Locating Eigenvalues of a Symmetric Matrix whose Graph is Unicyclic

R. O. BRAGA[1*], V. M. RODRIGUES[2] and R. O. SILVA[3]

ABSTRACT. We present a linear-time algorithm that computes in a given real interval the number of eigenvalues of any symmetric matrix whose underlying graph is unicyclic. The algorithm can be applied to vertex- and/or edge-weighted or unweighted unicyclic graphs. We apply the algorithm to obtain some general results on the spectrum of a generalized sun graph for certain matrix representations which include the Laplacian, normalized Laplacian and signless Laplacian matrices.

Keywords: symmetric matrix, eigenvalue location, unicyclic graph.

1 INTRODUCTION

Given a real symmetric matrix M we consider the *underlying graph* of M, i.e., the simple graph $G(M)$ whose vertices correspond to rows (or columns) of M with two vertices v_i and v_j adjacent whenever the (i, j)th entry of M is nonzero. For the purpose of this work each diagonal entry m_{ii} of M is assigned as a weight to the vertex v_i of $G(M)$ and each off-diagonal entry m_{ij} is assigned as an edge weight to the edge e_{ij} connecting vertices v_i and v_j of $G(M)$. Thus the underlying graph of M is seen as a vertex- and edge-weighted graph. We also say that a graph is *unweighted* if no real numbers are assigned to its edges or vertices.

Recall that the *adjacency matrix* $A = (a_{ij})$ of an unweighted simple graph G is the real symmetric matrix of order n with entries 0 or 1, where $a_{ij} = 1$ if and only if vertices v_i and v_j are adjacent. For an edge-weighted simple graph G, if vertices v_i and v_j are adjacent then $a_{ij} = \omega_{ij}$, where ω_{ij} is the weight of edge e_{ij}. Clearly, considering an unweighted graph as a graph where all edges have weight 1, any unweighted or edge-weighted simple graph G is the underlying graph of its adjacency matrix. The same is true for other matrices largely used to represent graphs like *Laplacian* (L), *normalized Laplacian* ($\mathcal{L}$) and *signless Laplacian* (Q).

*Corresponding author: Rodrigo Orsini Braga – E-mail: rbraga@ufrgs.br
[1]Departamento de Matemática Pura e Aplicada, IME - UFRGS, Av. Bento Gonçalves, 9500, 91509-900 Porto Alegre, RS, Brasil – E-mail: rbraga@ufrgs.br https://orcid.org/0000-0002-6156-2213
[2]Departamento de Matemática Pura e Aplicada, IME - UFRGS, Av. Bento Gonçalves, 9500, 91509-900 Porto Alegre, RS, Brasil – E-mail: vrodrig@mat.ufrgs.br https://orcid.org/0000-0002-1964-3327
[3]Departamento de Matemática Pura e Aplicada, IME - UFRGS, Av. Bento Gonçalves, 9500, 91509-900 Porto Alegre, RS, Brasil – E-mail: rafa.2604@hotmail.com https://orcid.org/0000-0001-6996-2142

The goal of this research is to present a linear-time algorithm to determine the number of eigenvalues of M in a given real interval in the case $G(M)$ is a *unicyclic graph*, that is, a connected graph containing exactly one cycle. In particular M could be any of the matrices mentioned above when they represent a unicyclic graph.

Algorithms for locating eigenvalues of some families of graphs have been developed over the last decade (see [4] and the references therein), starting with the algorithm for trees represented by their adjacency matrices developed by Jacobs and Trevisan in 2011 [7]. The applications of these algorithms have shown that they are a very useful tool in Spectral Graph Theory. Particularly, in 2017 Braga, Rodrigues and Trevisan [4] presented an algorithm that computes the number of eigenvalues in a given real interval of an unweighted unicyclic graph represented by its adjacency matrix, called *DiagonalizeUnicyclic*. Their algorithm was the main tool applied in [2] to reveal infinite families of *integral* unicyclic graphs, i.e. unicyclic graphs where all the eigenvalues of their adjacency matrices are integers.

Also in 2017, Braga and Rodrigues [3] developed an algorithm to locate eigenvalues of any *perturbed Laplacian* matrix of an edge-weighted tree, called *DiagonalizeW*. Given a real diagonal matrix D, the perturbed Laplacian matrix of a simple graph G with respect to D is the matrix $L^D(G) = D - A$, where $A = (a_{ij})$ is the adjacency matrix of G. The adjacency, Laplacian, normalized Laplacian and signless Laplacian matrices, among others, are examples of perturbed Laplacian matrices. Thus the algorithm DiagonalizeW has the advantage of generalizing previous localization algorithms for trees. We note that algorithm DiagonalizeW is in fact designed for any symmetric matrix whose underlying graph is a tree.

Recently, Belardo, Brunnetti and Trevisan [1] developed an algorithm for locating eigenvalues of *unbalanced unicyclic signed graphs*. A sign graph is a graph with edge-weights $+1$ or -1. Each edge weight is called a *signature* of the edge. A signed graph G is said to be unbalanced if there exists a cycle in G with an odd number of negatively signed edges.

In this work we extend the algorithm DiagonalizeUnicyclic to any symmetric matrix whose underlying graph is unicyclic. Hence our algorithm, called *DiagSMUnicyclic*, can be applied to unbalanced unicyclic signed graphs, also extending the work presented in [1]. We present algorithm DiagSMUnicyclic in Section 2. Besides preserving the practicality of the previous algorithms for locating eigenvalues of unicyclic graphs represented by their adjacency matrices, our method has the advantage of allowing vertex- and/or edge-weighted unicyclic graphs represented by any symmetric matrix whose underlying graph is unicyclic. In Section 3 we apply this method to obtain spectral properties of *generalized sun graphs* and give explicitly the spectrum of *balanced sun graphs* with respect to Laplacian, normalized Laplacian and signless Laplacian matrices.

2   ALGORITHM DIAGSMUNICYCLIC

In this section we present the algorithm *DiagSMUnicyclic* (Algorithm 1) that computes in a given real interval the number of eigenvalues of a symmetric matrix M whose underlying graph $G(M)$ is unicyclic. For a fixed real number α, the algorithm computes the diagonal values of a diagonal

matrix D_α *congruent* to $M + \alpha I$. We recall that two square matrices A and B of order n are *congruent* if there is an invertible matrix P such that $A = P^T B P$.

Algorithm DiagSMUnicyclic is a linear-time algorithm executed directly on the underlying unicyclic graph $G = G(M)$ so that the matrix M is not needed explicitly. At the end of the execution, the values $a(v)$ assigned to the each vertex v of G are the diagonal entries of the matrix D_α.

Algorithm 1 Algorithm $DiagSMUnicyclic(G, \alpha)$

```
Input: underlying unicyclic graph G = G(M) with ordered
       vertices v₁,...,vₙ and scalar α
Output: diagonal values a(v₁),...,a(vₙ)
```

1. Set $a(v_i) := \delta_i + \alpha$, where δ_i is the weight of vertex v_i, for all i.
2. Apply DiagonalizeW(T_k, α) (Algorithm 2) to each tree T_k, $1 \leq k \leq b$, skipping the initialization step already performed.
   If v_k has a child with zero value remove the edges on the cycle adjacent to v_k, if they haven't already been removed.
3. If an edge $v_{k-1}v_k$, for some $1 \leq k \leq b$, was removed in Step 2, then apply algorithm DiagonalizeW(P, α) to each path P that is not an isolated vertex choosing the endpoint of P with smaller index as the root.
4. If the cycle C_b was not disconnected in Step 2, then
   set $a_k := a(v_k)$, for $1 \leq k \leq b$;
   apply procedure DiagSMCycle($a_1, a_2, ..., a_b$) (Algorithm 3).

The general idea of the algorithm is to consider the unicyclic graph G as formed by a cycle C_b with pendant trees attached to the nodes of the cycle. Initially, the vertices of the cycle C_b are ordered $v_1, v_2, \cdots, v_b$. The remaining $n - b$ vertices of G are ordered considering vertex v_i, for $1 \leq i \leq b$, the root of the pendant tree T_i, which is the largest subgraph of G that is a tree and is connected to the cycle C_b at v_i. The vertices of T_i are ordered so that if v_j is a child of v_k, then $j > k$. We first order the vertices of T_1, then the vertices of T_2 and so on, until T_b.

We start algorithm DiagSMUnicyclic by assigning all vertices the weights $a(v_i) := \delta_i + \alpha$, where δ_i is the initial weight of vertex v_i of $G(M)$ (Step 1, Algorithm 1). Then the algorithm processes the vertices of each tree T_k bottom-up, towards the root v_k (Step 2, Algorithm 1), applying algorithm *DiagonalizeW* (Algorithm 2). The set of all children of a vertex v_j is denoted by S_j. After the diagonalization of T_k if its root v_k has a child with zero value, then the edges on the cycle adjacent to v_k are removed if they haven't already been removed. Finally, the vertices of the cycle C_b, from v_b to v_1, are processed (Step 3 and 4, Algorithm 1).

Theorem 2.1. *For the vertex-and edge-weighted unicyclic graph $G = G(M)$ with vertices $v_1, \cdots, v_n$ and a real number α, the algorithm $DiagSMUnicyclic(G, \alpha)$ computes the diagonal entries of a matrix D_α congruent to $M + \alpha I$.*

Algorithm 2 *DiagonalizeW(T, α)* [3]

```
Input: underlying tree T = T(M) with ordered vertices
       v₁,...,vₙ and scalar α
Output: diagonal values a(v₁),...,a(vₙ)
```

Initialize $a(v_i) := \delta_i$, where δ_i is the weight of vertex v_i, for all i.

For $k = n$ to 1

    if v_k is not a leaf then

        1. if $a(v_i) \neq 0$, for all $v_i \in S_k$, then

$$a(v_k) \leftarrow a(v_k) - \sum_{v_i \in S_k} \frac{(\omega_{ik})^2}{a(v_i)}.$$

        2. if $a(v_i) = 0$ for some $v_i \in S_k$, then

            choose a vertex v_j in S_k such that $a(v_j) = 0$;

$$a(v_k) \leftarrow -\frac{(\omega_{jk})^2}{2}; \quad a(v_j) \leftarrow 2;$$

            if v_k has a parent v_ℓ, remove the edge $v_k v_\ell$.

Print $a(v_1), a(v_2), ..., a(v_n)$.

Proof. Step 1 of Algorithm 1 initializes every vertex v_i of G with the diagonal value $a(v_i) = \delta_i + \alpha$, which is precisely the entry ii of $M + \alpha I$. We can arbitrarily order the vertices of G with $v_1, ..., v_b$ being the vertices of the unique cycle C_b in G, with v_k as the root of the pendant tree T_k, for $k = 1, ..., b$. We can also order the remaining vertices of G numbering first the vertices of T_1, then the vertices of T_2, and so on. Hence, matrix $M + \alpha I$ can be described as below

$$\begin{bmatrix} B & E_1 & E_2 & \cdots & E_b \\ (E_1)^T & B_1 & 0 & \cdots & 0 \\ (E_2)^T & 0 & B_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (E_b)^T & 0 & 0 & \cdots & B_b \end{bmatrix},$$

where B is $b \times b$ block of $M + \alpha I$ corresponding to the vertices of cycle C_b, B_k is the $|T_k - v_k| \times |T_k - v_k|$ block of $M + \alpha I$ corresponding to the vertices of $T_k - v_k$, for $k = 1, ..., b$, and E_k is the $b \times |T_k - v_k|$ rectangular block of $M + \alpha I$ with all entries zero except for the entries of line k corresponding to the adjacencies between v_k and its children in S_k, for $k = 1, ..., b$.

Since each T_k is a tree, we can apply algorithm *DiagonalizeW(T_k, α)*, for $k = 1, ..., b$. By [3, Theorem 2.1], the algorithm produces b diagonal matrices D_k congruent to B_k, for $k = 1, ..., b$. We recall that the second step of the algorithms DiagSMUnicyclic(G, α) and

---

Algorithm 3 Procedure $DiagSMCycle(a_1, a_2, ..., a_b)$

Set $TriangDiagonalized :=$ `false`; $z_2 := \omega_{12}$, $z_b := \omega_{1b}$, and $z_j := 0$, $3 \le j \le b-1$.

For $i = b$ to 3 do the following

    If $a_i \ne 0$ then

$$a_{i-1} \leftarrow a_{i-1} - \frac{(\omega_{i,i-1})^2}{a_i};\ a_1 \leftarrow a_1 - \frac{(z_i)^2}{a_i};\ z_{i-1} \leftarrow z_{i-1} - \frac{\omega_{i,i-1}}{a_i} z_i;\ \ //\text{case 1}$$

    else

$$\beta \leftarrow \frac{1}{2} \cdot \left(z_{i-1} - z_i \cdot \left(\frac{a_{i-1}}{2\,\omega_{i,i-1}} + \omega_{i,i-1} \right) \right); \gamma \leftarrow z_i \cdot \left(1 - \frac{a_{i-1}}{2(\omega_{i,i-1})^2} \right) + \frac{z_{i-1}}{\omega_{i,i-1}};$$

$$a_i \leftarrow 2;\quad a_{i-1} \leftarrow -\frac{(\omega_{i,i-1})^2}{2};\quad a_1 \leftarrow a_1 + 2\beta^2 \frac{1}{(\omega_{i,i-1})^2} - \frac{\gamma^2}{2};\quad //\text{case 2}$$

        if $i \ge 4$ then

$$z_{i-2} \leftarrow z_{i-2} - \frac{\omega_{i-1,i-2}}{\omega_{i,i-1}} \cdot z_i;$$

$$i \leftarrow i - 1;$$

        else

$$TriangDiagonalized \leftarrow \texttt{true};$$

end loop.

If $TriangDiagonalized =$ `false` and $z_2 \ne 0$ then

    if $a_2 \ne 0$ then

$$a_1 \leftarrow a_1 - \frac{(z_2)^2}{a_2};$$

    else

$$a_2 \leftarrow 2;\quad a_1 \leftarrow -\frac{(z_2)^2}{2}.$$

Return $(a_1, a_2, ..., a_b)$.

DiagonalizeUnicyclic(G, α) in [4] are exactly the same. Thus, we refer the reader to the proof of Theorem 2.2 in [4] to see that Step 2 transforms the matrix $M + \alpha I$ in the congruent matrix

$$\begin{bmatrix} \tilde{B} & 0 & 0 & \cdots & 0 \\ 0 & D_1 & 0 & \cdots & 0 \\ 0 & 0 & D_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & D_b \end{bmatrix},$$

where the $b \times b$ block $\tilde{B}$ represent the remaining adjacencies between the vertices of the original cycle C_b.

To complete the diagonalization process of $M + \alpha I$, it remains to perform the diagonalization of the block $\tilde{B}$.

If an edge of the cycle C_b was removed in Step 2, the original cycle will be now a single path or a union of disjoint paths. In this case, the algorithm for trees is applied to each path that is not an

isolated vertex, choosing the endpoint of the path with smaller index as the root of the path (Step 3). By [3, Theorem 2.1], this step completes the diagonalization process of the matrix.

If the cycle C_b was not disconnected by the application of algorithm DiagonalizeW (Algorithm 2) to each pendant tree during the execution of Step 2, then, in Step 4, procedure DiagSMCycle (Algorithm 3) is executed to diagonalize the block $\tilde{B}$.

This procedure denotes by z_i, for $1 = 2, ..., b$, the entries $(i,1)$ and $(1,i)$ of the original matrix $M + \alpha I$, which represent the adjacency between v_1 and v_i. Thus, initially, $z_2 = \omega_{12}$, $z_b = \omega_{1b}$ and $z_j = 0$, for $3 \le j \le b - 1$. It also writes $a_i = a(v_i)$, for all $i = 1, ..., b$ for simplicity. For fixed j, with $3 \le j \le b$, supposing that the rows and columns of $\tilde{B}$ from $j+1$ to b were diagonalized, let us diagonalize row j and column j of $\tilde{B}$, which has the form:

$$
\begin{array}{c}
1 \\
\vdots \\
j-1 \\
j \\
\vdots \\
b
\end{array}
\begin{bmatrix}
a_1 & \cdots & z_{j-1} & z_j & \cdots & z_b \\
\vdots & \ddots & \vdots & \vdots & & \vdots \\
z_{j-1} & \cdots & a_{j-1} & \omega_{j,j-1} & \cdots & 0 \\
z_j & \cdots & \omega_{j,j-1} & a_j & \cdots & 0 \\
\vdots & & \vdots & \vdots & \ddots & \vdots \\
z_b & \cdots & 0 & 0 & \cdots & a_b
\end{bmatrix}.
$$

If $a_j \ne 0$, then the following row and column operations

$$
R_{j-1} \leftarrow R_{j-1} - \frac{\omega_{j,j-1}}{a_j} R_j \qquad C_{j-1} \leftarrow C_{j-1} - \frac{\omega_{j,j-1}}{a_j} C_j
$$

$$
R_1 \leftarrow R_1 - \frac{z_j}{a_j} R_j \qquad C_1 \leftarrow C_1 - \frac{z_j}{a_j} C_j
$$

yield the congruent matrix

$$
\begin{array}{c}
1 \\
\vdots \\
j-1 \\
j \\
\vdots \\
b
\end{array}
\begin{bmatrix}
\tilde{a}_1 & \cdots & \tilde{z}_{j-1} & 0 & \cdots & 0 \\
\vdots & \ddots & \vdots & \vdots & & \vdots \\
\tilde{z}_{j-1} & \cdots & \tilde{a}_{j-1} & 0 & \cdots & 0 \\
0 & \cdots & 0 & a_j & \cdots & 0 \\
\vdots & & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & 0 & \cdots & a_b
\end{bmatrix},
$$

where $\tilde{a}_{j-1} = a_{j-1} - \frac{(\omega_{j,j-1})^2}{a_j}$, $\tilde{z}_{j-1} = z_{j-1} - \frac{\omega_{j,j-1}}{a_j} z_j$ and $\tilde{a}_1 = a_1 - \frac{(z_j)^2}{a_j}$. This process is repeated while $a_j \ne 0$, according to case 1 of Algorithm 3.

Suppose that $a_j = 0$, for some $3 \le j \le b$. If $j \ge 4$, the following operations

$$
R_{j-2} \leftarrow R_{j-2} - \frac{\omega_{j-2,j-1}}{\omega_{j,j-1}} R_j \quad \text{and} \quad C_{j-2} \leftarrow C_{j-2} - \frac{\omega_{j-2,j-1}}{\omega_{j,j-1}} C_j.
$$

annihilate the two entries representing the edge between v_{j-1} and v_{j-2}. These operations also change the value of z_{j-2} to $\tilde{z}_{j-2} = z_{j-2} - \frac{\omega_{j-2,j-1}}{\omega_{j,j-1}} z_j$. Next, the operations

$$
\begin{aligned}
R_{j-1} &\leftarrow R_{j-1} - \frac{a_{j-1}}{2\,\omega_{j,j-1}} R_j & C_{j-1} &\leftarrow C_{j-1} - \frac{a_{j-1}}{2\,\omega_{j,j-1}} C_j \\
R_j &\leftarrow R_j + \frac{1}{\omega_{j,j-1}} R_{j-1} & C_j &\leftarrow C_j + \frac{1}{\omega_{j,j-1}} C_{j-1} \\
R_{j-1} &\leftarrow R_{j-1} - \frac{\omega_{j,j-1}}{2} R_j & C_{j-1} &\leftarrow C_{j-1} - \frac{\omega_{j,j-1}}{2} C_j
\end{aligned}
$$

produce the submatrix

$$
\begin{array}{c}
1 \\ \vdots \\ j-2 \\ j-1 \\ j \\ \vdots \\ b
\end{array}
\left[
\begin{array}{cccccccc}
a_1 & \cdots & \tilde{z}_{j-2} & \beta & \gamma & \cdots & 0 \\
\vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\
\tilde{z}_{j-2} & \cdots & a_{j-2} & 0 & 0 & \cdots & 0 \\
\beta & \cdots & 0 & -\frac{(\omega_{j,j-1})^2}{2} & 0 & \cdots & 0 \\
\gamma & \cdots & 0 & 0 & 2 & \cdots & 0 \\
\vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & 0 & 0 & \cdots & a_b
\end{array}
\right] ,
$$

where

$$
\beta = \frac{1}{2} \cdot \left(z_{j-1} - z_j \cdot \left(\frac{a_{j-1}}{2\,\omega_{j,j-1}} + \omega_{j,j-1} \right) \right)
$$

and

$$
\gamma = z_j \cdot \left(1 - \frac{a_{j-1}}{2(\omega_{j,j-1})^2} \right) + \frac{z_{j-1}}{\omega_{j,j-1}}.
$$

Finally, the operations

$$
\begin{aligned}
R_1 &\leftarrow R_1 + 2\beta \frac{1}{(\omega_{j,j-1})^2} R_{j-1} & C_1 &\leftarrow C_1 + 2\beta \frac{1}{(\omega_{j,j-1})^2} C_{j-1} \\
R_1 &\leftarrow R_1 - \frac{\gamma}{2} R_j & C_1 &\leftarrow C_1 - \frac{\gamma}{2} C_j
\end{aligned}
$$

change the value of a_1 to $\tilde{a}_1 = a_1 + 2\beta^2 \frac{1}{(\omega_{j,j-1})^2} - \frac{\gamma^2}{2}$, yielding the submatrix

$$
\begin{array}{c}
1 \\ \vdots \\ j-2 \\ j-1 \\ j \\ \vdots \\ b
\end{array}
\left[
\begin{array}{cccccccc}
\tilde{a}_1 & \cdots & \tilde{z}_{j-2} & 0 & 0 & \cdots & 0 \\
\vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\
\tilde{z}_{j-2} & \cdots & a_{j-2} & 0 & 0 & \cdots & 0 \\
0 & \cdots & 0 & -\frac{(\omega_{j,j-1})^2}{2} & 0 & \cdots & 0 \\
0 & \cdots & 0 & 0 & 2 & \cdots & 0 \\
\vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & 0 & 0 & \cdots & a_b
\end{array}
\right] .
$$

At this point, both rows (and columns) j and $j-1$ are diagonalized, according to case 2 of Algorithm 3. We also observe that if $j = 3$ and $a_3 = 0$, the procedure changes the value of the boolean

variable *TriangDiagonalized* to true and the procedure ends. In this case, the diagonalization process is completed.

The proof that Algorithm 3 completes the diagonalization of $\tilde{B}$ when $j = 3$ and the boolean variable *TriangDiagonalized* remains false is identical to the proof of Theorem 2.2 in [4] and it will be omitted. □

Applying Theorem 2.1 and Sylvester's Law of Inertia, we obtain the following result.

Theorem 2.2. *Given a scalar* α *and a real symmetric* $n \times n$ *matrix* M *whose underlying graph* G *is unicyclic, let* D_α *be the diagonal matrix produced by the algorithm* *DiagSMUnicyclic*$(G(M), -\alpha)$. *Then the number of positive, negative and zero diagonal entries in* D_α *is equal to the number of eigenvalues of* M *which are greater, smaller and equal to* α, *respectively.*

Figure 1: Unicyclic graph G of Example 2.1.

Example 2.1. *Let be* G *the unweighted unicyclic graph of Figure 1. We will apply Algorithm 1 to determine the number of eigenvalues of the Laplacian matrix* L *of* G *in the interval* $(1,3)$. *Recall that the Laplacian matrix of an unweighted simple graph* G *is the matrix* $L = D_G - A$, *where* D_G *is the diagonal matrix whose diagonal entry* ii *is the degree* d_i *of vertex* v_i *of* G *and* A *is the adjacency matrix of* G. *In this case, the vertex weights of the underlying graph* $G(L)$ *are the vertex degrees and all edge weights are equal to* -1, *as shown in Figure 2.*

Next we will apply Algorithm 1 for $G(L)$ *and scalar* $\alpha = -3$ *to determine the number of eigenvalues of* L *that are grater, smaller or equal to* 3. *The left side of Figure 3 shows the initialization of the vertices with the value* $a(v) = \delta_v + \alpha = d_v - 3$ *(Step 1 of Algorithm 1).*

At Step 2, *since* v_9 *and* v_{12} *have the same values and two children, then their values become* $a(v_9) = a(v_{12}) = 0 - \left(\frac{1}{-2} + \frac{1}{-2}\right) = 1$. *Vertex* v_6 *has only one child, its value becomes* $a(v_6) = -1 - \frac{1}{-2} = -\frac{1}{2}$. *In the following, both* v_3 *and* v_4 *have only one child with diagonal value 1, hence their values are updated to* $a(v_3) = a(v_4) = 0 - \frac{1}{1} = -1$. *Since* v_2 *has two children, its value is*
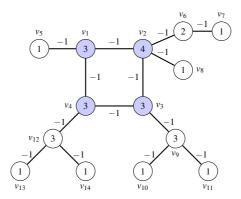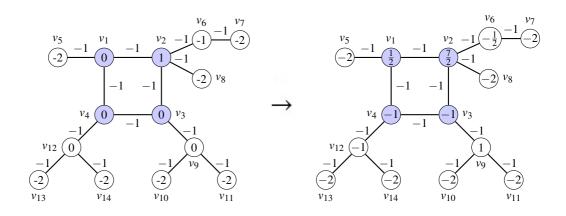
Figure 2: Vertex-and edge-weighted graph $G(L)$.

Figure 3: Step 1 and 2 of Algorithm 1 with $\alpha = -3$.

updated to $a(v_2) = 1 - \left(\frac{1}{-\frac{1}{2}} + \frac{1}{-2} \right) = \frac{7}{2}$. The value of vertex v_1 is updated to $a(v_1) = 0 - \frac{1}{-2} = \frac{1}{2}$. The right side of Figure 3 shows the diagonal values at the end of Step 2.

Since no vertex of the cycle has a child with a zero diagonal value, none of the edges of the cycle are removed in Step 2. Thus Algorithm 1 executes Algorithm 3 in Step 4. Figure 4 shows the current values of the vertices of the cycle after Step 2 is executed and the values z_j, for $j = 2, ..., 4$. The dashed line in Figure 4 represents that there is no edge between v_1 and v_3 but it will be changed during the execution of Algorithm 3.

In each iteration i, from $i = 4$ to 2, Algorithm 3 processes the vertices v_{i-1} and v_1, removing the edges that connect them to v_i. Since $a_4 = -1 \neq 0$, the diagonal value of v_3 becomes $a_3 = -1 - \frac{1}{a_4} = -1 - \frac{1}{-1} = 0$, the diagonal value of v_1 is changed to $a_1 = \frac{1}{2} - \frac{(z_4)^2}{a_4} = \frac{1}{2} - \frac{(-1)^2}{-1} = \frac{3}{2}$ and
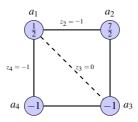
Figure 4: Initialization of procedure *DiagSMCycle* (Algorithm 3).

the value of z_3 becomes $z_3 = 0 + \frac{z_4}{a_4} = 0 + \frac{-1}{-1} = 1$. Figure 5 shows the results of the computations at this step of the procedure.
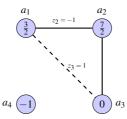

Figure 5: Iteration $i = 4$ of procedure *DiagSMCycle*.

Now, since $a_3 = 0$, the algorithm computes the values β and γ, obtaining

$$\beta = \frac{1}{2}\left(z_2 + z_3\left(1 + \frac{a_2}{2}\right)\right) = \frac{1}{2}\left(-1 + 1\left(1 + \frac{7}{2}\cdot\frac{1}{2}\right)\right) = \frac{7}{8};$$

$$\gamma = z_3\left(1 - \frac{a_2}{2}\right) - z_2 = 1\left(1 - \frac{7}{2}\cdot\frac{1}{2}\right) - (-1) = \frac{1}{4}$$

and then it assigns: $a_3 = 2$, $a_2 = -\frac{1}{2}$ and

$$a_1 = \frac{3}{2} + 2\beta^2 - \frac{\gamma^2}{2} = \frac{3}{2} + 2\left(\frac{7}{8}\right)^2 - \frac{1}{2}\left(\frac{1}{4}\right)^2 = 3.$$

Figure 6 shows the values obtained up to this point.

Since case 2 of Algorithm 3 is executed and $i = 3$, $TriangDiagonalized = true$. Therefore the procedure ends. Figure 7 shows the diagonal values at the end of the algorithm for $\alpha = -3$. As there are four positive diagonal values and all the other ten are negative, by Theorem 2.1, we conclude that four Laplacian eigenvalues of G are greater than 3 and all the other ones are smaller than 3.

To obtain the number of Laplacian eigenvalues in the interval $(1,3)$, we will apply algorithm DiagSMUnicyclic twice. Applying the algorithm for $G(L)$ with scalar $\alpha = -1$, we get that
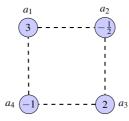
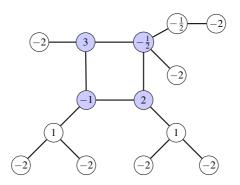Figure 6: Iteration $i = 3$ of procedure *DiagSMCycle*.

Figure 7: Final diagonal values for $G(L)$ and $\alpha = -3$.

seven Laplacian eigenvalues are greater than 1, *five Laplacian eigenvalues are smaller than* 1 *and two Laplacian eigenvalues are equal to* 1. *Therefore, we have three Laplacian eigenvalues in the interval* $(1,3)$. *In fact, the Laplacian spectrum of G is given by, approximately,* $\{0, 0.199, 0.231, 0.472, 0.74, 1, 1, 1.479,$
$2.351, 2.785, 3.363, 4.173, 4.445, 5.759\}$.

3   GENERALIZED SUN GRAPHS

In this section we apply algorithm DiagSMUnicyclic to obtain spectral properties of matrices L, $\mathscr{L}$ and Q of an infinite family of unicyclic graphs. Recall that the *signless Laplacian matrix* of an unweighted simple graph G is the matrix $Q = D_G + A$ and the *normalized Laplacian matrix* of G is the matrix $\mathscr{L} = (\ell_{ij})$, where $\ell_{ij} = 1$ if $i = j$ and $d_i > 0$, $\ell_{ij} = -\frac{1}{\sqrt{d_i \cdot d_j}}$, if vertices v_i and v_j are adjacent, and $\ell_{ij} = 0$, otherwise.

The *generalized sun graph* $\mathscr{S}(k_1, k_2, ..., k_b)$ is the graph obtained by taking a cycle C_b and adding $k_i \geq 1$ pendants to each node v_i of the cycle. If $k_1 = \cdots = k_b = p$, we say that $\mathscr{S}(p, p, ..., p)$ is a *balanced sun graph* and denote it by $\mathscr{S}_{b,p}$. Figure 8 illustrates graph $\mathscr{S}_{6,3}$.
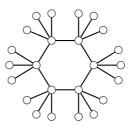
Figure 8: Balanced sun graph $\mathscr{S}_{6,3}$.

Next we present a general result for certain symmetric matrices whose underlying graph is a generalized sun graph.

Theorem 3.3. *Let $\mathscr{S}$ be a generalized sun graph with n vertices and cycle of length b and let β be a real number. If M is a symmetric matrix whose underlying graph is $\mathscr{S}$ and all diagonal entries of M corresponding to the pendants of $\mathscr{S}$ are equal to β, then M has b eigenvalues greater than β, b eigenvalues smaller than β and $n - 2b$ eigenvalues equal to β.*

Proof. Let us apply Algorithm 1 for $\mathscr{S}(M)$ with scalar $\alpha = -\beta$. Since every pendant is initialized with $\beta + \alpha = 0$, all the b nodes of the cycle receive a value $-\frac{(\omega_{ij})^2}{2} < 0$, exactly one child of each node receives value 2 and the other $n - 2b$ vertices remain with a zero value. Besides, all edges in the cycle are removed and the algorithm ends. Therefore, by Theorem 2.2, M has b eigenvalues greater than β, b eigenvalues smaller than β and and $n - 2b$ eigenvalues equal to β.
□

It is easy to see that the matrices L, Q and $\mathscr{L}$ satisfy the hypotheses of the Theorem 3.3 for $\beta = 1$. Applying this result together with Algorithm 1 we can obtain the spectrum of a balanced sun graph $\mathscr{S}_{b,p}$ in terms of the spectrum of the cycle C_b for these three matrices.

Theorem 3.4. *The spectra of the balanced sun graph $\mathscr{S}_{b,p}$ for the matrices L, Q and $\mathscr{L}$ are given in Table 1, where $i = 1, \ldots, b$ and $n = b(p+1)$ is the order of $\mathscr{S}_{b,p}$. In particular, if $i = b$, the corresponding Laplacian eigenvalues are 0 and $p+1$, and the normalized Laplacian eigenvalues are 0 and $\frac{2(p+1)}{p+2}$.*

Proof. By Theorem 3.3, 1 is a Laplacian eigenvalue of $\mathscr{S}_{b,p}$ with multiplicity $n - 2b$. To obtain the other $2b$ Laplacian eigenvalues, let us apply Algorithm 1 for $\mathscr{S}_{b,p}(L)$ with scalar $\alpha = -\lambda$, where $\lambda \neq 1$ is a Laplacian eigenvalue of $\mathscr{S}_{b,p}$. Since λ is an eigenvalue, the algorithm will produce at least one zero diagonal value.

Initially all vertices are assigned with diagonal value $d(v) - \lambda$. Since each node v_i of the cycle is adjacent to exactly p pendants and two other vertices of the cycle, then the algorithm assigns to v_i the value

$$a(v_i) = p + 2 - \lambda - \frac{p}{1 - \lambda},$$

Matrix	Spectrum
L	$\left\{1^{n-2b},\ \dfrac{p+\mu_i+1}{2}\pm\sqrt{\left(\dfrac{p+\mu_i+1}{2}\right)^2-\mu_i}\right\}$, where $\mu_i = 2 - 2\cos\left(\dfrac{2\pi i}{b}\right)$
Q	$\left\{1^{n-2b},\ \dfrac{p+\mu_i+1}{2}\pm\sqrt{\left(\dfrac{p+\mu_i+1}{2}\right)^2-\mu_i}\right\}$, where $\mu_i = 2 + 2\cos\left(\dfrac{2\pi i}{b}\right)$
$\mathscr{L}$	$\left\{1^{n-2b},\ \dfrac{\mu_i+1}{2}\pm\sqrt{\left(\dfrac{\mu_i-1}{2}\right)^2+\dfrac{p}{p+2}}\right\}$, where $\mu_i = 1 - \dfrac{2}{p+2}\cos\left(\dfrac{2\pi i}{b}\right)$

Table 1: The spectra of the $\mathscr{S}_{b,p}$ for matrices L, Q and $\mathscr{L}$

for $i = 1, \cdots, b$. At this point, since all $a(v_i)$ are equal, we can consider that we are applying the algorithm with scalar β to the cycle C_b, where $\beta = a(v_1) = \cdots = a(v_b)$. Thus, by Theorem 2.2, the algorithm will produce a zero diagonal value if and only if $\beta - 2 = -\mu$, where μ is a Laplacian eigenvalue of the cycle C_b. It is well-known that the Laplacian eigenvalues of cycle C_b are given by $2 - 2\cos\left(\frac{2\pi i}{n}\right)$, for $i = 1, \cdots, b$. Hence, $\lambda \neq 1$ is an eigenvalue of $\mathscr{S}_{b,p}$ if and only if $2 + p - \lambda - \frac{p}{1-\lambda} - 2 = -\mu$, where μ is a Laplacian eigenvalue of C_b. Besides, note that

$$2 + p - \lambda - \frac{p}{1-\lambda} - 2 = -\mu \quad \Leftrightarrow \quad \lambda^2 - (p+\mu+1)\lambda + \mu = 0$$

$$\Leftrightarrow \quad \lambda = \frac{p+\mu+1}{2} \pm \sqrt{\left(\frac{p+\mu+1}{2}\right)^2 - \mu}.$$

The proof for the signless Laplacian matrix Q is analogous. For the normalized Laplacian matrix $\mathscr{L}$, μ is an eigenvalue of the $b \times b$ matrix

$$
\begin{bmatrix}
1 & -\frac{1}{p+2} & 0 & 0 & \cdots & 0 & -\frac{1}{p+2} \\
-\frac{1}{p+2} & 1 & -\frac{1}{p+2} & 0 & \cdots & 0 & 0 \\
0 & -\frac{1}{p+2} & 1 & -\frac{1}{p+2} & \cdots & 0 & 0 \\
0 & \cdots & -\frac{1}{p+2} & 1 & -\frac{1}{p+2} & \cdots & 0 \\
\vdots & & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 1 & -\frac{1}{p+2} \\
-\frac{1}{p+2} & 0 & 0 & 0 & \cdots & -\frac{1}{p+2} & 1
\end{bmatrix},
$$

which is the normalized laplacian matrix of the cycle C_b replacing the entries $-\frac{1}{2}$ that represent the adjacencies of the vertices in the cycle by the entries $-\frac{1}{p+2}$ that represent the adjacencies of the vertices $v_1, \ldots, v_b$ in the balanced sun graph $\mathscr{S}_{b,p}$. The matrix above is a circulant matrix,

whose spectrum is given by $\mu_i = 1 - \frac{2}{p+2}\cos(\frac{2\pi i}{b})$, $i = 1,\ldots,b$ (see [6]). Hence, $\lambda \neq 1$ is a normalized Laplacian eigenvalue of $\mathscr{S}_{b,p}$ if and only if $1 - \lambda - \frac{p}{(1-\lambda)(p+2)} - 1 = -\mu$. Besides, note that

$$1 - \lambda - \frac{p}{(1-\lambda)(p+2)} - 1 = -\mu \quad \Leftrightarrow \quad \lambda^2 - (\mu+1)\lambda + \mu - \frac{p}{p+2} = 0$$

$$\Leftrightarrow \quad \lambda = \frac{\mu+1}{2} \pm \frac{1}{2}\sqrt{(\mu-1)^2 + \frac{4p}{p+2}}.$$

□

Applying Theorem 3.4, we can easily derive an upper bound for the largest Laplacian eigenvalue of a balanced sun graph, which is at the same time the exact value of the largest signless Laplacian eigenvalue for the same graph.

Corollary 3.4.1. *If λ_1 is the largest eigenvalue of the Laplacian matrix of the balanced sun graph $\mathscr{S}_{b,p}$, then*

$$\lambda_1 \leqslant \frac{p+5+\sqrt{(p+1)(p+9)}}{2},$$

with equality if and only if b is even. Moreover, if θ_1 is the largest eigenvalue of the signless Laplacian matrix of $\mathscr{S}_{b,p}$, then

$$\theta_1 = \frac{p+5+\sqrt{(p+1)(p+9)}}{2}.$$

Proof. The Laplacian eigenvalues of the cycle C_b are given by $\mu_i = 2 - 2\cos\left(\frac{2\pi i}{b}\right)$, for $i = 1,\ldots,b$. Since $\cos\left(\frac{2\pi i}{b}\right) \geqslant -1$, then $\mu_i = 2 - 2\cos\left(\frac{2\pi i}{b}\right) \leq 4$. Besides, $\mu_{\lceil \frac{b}{2}\rceil} = \mu_{\lfloor \frac{b}{2}\rfloor}$ is the largest Laplacian eigenvalue of the cycle C_b. Since $p \geqslant 1$, $f(\mu,p) = \frac{p+\mu+1}{2} + \sqrt{\left(\frac{p+\mu+1}{2}\right)^2 - \mu}$ is an increasing function of $\mu \geq 0$. Thus,

$$\begin{aligned}
\lambda_1 &= \frac{p+\mu_{\lfloor \frac{b}{2}\rfloor}+1}{2} + \sqrt{\left(\frac{p+\mu_{\lfloor \frac{b}{2}\rfloor}+1}{2}\right)^2 - \mu_{\lfloor \frac{b}{2}\rfloor}} \\
&\leqslant \frac{p+5}{2} + \sqrt{\left(\frac{p+5}{2}\right)^2 - 4} \\
&= \frac{p+5}{2} + \frac{1}{2}\sqrt{(p+1)(p+9)}.
\end{aligned}$$

Besides, $\mu_i = 4$ if and only if $\cos(\frac{2\pi i}{b}) = -1$, for $i = 1,\ldots,b$. Since $\frac{2\pi i}{b} = \pi \Leftrightarrow i = \frac{b}{2}$, the maximum value is achieved if and only if b is even. The result for the signless Laplacian follows directly from the fact that the maximum is achieved when $i = b$ and hence $\mu_b = 4$.   □

Remark 1. *Analogously, we can also easily derive that $\frac{p+3}{p+2} + \frac{\sqrt{1+p(p+2)}}{p+2} = 2$ is an upper bound for the largest normalized Laplacian eigenvalue of $\mathscr{S}_{b,p}$, which is achieved if and only if b is even. In fact, it is actually a more general result that the normalized Laplacian eigenvalues of any simple graph belong to interval $[0,2]$ and 2 is the largest eigenvalue if and only if a connected component of G is bipartite and nontrivial [5, Lemma 1.7], which is precisely the case when the length of the cycle of a unicyclic graph is even. Besides, it is also known that the spectrum of the signless Laplacian matrix is equal to the Laplacian spectrum if and only if the graph is bipartite (see for example [8]).*

The Laplacian eigenvalues of any graph are nonnegative. Besides, 0 is always an eigenvalue and it is simple if and only if the graph is connected (see [8] for details). The second smallest Laplacian eigenvalue of a graph G is called the *algebraic connectivity* of G. Using Theorem 3.4, we can derive the following result.

Corollary 3.4.2. *The algebraic connectivity of the balanced sun graph $\mathscr{S}_{b,p}$, denoted by $a_{b,p}$, satisfies*

$$a_{b,p} \leqslant \frac{(p+4) - \sqrt{p^2 + 8p + 4}}{2}$$

with equality if and only if $b = 3$.

Proof. The algebraic connectivity of the cycle C_b is given by $\mu_{b-1} = 2 - 2\cos(\frac{2\pi}{b})$. Since $b \geqslant 3$, then $0 \leqslant \frac{2\pi}{b} \leqslant \frac{2\pi}{3}$ and hence $\mu_{b-1} = 2 - 2\cos\left(\frac{2\pi}{b}\right) \leqslant 3$. Therefore,

$$
\begin{aligned}
a_{b,p} &= \frac{p + \mu_{b-1} + 1}{2} - \sqrt{\left(\frac{p + \mu_{b-1} + 1}{2}\right)^2 - \mu_{b-1}} \\
&\leqslant \frac{p+4}{2} - \sqrt{\left(\frac{p+4}{2}\right)^2 - 3} \\
&= \frac{p+4}{2} - \frac{1}{2}\sqrt{p^2 + 8p + 4}.
\end{aligned}
$$

The equality follows from the fact that $\mu_{b-1} = 3$ if and only if $b = 3$. $\square$

4   CONCLUSION

The algorithm DiagSMUnicyclic presented in this work is an efficient tool to obtain spectral properties of symmetric matrices associated to a unicyclic graph. This procedure is an extension of the algorithms for the adjacency matrix of unweighted unicyclic graphs and unbalanced unicyclic signed graphs. It allows to work with vertex- and/or edge-weighted unicyclic graphs. Our method also has the advantage of allowing any symmetric matrix representation of an unicyclic graph provided that the underlying graph of the matrix is unicyclic, as illustrated by the spectral properties of generalized sun graphs obtained in Section 3.

Acknowledgements

REFERENCES

[1] F. Belardo, M. Brunetti & V. Trevisan. Locating eigenvalues of unbalanced unicyclic signed graphs. *Appl. Math. Comput.*, **400** (2021), 126082. doi:10.1016/j.amc.2021.126082. URL `https://doi.org/10.1016/j.amc.2021.126082`.

[2] R.O. Braga, R.R. Del-Vecchio & V.M. Rodrigues. Integral unicyclic graphs. *Linear Algebra Appl.*, **614** (2021), 281–300. doi:10.1016/j.laa.2020.05.002. URL `https://doi.org/10.1016/j.laa.2020.05.002`.

[3] R.O. Braga & V.M. Rodrigues. Locating eigenvalues of perturbed Laplacian matrices of trees. *TEMA Tend. Mat. Apl. Comput.*, **18**(3) (2017), 479–491. doi:10.5540/tema.2017.018.03.479. URL `https://doi.org/10.5540/tema.2017.018.03.479`.

[4] R.O. Braga, V.M. Rodrigues & V. Trevisan. Locating eigenvalues of unicyclic graphs. *Appl. Anal. Discrete Math.*, **11**(2) (2017), 273–298. doi:10.2298/AADM1702273B. URL `https://doi.org/10.2298/AADM1702273B`.

[5] F.R.K. Chung. "Spectral Graph Theory". American Mathematical Society (1997).

[6] P.J. Davis. "Circulant matrices". John Wiley & Sons, New York-Chichester-Brisbane (1979), xv+250 p. A Wiley-Interscience Publication, Pure and Applied Mathematics.

[7] D.P. Jacobs & V. Trevisan. Locating the eigenvalues of trees. *Linear Algebra Appl.*, **434**(1) (2011), 81–88. doi:10.1016/j.laa.2010.08.006. URL `https://doi.org/10.1016/j.laa.2010.08.006`.

[8] R. Merris. Laplacian matrices of graphs: a survey. volume 197/198 (1994), p. 143–176. doi:10.1016/0024-3795(94)90486-3. URL `https://doi.org/10.1016/0024-3795(94)90486-3`. Second Conference of the International Linear Algebra Society (ILAS) (Lisbon, 1992).