Applicable Analysis and Discrete Mathematics
available online at http://pefmath.etf.rs

Appl. Anal. Discrete Math. **11** (2017), 273–298.
https://doi.org/10.2298/AADM1702273B

LOCATING EIGENVALUES OF UNICYCLIC GRAPHS

*Rodrigo O. Braga, Virgínia M. Rodrigues, Vilmar Trevisan**

In honor of Dragoš Cvetković on the occasion of his 75th birthday.

We present a linear time algorithm that computes the number of eigenvalues of a unicyclic graph in a given real interval. It operates directly on the graph, so that the matrix is not needed explicitly. The algorithm is applied to study the multiplicities of eigenvalues of *closed caterpillars*, obtain the spectrum of *balanced closed caterpillars* and give sufficient conditions for these graphs to be non-integral. We also use our method to study the distribution of eigenvalues of unicyclic graphs formed by adding a fixed number of copies of a path to each node in a cycle. We show that they are not integral graphs.

1. INTRODUCTION

Let G be a simple undirected graph with vertices $v_1, ..., v_n$. The *adjacency matrix* $A = (a_{ij})$ of G is the $0 - 1$ matrix of order n, where $a_{ij} = 1$ if and only if v_i is adjacent to v_j. Since A is a real symmetric matrix, its eigenvalues are real numbers. The multiset of eigenvalues of A is called the *spectrum* of G. A graph G is *integral* if its spectrum consists entirely of integers. A connected graph without any cycles is called a tree, while a connected graph containing exactly one cycle is called *unicyclic*.

Jacobs and Trevisan [**9**] developed a simple algorithm, called *Diagonalize*, to compute the number of eigenvalues of any tree T lying in a given real interval, in linear time. Operating directly on the tree, their algorithm computes the diagonal values of a matrix congruent to $A+\alpha I$, where A is the adjacency matrix of the tree

*Corresponding author. Vilmar Trevisan
2010 Mathematics Subject Classification. 05C50, 05C85, 15A18.
Keywords and Phrases. Unicyclic graphs, localization of eigenvalues, closed caterpillars.

T, I is the identity matrix and α is a real number. Consequently, by Sylvester's Law of Inertia, the number of positive, negative or zero values obtained at the end of the algorithm are the number of eigenvalues of T greater than, smaller than or equal to α, respectively.

The algorithm Diagonalize and its extensions to other matrices became a very useful tool for studying spectral properties of trees. In [**9**], it was applied to study the multiplicities of the eigenvalues of *caterpillars*, which they considered as trees formed by taking a path P_b on $b \geqslant 2$ vertices and adding at least one pendant vertex to each node in the path. The authors also showed that under certain conditions a caterpillar is not integral. In [**6**], Fritscher *et al.* used the algorithm adapted for the *Laplacian matrix* of a tree, defined as $L = D - A$, where D is the diagonal matrix whose $(i,i)-$entry is the degree of vertex v_i and A is the adjacency matrix of the tree. They applied the algorithm to derive a new upper bound on the sum of the k largest Laplacian eigenvalues of a tree. This allowed them to prove that among all trees with n vertices, the star S_n has the highest Laplacian energy, which was conjectured by Radenković and Gutman [**11**]. In [**3**], Braga *et al.* applied the algorithm to the Laplacian matrix to study how the number of small Laplacian eigenvalues behaves when some vertex transformations are performed on a tree.

The algorithm Diagonalize was adapted in [**4**] to the *normalized Laplacian matrix* of a tree, defined by Chung [**5**] as the matrix $\mathcal{L}$ with rows and columns indexed by the vertices of a graph $G(V,E)$, whose entries are

$$\mathcal{L}(u,v) = \begin{cases} 1, & \text{if } u = v \text{ and } d_u > 0, \\ -\frac{1}{\sqrt{d_u \cdot d_v}}, & \text{if } \{u,v\} \in E, \\ 0, & \text{otherwise}, \end{cases}$$

where d_v is the degree of a vertex $v \in V$. This variation of the algorithm was applied in [**4**] to study the multiplicity of normalized Laplacian eigenvalues of small diameter trees, which allowed the authors to characterize the trees that have 4 or 5 normalized Laplacian eigenvalues.

A natural extension of the algorithm for locating eigenvalues of trees is to consider unicyclic graphs. This is the aim of this work, where we present, in Section 2, an algorithm to determine the number of eigenvalues of a unicyclic graph in a given real interval. The algorithm is linear in n, the number of vertices, and is based on the diagonalization of $A+\alpha I$. The proposed method is simple and can be implemented on the graph itself, meaning that the necessary storage is also linear in n. Examples to illustrate the algorithm are given in Section 3.

In Section 4 we apply our algorithm to study the multiplicity of eigenvalues of *closed caterpillars*, which are graphs where the removal of all pendant vertices gives a cycle. For purposes of this paper, we consider that each vertex in the cycle of a closed caterpillar has at least one pendant vertex. We also give sufficient conditions for a closed caterpillar to be non-integral. In addition, we use the algorithm to obtain the spectrum of closed caterpillars where each node in the cycle has the same degree, called *balanced closed caterpillars*. Besides, we study the distribution of eigenvalues of a family of unicyclic graphs, formed by adding a fixed number of copies of the path P_ℓ to each node in a cycle C_b. We show that every graph in this family is not integral.

K. Balińska *et al.* noted in their survey [**1**] that integral graphs are very rare and difficult to find. In particular, G. Omidi [**10**] studied integral graphs with few cycles. He proved that an integral unicyclic graph with no eigenvalue 0 is either C_3 or C_6, and that there is no non-bipartite integral unicyclic graph with exactly one eigenvalue 0. He also noticed that C_4 is the smallest integral unicyclic graph with two eigenvalues 0. In Section 5 we present the result of a computer search for all integral unicyclic graphs up to 21 vertices. We found three of those graphs in this range, different from C_3, C_4 and C_6. Interestingly, two of them are nonisormorphic cospectral.

2. ALGORITHM FOR LOCATING EIGENVALUES OF UNICYCLIC GRAPHS

In this section we present the algorithm *DiagonalizeUnicyclic* that computes the number of eigenvalues of a unicyclic graph lying in a given real interval. For a unicyclic graph G of order n containing a cycle C_b and a real number α, the algorithm computes the diagonal values of a diagonal matrix D *congruent* to $A(G)+\alpha I$, where $A(G)$ is the adjacency matrix of G. The algorithm is executed directly on the unicyclic graph, so that the matrix is not needed explicitly. At the end of the execution the values $a(v)$ assigned to the vertices of G are the diagonal entries of the matrix D.

We recall that two real symmetric matrices B and C are congruent if there exists an invertible matrix P such that $B = P^TCP$. Since the matrix D produced by algorithm DiagonalizeUnicyclic is congruent to $A(G)+\alpha I$, by Sylvester's Law of Inertia (see [**7**, Theorem 4.5.8]) the number of positive values $a(v)$ assigned to the vertices is equal to the number of positive eigenvalues of $A(G)+\alpha I$, which are the eigenvalues of $A(G)$ that are greater than $-\alpha$. Likewise for the number of negative and zero values $a(v)$.

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. We apply the algorithm Diagonalize of Jacobs and Trevisan [**9**] to each pendant tree and then process the cycle.

Initially, the vertices of the cycle C_b are ordered $v_1, v_2, \ldots, v_b$. The remaining $n-b$ vertices of G are ordered considering vertex v_i, for $1 \le i \le b$, 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. Figure 1 shows a unicyclic graph containing a cycle C_3 and pendant trees T_1, T_2 and T_3.

Next we order the vertices of each tree T_i, for $i = 1, ..., b$. We say that the vertices of T_i adjacent to the root v_i are its children. The vertices adjacent to the children of the root are their own children, and so forth. Every vertex v of T_i, except for the root v_i, has a parent, that is the only neighbor of v that is not a child of v, and the only vertices that don't have children are the leaves. The vertices of

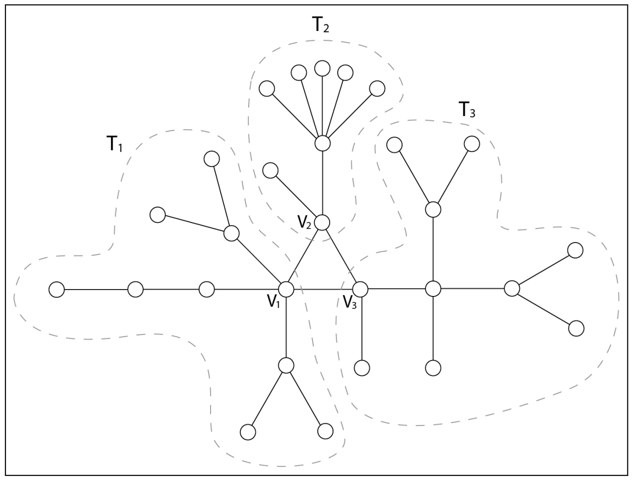


Figure 1: A unicyclic graph

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.

Figure 2 presents the algorithm Diagonalize that will be applied to process the vertices of the pendant trees T_i, which corresponds to diagonalize the portion of the matrix $A(G) + \alpha I$ corresponding to the T_i's, for $1 \leq i \leq b$. For a vertex v_k, we denote by S_k the set of all children of v_k.

```
Input:  tree T, scalar α
Output:  diagonal matrix D congruent to A + αI

Algorithm Diagonalize(T, α)
   set a(v) := α, for all vertices v of T
   order vertices so that if v_j is a child of v_k, then j > k
   for k = n to 1
         if v_k is not a leaf, then
               1.  if a(v) ≠ 0, for all v ∈ S_k, then
                     a(v_k) ← a(v_k) − Σ_{v∈S_k} 1/a(v);
               2.  if a(v) = 0 for some v ∈ S_k, then
                     choose a vertex v_j in S_k such that a(v_j) = 0;
                     a(v_k) ← −1/2;  a(v_j) ← 2;
                     remove the edge v_k v_ℓ if v_k has a parent v_ℓ.
```

Figure 2: Algorithm Diagonalize(T, α)

Theorem 2.1. *[**9**, Theorem 1] For inputs T, α, where T is a tree with adjacency* *matrix A, algorithm Diagonalize computes a diagonal matrix D, which is congruent to $A + \alpha I$.*

The algorithm DiagonalizeUnicyclic is presented in Figure 3. After applying the algorithm Diagonalize to the pendant trees, our method diagonalizes the block of the matrix $A(G) + \alpha I$ corresponding to the cycle C_b, whose vertices are processed with initial values that were computed by the algorithm Diagonalize.

We start our algorithm assigning all vertices the value $a(v) := \alpha$ (Step 1, Fig. 3). Then the algorithm processes the vertices of each tree T_i bottom-up, towards the root v_i (Step 2, Fig. 3). Note that, for each tree T_i, both vertices of the cycle C_b that are adjacent to v_i are considered its parents. Finally, the vertices of the cycle C_b, from v_b to v_1, are processed (Steps 3 and 4, Fig. 3).

```
Input:  unicyclic graph G, with ordered vertices v_1,...,v_n, scalar α
Output:  values a(v_1),...,a(v_n)

Algorithm DiagonalizeUnicyclic(G,α)
1.  Set a(v) := α, for all vertices v of G.
2.  Apply algorithm Diagonalize(T_i,α) to each tree T_i, for 1 ≤ i ≤ b,
     considering both v_{i-1} and v_{i+1} parents of v_i, where v_{i-1} = v_b,
     if i = 1, and v_{i+1} = v_1, if i = b.
3.  If an edge v_{i-1}v_i, for some 1 ≤ i ≤ b, was removed in Step 2, then
     apply algorithm Diagonalize(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_i := a(v_i), for 1 ≤ i ≤ b;
     apply procedure DiagCycle(a_1,a_2,...,a_b), described in Figure 4
```

Figure 3: Algorithm DiagonalizeUnicyclic(G, α)

Theorem 2.2. *Let G be a unicyclic graph of order n and let α be a real number. The algorithm DiagonalizeUnicyclic(G, α) computes the diagonal entries of a matrix D congruent to $A(G) + \alpha I$, where $A(G)$ is the adjacency matrix of G.*

Proof. We order the vertices of G as described above, with $v_1, ..., v_b$ the vertices of the cycle C_b of G, and v_i the root of the pendant tree T_i, for $1 \leq i \leq b$. In Step 1, algorithm DiagonalizeUnicyclic initializes every vertex v of G with the diagonal value $a(v) = \alpha$. Thus the matrix $A(G) + \alpha I$ is of the form

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

```
Procedure DiagCycle(a_1, a_2, ..., a_b)
   Set TriangDiagonalized := false; w_2 := 1, w_b := 1 and w_j := 0,
      for 3 ≤ j ≤ b − 1.
   For i = b to 3 do the following.
      If a_i ≠ 0 then
         a_{i−1} ← a_{i−1} − 1/a_i;  a_1 ← a_1 − (w_i)^2/a_i;  w_{i−1} ← w_{i−1} − w_i/a_i;   //case 1
      else
         β ← 1/2 · (w_{i−1} − w_i · (1 + a_{i−1}/2));  γ ← w_{i−1} + w_i · (1 − a_{i−1}/2);
         a_i ← 2;   a_{i−1} ← −1/2;   a_1 ← a_1 + 2β^2 − γ^2/2;              //case 2
         if i ⩾ 4 then
            w_{i−2} ← w_{i−2} − w_i;
            i ← i − 1;
         else
            TriangDiagonalized ← true;
   end loop.
   If TriangDiagonalized = false and w_2 ≠ 0 then
      if a_2 ≠ 0 then
         a_1 ← a_1 − (w_2)^2/a_2;
      else
         a_2 ← 2;   a_1 ← −(w_2)^2/2.
   Return (a_1, a_2, ..., a_b).
```

Figure 4: Procedure DiagCycle$(a_1, a_2, ..., a_b)$

where $B = A(C_b) + \alpha I_b$, with $A(C_b)$ the adjacency matrix of the cycle C_b; $B_i = A(T_i - v_i) + \alpha I_{|T_i - v_i|}$, where $T_i - v_i$ is the subgraph of G induced by the vertices of $T_i - v_i$ and $A(T_i - v_i)$ is its adjacency matrix, for $i = 1, ..., b$; and E_i is the $b \times |T_i - v_i|$ rectangular block of $A(G)$ with zero entries except for the entries of line i that are equal to one, which represent the adjacencies between v_i and its children in S_i, for $i = 1, ..., b$.

In Step 2 of algorithm DiagonalizeUnicyclic, since each T_i is a tree and there is no edge between $T_i - v_i$ and $T_j - v_j$, for $i \neq j$, when algorithm Diagonalize(T_i, α) processes the vertices of subgraph $T_i - v_i$, the adjacencies in E_i are not affected, for $i = 1, ..., b$. By Theorem 2.1, the algorithm produces b diagonal matrices D_i, for $1 \leq i \leq b$, congruent to B_i. Since there are no edges between the subgraphs $T_i - v_i$ and the cycle C_b, the block B remains the same, yielding the matrix

$$\begin{bmatrix} B & E_1 & E_2 & \cdots & E_b \\ (E_1)^T & D_1 & \mathbf{0} & \cdots & \mathbf{0} \\ (E_2)^T & \mathbf{0} & D_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (E_b)^T & \mathbf{0} & \mathbf{0} & \cdots & D_b \end{bmatrix}.$$

Then algorithm Diagonalize(T_i, α) processes the root v_i of pendant tree T_i, considering that v_{i-1} and v_{i+1} are the parents of v_i, where $v_{i-1} = v_b$, if $i = 1$, and $v_{i+1} = v_1$, if $i = b$. If $a(v) \neq 0$, for all $v \in S_i$, then algorithm Diagonalize(T_i, α) assigns to v_i the value

$$a(v_i) \leftarrow a(v_i) - \sum_{v \in S_i} \frac{1}{a(v)},$$

and the entries of E_i, which represent the adjacencies between v_i and its children in S_i, become zero.

If $a(v) = 0$, for some $v \in S_i$, algorithm Diagonalize(T_i, α) assigns to v_i and one of its zero children v_j the diagonal values $-\frac{1}{2}$ and 2, respectively, and the entries in E_i, which represent the adjacencies between v_i and its children in S_i, as well as the entries that represent the adjacencies between v_i and its parents v_{i+1} and v_{i-1}, become zero. To show that these operations in fact diagonalize this portion of the matrix, let us consider the submatrix below which shows the entries corresponding to v_j, v_i, v_{i+1} and v_{i-1}:

$$\begin{array}{c} \\ i-1 \\ i \\ i+1 \\ \\ j \\ \\ \end{array} \left[\begin{array}{ccccccc} & \vdots & & & & \vdots & \\ \cdots & \alpha & 1 & 0 & \cdots & 0 & \cdots \\ \cdots & 1 & \alpha & 1 & \cdots & 1 & \cdots \\ \cdots & 0 & 1 & \alpha & \cdots & 0 & \cdots \\ & \vdots & \vdots & \vdots & \ddots & \vdots & \\ \cdots & 0 & 1 & 0 & \cdots & 0 & \cdots \\ & \vdots & \vdots & \vdots & & \vdots & \end{array} \right].$$

For $k \in \{i-1, i+1\}$, the row and column operations

$$R_k \leftarrow R_k - R_j$$

and

$$C_k \leftarrow C_k - C_j,$$

annihilate the entries $(i-1, i)$ and $(i, i-1)$, when $k = i-1$, and annihilate the entries $(i+1, i)$ and $(i, i+1)$, when $k = i+1$, yielding the submatrix

$$\begin{array}{c} \\ i-1 \\ i \\ i+1 \\ \\ j \\ \\ \end{array} \left[\begin{array}{ccccccc} & \vdots & & & & \vdots & \\ \cdots & \alpha & 0 & 0 & \cdots & 0 & \cdots \\ \cdots & 0 & \alpha & 0 & \cdots & 1 & \cdots \\ \cdots & 0 & 0 & \alpha & \cdots & 0 & \cdots \\ & \vdots & \vdots & \vdots & \ddots & \vdots & \\ \cdots & 0 & 1 & 0 & \cdots & 0 & \cdots \\ & \vdots & \vdots & \vdots & & \vdots & \end{array} \right].$$

These operations effectively remove the edges between v_i and its parents v_{i-1} and v_{i+1} in the cycle $\mathcal{C}_b$, disconnecting the cycle if it was not disconnected before.

Performing the same operations for all $k \neq j$ such that v_k is a child of v_i in S_i, the entries (k, i) and (i, k) of the matrix are also annihilated. Finally, the following operations remove the connection between v_i and v_j, assigning to v_i and v_j the diagonal values $-1/2$ and 2, respectively.

$$\begin{array}{rcl} R_i & \leftarrow & R_i - \dfrac{\alpha}{2} R_j \\ C_i & \leftarrow & C_i - \dfrac{\alpha}{2} C_j \\ R_j & \leftarrow & R_j + R_i, \\ C_j & \leftarrow & C_j + C_i, \\ R_i & \leftarrow & R_i - \dfrac{1}{2} R_j, \\ C_i & \leftarrow & C_i - \dfrac{1}{2} C_j. \end{array}$$

Note that the children of v_i in S_i, including those with a zero diagonal value, are not affected by the operations above. This completes the diagonalization of the portion of the matrix that corresponds to the adjacencies between v_i and its neighbors, for $i = 1, \ldots, b$.

Therefore, at the end of Step 2, the algorithm DiagonalizeUnicyclic produces a matrix congruent to $A(G) + \alpha I$ that is diagonalized for all vertices v_j, with $b + 1 \leq j \leq n$:

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

where block $\tilde{B}$ of dimension $b \times b$ represents the remaining adjacencies among the vertices of the original cycle $\mathcal{C}_b$.

To complete the diagonalization process of $A(G) + \alpha I$, it remains to diagonalize the block $\tilde{B}$.

If an edge of the cycle C_b was removed in Step 2, we have 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 root (Step 3). By Theorem 2.1, this step completes the diagonalization of the matrix.

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

The general idea is to perform the diagonalization of $\tilde{B}$ from vertex v_b towards vertex v_1, similarly to the algorithm for trees, with the difference that in each step the diagonal value of v_1 is updated, since its value is affected by the computations for the other vertices of the cycle.

The procedure denotes by w_i, for $i = 2, ..., b$, the entries $(i, 1)$ and $(1, i)$ of the original matrix $A(G)+\alpha I$, which represent the adjacency between v_1 and v_i. Thus, initially, $w_2 = w_b = 1$ and $w_j = 0$, for $3 \leq j \leq b-1$. It also writes $a_i = a(v_i)$, for all $i = 1, ..., b$, for simplicity. With this notation, the submatrix $\tilde{B}$ has the form:

$$\begin{array}{c} 1 \\ \vdots \\ j-1 \\ j \\ \vdots \\ b \end{array} \begin{bmatrix} a_1 & \cdots & w_{j-1} & w_j & \cdots & w_b \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ w_{j-1} & \cdots & a_{j-1} & 1 & \cdots & 0 \\ w_j & \cdots & 1 & a_j & \cdots & 0 \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ w_b & \cdots & 0 & 0 & \cdots & a_b \end{bmatrix}.$$

For fixed j, with $3 \leq j \leq b$, 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 $\tilde{B}$. If $a_j \neq 0$, then the following row and column operations annihilate the entries $(j-1, j)$ and $(j, j-1)$:

$$R_{j-1} \leftarrow R_{j-1} - \frac{1}{a_j} R_j$$

and

$$C_{j-1} \leftarrow C_{j-1} - \frac{1}{a_j} C_j.$$

After these operations, the diagonal value of v_{j-1} becomes $\tilde{a}_{j-1} = a_{j-1} - \frac{1}{a_j}$ and the value of w_{j-1} becomes $\tilde{w}_{j-1} = w_{j-1} - \frac{w_j}{a_j}$, yielding the matrix

$$\begin{array}{c} 1 \\ \vdots \\ j-1 \\ j \\ \vdots \\ b \end{array} \begin{bmatrix} a_1 & \cdots & \tilde{w}_{j-1} & w_j & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ \tilde{w}_{j-1} & \cdots & \tilde{a}_{j-1} & 0 & \cdots & 0 \\ w_j & \cdots & 0 & a_j & \cdots & 0 \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & a_b \end{bmatrix}.$$

Besides, the following row and column operations annihilate the entries w_j of the first row and column of the matrix:

$$R_1 \leftarrow R_1 - \frac{w_j}{a_j} R_j$$

and

$$C_1 \leftarrow C_1 - \frac{w_j}{a_j} C_j.$$

The value of v_1 becomes $\tilde{a}_1 = a_1 - \frac{(w_j)^2}{a_j}$, and the resulting matrix is

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

This process is repeated while $a_j \neq 0$, according to case 1 of procedure DiagCycle.

Suppose that $a_j = 0$, for some $3 \leq j \leq b$:

$$\begin{array}{c} 1 \\ \vdots \\ j-2 \\ j-1 \\ j \\ \vdots \\ b \end{array} \begin{bmatrix} a_1 & \cdots & w_{j-2} & w_{j-1} & w_j & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ w_{j-2} & \cdots & a_{j-2} & 1 & 0 & \cdots & 0 \\ w_{j-1} & \cdots & 1 & a_{j-1} & 1 & \cdots & 0 \\ w_j & \cdots & 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & a_b \end{bmatrix}.$$

If $j \geq 4$, the vertex v_j is used to annihilate the two entries representing the edge between v_{j-1} and its parent v_{j-2}, by performing the following operations:

$$R_{j-2} \leftarrow R_{j-2} - R_j$$

and

$$C_{j-2} \leftarrow C_{j-2} - C_j.$$

These operations change the value of w_{j-2} to $\tilde{w}_{j-2} = w_{j-2} - w_j$ (case 2 of the procedure DiagCycle), yielding the submatrix

$$\begin{array}{c} 1 \\ \vdots \\ j-2 \\ j-1 \\ j \\ \vdots \\ b \end{array} \begin{bmatrix} a_1 & \cdots & \tilde{w}_{j-2} & w_{j-1} & w_j & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ \tilde{w}_{j-2} & \cdots & a_{j-2} & 0 & 0 & \cdots & 0 \\ w_{j-1} & \cdots & 0 & a_{j-1} & 1 & \cdots & 0 \\ w_j & \cdots & 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & a_b \end{bmatrix}.$$

In the case that $j = 3$, row $j-2$ of $\tilde{B}$ is the first row and the adjacency between v_{j-1} and its parent v_{j-2} will be removed later.

Next, the operations

$$R_{j-1} \leftarrow R_{j-1} - \frac{a_{j-1}}{2} R_j$$

and

$$C_{j-1} \leftarrow C_{j-1} - \frac{a_{j-1}}{2} C_j$$

annihilate the value a_{j-1}, the value of w_{j-1} is changed to $\tilde{w}_{j-1} = w_{j-1} - \frac{w_j \cdot a_{j-1}}{2}$, and the submatrix becomes

$$\begin{array}{c} 1 \\ \vdots \\ j-2 \\ j-1 \\ j \\ \vdots \\ b \end{array} \begin{bmatrix} a_1 & \cdots & \tilde{w}_{j-2} & \tilde{w}_{j-1} & w_j & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ \tilde{w}_{j-2} & \cdots & a_{j-2} & 0 & 0 & \cdots & 0 \\ \tilde{w}_{j-1} & \cdots & 0 & 0 & 1 & \cdots & 0 \\ w_j & \cdots & 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & a_b \end{bmatrix}.$$

Besides, the operations

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

produce the submatrix

$$\begin{array}{c} 1 \\ \vdots \\ j-2 \\ j-1 \\ j \\ \vdots \\ b \end{array} \begin{bmatrix} a_1 & \cdots & \tilde{w}_{j-2} & \beta & \gamma & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ \tilde{w}_{j-2} & \cdots & a_{j-2} & 0 & 0 & \cdots & 0 \\ \beta & \cdots & 0 & -1/2 & 0 & \cdots & 0 \\ \gamma & \cdots & 0 & 0 & 2 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & a_b \end{bmatrix},$$

where $\beta = \tilde{\tilde{w}}_{j-1} = \frac{1}{2} \cdot \left(w_{j-1} - w_j \cdot \left(1 + \frac{a_{j-1}}{2}\right)\right)$ and $\gamma = \tilde{w}_j = w_{j-1} + w_j \cdot \left(1 - \frac{a_{j-1}}{2}\right)$.

Finally, the operations

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

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

$$\begin{array}{c} 1 \\ \vdots \\ j-2 \\ j-1 \\ j \\ \vdots \\ b \end{array} \left[\begin{array}{ccccccc} \tilde{a}_1 & \cdots & \tilde{w}_{j-2} & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ \tilde{w}_{j-2} & \cdots & a_{j-2} & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & -1/2 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 2 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & a_b \end{array} \right].$$

Note that, at this point, both rows (and columns) j and $j-1$ are diagonalized, according to case 2 of the procedure DiagCycle. We also observe that if $j = 3$, i.e. $a_3 = 0$, the procedure changes the value of the boolean variable *TriangDiagonalized* to *true* and the procedure ends. In this case, the diagonalization process is completed.

To see that procedure DiagCycle completes the diagonalization of $\tilde{B}$, we consider the case when $j = 2$ and it remains to diagonalize the 2×2 block

$$\left[\begin{array}{cc} a_1 & w_2 \\ w_2 & a_2 \end{array} \right].$$

Note that if $w_2 = 0$, the 2×2 block above is already diagonalized. If $w_2 \neq 0$, it is easy to see that if $a_2 \neq 0$, the operations

$$R_1 \leftarrow R_1 - \frac{w_2}{a_2} R_2$$

and

$$C_1 \leftarrow C_1 - \frac{w_2}{a_2} C_2,$$

complete the diagonalization process, with the value a_1 changed to $\tilde{a}_1 = a_1 - \frac{(w_2)^2}{a_2}$. If $w_2 \neq 0$ and $a_2 = 0$, then the following row and column operations

$$\begin{aligned} R_1 &\leftarrow R_1 - \frac{a_2}{2 \cdot w_2} R_2, \\ C_1 &\leftarrow C_1 - \frac{a_2}{2 \cdot w_2} C_2, \\ R_2 &\leftarrow R_2 + \frac{1}{w_2} R_1, \\ C_2 &\leftarrow C_2 + \frac{1}{w_2} C_2, \\ R_1 &\leftarrow R_1 - \frac{w_2}{2} R_2, \\ C_1 &\leftarrow C_1 - \frac{w_2}{2} C_2 \end{aligned}$$

yield the diagonalized 2×2 block

$$\begin{bmatrix} -\frac{w_2{}^2}{2} & 0 \\ 0 & 2 \end{bmatrix},$$

which completes the proof. □

Remark: The algorithm DiagonalizeUnicyclic can be implemented with $O(n)$ operations, since it processes each vertex exactly once, performing a fixed number of operations every time. The matrix itself is not necessary and the structure of the graph may be used to store intermediate values. As in the examples below, the diagonal elements are stored in the vertices. Hence, the space needed for the algorithm is also $O(n)$.

Applying Theorem 2.2 and Sylvester's Law of Inertia, we obtain the following result.

Theorem 2.3. *Let G be a unicyclic graph and α a real number. Let D be the diagonal matrix produced by the algorithm DiagonalizeUnicyclic(G, α) with $\alpha = -\beta$. The following assertions hold.*

(i) The number of eigenvalues of G that are greater than β is the number of positive entries in D.

(ii) The number of eigenvalues of G that are smaller than β is the number of negative entries in D.

(iii) The multiplicity of β as an eigenvalue of G is the number of zero entries in D.

Next, we observe that we may determine the number of eigenvalues of a unicyclic graph G in a finite real interval by making two calls of algorithm DiagonalizeUnicyclic.

Corollary 2.4. *The number of eigenvalues of a unicyclic graph G in an interval (α_1, α_2) is the number of positive diagonal values produced by DiagonalizeUnicyclic(G, α_1) minus the number of positive and zero diagonal values produced by DiagonalizeUnicyclic(G, α_2).*

3. EXAMPLE

To illustrate the execution of algorithm DiagonalizeUnicyclic, let us consider the unicyclic graph G of Figure 5.

We will apply the algorithm to G with scalar $\alpha = -2$. The left side of Figure 6 shows the initialization of the vertices with the value -2 (Step 1 of the algorithm).

At Step 2, since v_{12} has only one child, which has a nonzero value, its value becomes $a(v_{12}) = -2 - \frac{1}{-2} = -\frac{3}{2}$. The same happens when vertices v_1 and v_6

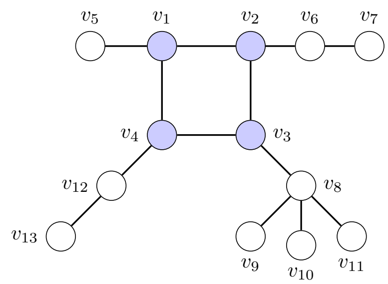


Figure 5: Unicyclic graph G

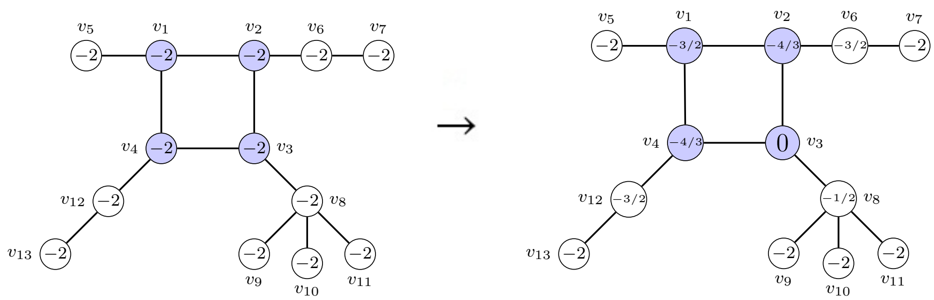


Figure 6: Steps 1 and 2 with $\alpha = -2$

are processed. Vertex v_8 has three children, all with nonzero values, then its value becomes $a(v_8) = -2 - \frac{3}{-2} = -\frac{1}{2}$ and the value of its parent v_3 becomes $a(v_3) = -2 - \frac{1}{-1/2} = 0$. Both v_4 and v_2 have only one child with diagonal value $-\frac{3}{2}$, hence their values are updated to $-2 - \frac{1}{-3/2} = -\frac{4}{3}$. The right side of Figure 6 shows the diagonal values at the end of Step 2.

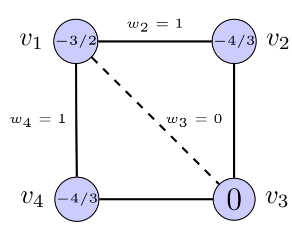


Figure 7: Initialization of procedure DiagCycle

Since no vertex v_i of the cycle had a child with a zero diagonal value, none of the edges of the cycle were removed in Step 2. Thus the algorithm executes the procedure DiagCycle in Step 4. Figure 7 shows the initial values of the vertices of the cycle, computed in Step 2, and the values w_i, $i = 2, 3, 4$.

In each iteration i, from $i = 4$ to 2, the procedure DiagCycle processes the vertices v_{i-1} and v_1, removing the edges that connect them to v_i . Since $a_4 = -\frac{4}{3} \neq 0$, the diagonal value of v_3 becomes $a_3 = 0 - \frac{1}{-4/3} = \frac{3}{4}$, the diagonal value of v_1 is changed to $a_1 = -\frac{3}{2} - \frac{(w_4)^2}{a_4} = -\frac{3}{2} - \frac{1^2}{-4/3} = -\frac{3}{4}$ and the value of w_3 becomes $w_3 = 0 - \frac{w_4}{a_4} = 0 - \frac{1}{-4/3} = \frac{3}{4}$. Figure 8 shows the results of the computations at this step of the procedure.

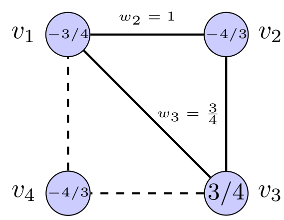


Figure 8: Iteration $i = 4$ of the procedure DiagCycle

Since now $a_3 = \frac{3}{4} \neq 0$, the algorithm changes the value of v_2 to $a_2 = -\frac{4}{3} - \frac{1}{3/4} = -\frac{8}{3}$, the value of v_1 to $a_1 = -\frac{3}{4} - \frac{(w_3)^2}{a_3} = -\frac{3}{4} - \frac{\left(\frac{3}{4}\right)^2}{3/4} = -\frac{3}{2}$ and the value of w_2 to $w_2 = 1 - \frac{w_3}{a_3} = 1 - \frac{3/4}{3/4} = 0$. Figure 9 shows the values obtained up to this point.

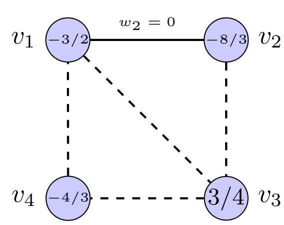


Figure 9: Iteration $i = 3$ of procedure DiagCycle

The case 2 of the procedure DiagCycle was not executed, hence *TriangDiagonalized* = false. Then, since $w_2 = 0$, the procedure ends.

Figure 10 shows the diagonal values at the end of the algorithm for $\alpha = -2$. Since $a(v_3)$ is the only positive diagonal value and all the others are negative, by Theorem 2.3, we conclude that one eigenvalue of G is greater than 2 and all the others are smaller than 2.

Now let us apply the algorithm DiagonalizeUnicyclic to G with scalar $\alpha = 0$. The left side of Figure 11 shows the initialization of the vertices.

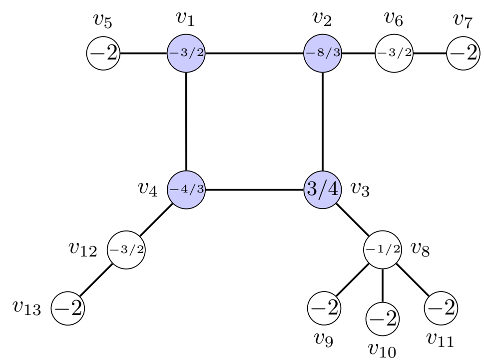


Figure 10: Final diagonal values with $\alpha = -2$

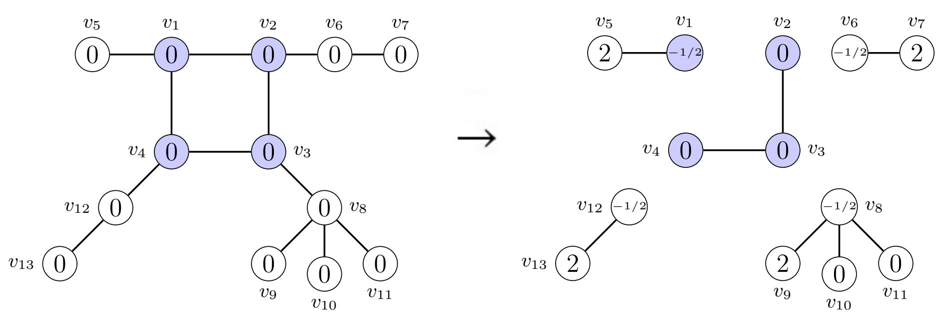


Figure 11: Steps 1 and 2 with $\alpha = 0$

Since all vertices are initialized with a zero value, at Step 2 of the procedure the value of each vertex v_i with a zero child becomes $-\frac{1}{2}$ and the value of one zero child of v_i is replaced by 2. Besides, the edge connecting v_i to its parent is removed. Hence, the edges connecting v_1 to its adjacent vertices in the cycle, v_2 and v_4, are removed. The corresponding graph with the diagonal values obtained so far are shown in the right side of Figure 11.

Next, Step 3 of the algorithm is executed and the path $v_2v_3v_4$ with root v_2 is processed by the algorithm for trees. Since v_4 has a zero diagonal value, the algorithm replaces the values of v_3 and v_4 by $-\frac{1}{2}$ and 2, respectively, and the edge connecting v_3 to v_2 is removed. Thus v_2 becomes an isolated vertex and the algorithm ends. Figure 12 shows the diagonal values at the end of the algorithm.

By Theorem 2.3, we conclude that G has five positive eigenvalues, five negative eigenvalues and zero is an eigenvalue of G with multiplicity 3.

Combining the two cases above, it follows from Corollary 4.9 that G has four eigenvalues in the interval $(0, 2)$. In fact, computing the spectrum of this graph we

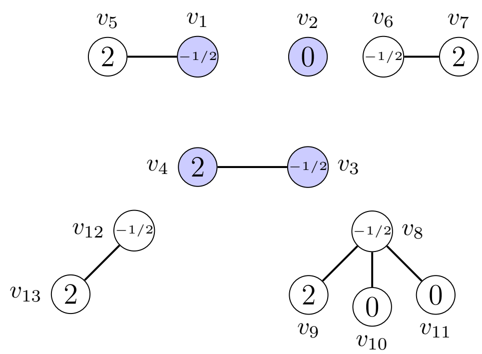


Figure 12: Final diagonal values with $\alpha = 0$

obtain that the nonzero eigenvalues of G are ± 1, $\pm\sqrt{2}$, ± 0.53128, ± 1.82357 and ± 2.52831.

4. APPLICATIONS

In this section, applying the algorithm DiagonalizeUnicyclic we obtain spectral properties of certain families of unicyclic graphs.

We first consider a graph obtained by taking a cycle C_b and adding at least one pendant to each node in the cycle, called a *closed caterpillar*. The nodes in the cycle are called *back nodes*. The following result gives the inertia of a closed caterpillar.

Theorem 4.5. *If C is a closed caterpillar with n vertices and b back nodes, then C has b positive eigenvalues and b negative eigenvalues. Besides, the multiplicity of zero as an eigenvalue of C is $n - 2b$ and the nonzero eigenvalues have multiplicity at most* 2.

Proof. Let us apply the algorithm DiagonalizeUnicyclic for the caterpillar C with scalar $\alpha = 0$. Since every pendant is initialized with a zero, all the b back nodes receive the value $-\frac{1}{2}$, one child of each back node is assigned the 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 2.3, C has b positive eigenvalues, b negative eigenvalues and the multiplicity of zero as an eigenvalue of C is $n - 2b$.

Next we apply the algorithm DiagonalizeUnicyclic for C with scalar $\alpha = -\lambda$, where λ is a nonzero eigenvalue of C. By Theorem 2.3(iii), at the end of the execution of the algorithm at least one diagonal value must be zero. Note that every pendant remains with diagonal value $-\lambda \neq 0$, which is the initial value assigned to all vertices. Besides, for $i = 3, \ldots, b$, if a zero is assigned to a back node

v_i of C during the execution, the algorithm replaces this value by 2. Hence, at the end of the execution v_1 or v_2 (perhaps both) have diagonal value zero. It follows that λ has multiplicity at most 2. □

We give below sufficient conditions for a closed caterpillar to be non-integral.

Theorem 4.6. *A closed caterpillar that contains a vertex of degree* 3 *is not integral.*

Proof. Let C be a closed caterpillar with b back nodes that contains a vertex of degree 3. Note that we can order the vertices so that back node v_b has degree 3. We apply the algorithm DiagonalizeUnicyclic for C with scalar $\alpha = -1$. Initially, every vertex is assigned the value -1. Since v_b is adjacent to exactly one pendant, its diagonal value becomes $a(v_b) = -1 - \frac{1}{-1} = 0$. Then the algorithm processes v_{b-1}, whose other children are pendants, so they have nonzero values. Hence the algorithm assigns 2 to v_b and $-1/2$ to its parent v_{b-1}. These two diagonal values and the initial values assigned to the pendants remain the same until the end of the execution of the algorithm. Therefore, the algorithm produces at most $b-1$ non-negative diagonal values, which implies that C has at most $b-1$ eigenvalues that are greater than or equal to 1. On the other hand, by Theorem 4.5, C has exactly b positive eigenvalues. Hence C has at least one eigenvalue in the interval $(0,1)$, which proves that it is not integral. □

Theorem 4.7. *A closed caterpillar with b back nodes and maximum degree Δ is not integral if $b > 4\sqrt{\Delta - 1}$.*

Proof. By Theorem 4.5, a caterpillar C with b back nodes has b positive eigenvalues and any nonzero eigenvalue of C has multiplicity at most 2, which implies that it has at least $\lceil \frac{b}{2} \rceil$ positive distinct eigenvalues. By [**8**, Theorem 1], these eigenvalues are bounded by $2\sqrt{\Delta - 1}$, and there are exactly $\lfloor 2\sqrt{\Delta - 1} \rfloor$ positive integers in this range.

Thus, by the pigeon-hole principle, if $b > \lfloor 4\sqrt{\Delta - 1} \rfloor$, at least one eigenvalue of C must be non-integral. □

We now consider closed caterpillars where all back nodes have the same degree, called *balanced closed caterpillars*. We write $C_{b,p}$ to represent the balanced closed caterpillar with b back nodes, each with $p \geqslant 1$ pendants. Figure 13 shows the balanced closed caterpillar $C_{4,3}$.

The algorithm DiagonalizeUnicyclic allows us to obtain the spectrum of a balanced closed caterpillar $C_{b,p}$ in terms of the spectrum of the cycle C_b.

Theorem 4.8. *The spectrum of the balanced closed caterpillar $C_{b,p}$ is given by*

$$\left\{ 0^{n-2b},\ \frac{r_i}{2} \pm \sqrt{p + \left(\frac{r_i}{2}\right)^2},\ i = 1, \ldots, b \right\},$$

where $n = b(p+1)$ is the order of $C_{b,p}$ and $r_i = 2\cos\left(\frac{2\pi i}{b}\right)$, $i = 1, \ldots, b$, are the eigenvalues of the cycle C_b.

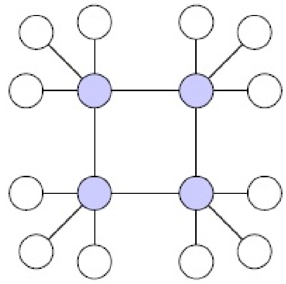


Figure 13: $C_{4,3}$

Proof. By Theorem 4.5, zero is an eigenvalue of $C_{b,p}$ with multiplicity $n-2b$. To obtain the other $2b$ eigenvalues, let us apply the algorithm DiagonalizeUnicyclic for $C_{b,p}$ with scalar $\alpha = -\lambda$, where λ is a nonzero eigenvalue of the caterpillar. Since λ is an eigenvalue, the algorithm will produce at least one zero diagonal value.

Initially all vertices are assigned diagonal value $-\lambda$. Each back node v_i has p pendants, all with nonzero values, hence the algorithm assigns to v_i the value

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

for $i = 1, ..., b$. 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 2.3 (iii), the algorithm will produce a zero diagonal value if and only if $\beta = -r$, where r is an eigenvalue of the cycle C_b. Hence, λ is a nonzero eigenvalue of $C_{b,p}$ if and only if $\frac{\lambda^2-p}{-\lambda} = -r$, where r is an eigenvalue of C_b. Also note that

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

If G_1 and G_2 are two graphs on disjoint sets of n and m vertices, respectively, the *corona* $G_1 \circ G_2$ of G_1 and G_2 is the graph obtained by taking one copy of G_1 and n copies of G_2, and then joining the ith vertex of G_1 to every vertex in the ith copy of G_2. Note that the balanced closed caterpillar $C_{b,p}$ is the corona of the cycle C_b and the graph consisting of p isolated vertices. Thus the spectrum of $C_{b,p}$ can also be deduced from [**2**, Theorem 3.1], where the spectrum of the corona of a graph G_1 and a regular graph G_2 is given in terms of the spectrum of G_1 and G_2.

Corollary 4.9. *The index of the balanced closed caterpillar $C_{b,p}$ is $1+\sqrt{p+1}$ and its greatest negative eigenvalue is $1-\sqrt{p+1}$, both simple. Furthermore, the positive eigenvalues of $C_{b,p}$ lie in the interval $[-1+\sqrt{p+1},\, 1+\sqrt{p+1}]$ and the negative eigenvalues belong to the interval $[-1-\sqrt{p+1},\, 1-\sqrt{p+1}]$.*

Proof. Since $\frac{r}{2} \pm \sqrt{p + \left(\frac{r}{2}\right)^2}$ are both increasing one-to-one functions of r, and 2 is the index the cycle C_b, it follows from Theorem 4.8 that $\frac{2}{2} + \sqrt{p + \left(\frac{2}{2}\right)^2} = 1 + \sqrt{p+1}$ is the greatest positive eigenvalue and $\frac{2}{2} - \sqrt{p + \left(\frac{2}{2}\right)^2} = 1 - \sqrt{p+1}$ is the greatest negative eigenvalue of the closed caterpillar $C_{b,p}$. Additionally, it also follows from Theorem 4.8 that the smallest positive eigenvalue of $C_{b,p}$ is

$$\frac{a}{2} + \sqrt{p + \left(\frac{a}{2}\right)^2},$$

where a is the smallest eigenvalue of the cycle C_b. If b is even, $a = -2$, and if b is odd, $a = 2\cos\left(\frac{(b-1)\pi}{b}\right) = -2\cos\left(\frac{\pi}{b}\right) > -2$. Thus, if b is even, $-1 + \sqrt{p+1}$ is the smallest positive eigenvalue of $C_{b,p}$. If b is odd, note that

$$\begin{aligned}
\frac{a}{2} + \sqrt{p + \left(\tfrac{a}{2}\right)^2} > -1 + \sqrt{p+1} \quad & \Leftrightarrow \sqrt{p + \left(\tfrac{a}{2}\right)^2} > -\left(1 + \tfrac{a}{2}\right) + \sqrt{p+1} \\
& \Leftrightarrow p + \left(\tfrac{a}{2}\right)^2 > \left(1 + \tfrac{a}{2}\right)^2 - 2\left(1 + \tfrac{a}{2}\right)\sqrt{p+1} + p + 1 \\
& \Leftrightarrow 0 > 2 + a - (a+2)\sqrt{p+1} \\
& \Leftrightarrow 0 > (2+a)(1 - \sqrt{p+1}),
\end{aligned}$$

which is true, since $a + 2 > 0$ and $1 - \sqrt{p+1} < 0$. Hence, the positive eigenvalues of $C_{b,p}$ belong to the interval $[-1 + \sqrt{p+1},\ 1 + \sqrt{p+1}]$. The proof for the negative eigenvalues is analogous. □

Corollary 4.10. *No balanced closed caterpillar is integral.*

Proof. By Theorem 4.5 and Corollary 4.9, if $b \geqslant 6$ the balanced closed caterpillar $C_{b,p}$ has at least 4 distinct positive eigenvalues in the interval $\left[-1 + \sqrt{p+1}, 1 + \sqrt{p+1}\right]$, since its largest eigenvalue is simple. On the other hand, there are at most three integers in $\left[-1 + \sqrt{p+1}, 1 + \sqrt{p+1}\right]$. Hence, by the pigeon-hole principle, at least one positive eigenvalue of $C_{b,p}$ is not an integer. If $b \in \{3, 4, 5\}$, one can show that the spectrum of $C_{b,p}$ given by Theorem 4.8 also has non-integral values. □

Next we consider unicyclic graphs obtained by adding a fixed number of copies of the path P_ℓ, with $\ell \geqslant 2$, to each node in a cycle. The nodes in the cycle are also called *back nodes*. We write $C_{b,p,\ell}$ to represent a unicyclic graph with b back nodes, where $p \geqslant 1$ copies of P_ℓ are attached to each back node. Figure 14 shows $C_{3,3,4}$.

Applying algorithm DiagonalizeUnicyclic we obtain the inertia of the unicyclic graph $C_{b,p,\ell}$ as well as the distribution of eigenvalues of this graph with respect to 1, in terms of the distribution of eigenvalues of the cycle C_b and the balanced closed caterpillar $C_{b,p}$. The inertia of $C_{b,p}$ is given in Theorem 4.5.

For a real number α and a graph G, we denote by $In_\alpha(G) = (i_\alpha(G), i_\alpha^+(G), i_\alpha^-(G))$ the triple composed of the number of eigenvalues of G that are equal to, greater than and smaller than α, respectively, which corresponds to the inertia of the matrix $A(G) - \alpha I$. Note that $In_0(G)$ is the inertia

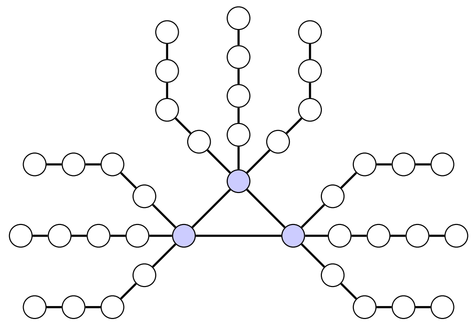


Figure 14: $C_{3,3,4}$

of the graph G. We also write $m_G(I)$ to represent the number of eigenvalues of G in the interval I.

Lemma 4.11. *For the balanced closed caterpillar $C_{b,p}$,*

$$In_1(C_{b,p}) \;=\; (i_{1-p}(C_b),\;\; i^+_{1-p}(C_b),\;\; i^-_{1-p}(C_b) + pb).$$

In particular, if $p \geqslant 4$, $\;In_1(C_{b,p}) = (0, b, pb)$.

Proof. We apply the algorithm DiagonalizeUnicyclic to $C_{b,p}$ with scalar $\alpha = -1$. Initially all vertices are assigned diagonal value -1. Since each back node v_i has p pendants, the algorithm assigns to v_i the value

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

for $i = 1, ..., b$. Since all $a(v_i)$ are equal, we can consider that we are applying the algorithm with scalar $\beta = -(1-p)$ to the cycle C_b. By Theorem 2.3, at the end of the execution of the algorithm the number of positive, negative and zero diagonal values $a(v_i)$ is equal to the number of eigenvalues of C_b that are greater than, smaller than and equal to $-\beta = 1-p$, respectively. Hence, since all pb pendants remain with the negative value -1, we have

$$i_1(C_{b,p}) = i_{1-p}(C_b), \quad i^+_1(C_{b,p}) = i^+_{1-p}(C_b), \quad i^-_1(C_{b,p}) = i^-_{1-p}(C_b) + pb.$$

In particular, since all eigenvalues of the cycle C_b lie in the interval $[-2, 2]$, for $p \geqslant 4$ we get

$$i_1(C_{b,p}) = 0, \quad i^+_1(C_{b,p}) = b, \quad i^-_1(C_{b,p}) = pb,$$

which concludes the proof. $\square$

Lemma 4.12. *For the cycle C_b,*

$$i_0(C_b)=\begin{cases}2, & \textit{if } b\equiv 0 \mod 4\\ 0, & \textit{if } b\not\equiv 0 \mod 4\end{cases},\quad i_0^+(C_b)=\begin{cases}\frac{b-2}{2}, & \textit{if } b\equiv 0 \mod 4\\ \frac{b+1}{2}, & \textit{if } b\equiv 1 \mod 4\\ \frac{b}{2}, & \textit{if } b\equiv 2 \mod 4\\ \frac{b-1}{2}, & \textit{if } b\equiv 3 \mod 4\end{cases},\quad i_0^-(C_b)=\begin{cases}\frac{b-2}{2}, & \textit{if } b\equiv 0 \mod 4\\ \frac{b-1}{2}, & \textit{if } b\equiv 1 \mod 4\\ \frac{b}{2}, & \textit{if } b\equiv 2 \mod 4\\ \frac{b+1}{2}, & \textit{if } b\equiv 3 \mod 4\end{cases}$$

$$i_1(C_b)=\begin{cases}2, & \textit{if } b\equiv 0 \mod 6\\ 0, & \textit{if } b\not\equiv 0 \mod 6\end{cases},\quad i_1^+(C_b)=\begin{cases}\frac{b-3}{3}, & \textit{if } b\equiv 0 \mod 6\\ \frac{b+2}{3}, & \textit{if } b\equiv 1 \mod 6\\ \frac{b+1}{3}, & \textit{if } b\equiv 2 \mod 6\\ \frac{b}{3}, & \textit{if } b\equiv 3 \mod 6\\ \frac{b-1}{3}, & \textit{if } b\equiv 4 \mod 6\\ \frac{b-2}{3}, & \textit{if } b\equiv 5 \mod 6\end{cases},\quad i_1^-(C_b)=\begin{cases}\frac{2b-3}{3}, & \textit{if } b\equiv 0 \mod 6\\ \frac{2b-2}{3}, & \textit{if } b\equiv 1 \mod 6\\ \frac{2b-1}{3}, & \textit{if } b\equiv 2 \mod 6\\ \frac{2b}{3}, & \textit{if } b\equiv 3 \mod 6\\ \frac{2b+1}{3}, & \textit{if } b\equiv 4 \mod 6\\ \frac{2b+2}{3}, & \textit{if } b\equiv 5 \mod 6\end{cases}$$

$$i_{-1}(C_b)=\begin{cases}2, & \textit{if } b\equiv 0 \mod 3\\ 0, & \textit{if } b\not\equiv 0 \mod 3\end{cases},\quad i_{-1}^+(C_b)=\begin{cases}\frac{2b-3}{3}, & \textit{if } b\equiv 0 \mod 3\\ \frac{2b+1}{3}, & \textit{if } b\equiv 1 \mod 3\\ \frac{2b-1}{3}, & \textit{if } b\equiv 2 \mod 3\end{cases},\quad i_{-1}^-(C_b)=\begin{cases}\frac{b-3}{3}, & \textit{if } b\equiv 0 \mod 3\\ \frac{b-1}{3}, & \textit{if } b\equiv 1 \mod 3\\ \frac{b+1}{3}, & \textit{if } b\equiv 2 \mod 3\end{cases}$$

$$i_{-2}(C_b)=\begin{cases}1, & \textit{if } b\equiv 0 \mod 2\\ 0, & \textit{if } b\not\equiv 0 \mod 2\end{cases},\quad i_{-2}^+(C_b)=\begin{cases}b-1, & \textit{if } b\equiv 0 \mod 2\\ b, & \textit{if } b\equiv 1 \mod 2\end{cases},\quad i_{-2}^-(C_b)=0.$$

Proof. The eigenvalues of the C_b are $r_i = 2\cos\left(\frac{2\pi i}{b}\right)$, $i = 1,\dots,b$. Thus all eigenvalues lie in the interval $[-2,2]$ and

$$0\leqslant r_i\leqslant 2 \;\Leftrightarrow\; 0<\frac{2\pi i}{b}\leqslant\frac{\pi}{2} \text{ or } \frac{3\pi}{2}\leqslant\frac{2\pi i}{b}\leqslant 2\pi \;\Leftrightarrow\; 0<i\leqslant\frac{b}{4} \text{ or } \frac{3b}{4}\leqslant i\leqslant b,$$

which implies that $m_{C_b}[0,2] = 1+2\lfloor\frac{b}{4}\rfloor$. Besides, $r_i = 0$ if and only if $i=\frac{b}{4}$ or $i=\frac{3b}{4}$. Analogously, $m_{C_b}[1,2] = 1+2\lfloor\frac{b}{6}\rfloor$ and $r_i = 1$ if and only if $i=\frac{b}{6}$ or $i=\frac{5b}{6}$. In addition, $m_{C_b}[-1,2] = 1+2\lfloor\frac{b}{3}\rfloor$ and $r_i=-1$ if and only if $i=\frac{b}{3}$ or $i=\frac{2b}{3}$. Besides, -2 is an eigenvalue of C_b if and only if b is even, which concludes the proof. □

Theorem 4.13. *The unicyclic graph $C = C_{b,p,\ell}$ satisfies:*

$$(i)\;\; In_0(C)=\begin{cases}\left(i_0(C_b),\; i_0^+(C_b)+\frac{pb\ell}{2},\; i_0^-(C_b)+\frac{pb\ell}{2}\right), & \textit{if } \ell \textit{ is even}\\ \left((p-1)b,\; \frac{pb(\ell-1)}{2}+b,\; \frac{pb(\ell-1)}{2}+b\right), & \textit{if } \ell \textit{ is odd}\end{cases}$$

$$(ii)\;\; In_1(C)=\begin{cases}\left(i_1(C_b),\; i_1^+(C_b)+\frac{pb\ell}{3},\; i_1^-(C_b)+\frac{2pb\ell}{3}\right), & \textit{if } \ell\equiv 0 \mod 3\\ \left(i_1(C_{b,p}),\; i_1^+(C_{b,p})+\frac{pb(\ell-1)}{3},\; i_1^-(C_{b,p})+\frac{2pb(\ell-1)}{3}\right), & \textit{if } \ell\equiv 1 \mod 3\\ \left((p-1)b,\; \frac{pb(\ell-2)}{3}+b,\; \frac{2pb(\ell-2)}{3}+(p+1)b\right), & \textit{if } \ell\equiv 2 \mod 3\end{cases}$$

Proof. We apply the algorithm DiagonalizeUnicyclic to C with scalar $\alpha = 0$. Let us consider one of the paths P_ℓ attached to the cycle C_b, with vertices $u_1, u_2, \dots, u_\ell$, where u_1 is adjacent to a back node and u_ℓ is a pendant. Since every vertex of C is initialized with a zero, the algorithm assigns 2 to u_ℓ and $-1/2$ to $u_{\ell-1}$. If $\ell = 2$, the edge that connects P_ℓ to the cycle is removed, otherwise, the edge between $u_{\ell-1}$ and $u_{\ell-2}$ is removed. Note that this is repeated for the next two vertices in the path. Thus, if ℓ is even, $\ell/2$ vertices in the path receive the value 2, $\ell/2$ receive the value $-1/2$ and the edge that connects P_ℓ to the cycle is removed. The same happens for all paths attached to the cycle C_b. Therefore, if ℓ is even, it follows from Theorem 2.3 that

$$In_0(C) = \left(i_0(C_b),\ i_0^+(C_b) + \tfrac{pb\ell}{2},\ i_0^-(C_b) + \tfrac{pb\ell}{2}\right).$$

If ℓ is odd, in each path P_ℓ attached to the cycle C_b, $(\ell-1)/2$ vertices receive value 2, $(\ell-1)/2$ receive value $-1/2$, and the vertex adjacent to the cycle remains with a zero value. Hence all back nodes receive the value $-\frac{1}{2}$, one child of each back node is assigned the value 2 and the other $p-1$ children of each back node remain with the zero value. Besides, all edges in the cycle are removed and the algorithm ends. Therefore,

$$In_0(C) = \left((p-1)b,\ \tfrac{pb(\ell-1)}{2} + b,\ \tfrac{pb(\ell-1)}{2} + b\right),$$

if ℓ is odd.

Now we apply the algorithm to C with scalar $\alpha = -1$. Again, let us consider one of the paths P_ℓ attached to the cycle C_b, with vertices $u_1, u_2, \ldots, u_\ell$, where u_1 is adjacent to a vertex in the cycle and u_ℓ is a pendant. Every vertex is initialized with -1, then the algorithm assigns to $u_{\ell-1}$ the value

$$a(u_{\ell-1}) = -1 - \frac{1}{a(u_\ell)} = -1 - \frac{1}{-1} = 0.$$

If $\ell \geqslant 3$, since $u_{\ell-1}$ has a zero value, the algorithm assigns 2 to $u_{\ell-1}$ and $-1/2$ to $u_{\ell-2}$. If $\ell = 3$, the edge that connects P_ℓ to the cycle is removed, otherwise, the edge between $u_{\ell-2}$ and $u_{\ell-3}$ is removed. This is repeated for the next three vertices in the path. Hence, if $\ell \equiv 0 \mod 3$, $\ell/3$ vertices in the path receive the positive value 2, while $2\ell/3$ vertices receive a negative value (remain with the initial value -1 or receive the value $-1/2$) and the edge that connects P_ℓ to the cycle is removed. The same occurs with all the paths. Therefore, if $\ell \equiv 0 \mod 3$,

$$In_1(C) = \left(i_1(C_b),\ i_1^+(C_b) + \tfrac{pb\ell}{3},\ i_1^-(C_b) + \tfrac{2pb\ell}{3}\right).$$

If $\ell \equiv 1 \mod 3$, in each path P_ℓ, $(\ell-1)/3$ vertices receive the positive value 2, $2(\ell-1)/3$ vertices receive a negative value, the edge that connects u_2 to u_1 is removed and u_1 remains with the initial value -1. Thus, in this case, the remaining graph to be processed by the algorithm is the balanced closed caterpillar $C_{b,p}$, whose triple $In_1(C_{b,p})$ is given in Theorem 4.11. Therefore, if $\ell \equiv 1 \mod 3$,

$$In_1(C) = \left(i_1(C_{b,p}),\ i_1^+(C_{b,p}) + \tfrac{pb(\ell-1)}{3},\ i_1^-(C_{b,p}) + \tfrac{2pb(\ell-1)}{3}\right).$$

Finally, if $\ell \equiv 2 \mod 3$ (which includes the case $\ell = 2$), $(\ell-2)/3$ vertices of each path P_ℓ receive the positive value 2, $2(\ell-2)/3$ receive a negative value, u_2 remains with the initial value -1 and u_1 receives a zero value. Hence the algorithm assigns the value $-\frac{1}{2}$ to all back nodes, one child of each back node receives the value 2 and the other $p-1$ children of each back node remain with the zero value. Besides, all edges in the cycle are removed and the algorithm ends. Therefore,

$$In_1(C) = \left((p-1)b,\ \tfrac{pb(\ell-2)}{3} + b,\ \tfrac{2pb(\ell-2)}{3} + (p+1)b\right),$$

if $\ell \equiv 2 \mod 3$. □

Corollary 4.14. *The unicyclic graph $C_{b,p,\ell}$ is not integral.*

Proof. Let $C = C_{b,p,\ell}$ and let us suppose that ℓ is even. The case ℓ is odd is analogous.

If $\ell \equiv 0 \mod 3$, by Theorem 4.13,

$$m_C(0,1) = i_0^+(C_b) + \tfrac{pb\ell}{2} - i_1(C_b) - i_1^+(C_b) - \tfrac{pb\ell}{3} = \tfrac{pb\ell}{6} + m_{C_b}(0,1) \geqslant pb + m_{C_b}(0,1),$$

since $\ell \geqslant 6$. Hence, $m_C(0,1) > 0$.

If $\ell \equiv 1 \mod 3$, by Theorem 4.13,

$$\begin{aligned} m_C(0,1) &= i_0^+(C_b) + \tfrac{pb\ell}{2} - i_1(C_{b,p}) - i_1^+(C_{b,p}) - \tfrac{pb(\ell-1)}{3} \\ &= \tfrac{pb(\ell+2)}{6} + i_0^+(C_b) - m_{C_{b,p}}[1,\infty). \end{aligned}$$

By Theorem 4.5, $m_{C_{b,p}}[1,\infty) \leqslant b$, hence

$$m_C(0,1) \geqslant \tfrac{pb(\ell+2)}{6} - b + i_0^+(C_b) \geqslant i_0^+(C_b) > 0,$$

since $\ell \geqslant 4$ and C_b has at least one positive eigenvalue.

If $\ell \equiv 2 \mod 3$, by Theorem 4.13,

$$m_C(0,1) = i_0^+(C_b) + \tfrac{pb\ell}{2} - (p-1)b - \tfrac{pb(\ell-2)}{3} - b = \tfrac{pb(\ell-2)}{6} + i_0^+(C_b) > 0,$$

since $\ell \geqslant 2$ and the cycle C_b has at least one positive eigenvalue. □

5. CONCLUDING REMARKS

In the previous section we presented several non-integral unicyclic graphs. In fact, integral unicyclic graphs seem to be rare. We performed a computer search for all integral unicyclic graphs up to 21 vertices and, besides the cycles C_3, C_4 and C_6, we found only three integral unicyclic graphs, shown in Table 1. Note that the two integral uniclyclic graphs with 20 vertices are nonisomorphic cospectral.

Acknowledgments. The authors would like to thank Dragan Stevanović who kindly offered a computer program he wrote to search for integral unicyclic graphs. This work is part of the doctoral studies of Rodrigo O. Braga. The third author is partially supported by CNPq, grants 409746/2016-9 and 303334/2016-9.

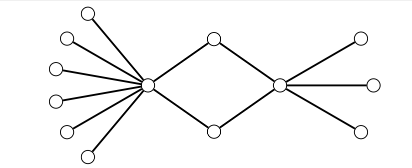
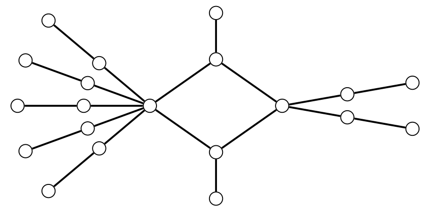
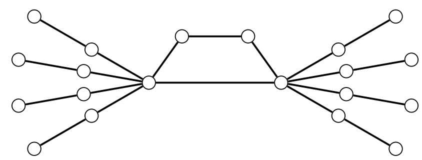
n	*Spectrum*	*Integral Unicyclic Graph*
13	$-3, -2, 0^9, 2, 3$	
20	$-3, -2, -1^7, 0^2, 1^7, 2, 3$	
20	$-3, -2, -1^7, 0^2, 1^7, 2, 3$	

Table 1: Integral unicyclic graphs up to 21 vertices that are not cycles

REFERENCES

1. K. Balińska, D. Cvetković, Z. Radosavljević, S. Simić, D. Stevanović: *A survey on integral graphs.* Univ. Beograd. Publ. Elektrotehn. Fak. Ser. Mat., **13** (2002), 42–65.
2. S. Barik, S. Pati, B. K. Sarma: *The spectrum of the corona of two graphs.* SIAM J. Discrete Math., **21** (1) (2007), 47–56.
3. R. O. Braga, V. M. Rodrigues, V. Trevisan: *On the distribution of Laplacian eigenvalues of trees.* Discrete Math., **313** (2013), 2382–2389.
4. R. O. Braga, R. R. Del-Vecchio, V. M. Rodrigues, V. Trevisan: *Trees with 4 or 5 distinct normalized Laplacian eigenvalues.* Linear Algebra Appl., **471** (2015), 615–635.
5. F. R. K. Chung: *Spectral Graph Theory.* CBMS Reg. Conf. Ser. Math., Vol. 92, Amer. Math. Soc., Providence, RI, 1997.
6. E. Fritscher, C. Hoppen, I. Rocha, V. Trevisan: *On the sum of the laplacian eigenvalues of a tree.* Linear Algebra Appl., **435** (2011), 371–399.
7. R. Horn, C. R. Johnson: *Matrix Analysis.* Cambridge University Press, 1985.
8. S. Hu: *The largest eigenvalue of unicyclic graphs.* Discrete Math., **307** (2007), 280–284.
9. D. P. Jacobs, V. Trevisan: *Locating the eigenvalues of trees.* Linear Algebra Appl., **434** (1) (2011), 81–88.

10. G. R. Omidi: *On Integral Graphs with Few Cycles.* Graphs Combin., **25** (2009), 841–849.

11. S. Radenković, I. Gutman: *Total pi-electron energy and Laplacian energy: how far the analogy goes?* J. Serb. Chem. Soc., **72** (12) (2007), 1343–1350.

(Received 08.10.2016.)
(Revised 19.10.2017.)

Rodrigo O. Braga
Instituto de Matemática,
Universidade Federal do Rio Grande do Sul,
Bento Gonçalves 9500 Porto Alegre,
Brazil
E-mail: `rbraga@ufrgs.br`

Virgínia M. Rodrigues
Instituto de Matemática,
Universidade Federal do Rio Grande do Sul,
Bento Gonçalves 9500 Porto Alegre,
Brazil
E-mail: `vrodrig@mat.ufrgs.br`

Vilmar Trevisan
Instituto de Matemática,
Universidade Federal do Rio Grande do Sul,
Bento Gonçalves 9500 Porto Alegre,
Brazil
E-mail: `trevisan@mat.ufrgs.br`