Multiplicity of eigenvalues of cographs

Luiz Emilio Allem$^a$, Fernando Tura$^b$,  

$^a$ Instituto de Matemática, UFRGS, Porto Alegre, RS, 91509-900, Brazil  
emilio.allem@ufrgs.br

$^b$ Departamento de Matemática, UFSM, Santa Maria, RS, 97105-900, Brazil  
ftura@mail.ufsm.br

Abstract

Motivated by the linear time algorithm that locates the eigenvalues of a cograph $G$ [10], we investigate the multiplicity of eigenvalue $\lambda$ for $\lambda \neq 0, -1$. For cographs with balanced cotrees we determine explicitly the highest value for the multiplicity.

The energy of a graph is defined as the sum of absolute values of the eigenvalues. A graph $G$ on $n$ vertices is said to be borderenergetic if its energy equals the energy of the complete graph $K_n$. We present families of non-cospectral and borderenergetic cographs.

1 Introduction

We recall that the spectrum of a graph $G$ is the multiset of the eigenvalues of its adjacency matrix. The main goal of this paper is to discuss the multiplicity of eigenvalues of cographs.

Cographs is an important class of graphs for its many applications. They have several alternative characterizations, for example, a cograph is graph which contains no path of length four as an induced subgraph [5] and because of this they are often simply called $P_4$ free graph in the literature. In particular it well known that any cograph has a canonical tree representation, called the cotree [2]. The cotree will be relevant to this paper and will be described later.

Our original motivation for considering cographs is to study the distribution of eigenvalues of graphs. It is known, for example, that any interval of the real line contains some eigenvalues of graphs, since, more generally, any root of a real-rooted monic polynomial with integer coefficients occurs as an eigenvalue of some tree [18]. On the other hand it was proved (see [15]) that no cograph has eingenvalues in the interval $(-1, 0)$, a surprising result.

In this paper, we turn to study the multiplicities of eigenvalues of cographs. In [12] it was proved that all eigenvalues of threshold graphs (a subclass of cographs), except $-1$
and 0 are simple. This motivates us to investigate further the multiplicities of cograph eigenvalues. Since the multiplicities of the eigenvalues $-1$ and 0 are known [2] we deal with eigenvalues that are different from 0 and -1.

The multiplicities of graph eigenvalues are extensively studied by several authors. Bell et al. [1] determined upper bound for the multiplicities of graphs. Later, Rowlinson in [16] studied the multiplicities of eigenvalues in trees. Recently, Bu et al. [3] studied the multiplicities in graphs attaching one pendent path, generalizing some known results for trees and unicyclic graphs [16].

Different from the star complement technique used in the works above, our technique is based on an algorithm called Diagonalization, presented in [10]. The Diagonalization finds, in $O(n)$ time, the number of eigenvalues of a cograph, by operating directly on the cotree of the cograph. The algorithm and the technique will be explained in the next section.

We study cographs whose cotree is balanced (see definition in Section 3) and determine the multiplicity of some eigenvalues and an upper bound for the multiplicity of other eigenvalues.

As an application of these results, we study the energy of families of cographs. Recall that if $G$ is a graph having eigenvalues $\lambda_1, \ldots, \lambda_n$, its energy, denoted $E(G)$ is defined [8,14] as $\sum_{i=1}^{n} |\lambda_i|$. There are many results on energy and its applications in several areas, including in chemistry see [14] for more details and the references therein.

It is well known that the complete graph $K_n$ has $E(K_n) = 2n - 2$ and it is a natural and important research problem to determine graphs that have the same energy of the complete graph $K_n$. A graph $G$ on $n$ vertices is said to be borderenergetic if its energy equals the energy of the complete graph $K_n$. Some recent results on borderenergetic graphs are the following.

In [8], it was shown that there exists borderenergetic graphs on order $n$ for each integer $n \geq 7$, and all borderenergetic graphs with 7, 8, and 9 vertices were determined. In [11] it was considered the classes of borderenergetic threshold graphs. For each $n \geq 3$, it was determined $n - 1$ threshold graphs on $n^2$ vertices, pairwise non-cospectral and equienergetic to the complete graph $K_{n^2}$.

Recently, Hou and Tao [9], showed that for each $n \geq 2$ and $p \geq 1$ ($p \geq 2$ if $n = 2$), there are $n - 1$ threshold graphs on $pn^2$ vertices, pairwise non-cospectral and equienergetic
with the complete graph $K_{p_n^2}$, generalizing the results in [11].

In this paper, we continue this investigation in the class of cographs. More precisely, we determine two infinite families of cographs that are borderenergetic.

Here is an outline of the remainder of this paper. In Section 2, we mention the representation of cographs by a cotree and explain the Diagonalization algorithm. In Section 3, we determine explicitly the multiplicity $m(\lambda)$ for some classes of cographs, except $0, -1$ and an upper bound for the remaining eigenvalues. In Section 4, as application, we present two families of integral non-cospectral and borderenergetic cographs.

2 Notation and Preliminaries

Let $G = (V, E)$ be an undirected graph with vertex set $V$ and edge set $E$, without loops or multiple edges. We denote the open neighborhood of $v$, by

$$N(v) = \{ w | \{v, w\} \in E \}$$

and its closed neighborhood by

$$N[v] = N(v) \cup \{ v \}.$$ 

The adjacency matrix of $G$, denoted by $A = [a_{ij}]$, is a matrix whose rows and columns are indexed by the vertices of $G$, and is defined to have entries

$$a_{ij} = \begin{cases} 1 & \text{if } v_i v_j \in E \\ 0 & \text{otherwise} \end{cases}$$

A value $\lambda$ is an eigenvalue of $G$ if $\det(A - \lambda I_n) = 0$, and since $A$ is real and symmetric, its eigenvalues are real numbers. We denote $m(\lambda)$ the multiplicity of the eigenvalue $\lambda$ of $A$.

2.1 Cotrees

A cograph has been rediscovered independently by several authors since the 1960’s. Corneil, Lerchs and Burlingham [5] define cographs recursively by the following rules:

(i) a graph on a single vertex is a cograph,

(ii) a finite union of cographs is a cograph,

(iii) a finite join of cographs is a cograph.
In this note, we focus on representing the recursive construction of a cograph using its cotree, that we describe below.

A cotree $T_G$ of a cograph $G$ is a rooted tree in which any interior vertex $w$ is either of $\cup$ type (corresponding to disjoint union) or $\otimes$ type (corresponding to join). The terminal vertices (leaves) are typeless and represent the vertices of the cograph $G$. We say that the depth of the cotree is the number of edges of the longest path from the root to a leaf. To build a cotree for a connected cograph, we simply place a $\otimes$ at the tree’s root, placing $\cup$ on interior vertices with odd depth, and placing $\otimes$ on interior vertices with even depth. To build a cotree for a disconnected cograph, we place $\cup$ at the root, and place $\otimes$ at odd depths, and $\cup$ at even depths. All interior vertices have at least two children. In [2] this structure is called minimal cotree, but throughout this paper we call it simply a cotree. Figure 1 shows a cograph and its cotree with depth equals to 4.

Figure 1: The cograph $G = (((v_1 \cup v_2) \cup v_3) \lor v_4) \lor (((v_5 \lor v_6) \lor v_7) \cup v_8)) \lor v_9$ and its cotree.

Two vertices $u$ and $v$ are duplicates if $N(u) = N(v)$ and coduplicates if $N[u] = N[v]$. In a cograph, any collection of mutually coduplicates (resp. duplicates) vertices, e.g. with the same neighbors and adjacent (resp. not adjacent) have a common parent of type $\otimes$ (resp. $\cup$). In Figure 1 for example, we have that $v_1$ and $v_2$ are duplicates because $N(v_1) = N(v_2)$, while $v_5, v_6$ and $v_7$ are coduplicates. In fact, a recursive characterization of cographs in terms of the vertex duplication and co-duplication operations is given in [15].

2.2 Diagonalization

An algorithm for constructing a diagonal matrix congruent to $A + xI$, where $A$ is the adjacency matrix of a cograph, and $x$ is an arbitrary scalar, using $O(n)$ time and space
was developed in \cite{10}. This algorithm will be the main tool of this article and, hence, we will make a brief review of the method. For more information, see \cite{10}.

The algorithm’s input is the cotree $T_G$ and $x$. Each leaf $v_i, i = 1, \ldots, n$ has a value $d_i$ that represents the diagonal element of $A + xI$. It initializes all entries $d_i$ with $x$. Even though the operations represent rows and columns operations on the matrix $A + xI$, the algorithm is performed on the cotree itself and matrix is never actually used.

In each iteration of the procedure, a pair ${v_k, v_l}$ of duplicates or coduplicates vertices with maximum depth is selected. Then the pair is processed, that is, assignments are given to $d_k$ and $d_l$, such that either one or both rows (columns), corresponding to this vertices, are diagonalized. When a $k$ row(column) corresponding to vertex $v_k$ has been diagonalized then $v_k$ is removed from the cotree $T_G$, it means that $d_k$ has a permanent final value. Then the algorithm moves to the cotree $T_G - v_k$. The algorithm is shown in Figure 2.

\begin{algorithm}
\caption{Diagonalization algorithm}
\begin{algorithmic}
  \State INPUT: cotree $T_G$, scalar $x$
  \State OUTPUT: diagonal matrix $D = [d_1, d_2, \ldots, d_n]$ congruent to $A(G) + xI$
  \Procedure{Algorithm Diagonal}{$(T_G, x)$}
    \State initialize $d_i := x$, for $1 \leq i \leq n$
    \While{$T_G$ has $\geq 2$ leaves}
      \State select a pair $(v_k, v_l)$ (co)duplicate of maximum depth with parent $w$
      \State $\alpha \leftarrow d_k$, $\beta \leftarrow d_l$
      \If{$w = \otimes$
        \If{$\alpha + \beta \neq 2$
          \State $d_l \leftarrow \frac{\alpha \beta - 1}{\alpha + \beta - 2}$; \quad $d_k \leftarrow \alpha + \beta - 2$; \quad $T_G = T_G - v_k$
        \ElseIf{$\beta = 1$
          \State $d_l \leftarrow 1$ \quad $d_k \leftarrow 0$; \quad $T_G = T_G - v_k$
        \Else
          \State $d_l \leftarrow 1$ \quad $d_k \leftarrow -(1 - \beta)^2$; \quad $T_G = T_G - v_k$; \quad $T_G = T_G - v_l$
        \EndIf
      \ElseIf{$w = \cup$
        \If{$\alpha + \beta \neq 0$
          \State $d_l \leftarrow \frac{\alpha \beta}{\alpha + \beta}$; \quad $d_k \leftarrow \alpha + \beta$; \quad $T_G = T_G - v_k$
        \ElseIf{$\beta = 0$
          \State $d_l \leftarrow 0$; \quad $d_k \leftarrow 0$; \quad $T_G = T_G - v_k$
        \Else
          \State $d_l \leftarrow \beta$; \quad $v_k \leftarrow -\beta$; \quad $T_G = T_G - v_k$; \quad $T_G = T_G - v_l$
        \EndIf
      \EndIf
    \EndWhile
  \EndProcedure
\end{algorithmic}
\end{algorithm}

Figure 2: Diagonalization algorithm

Now, we will present a few results from \cite{10} that will be used throughout the note. The following theorem is based on Sylvester’s Law of Inertia.
Theorem 1 [10] Let $D = [d_1, d_2, \ldots, d_n]$ be the diagonal returned by the diagonalization algorithm $(T_G, -x)$, and assume $D$ has $k_+ \; \text{positive values}$, $k_0 \; \text{zeros}$ and $k_- \; \text{negative values}$. 

i The number of eigenvalues of $G$ that are greater than $x$ is exactly $k_+$. 

ii The number of eigenvalues of $G$ that are less than $x$ is exactly $k_-$. 

iii The multiplicity of $x$ is $k_0$.

The following two lemmas show that, under certain conditions, we can control the assignments made at each iteration.

Lemma 1 [10] If $v_1, \ldots, v_m$ have parent $w = \otimes$, each with diagonal value $y \neq 1$, then the algorithm performs $m - 1$ iterations of subcase 1a assigning, during iteration $j$:

$$d_k \leftarrow \frac{j + 1}{j} (y - 1) \quad (1)$$

$$d_l \leftarrow \frac{y + j}{j + 1} \quad (2)$$

Lemma 2 [10] If $v_1, \ldots, v_m$ have parent $w = \cup$, each with diagonal value $y \neq 0$, then the algorithm performs $m - 1$ iterations of subcase 2a assigning, during iteration $j$:

$$d_k \leftarrow \frac{(j + 1)}{j} y \quad (3)$$

$$d_l \leftarrow \frac{y}{j + 1} \quad (4)$$

The next three lemmas show that if we start an iteration with some known value then we can control the exit values.

Lemma 3 [10] If $\{v_k, v_l\}$ is a pair of coduplicate vertices processed by Diagonalization with assignments $0 \leq d_k, d_l < 1$, then $d_k$ becomes permanently negative, and $d_l$ is assigned a value in $(0, 1)$.

Lemma 4 If $\{v_k, v_l\}$ is a pair of duplicate vertices processed by Diagonalization with the assignments $0 < d_k, d_l \leq 1$, then $d_k$ becomes permanently positive, and $d_l$ is assigned a value in $(0, 1)$.

Proof: We notice that the algorithm executes subcase 2a, meaning that $d_k = \alpha + \beta > 0$ and $d_l = \alpha \beta / (\alpha + \beta)$. The fact that $d_l > 0$ is obvious. To see that $d_l < 1$, we observe that if $\alpha = \beta = 1$, then $d_k = 1/2$. If either (but not both) $\alpha$ or $\beta = 1$, then it is clear that $d_l = \alpha / (\alpha + 1) < 1$. Now if $0 < \alpha, \beta < 1$, then $d_l < 1$ follows from Lemma 3 of [3].
Lemma 5 During the execution of Diagonalize \((T_G, x)\) with \(x \in (0, 1)\), all diagonal values of vertices remaining on the cotree are in \((0, 1)\). Furthermore, if \(d_k\) corresponds to a permanent value of a removed vertex on \(T_G - v_k\), then \(d_k \neq 0\).

**Proof:** Let \(G\) be a cograph and \(T_G\) its cotree. Initially all vertices on \(T_G\) are in \((0, 1)\). Suppose after \(m\) iterations of Diagonalize all diagonal values of the cotree are in \((0, 1)\) and no zero is assigned. Now consider iteration \(m + 1\) with a pair \(\{v_k, v_l\}\) and parent \(w\). If \(w = \otimes\) then Lemma 3 guarantees the vertex \(d_l\) remaining on the cotree is assigned a value in \((0, 1)\) and the vertex \(d_k\) is assigned a permanently negative value. If \(w = \cup\) then Lemma 4 guarantees the vertex \(d_l\) remaining on the cotree is assigned a value in \((0, 1)\) and the vertex \(d_k\) is assigned a permanently positive value, completing the proof.

The next result follows from Lemma 5.

**Theorem 2** No cograph \(G\) has eigenvalue in the interval \((-1, 0)\).

![Figure 3: The cograph with cotree \(T_G(3, 2, 0|0, 0, 2)\).](image)

3 On the multiplicities of eigenvalues in balanced cotrees

In this section we study the eigenvalues of cographs that have balanced cotrees.

We say that a cograph \(G\) has a balanced cotree \(T_G\) with depth \(r\) if every interior vertex with depth \(i\) in \(T_G\) has the same number of interior vertices and the same number of leaves as direct successors, for \(i \in \{1, \ldots, r-1\}\). We will use the notation \(T_G(a_1, \ldots, a_r|b_1, \ldots, b_r)\) to represent a balanced cotree of a cograph \(G\), where the root of \(T_G\) has exactly \(a_1\) immediate interior vertices and \(b_1\) leaves. An interior vertex successor of the root has exactly \(a_2\) immediate interior vertices and \(b_2\) leaves, and so on. Thus, we will assume that \(a_1, \ldots, a_{r-1}\) are positive integers and \(a_r = 0\). Additionally, we assume that \(b_1, \ldots, b_{r-1}\) are non-negative integer values and \(b_r \geq 2\). Figure 3 shows the balanced cotree \(T_G(3, 2, 0|0, 0, 2)\).
3.1 Regular balanced cotrees

Here we will study eigenvalues of (regular) cographs $G$ that have balanced cotrees of the type $T_G(a_1, \ldots, a_{r-1}, 0|0, \ldots, 0, b_r)$, whose order is $n = a_1a_2 \ldots a_{r-1}b_r$. We show in Figure 4 a representation of general regular balanced cotree with odd $r$, meaning that level $r - 1$ has vertices of type $\otimes$.

$$\begin{align*}
    a_1 \otimes \cdots \otimes a_1 \\
    a_2 \otimes \cdots \otimes a_1a_2 \\
    \vdots \\
    a_{r-3} \otimes a_1 \cdots a_{r-3} \\
    a_{r-2} \otimes \cdots \otimes a_1 \cdots a_{r-2} \\
    a_{r-1} \otimes \cdots \otimes a_1 \cdots a_{r-1} \\
    b_r \bullet \cdots \bullet a_1 \cdots a_{r-1}b_r
\end{align*}$$

Figure 4: Cotree $T_G(a_1, \ldots, a_{r-1}, 0|0, \ldots, 0, b_r)$ with $r$ odd.

The next two theorems are known results and can be found, for example in [2, 10].

**Theorem 3** Let $G$ be a cograph with cotree $T_G$ having $\otimes$-nodes $\{w_1, \ldots, w_m\}$, where $w_i$ has $t_i \geq 1$ terminal children. Then $m(-1) = \sum_{i=1}^{m}(t_i - 1)$.

**Theorem 4** Let $G$ be a cograph with cotree $T_G$ having $\cup$-nodes $\{w_1, \ldots, w_m\}$, where $w_i$ has $t_i \geq 1$ terminal children. If $G$ has $j \geq 0$ isolated vertices then $m(0) = j + \sum_{i=1}^{m}(t_i - 1)$.

Using the above results we can easily prove the next corollary.

**Corollary 1** Let $G$ be a cograph with balanced cotree $T_G(a_1, \ldots, a_{r-1}, 0|0, \ldots, 0, b_r)$ of order $n = a_1a_2 \ldots a_{r-1}b_r$.

(i) If $r$ is odd then $G$ has the eigenvalue $-1$ with multiplicity $a_1a_2 \ldots a_{r-1}(b_r - 1)$.

(ii) If $r$ is even then $G$ has the eigenvalue $0$ with multiplicity $a_1a_2 \ldots a_{r-1}(b_r - 1)$. 
Corollary 2 Let $G$ be a balanced cotree $T_G(a_1, \ldots, a_{r-1}, 0|0, \ldots, 0, b_r)$ of a cograph $G$ of order $n = a_1 a_2 \ldots a_{r-1} b_r$. If $r$ is odd (even) then, counting multiplicities, the number of eigenvalues of $G$ other than $-1$ (0) is equal to

$$a_1 a_2 \ldots a_{r-1}.$$  \hfill (5)

Proof: Suppose $r$ is odd. Since $n = a_1 a_2 \ldots a_{r-1} b_r$ and $G$ has $a_1 a_2 \ldots a_{r-1} (b_r - 1)$ coduplicates vertices, it follows that the number of eigenvalues that are distinct from $-1$ is equal to $n - a_1 a_2 \ldots a_{r-1} (b_r - 1) = a_1 a_2 \ldots a_{r-1}$. The case $r$ even is similar.

Lemma 6 Let $G$ be a cograph with balanced cotree $T_G(a_1, \ldots, a_{r-1}, 0|0, \ldots, 0, b_r)$ of order $n = a_1 a_2 \ldots a_{r-1} b_r$.

(i) If $r$ is odd then $G$ has the eigenvalue $b_r - 1$ with multiplicity $a_1 a_2 \ldots (a_{r-1} - 1)$.

(ii) If $r$ is even then $G$ has the eigenvalue $-b_r$ with multiplicity $a_1 a_2 \ldots (a_{r-1} - 1)$.

Proof: We assume that $r$ is odd. The case even is similar. Consider $x = -(b_r - 1)$ and execute the algorithm Diagonalization with input $(T_G, x)$. By Theorem 1 we have to prove that the algorithm creates $a_1 a_2 \ldots (a_{r-1} - 1)$ null permanent values. Since $G$ has coduplicate vertices, see Figure 4 and $x \neq 1$, we apply Lemma 1 and after $b_r - 1$ iterations for each $\otimes$ vertice at level $r - 1$, the remaining vertices on the cotree receive

$$d_t \leftarrow \frac{-(b_r - 1) + b_r - 1}{b_r - 1 + 1} = 0,$$

and the removed vertices receive

$$d_k \leftarrow -\frac{j + 1}{j} b_r < 0$$

for $j = 1, \ldots, b_r - 1$. 

Figure 5: Processing deepest $\otimes$ level.
This is illustrated on the left of Figure 5.

Now the leaves at level $r$ move up to the $\cup$ vertices as on the right of Figure 5 and we process them. Notice that we have duplicate leaves with null value. Then the algorithm performs subcase 2b at the leaves in each vertex $\cup$ and it creates $a_1a_2\ldots(a_{r-1} - 1)$ permanent zeros in the removed vertices. The remaining vertices keep the value zero, as shown on the left of Figure 6. So $m(b_r - 1) \geq a_1a_2\ldots(a_{r-1} - 1)$.

Now we show that no more permanent zeros are created. The zero value vertices now move up to the next $\otimes$ level. Notice that, see right of Figure 6, we have coduplicate vertices in the remaining tree with assignments equal to 0. Using Lemma 5 once and then Lemma 3, we know that no null value will be generated and it proves that $m(b_r - 1) = a_1a_2\ldots(a_{r-1} - 1)$.

In the next theorem we present a bound for the eigenvalues of regular balanced cotrees.

**Theorem 5** Let $G$ be a cograph with balanced cotree $T_G(a_1, \ldots, a_{r-1}, 0|0, \ldots, 0, b_r)$ order $n = a_1a_2\ldots a_{r-1}b_r$.

(i) If $r$ is odd and $\lambda \neq -1$, $b_r - 1$ then $m(\lambda) \leq a_1\cdots a_{r-2}$;

(ii) If $r$ is even and $\lambda \neq 0$, $-b_r$ then $m(\lambda) \leq a_1\cdots a_{r-2}$.

**Proof:** Suppose that $r$ is odd. Then $m(\lambda) \leq n - m(-1) - m(b_r - 1) = a_1a_2\ldots a_{r-1}b_r - a_1a_2\cdots a_{r-1}(b_r - 1) - a_1a_2\ldots(a_{r-1} - 1) = a_1\cdots a_{r-2}$.

3.2 Non-regular balanced cotrees

We now define two types of cotrees depending on whether its depth $r$ is even or odd. Let $T_G(a_1, \ldots, a_{r-1}, 0|b_1, b_2, \ldots, b_r)$ be a balanced cotree defined as follows:
If $r$ is even then, for $1 \leq i \leq r - 1$, \[
\begin{cases} 
 b_i = 0 & \text{if } i \text{ is odd;} \\
 b_i \geq b_r & \text{if } i \text{ is even.} 
\end{cases}
\]

If $r$ is odd then, for $1 \leq i \leq r - 1$, \[
\begin{cases} 
 b_i = 0 & \text{if } i \text{ is even;} \\
 b_i \geq b_r & \text{if } i \text{ is odd.} 
\end{cases}
\]

**Theorem 6** Let $G$ be a cograph with balanced cotree $T_G(a_1, \ldots, a_{r-1}, 0|b_1, b_2, \ldots, b_r)$ defined above.

(i) If $r$ is odd then $G$ has the eigenvalue $b_r - 1$ with multiplicity $a_1 a_2 \ldots (a_{r-1} - 1)$.

(ii) If $r$ is even then $G$ has the eigenvalue $-b_r$ with multiplicity $a_1 a_2 \ldots (a_{r-1} - 1)$.

**Proof:** We assume that $r$ is even. The case odd is similar. The illustration of the initial configuration is given on the left of Figure 7. Consider $x = -b_r$ and execute the algorithm Diagonalization with input $(T_G, x)$. By Theorem 1 we have to prove that the algorithm creates at least $a_1 a_2 \ldots (a_{r-1} - 1)$ permanent null values. Applying Lemma 2 at each vertex $\cup$ at level $r - 1$, the following assignments are made

\[
\begin{align*}
 d_k & \leftarrow \frac{(j+1)b_r}{j} b_k > 0, \quad j = 1, \ldots, b_r - 1; \\
 d_l & \leftarrow \frac{b_r}{b_r - 1 + 1} = 1.
\end{align*}
\]

The removed leaves have a permanent positive value and the remaining vertices have value 1, as illustrated on the right of Figure 7.

![Diagram](image)

Figure 7: Processing the deepest $\cup$ level

Now the vertices remaining (with value 1) are moved up and become leaves of a $\otimes$ vertex, as seen on the left of Figure 8. We perform subcase 1b and then the $a_1 \cdots a_{r-2}(a_{r-1} - 1)$ removed vertices receive the value 0 and the remaining leaves receive 1 as shown on the right of Figure 8 and so $m(-b_r) \geq a_1 a_2 \ldots (a_{r-1} - 1)$.

Now the vertices with value 1 move to level $r - 2$ as shown on the left of Figure 9. At each vertex $\cup$ at level $r - 3$ we start processing the vertices with value 1, and by Lemma
Then we process the vertices with value \( b_r \) using Lemma 2:

\[
d_k \leftarrow \frac{(j+1)}{j} b_r > 0, \quad j = 1, \ldots, b_r - 2 - 1;
\]

\[
dl \leftarrow \frac{b_r}{b_r - 2 + 1} = \frac{b_r}{b_r - 2}
\]

The right of Figure 9 represents the last iteration in each vertex \( \cup \) at level \( r - 2 \). Notice that each remaining leaf has a value in \((0, 1]\) and using the same argument as in Lemma 3 we can prove that no more zeros are assigned and the remaining vertices on the cotree are in \((0, 1]\), proving that \( m(-b_r) = a_1 a_2 \ldots (a_{r-1} - 1) \).

\[\frac{1}{a_{r-2}} < 1 \quad b_r \quad \frac{b}{b_{r-2}} \leq 1\]

**Figure 9: Processing level \( r - 2 \).**

### 4 Borderenergetic Cographs

In this section we present some families of non-cospectral and borderenergetic cographs.

Consider the cograph \( G = K_a \otimes (a - 1)(b - 1)K_b \), of order \( n = a + b(a - 1)(b - 1) \).

We observe that \( G \) has the balanced cotree \( T_G(1, (a - 1)(b - 1), 0|a, 0, b) \), represented in Figure 10.

**Lemma 7** Let \( G = K_a \otimes (a - 1)(b - 1)K_b \) be the cograph \( G \) of Figure 10 of order \( n = a + b(a - 1)(b - 1) \), for fixed values \( a \geq b \geq 2 \). The spectrum of \( G \) is

\[ -(a - 1)(b - 1); -1; b - 1; ab - 1 \]

with multiplicity

\[ 1; (a - 1)[(b - 1)^2 + 1]; (a - 1)(b - 1) - 1; 1, \]
respectively.

**Proof:** Using Theorem with $m = (a - 1)(b - 1) + 1$, $t_1 = \cdots = t_{m-1} = b$ and $t_m = a$. We compute the multiplicity of $-1$:

$$m(-1) = \sum_{i=1}^{m} (t_i - 1) = (m - 1)(b - 1) + (a - 1) = (a - 1)[(b - 1)^2 + 1].$$

Since $T_G(1, (a - 1)(b - 1), 0|a, 0, b)$ has a non regular balanced cotree, by Lemma $b - 1$ is an eigenvalue with multiplicity $(a - 1)(b - 1) - 1$. Now, we will prove, by Theorem that $ab - 1$ is an eigenvalue of multiplicity 1 by showing that the algorithm Diagonalize with input $(T_G, -ab + 1)$, creates a single zero in the $T_G$.

We initialize the leaves with value $-ab + 1 \neq 1$. Then we can use Lemma and for each $\otimes$ vertex, we have that

$$d_k \leftarrow \frac{(j+1)}{j}(-ab), \quad j = 1, \ldots, b - 1;$$
$$d_l \leftarrow -a + 1$$

where $d_k$ represents the removed leaves and $d_l$ the remaining ones. The left of Figure represents the cotree yet to be processed. Now, the leaves at depth 3 move up to the $\cup$ vertices at depth 2, as on the right of Figure.
In the next step we use Lemma 2 because the duplicate vertices at depth 2 have assignments equal to \(-a + 1 \neq 0\). We obtain
\[
d_k \leftarrow \frac{j+1}{2}(-a + 1), \quad j = 1, \ldots, (a - 1)(b - 1) - 1;
\]
\[
d_i \leftarrow \frac{1}{b-1}.
\]
As the left of Figure 12 shows, the remaining leaf at depth 2 moves up to depth 1, as on the right of Figure 12.

Figure 12: Processing level 2

At depth one, there are \(a + 1\) coduplicate vertices. \(a\) with value \(-ab + 1\) and one with value \(\frac{1}{b-1}\) as the right of Figure 12. The algorithm processes, by Lemma 1, the leaves with value \(-ab + 1\) first and it generates the following assignments
\[
d_k \leftarrow \frac{j+1}{2}(-ab), \quad j = 1, \ldots, (a - 1);
\]
\[
d_i \leftarrow -b + 1.
\]

The last step of the algorithm is to process the two remaining vertices whose values are \(\alpha = -b + 1\) and \(\beta = \frac{1}{b-1}\). Since \(\alpha, \beta < 0\) then the algorithm performs subcase 1a and assigns
\[
d_k \leftarrow \alpha + \beta - 2 = \frac{-b^2}{b-1} \quad \text{and} \quad d_i \leftarrow \frac{\alpha \beta - 1}{\alpha + \beta - 2} = 0,
\]
creating a negative value and a zero for the last two diagonal entries, so \(m(ab - 1) = 1\).

Using the fact that sum of eigenvalues must be zero, we obtain the remaining eigenvalue \(-(a - 1)(b - 1)\) of \(G\), proving the result.

The following theorem follows directly from Lemma 7 and summarizes the results for the family of cographs represented in Figure 10.

**Theorem 7** Let \(G = K_a \otimes (a - 1)(b - 1)K_b\) be the cograph of order \(n = a + b(a - 1)(b - 1)\), for fixed values \(a \geq b \geq 2\) represented in Figure 10. Then \(G\) is an integral cograph, non-cospectral and borderenergetic to \(K_n\).
Proof: It is well known that the \( \text{Spec}(K_n) = \{ (-1)^{n-1}, (n-1)^1 \} \) and, hence, \( E(K_n) = 2(n-1) \). Using Lemma 7 we can compute the energy of \( G \) as follows
\[
E(G) = (a-1)(b-1) + (1)(a-1)[(b-1)^2+1] + (b-1)[(a-1)(b-1)-1] + (ab-1) = 2(n-1).
\]

Consider now the cograph \( G = (p+1)K_2 \otimes (p+1)K_2 \), of order \( n = 4p + 4 \), whose regular balanced cotree \( T_G(2, p+1, 0|0, 0, 2) \) is represented in Figure 13.

![Figure 13: The cotree \( T_G \)](image)

Lemma 8 Let \( G = (p+1)K_2 \otimes (p+1)K_2 \) be a cograph of order \( n = 4p + 4 \), for a fixed value \( p \geq 1 \). Then the spectrum of \( G \) is
\[
-(2p+1); -1; 2p+3
\]
with multiplicity
\[
1; 2(p+1); 2p; 1,
\]
respectively.

Proof: Notice that, using Theorem 3 we can consider that \( m = 2(p+1), t_1 = \cdots = t_m = 2 \). Then the multiplicity of \(-1\) is
\[
\sum_{i=1}^{m} (t_i - 1) = 2(p+1).
\]
Noticing that \( T_G(2, p+1, 0|0, 0, 2) \) is a regular balanced cotree, we can apply Lemma 6 to obtain that \( m(1) = 2p \). To obtain that \( m(-(2p+1)) = 1 \) we just execute the algorithm diagonalize with input \( (T_G, 2p+1) \) and observe that it creates a single zero on the \( T_G \).
The eigenvalue \( 2p+3 \) is determined by the fact that the eigenvalues must sum zero.

Theorem 8 Let \( G = (p+1)K_2 \otimes (p+1)K_2 \) be the cograph of order \( n = 4p + 4 \), represented in Figure 13, for a fixed value \( p \geq 1 \). Then \( G \) is integral, non-cospectral and borderenergetic to \( K_n \).
Proof: Using Lemma 8 we have that $E(G) = 8p + 6$ and $E(K_n) = 2(n - 1) = 8p + 6$. And the result follows.

References

[1] F.K. Bell, P. Rawlinson, On the multiplicities of graph eigenvalues, *Bull. Lond. Math. Soc.* 35 (2003) 401–408.

[2] T. Bıyıkoğlu, S. K. Simić, Z. Stanić, Some notes on spectra of cographs, *Ars Comb.* 100 (2011) 421–434.

[3] C. Bu, X. Zhang, J. Zhou, A note on the multiplicities of graph eigenvalues, *Lin. Algebra Appl.* 442 (2014) 69–74.

[4] Gerard. J. Chang, Liang-Hao Huang, Hong-Gwa Yeh, On the rank of a cograph, *Lin. Algebra Appl.* 429 (2008), 601-605.

[5] D.G. Corneil, H. Lerchs and L. Stewart Birmingham; Complement reducible graphs, *Discr. Appl. Math.* 3 (1981) 163–174.

[6] B. Deng, X. Li and I. Gutman, More on borderenergetic graphs, *Lin. Algebra Appl.* 497 (2016) 199-208.

[7] C. Godsil, G. Royle, *Algebraic Graph Theory*, Springer-Verlag, New York, 2001.

[8] S. C. Gong, X. Li, G. H. Xu, I. Gutman, B. Furtula, Borderenergetic graphs, *MATCH Commun. Math. Comput. Chem.* 74 (2015) 321-332.

[9] Y. Hou, Q. Tao, Borderenergetic threshold graphs, *MATCH Commun. Math. Comput. Chem.* 75 (2016) 253–262.

[10] D. P. Jacobs, V. Trevisan, F. C. Tura, Eigenvalue location in cographs, *Discr. Appl. Math.* (2017), [http://dx.doi.org/10.1016/j.dam.2017.02.007](http://dx.doi.org/10.1016/j.dam.2017.02.007).

[11] D. P. Jacobs, V. Trevisan, F. Tura, Eigenvalues and energy in threshold graphs, *Lin. Algebra Appl.* 465 (2015) 412–425.

[12] D. P. Jacobs, V. Trevisan, F. Tura, Eigenvalue location in threshold graphs, *Lin. Algebra Appl.* 439 (2013) 2762–2773.
[13] X. Li, M. Wei, S. Gong, A computer search for the borderenergetic graphs of order 10, *MATCH Commun. Math. Comput. Chem.* **74** (2015) 333-342.

[14] X. Li, Y. Shi, I. Gutman, *Graph Energy*, Springer, New York, 2012.

[15] A. Mohammadian, V. Trevisan, Some spectral properties of cographs, *Discr. Math.* **339** (2016) 1261–1264.

[16] P. Rowlinson, On multiple eigenvalues of tree, *Lin. Algebra Appl.* **432** (2010), 3007-3011.

[17] G. F. Royle, The rank of cographs, *Elec. J. Comb.* **10** (2003) N0 1. Research paper N 11:7 p.

[18] J. Salez, Every totally real algebraic integer is a tree eigenvalue, *J. Comb. Theory Ser. B* **111** (2015) 249-256.

[19] I. Sciriha, S. Farrugia, On the spectrum of threshold graphs, *ISRN Discr. Math.* (2011) doi:10.5402/2011/108509.

[20] Z. Shao, F. Deng, Correcting the number of borderenergetic graphs of order 10, *MATCH Commun. Math. Comput. Chem.* **75** (2016) 263-266.