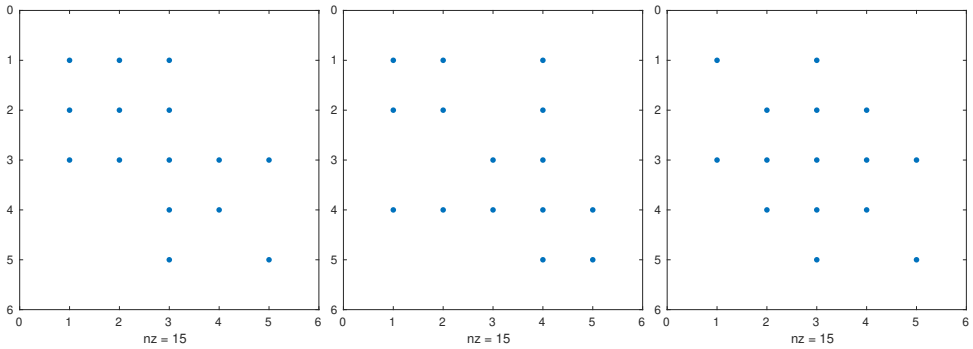

Fig. 6.2: Modified star graph with $n = 5$ nodes: spy plot of the reorderings of the similarity matrix S produced by 3 of the 18 admissible permutations.

When the Fiedler value is a simple eigenvalue of the Laplacian matrix associated to the graph, the spectral algorithm introduced in [2] is particularly effective in enumerating all the solutions to the seriation problem, that is, the index permutations that realize the best ordering of a set of units according to the given similarity rule. When the multiplicity of the Fiedler value is larger than one, such algorithm is not applicable.

We examine in detail three case studies characterized by a double Fiedler value, for which it is possible to draw conclusions about the solution of the problem, and we show that the number of admissible permutations is smaller than the maximum allowed number of permutations. In fact, it varies in the three cases considered, and it appears to depend on the structure of the underlying bipartite graph.

We propose a graphical method to list the admissible permutations and we compare it to a randomized algorithm. Numerical experiments show that the graphical method is able to identify all the admissible permutations for the considered case

studies, and that its performance is superior to the randomized approach. The reason for this is that some admissible permutations are produced by particular fixed values of the coefficients in the linear combination of Fiedler vectors, and such coefficients are not likely to be reproduced by a random sampling.

We believe that further work is needed to study the case of a Fiedler value with multiplicity larger than two and to optimize the complexity and data storage of the algorithms. Such development will be essential for the application to large scale problems such as, for example, seriation problems derived from genome sequencing and sparse matrix reordering problems.

Acknowledgment. The authors are indebted to the two Reviewers for their care in reviewing the manuscript. Their remarks led to a substantial improvement in the content of the paper and suggested possible future research development.

Funding. Anna Concas, Caterina Fenu, and Giuseppe Rodriguez were partially supported by Regione Autonoma della Sardegna research project "Algorithms and Models for Imaging Science [AMIS]" (RASSR57257, intervento finanziato con risorse FSC 2014-2020 - Patto per lo Sviluppo della Regione Sardegna) and by the INdAM-GNCS research project "Tecniche numeriche per l'analisi delle reti complesse e lo studio dei problemi inversi". Caterina Fenu also gratefully acknowledges Regione Autonoma della Sardegna for the financial support provided under the Operational Programme P.O.R. Sardegna F.S.E. (European Social Fund 2014-2020 - Axis III Education and Formation, Objective 10.5, Line of Activity 10.5.12). The research of Raf Vandebril was partially supported by the Research Council KU Leuven, project C16/21/002 (Manifactor: Factor Analysis for Maps into Manifolds).

REFERENCES

[1] S. Armstrong, C. Guzmán, and C. A. Sing Long, *An optimal algorithm for strict circular seriation*, SIAM J. Math. Data Sci., 3 (2021), pp. 1223–1250.
[2] J. E. Atkins, E. G. Boman, and B. Hendrickson, *A spectral algorithm for seriation and the consecutive ones problem*, SIAM J. Comput., 28 (1998), pp. 297–310.
[3] S. T. Barnard, A. Pothen, and H. Simon, *A spectral algorithm for envelope reduction of sparse matrices*, Numer. Linear Algebra Appl., 2 (1995), pp. 317–334.
[4] K. S. Booth and G. S. Lueker, *Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms*, J. Comput. Syst. Sci., 13 (1976), pp. 335–379.
[5] G. W. Brainerd, *The place of chronological ordering in archaeological analysis*, Am. Antiq., 16 (1951), pp. 301–313.
[6] M. J. Brusco and D. Steinley, *Clustering, seriation, and subset extraction of confusion data*, Psychol. Methods, 11 (2006), pp. 271–286.
[7] M. Carmona, V. Chepoi, G. Naves, and P. Préa, *A simple and optimal algorithm for strict circular seriation*, arXiv preprint arXiv:2205.04694, (2022).
[8] V. Chepoi and B. Fichet, *Recognition of Robinsonian dissimilarities*, J. Classif., 14 (1997), pp. 311–325.
[9] R. R. Coifman, Y. Shkolnisky, F. J. Sigworth, and A. Singer, *Graph Laplacian tomography from unknown random projections*, IEEE Trans. Image Process., 17 (2008), pp. 1891–1899.
[10] A. Concas, C. Fenu, and G. Rodriguez, *PQser: a Matlab package for spectral seriation*, Numer. Algorithms, 80 (2019), pp. 879–902.
[11] H. S. M. Coxeter, *Self-dual configurations and regular graphs*, Bull. Amer. Math. Soc., 56 (1950), pp. 413–455.
[12] P. J. Davis, *Circulant Matrices*, Wiley, New York, 1979.
[13] N. M. M. De Abreu, *Old and new results on algebraic connectivity of graphs*, Linear Algebra Appl., 423 (2007), pp. 53–73.
[14] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, *Cluster analysis and display of genome-wide expression patterns*, P. Natl. Acad. Sci. U.S.A., 95 (1998), pp. 14863–14868.

[15] X. Evangelopoulos, A. J. Brockmeier, T. Mu, and J. Y. Goulermas, *Continuation methods for approximate large scale object sequencing*, Mach. Learn., 108 (2019), pp. 595–626.
[16] ———, *Circular object arrangement using spherical embeddings*, Pattern Recognit., 103 (2020), p. 107192.
[17] M. Fiedler, *Algebraic connectivity of graphs*, Czech. Math. J., 23 (1973), pp. 298–305.
[18] ———, *A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory*, Czech. Math. J., 25 (1975), pp. 619–633.
[19] ———, *Laplacian of graphs and algebraic connectivity*, Banach Center Publ., 25 (1989), pp. 57–70.
[20] F. Fogel, R. Jenatton, F. Bach, and A. d'Aspremont, *Convex relaxations for permutation problems*, in Advances in Neural Information Processing Systems 26 (NIPS 2013), C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, eds., 2013.
[21] M. Hahsler, K. Hornik, and C. Buchta, *Getting things in order: an introduction to the R package seriation*, J. Stat. Softw., 25 (2008), pp. 1–34.
[22] F. R. Hodson, D. G. Kendall, and P. Tautu, *Mathematics in the Archaeological and Historical Sciences*, Edinburgh University Press, Edinburgh, 1971.
[23] L. Hubert, P. Arabie, and J. Meulman, *Graph-theoretic representations for proximity matrices through strongly-anti-Robinson or circular strongly-anti-Robinson matrices*, Psychometrika, 63 (1998), pp. 341–358.
[24] A. Jamakovic and P. V. Mieghem, *On the robustness of complex networks by using the algebraic connectivity*, in International conference on research in networking, Springer, 2008, pp. 183–194.
[25] A. Jamakovic and S. Uhlig, *On the relationship between the algebraic connectivity and graph's robustness to node and link failures*, in 2007 Next Generation Internet Networks, IEEE, 2007, pp. 96–102.
[26] A. Kaveh and H. Rahami, *Block circulant matrices and applications in free vibration analysis of cyclically repetitive structures*, Acta Mech., 217 (2011), pp. 51–62.
[27] D. G. Kendall, *A statistical approach to Flinders–Petries sequence-dating*, Bull. Int. Stat. Inst., 40 (1963), pp. 657–681.
[28] ———, *A mathematical approach to seriation*, Philos. Trans. R. Soc. A-Math. Phys. Eng. Sci., 269 (1970), pp. 125–134.
[29] M. Laurent and M. Seminaroti, *A Lex-BFS-based recognition algorithm for Robinsonian matrices*, Discret. Appl. Math., 222 (2017), pp. 151–165.
[30] ———, *Similarity-First Search: a new algorithm with application to Robinsonian matrix recognition*, SIAM Discret. Math., 31 (2017), pp. 1765–1800.
[31] C. H. Lim and S. Wright, *A box-constrained approach for hard permutation problems*, in International Conference on Machine Learning, PMLR, 2016, pp. 2454–2463.
[32] B. G. Mirkin and S. N. Rodin, *Graphs and Genes*, vol. 11 of Biomathematics, Springer-Verlag, Berlin, 1984.
[33] J. M. Ortega, *On Sturm sequences for tridiagonal matrices*, J. ACM, 7 (1960), pp. 260–263.
[34] P. M. Pardalos, F. Rendl, and H. Wolkowicz, *The quadratic assignment problem: A survey and recent developments*, in Quadratic Assignment and Related Problems, P. M. Pardalos and H. Wolkowicz, eds., vol. 16 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, New York, 1994, pp. 1–42.
[35] W. M. F. Petrie, *Sequences in prehistoric remains*, J. R. Anthropol. Inst., 29 (1899), pp. 295–301.
[36] P. Piana Agostinetti and M. Sommacal, *Il problema della seriazione in archeologia*, Rivista di Scienze Preistoriche, LV (2005), pp. 29–69.
[37] P. Préa and D. Fortin, *An optimal algorithm to recognize Robinsonian dissimilarities*, J. Classif., 31 (2014), p. 351.
[38] A. Recanati, T. Kerdreux, and A. d'Aspremont, *Reconstructing latent orderings by spectral clustering*. 2018.
[39] A. Recanati, N. Servant, J.-P. Vert, and A. d'Aspremont, *Robust seriation and applications to cancer genomics*, arXiv preprint arXiv:1806.00664, (2018).
[40] W. S. Robinson, *A method for chronologically ordering archaeological deposits*, Am. Antiq., 16 (1951), pp. 293–301.
[41] M. Seston, *Dissimilarités de Robinson: algorithmes de reconnaissance et d'approximation*, PhD thesis, Aix Marseille 2, 2008.
[42] M. E. Watkins, *A theorem on tait colorings with an application to the generalized Petersen graphs*, J. Comb. Theory, 6 (1969), pp. 152–164.
[43] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, vol. 87, Clarendon Press, Oxford, 1965.