Generating I-eigenvalue free threshold graphs

Luiz Emilio Allem*

Instituto de Matemática e Estatística
UFRGS
Porto Alegre, Rio Grande do Sul, Brazil, 91509-900.

emilio.allem@ufrgs.br

Elismar R. Oliveira

Instituto de Matemática e Estatística
UFRGS
Porto Alegre, Rio Grande do Sul, Brazil, 91509-900.

elismar.oliveira@ufrgs.br

Fernando Tura †

Departamento de Matemática
UFSM
Santa Maria, Rio Grande do Sul, Brazil, 97105-900.

fernando.tura@ufsm.br

Submitted: May 4, 2022; Accepted: Mar 9, 2023; Published: May 19, 2023

Abstract

A graph is said to be I-eigenvalue free if it has no eigenvalues in the interval I with respect to the adjacency matrix A. In this paper we present two algorithms for generating I-eigenvalue free threshold graphs.

Mathematics Subject Classifications: 05C50, 05C75,05C85

1 Introduction

In 1972 A. J. Hoffman [7] presented a remarkable result about the density of eigenvalues on the real line. Let $\mathcal{A}$ be the set of all symmetric matrices of all orders, every entry of which is a non-negative integer and $R = \{\rho : \rho = \rho(A) \text{ for some } A \in \mathcal{A}\}$ where $\rho(A)$ is

*Supported by FAPERGS grant 21/2551- 0002053-9.

†Supported by CNPQ grant 200716/2022-0.

the largest eigenvalue of A. He determined all limit points of $R \leqslant \sqrt{2+\sqrt{5}}$. Then, in 1989, Shearer [12] extended this result. He showed that every point $\geqslant \sqrt{2+\sqrt{5}}$ is a limit point of R. Since then, several interesting results have been obtained on this topic as you can see in [13], where it is proved that every real number is a limit point of eigenvalues of graphs.

Given that, it seems counterintuitive that some classes of graphs have no eigenvalues in a prescribed, arbitrary size interval. Nevertheless, Jacobs et al. [8] showed that threshold graphs are $(-1,0)$-eigenvalue free. And, in [1], Aguilar et al. proved that anti-regular graphs, a subclass of threshold graphs, have no eigenvalues in the interval $\Omega = \left[\frac{-1-\sqrt{2}}{2}, \frac{-1+\sqrt{2}}{2}\right]$ except the trivial eigenvalues -1 and 0. This result was extended by Ghorbani in [4] by showing that threshold graphs have no eigenvalues in Ω except the trivial ones.

Searching for graphs that have no eigenvalues in some interval, in this paper, we provide two algorithms for generating infinite families of I-eigenvalue free threshold graphs. More specifically, given a positive real number N and a natural number r one algorithm constructs a threshold graph G with associated cotree $T_G(a_1, a_2, \ldots, a_r)$ of depth r, which will be the initial threshold graph for generating infinite families of $(0, N]$-eigenvalue free threshold graphs. And, the other one provides a similar result for the interval $[M, -1)$ where M is a negative real number. We show that our approach also can be used to prove that threshold graphs are $\left(\frac{-1-\sqrt{2}}{2}, -1\right]$ and $\left[0, \frac{-1+\sqrt{2}}{2}\right)$-eigenvalue free which shows that we have a generalization of the above mentioned results.

Here is an outline of the paper. In Section 2 we describe the cotrees associated to threshold graphs, and in Section 3 we recall an algorithm for locating eigenvalues in a cograph G developed in [9] and present some of its applications that include determining the inertia of G just using its associated cotree. In Section 4 we explain our strategy based on the inertia of the graph. The algorithms for generating infinite families of I-eigenvalue free threshold graphs are given in Section 5 and 6. In Section 7 we exhibit such families. Our method is used to prove that threshold graphs are $\left(\frac{-1-\sqrt{2}}{2}, -1\right]$ and $\left[0, \frac{-1+\sqrt{2}}{2}\right)$-eigenvalue free in Section 8. Finally, in Section 9 we discuss about some open problems.

2 Threshold representation

Let $G = (V, E)$ be an undirected graph with vertex set V and edge set E. If $|V| = n$, then its adjacency matrix $A(G) = [a_{ij}]$ is the $n \times n$ matrix of zeros and ones such that $a_{ij} = 1$ if and only if v_i is adjacent to v_j (that is, there is an edge between v_i and v_j). For $v \in V$, $N(v)$ denotes the open neighborhood of v, that is $\{w | \{v, w\} \in E\}$ and $N[v] = N(v) \cup \{v\}$ the closed neighborhood. A value λ is an eigenvalue if $\det(\lambda I_n - A) = 0$, and since A is real symmetric its eigenvalues are real. In this paper, a graph's eigenvalues are the eigenvalues of its adjacency matrix.

The class of threshold graphs was introduced by Chvátal and Hammer [3] and Henderson and Zalestein [6] in 1977 and their numerous applications go from computer science to psychology [10].

A threshold graph can be constructed through an iterative process which starts with an isolated vertex, and at each step either a new isolated vertex is added or a dominating vertex is added, i.e., a vertex adjacent to all previous vertices is added. More specifically, we may represent a threshold graph on n vertices using a binary sequence $(b_1, b_2, \ldots, b_n)$ as follows: $b_i = 0$ if vertex v_i is added as an isolated vertex, and $b_i = 1$ if v_i is added as a dominating vertex. In Figure 1 we show the threshold graph G with binary sequence $b = (1, 1, 1, 1, 0, 0, 0, 1, 1)$. It can also be seen as consecutive blocks of 0's and 1's, for instance $b = (1, 1, 1, 1, 0, 0, 0, 1, 1) = 1^4 0^3 1^2$. Notice that we order the vertices of G in the same way they are given in their sequence.

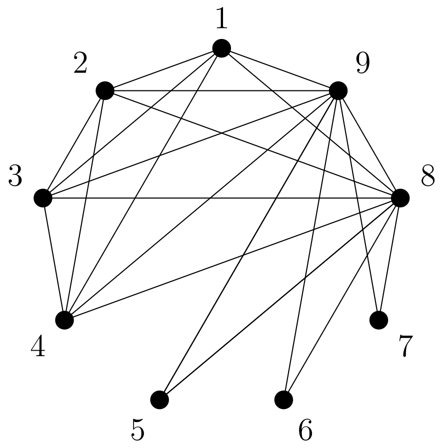

Figure 1: $b = (1, 1, 1, 1, 0, 0, 0, 1, 1) = 1^4 0^3 1^2$

Threshold graphs are a subclass of cographs and each cograph can be represented by a cotree, for more details see [2, 9]. It is interesting that for threshold graphs, the cotree is a caterpillar, as shown in [11].

In this note, we focus on representing the threshold graph using its cotree (caterpillar), that we describe below. A cotree T_G of a threshold graph is a rooted path in which any interior vertex w is either of union $\cup$ type (corresponding to a block of 0's) or join $\otimes$ type (corresponding to a block of 1's). The terminal vertices (leaves) are typeless and represent the vertices of the threshold. Since we work only with connected thresholds graphs in this paper, our cotree is basically defined by placing a $\otimes$ node at the trees's root. And then, placing $\cup$ on interior nodes with odd depth, and placing $\otimes$ on interior nodes with even depth.

Notice that, if a cotree T_G associated to a threshold graph has an even depth then its final interior node is a $\cup$ type, and if its depth is odd then it is a $\otimes$ type as in Figure 2.

The cotree denoted by $T_G(a_1, a_2, \ldots, a_r)$ with $a_i \geqslant 1$ for $1 \leqslant i \leqslant r-1$, $a_r \geqslant 2$ and r odd is depicted in Figure 2. Notice that following our notation each interior node at depth i has a_i terminal vertices (leaves).

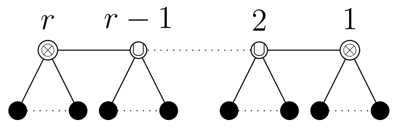

Figure 2: Cotree $T_G(a_1, a_2, \ldots, a_r)$.

The binary representation corresponding to the cotree in Figure 2 is given by $b = \left(1^{a_r}, b_2^{a_{r-1}}, \ldots, b_{r-1}^{a_2}, 1^{a_1}\right)$. For instance, the threshold with binary representation $b = (1,1,1,1,0,0,0,1,1) = 1^4 0^3 1^2$ has the cotree representation $T_G(2,3,4)$ depicted in Figure 3.

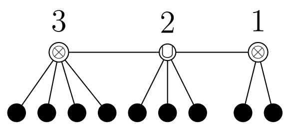

Figure 3: Cotree $T_G(2,3,4)$.

We would like to point out that in this paper we only deal with connected threshold graphs. It means that in the binary representation we always have $b_n = 1$ and in the cotree representation our first node always is a $\otimes$.

Finally, we define siblings vertices given its important role in the structure of cographs and in the following algorithm. Two vertices u and v are duplicates if $N(u) = N(v)$ and coduplicates if $N[u] = N[v]$. Therefore, we call u and v siblings if they are either duplicates or coduplicates.

3 Algorithm Diagonalize

The algorithm presented in this section was developed in [9]. Basically, it constructs a diagonal matrix D congruent to $A + xI$, where A is the adjacency matrix of a cograph G.

Algorithm 1 Diagonalize (T_G, x)

Input: cotree T_G, scalar x
Output: diagonal matrix $D = [d_1, d_2, \ldots, d_n]$ congruent to $A(G) + xI$

initialize $d_i := x$, for $1 \leqslant i \leqslant n$
while T_G has $\geqslant 2$ leaves **do**
  select siblings $\{v_k, v_l\}$ of maximum depth with parent w
  $\alpha \leftarrow d_k$ $\beta \leftarrow d_l$
  if $w = \otimes$ **then**
    if $\alpha + \beta \neq 2$ **then** `//subcase 1a`
      $d_l \leftarrow \frac{\alpha\beta - 1}{\alpha + \beta - 2}$; $d_k \leftarrow \alpha + \beta - 2$; $T_G = T_G - v_k$
    else if $\beta = 1$ **then** `//subcase 1b`
      $d_l \leftarrow 1$ $d_k \leftarrow 0$; $T_G = T_G - v_k$
    else `//subcase 1c`
      $d_l \leftarrow 1$ $d_k \leftarrow -(1-\beta)^2$; $T_G = T_G - v_k$; $T_G = T_G - v_l$
    end if
  else if $w = \cup$ **then**
    if $\alpha + \beta \neq 0$ **then** `//subcase 2a`
      $d_l \leftarrow \frac{\alpha\beta}{\alpha + \beta}$; $d_k \leftarrow \alpha + \beta$; $T_G = T_G - v_k$
    else if $\beta = 0$ **then** `//subcase 2b`
      $d_l \leftarrow 0$ $d_k \leftarrow 0$; $T_G = T_G - v_k$
    else `//subcase 2c`
      $d_l \leftarrow \beta$ $d_k \leftarrow -\beta$; $T_G = T_G - v_k$; $T_G = T_G - v_l$
    end if
  end if
end while

We would like to point out that the Diagonalize(T_G, x) works bottom up since the cotree is represented in the same way. In this note we just work with threshold graphs so the cotrees we use are depicted in Figure 2. Therefore, throughout the text we represent the steps bottom up by steps from left to right.

In this article given a graph G and a scalar $a \in \mathbb{R}$ we define the triple (a_+, a_0, a_-), where a_+ denotes the number of eigenvalues of G that are greater than a, a_0 the multiplicity of a and a_- the number of eigenvalues of G that are less than a. Therefore, the inertia of a graph G, using our notation, is the triple $(0_+, 0_0, 0_-)$. The following results presented in this section are from [9] and will be used throughout the note. The next one shows that the Algorithm 1 computes the triple (a_+, a_0, a_-).

Theorem 1. *Let $D = [d_1, d_2, \ldots, d_n]$ be the diagonal returned by Diagonalize$(T_G, -a)$, and assume D has a_+ positive values, a_0 zeros and a_- negative values. Then*

i: The number of eigenvalues of G that are greater than a is exactly a_+.

ii: The number of eigenvalues of G that are less than a is exactly a_-.

iii: The multiplicity of a is a_0.

The following three results show that we can obtain information about the localization of certain eigenvalues of the cograph G just by analysing its associated cotree T_G.

Theorem 2. *Let G be a cograph with cotree T_G having $\otimes$-nodes $\{w_1, \ldots, w_j\}$, and assume each w_i has t_i children in T_G. Then*

$$n_-(G) = \sum_{i=1}^{j} (t_i - 1).$$

Theorem 3. *Let G be a cograph with cotree T_G having $\cup$-nodes $\{w_1, \ldots, w_m\}$, where w_i has $t_i > 0$ terminal children. If G has $j \geqslant 0$ isolated vertices, then*

$$n_0(G) = j + \sum_{i=1}^{m} (t_i - 1).$$

Theorem 4. *Let G be a cograph with cotree T_G having $\otimes$-nodes $\{w_1, \ldots, w_m\}$, where w_i has $t_i > 0$ terminal children. Then the multiplicity of -1 is*

$$\sum_{i=1}^{m} (t_i - 1).$$

The next two lemmas provide an alternative initialization of the algorithm i.e., first we perform assignments to the leaves with identical value of the cotree and then we move on with the specialized cotree.

Lemma 5. *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:*

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

$$d_l \leftarrow \frac{y+j}{j+1}. \tag{2}$$

Lemma 6. *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:*

$$d_k \leftarrow \frac{j+1}{j}\, y \tag{3}$$

$$d_l \leftarrow \frac{y}{j+1}. \tag{4}$$

One advantage of the algorithm is that it is applied directly in the cotree so the diagonal values processed during the iterations will be labeled in the vertices. As pointed out before, the algorithm progresses bottom up but in this article we work only with threshold graphs so we will represent these iterations from left to right. The next example elucidates these observations.

Example 7. We will apply Diagonalize to the cograph G with cotree $T_G(2,3,4)$ and $x=-1$. Since we process the algorithm directly in the cotree, the diagonal values d_i's will appear at the terminal vertices v_i's during the execution. Each leaf is initialized with the value $d_i=-1$ as in Figure 4. Since multiple leaves of the same parent ($\otimes$ or $\cup$) have the same diagonal value $y=-1$ at the initialization, we can begin performing assignments to the terminal children (leaves) of the cotree using Lemmas 5 and 6 as follows.

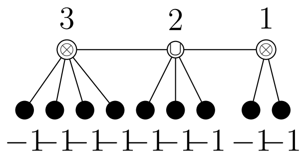

Figure 4: Diagonalize($T_G(2,3,4),-1$).

At depth 3, there are 4 leaves all with value $y=-1$, and using Lemma 5, 3 iterations are performed. According to equation (1), -4, -3, $\frac{-8}{3}$ are the permanent diagonal values assigned. And, by (2), the final remaining value is

$$d_l \leftarrow \frac{1}{2}.$$

Figure 5 depicts the cotree after Lemma 5 being applied at $\otimes$-nodes at depth 1 and 3, and Lemma 6 at $\cup$-node at depth 2.

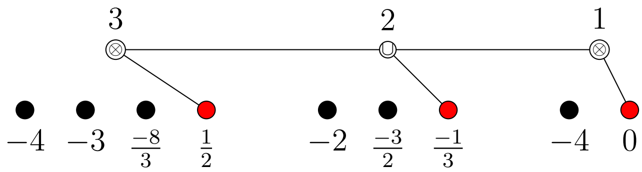

Figure 5: Specialized cotree $T_G(2,3,4)$.

Notice that the vertices associated to permanent values are removed and we proceed with the cotree with the remaining vertices and its values as depicted in Figure 6 left. The last vertex with value $\frac{1}{2}$ is relocated to the next level as in Figure 6 right.

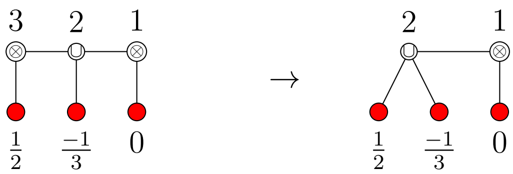

Figure 6: Applying **subcase 2a**.

Select the sibling pair $\{v_k, v_l\}$ labeled with $d_k=\frac{1}{2}$ and $d_l=\frac{-1}{3}$ at the $\cup$-node at depth 2 and initialize $\alpha \leftarrow d_k=\frac{1}{2}$ and $\beta \leftarrow d_l=\frac{-1}{3}$ as illustrated in Figure 6 right. Then

subcase 2a is executed and the assignments $d_k \leftarrow \frac{1}{6}$ and $d_l \leftarrow -1$ are made as shown in Figure 7 left. Then, the remaining vertex is relocated to the next level as in Figure 7 right.

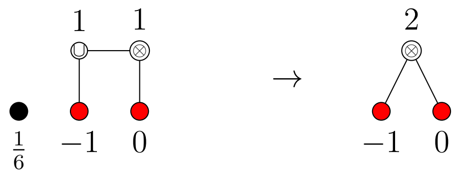

Figure 7: Applying **subcase 1a**.

Now, as $\alpha = -1$ and $\beta = 0$, **subcase 1a** is executed at the last iteration and the assignments $d_k \leftarrow -3$ and $d_l \leftarrow \frac{1}{3}$ are made as shown in Figure 8.

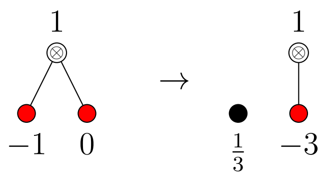

Figure 8: Final remaining value.

The algorithm stops, and the final diagonal is formed by the permanent values and the last remaining value as shown in Figure 9. Therefore, according to Theorem 1 $1_+ = 2$, $1_- = 7$ and $1_0 = 0$.

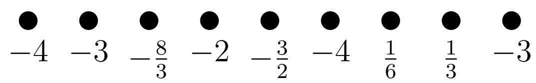

Figure 9: Final diagonal.

4 Our strategy

We would like to begin this section with two observations. First, suppose we have a threshold graph G such that $0_+ = N_+$ for a positive real number N, that is, the number of eigenvalues of G that are greater than 0 is equal to the number of eigenvalues of G that are greater than N. Therefore, $0_+ = N_+$ implies that G is $(0, N]$-eigenvalue free. Second, suppose we have a threshold graph G such that $M_+ = (-1)_0 + 0_0 + 0_+$ for a negative real number $M < -1$. Then, the number of eigenvalues of G that are greater than M is equal to the number of eigenvalues of G that are greater than or equal to -1. That is, $M_+ = (-1)_0 + 0_0 + 0_+$ implies that G is $[M, -1)$-eigenvalue free.

Next, we explain our strategy for generating a threshold graph G that is $(0, N]$-eigenvalue free for a real number $N > 0$. Our method consists in constructing a threshold graph G with cotree $T_G(a_1, a_2, \ldots, a_r)$, by choosing each $a_i \geqslant 1$ for $1 \leqslant i \leqslant r-1$ and $a_r \geqslant 2$, and showing that this threshold graph is $(0, N]$-eigenvalue free using the fact that $N_+ = 0_+$. Next we compute 0_+ as a function of the number of crosses and unions in T_G for the cases r odd and even.

Let G be a threshold graph with cotree $T_G(a_1, a_2, \ldots, a_r)$ with $a_r \geqslant 2$ and r odd. When we apply the algorithm $\mathrm{Diag}(T_G, 0)$ we are able to compute the inertia $(0_+, 0_0, 0_-)$ of G. First, note that $r = |\otimes| + |\cup| = 2|\otimes| - 1$ since $|\cup| = |\otimes| - 1$, where $|\otimes|$ and $|\cup|$ denotes, respectively, the number of crosses and unions in G w.r.t. to the cotree representation. Hence, by Theorem 2 we obtain the following

$$0_- = \left(\sum_{i=1}^{|\otimes|-1} a_{2i-1} + 1 - 1\right) + a_r - 1 = \left(\sum_{i=1}^{|\otimes|} a_{2i-1}\right) - 1.$$

And, by Theorem 3

$$0_0 = \sum_{i=1}^{|\cup|} (a_{2i} - 1) = \sum_{i=1}^{|\cup|} (a_{2i}) - |\cup|.$$

Finally

$$0_+ = n - 0_- - 0_0 = n - \left(\left(\sum_{i=1}^{|\otimes|} a_{2i-1}\right) - 1\right) - \left(\sum_{i=1}^{|\cup|} (a_{2i}) - |\cup|\right)$$

$$0_+ = n - \underbrace{\left(\left(\sum_{i=1}^{|\otimes|} a_{2i-1}\right) + \sum_{i=1}^{|\cup|} (a_{2i})\right)}_{n} + |\cup| + 1$$

$$0_+ = |\cup| + 1.$$

If r is even, then $r = |\otimes| + |\cup|$ with $|\otimes| = |\cup|$. Therefore, as before, by Theorem 2

$$0_- = \sum_{i=1}^{|\otimes|}(a_{2i-1} + 1 - 1) = \sum_{i=1}^{|\otimes|}(a_{2i-1})$$

and by Theorem 3

$$0_0 = \sum_{i=1}^{|\cup|}(a_{2i} - 1) = \sum_{i=1}^{|\cup|}(a_{2i}) + |\cup|.$$

So,

$$0_+ = n - 0_- - 0_0 = n - \left(\sum_{i=1}^{|\otimes|} (a_{2i-1})\right) - \left(\sum_{i=1}^{|\cup|} (a_{2i}) + |\cup|\right) = |\cup|.$$

Therefore, as aforementioned, our approach is to construct a threshold graph such that $N_+ = |\cup| + 1$ if r is odd and $N_+ = |\cup|$ if r is even.

Now, we present our strategy for generating a threshold graph G that is $[M, -1)$-eigenvalue free for a real number $M < -1$. As before, our method consists in constructing a threshold graph G with cotree $T_G(a_1, a_2, \ldots, a_r)$, by choosing each $a_i \geqslant 1$ for $1 \leqslant i \leqslant r-1$ and $a_r \geqslant 2$, and showing that this threshold graph is $[M, -1)$-eigenvalue free using the fact that $M_+ = (-1)_0 + 0_0 + 0_+$. Next we compute $(-1)_0 + 0_0 + 0_+$ for the cases r odd and even.

Let G be a threshold graph with cotree representation $T_G(a_1, a_2, \ldots, a_r)$ with $a_r \geqslant 2$ and r odd. Then $r = |\otimes| + |\cup|$ where $|\otimes| = |\cup| + 1$. Using Theorem 4 we obtain that $(-1)_0 = \sum_{i=1}^{|\otimes|}(a_{2i-1} - 1)$, so

$$\begin{aligned}
(-1)_0 + 0_0 + 0_+ &= \sum_{i=1}^{|\otimes|}(a_{2i-1} - 1) + 0_0 + (n - 0_- - 0_0) \\
&= \sum_{i=1}^{|\otimes|}(a_{2i-1} - 1) + n - \left(\sum_{i=1}^{|\otimes|}(a_{2i-1}) - 1\right) \\
&= n - |\otimes| + 1 \\
&= (n - r) + (|\cup| + 1),
\end{aligned}$$

since $r = |\cup| + |\otimes|$.

And, if r is even then $0_- = \sum_{i=1}^{|\otimes|}(a_{2i-1})$. Hence,

$$(-1)_0 + 0_0 + 0_+ = n - |\otimes| = (n - r) + |\cup|.$$

In this case, our approach is to construct an threshold graph such that $M_+ = (n - r) + |\cup| + 1$ if r is odd and $M_+ = (n - r) + |\cup|$ if r is even.

5 Diag (T_G,-N)

Instead of directly applying the algorithm $\text{Diag}(T_G(a_1, a_2, \ldots, a_r), -N)$ for $N > 0$, we initialize the process using Lemmas 5 and 6 in the leaves of the cotree with identical value as described in Example 7.

First, Lemma 5 is applied at each $\otimes$-node at depth i of the cotree. We perform $a_i - 1$ iterations that leave negative permanent assignments $\frac{j+1}{j}(-N-1)$ for $j = 1 \ldots a_i - 1$ by (1) and by (2) a remaining value

$$d_l \leftarrow \frac{-N + a_i - 1}{a_i} = 1 - \frac{N+1}{a_i}$$

at the last iteration, as represented in Figure 10.

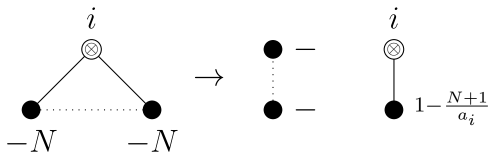

Figure 10: Join.

And, following Lemma 6, for each $\cup$-node at depth i of the cotree we perform $a_i - 1$ iterations that leave negative permanent assignments $\frac{j+1}{j}(-N)$ for $j = 1 \ldots a_i - 1$ by (3) and by (4) a remaining value

$$d_l \leftarrow \frac{-N}{a_i}$$

at the last iteration, as depicted in Figure 11.

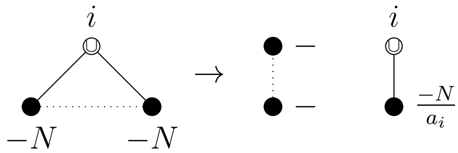

Figure 11: Union.

We would like to emphasize that, as described in Example 7, the vertices associated to permanent values are removed from the cotree. Therefore, after the specialization aforementioned in the terminal children of the cotree with identical value, we have $\sum_{i=1}^{r}(a_i - 1) = n - r$ negative permanent assignments. And, after that we will continue processing the specialized cotree in Figure 12 using Algorithm Diagonalize from left to right. Our strategy is to show that each $\cup$-node assigns a positive permanent value and each $\otimes$-node a negative permanent value with a positive remaining value. Consequently, it will imply that $N_+ = |\cup| + 1$ if r is odd and $N_+ = |\cup|$ if r is even.

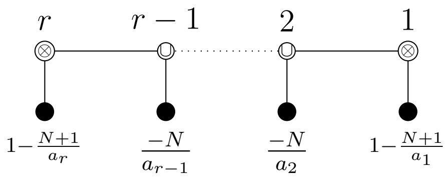

Figure 12: Specialized cotree.

It is important to point out that throughout this paper we execute Diagonalize in two main steps. In the first one we process the leaves with identical value and obtain what we have called specialized cotree as illustrated in Figure 12. Then, we proceed with the algorithm on the specialized cotree as in Example 7.

Consider the sequence of remaining values $\left\{1 - \frac{N+1}{a_r}, \frac{-N}{a_{r-1}}, \ldots, \frac{-N}{a_2}, 1 - \frac{N+1}{a_1}\right\}$ at the specialized cotree in Figure 12 and the functions

$$f(X,Y) = \frac{X\,Y - 1}{X + Y - 2} \tag{5}$$

$$g(X,Y) = \frac{X\,Y}{X + Y}. \tag{6}$$

If we choose the initial value $s_{r+1} = 1 - \frac{N+1}{a_r}$ and for each new value we pick g for a union and f for a join, then we obtain recursively $s_r = g(s_{r+1}, \frac{-N}{a_{r-1}})$, $s_{r-1} = f(s_r, 1 - \frac{N+1}{a_{r-2}})$ and so on, until we reach s_1. These are the assignments we obtain by applying the **subcase 1a** and **subcase 2a** from $\mathrm{Diag}(T_G, -N)$, which is performed from left to right. That is why we produce s_i in the inverse order, from s_{r+1} to s_1.

In the next two lemmas we show that by choosing the number of terminal children we can control the sign of the permanent and remaining assignments.

Lemma 8. *Suppose we have a $\cup$-node with leaves having values $s_{i+1} > 0$ and $\frac{-N}{a_i}$ with $a_i > \frac{N}{s_{i+1}}$ as illustrated in Figure 13. Then we obtain a positive permanent value p_i and a negative remaining value s_i.*

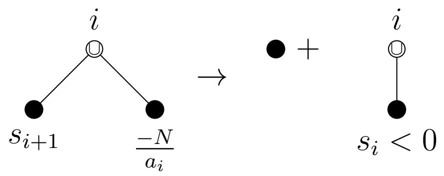

Figure 13: Union step.

Proof. Since $\alpha = s_{i+1} > \frac{N}{a_i}$ and $\beta = -\frac{N}{a_i}$, we can execute **subcase 2a** because $\alpha + \beta > 0$. It returns a positive permanent assignment

$$p_i = d_k \leftarrow s_{i+1} - \frac{N}{a_i} > 0,$$

and a negative remaining value

$$s_i = d_l \leftarrow g\left(s_{i+1}, -\frac{N}{a_i}\right) = \frac{s_{i+1}\left(\frac{-N}{a_i}\right)}{p_i} < 0.$$

$\square$

Lemma 9. *Suppose we have a $\otimes$-node with leaves having values $s_{i+1} < 0$ and $1 - \frac{(N+1)}{a_i}$ with $a_i > \frac{N+1}{1-\left(\frac{1}{s_{i+1}}\right)}$ as illustraded in Figure 14. Then we obtain a negative permanent value p_i and a positive remaining value s_i.*

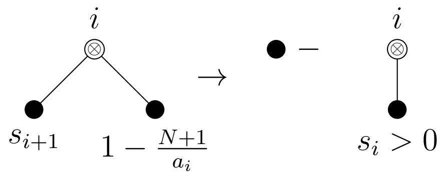

Figure 14: Join step.

Proof. Since $\alpha = s_{i+1} < 0$ and $\beta = 1 - \frac{N+1}{a_i}$ then we can execute **subcase 1a** because $\alpha + \beta - 2 = s_{i+1} + 1 - \frac{N+1}{a_i} - 2 = s_{i+1} - 1 - \frac{N+1}{a_i} < 0$. It returns a negative permanent assignment

$$p_i = d_k \leftarrow s_{i+1} - 1 - \frac{N+1}{a_i} < 0,$$

and a positive remaining value

$$s_i = d_l \leftarrow f\left(s_{i+1}, 1 - \frac{N+1}{a_i}\right) = \frac{s_{i+1}\left(1 - \frac{N+1}{a_i}\right) - 1}{p_i} > 0$$

if and only if

$$a_i > \frac{N+1}{1 - \left(\frac{1}{s_{i+1}}\right)},$$

as our hypothesis holds. □

Next, we will show that the two lemmas above used together produce an algorithm to construct threshold graphs that are $(0, N]$-eigenvalue free.

Initial step: Considering r odd, after the specialization, the leaf at the $\otimes$-node of depth r has the assignment

$$s_r = 1 - \frac{N+1}{a_r} > 0$$

if and only if $a_r > N + 1$. Then, using $a_r > N + 1$ and Lemmas 8 and 9 we can construct a threshold graph of depth r that is $(0, N]$-eigenvalue free.

And for r even, after the specialization at the node of depth r we have a leaf with assignment

$$s_r = \frac{-N}{a_r} < 0$$

for $a_r \geqslant 2$. Then, using $a_r \geqslant 2$ and lemmas 8 and 9 we can construct a threshold graph of depth r that is $(0, N]$-eigenvalue free.

The next two results sum up the above observations. We would like to point out that in the next theorem we begin with a generic cotree $T_G(a_1, a_2, \ldots, a_r)$ and, at each step, we choose specific values for a_i, $1 \leqslant i \leqslant r$.

Theorem 10. *Let $N > 0$ be a fixed number and r an odd number, if we choose natural numbers $a_r > N + 1$, $a_{r-1} > \frac{N}{s_r}$, $a_{r-2} > \frac{N+1}{1-\left(\frac{1}{s_{r-1}}\right)}$, etc., then $T_G(a_1, a_2, \ldots, a_r)$ is $(0, N]$-eigenvalue free.*

Proof. Considering the case r odd we have to prove that $N_+ = 0_+ = |\cup| + 1$. We start the $\mathrm{Diag}(T_G, -N)$ using Lemmas 5 and 6 to process the terminal children with identical value. As afore-explained this initial process produces $(n-r)$ negative permanent diagonal assignments, adding $(n-r)$ to N_-. Then we proceed the Algorithm Diagonalize on the specialized cotree in Figure 12. The last $\otimes$-node at depth r has a leaf with assignment $s_r = 1 - \frac{N+1}{a_r} > 0$ which will be relocated to the next level at the $\cup$-node as shown in Figure 15.

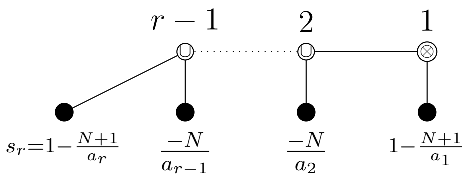

Figure 15: Specialized cotree relocated.

Using Lemma 8 at the $\cup$-node at depth $r-1$ we obtain a positive permanent diagonal value $p_{r-1} > 0$ and a remaining assignment $s_{r-1} < 0$. The leaf having the value s_{r-1} is then relocated to the next $\otimes$-node at level $r-2$ as depicted in Figure 16. At the $\otimes$-node at depth $r-2$ we apply Lemma 9 and it creates a negative permanent diagonal value $p_{r-2} < 0$ and a remaining value $s_{r-2} > 0$ which will be relocated to the next level $r-3$.

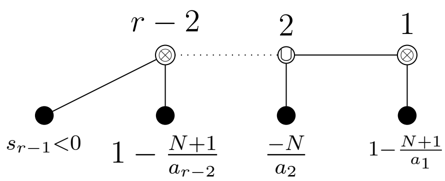

Figure 16: Specialized cotree relocated.

Continuing this process, the $\otimes$-node at level 1 will have two leafs with assignments s_2 and $1 - \frac{N+1}{a_1}$ as illustrated in Figure 17 left. Once we apply Lemma 9 we process the two remaining vertices whose diagonal values will be $p_1 < 0$ and $s_1 > 0$.

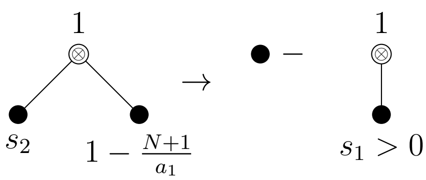

Figure 17: Last iteration.

The algorithm stops, and then each $\cup$-node has produced a positive permanent diagonal assignment, adding $|\cup|$ to N_+. And each $\otimes$-node, except the one at depth r, has produced a negative permanent diagonal value, adding $|\otimes|-1$ to N_-. And the final permanent diagonal value is $s_1>0$, adding $+1$ to N_+. Therefore $N_-=(n-r)+|\otimes|-1$, $N_0=0$ and $N_+=|\cup|+1$. $\square$

Theorem 11. *Let $N>0$ be a fixed number and r an even number, if we choose natural numbers $a_r \geqslant 2$, $a_{r-1} > \frac{N+1}{1-(\frac{1}{s_r})}$, $a_{r-2} > \frac{N}{s_{r-1}}$, etc., then $T_G(a_1, a_2, \ldots, a_r)$ is $(0,N]$-eigenvalue free.*

Proof. Considering the case r even we have to prove that $N_+ = 0_+ = |\cup|$. We start the $\mathrm{Diag}(T_G, -N)$ using Lemmas 5 and 6 to process the terminal children with identical value. As afore-explained this initial process produces $(n-r)$ negative permanent diagonal assignments, adding $(n-r)$ to N_-. Then we proceed the Algorithm Diagonalize on the specialized cotree in Figure 18 left. The last $\cup$-node at depth r has a leaf with assignment $s_r=-\frac{N}{a_r}>0$ which will be relocated to the next level at the $\otimes$-node as shown in Figure 18 right.

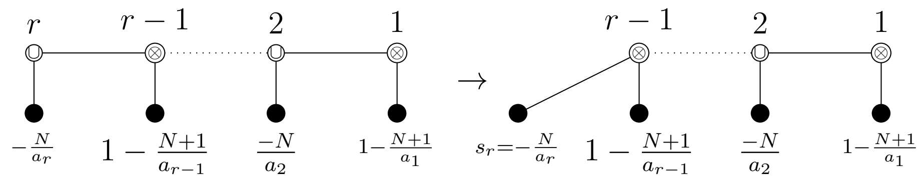

Figure 18: Specialized cotree relocated.

Using Lemma 9 at the $\otimes$-node at depth $r-1$ we obtain a negative permanent diagonal value $p_{r-1}>0$ and a remaining assignment $s_{r-1}>0$. The leaf having the value s_{r-1} is then relocated to the next $\cup$-node at level $r-2$ as depicted in Figure 19. At the $\cup$-node at depth $r-2$ we apply Lemma 8 and it creates a positive permanent diagonal value $p_{r-2}<0$ and a remaining value $s_{r-2}<0$ which will be relocated to the next level $r-3$.

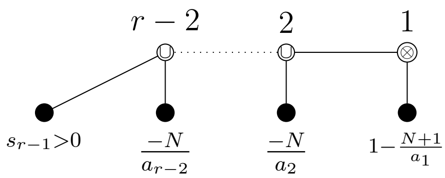

Figure 19: Specialized cotree relocated.

Continuing this process, the $\otimes$-node at level 1 will have two leafs with assignments s_2 and $1-\frac{N+1}{a_1}$ as illustrated in Figure 17 left. Once we apply Lemma 9 we process the two remaining vertices whose diagonal values will be $p_1<0$ and $s_1>0$.

The algorithm stops, and then each $\cup$-node, except the one at level r, has produced a positive permanent diagonal assignment, adding $|\cup|-1$ to N_+. And each $\otimes$-node has produced a negative permanent diagonal value, adding $|\otimes|$ to N_-. And the final permanent diagonal value is $s_1 > 0$, adding $+1$ to N_+. Therefore $N_- = (n-r) + |\otimes|$, $N_0 = 0$ and $N_+ = |\cup|$. $\square$

Remark 12. The values $a_1, a_2, \ldots, a_r$ are natural numbers so $a_{r-1} > \frac{N}{s_r}$ for r odd (case r even is similar), it means that $a_{r-1} \geqslant \left\lceil \frac{N}{s_r} \right\rceil$ if $\frac{N}{s_r}$ is not an integer number, otherwise $a_{r-1} \geqslant \left\lceil \frac{N}{s_r} \right\rceil + 1$. To avoid this confusion we can always choose $a_r \geqslant 1 + \lfloor N+1 \rfloor$, $a_{r-1} \geqslant 1 + \left\lfloor \frac{N}{s_r} \right\rfloor$, $a_{r-2} \geqslant 1 + \left\lfloor \frac{N+1}{1-\left(\frac{1}{s_{r-1}}\right)} \right\rfloor$, etc., to construct $T_G(a_1, a_2, \ldots, a_r)$.

Now we can define an algorithm to produce threshold graphs $T_G(a_1, a_2, \ldots, a_r)$ satisfying the Theorem 10 or 11:

Algorithm 2 The Right Free Interval Algorithm: RFI(N, r)

Input: a positive real number N and a positive integer r
Output: cotree $T_G(a_1, a_2, \ldots, a_r)$
  if r is odd **then**
    Choose $a_r \geqslant 1 + \lfloor N+1 \rfloor$
    $s_r \leftarrow 1 - \frac{1+N}{a_r}$
  else if r is even **then**
    Choose $a_r \geqslant 2$
    $s_r \leftarrow -\frac{N}{a_r}$
  end if
  for $i = r-1$ to 1 **do**
    if i is odd **then**
      Choose $a_i \geqslant 1 + \left\lfloor \frac{N+1}{1-\left(\frac{1}{s_{i+1}}\right)} \right\rfloor$
      $p_i \leftarrow s_{i+1} - 1 - \left(\frac{N+1}{a_i}\right)$
      $s_i \leftarrow f\left(s_{i+1}, 1 - \left(\frac{N+1}{a_i}\right)\right)$
    else if i is even **then**
      Choose $a_i \geqslant 1 + \left\lfloor \frac{N}{s_{i+1}} \right\rfloor$
      $p_i \leftarrow s_{i+1} - \frac{N}{a_i}$
      $s_i \leftarrow g\left(s_{i+1}, -\frac{N}{a_i}\right)$
    end if
  end for

Definition 13. Let $N > 0$ be a fixed number and r an odd number (case r even is similar), if we choose natural numbers $a_r = 1 + \lfloor N+1 \rfloor$, $a_{r-1} = 1 + \left\lfloor \frac{N}{s_{r+1}} \right\rfloor$, $a_{r-2} = 1 + \left\lfloor \frac{N+1}{1-\left(\frac{1}{s_r}\right)} \right\rfloor$, etc., then $T_G(a_1, a_2, \ldots, a_r)$ is called the initial threshold w.r.t. N, having no eigenvalues in the interval $(0, N]$.

Notice that the initial threshold is obtained by making the smallest choice for each a_i at RFI(N, r).

Next, we denote the first positive eigenvalue of a graph G by a $+$ sign in the exponent, as $\theta^+(G)$.

Example 14. Given $N = 4.8$ and $r = 5$ then RFI(4.8, 5) generates the initial threshold graph with cotree $T_G(5, 145, 5, 145, 6)$ which is $(0, 4.8]$-eigenvalue free. Indeed, a direct computation shows that $\theta^+(G) = 4.80000053517$.

Example 15. Given $N = 4.8$ and $r = 6$ then RFI(4.8, 6) generates the initial threshold graph with cotree $T_G(6, 44, 6, 36, 5, 2)$ which is $(0, 4.8]$-eigenvalue free. Indeed, a direct computation shows that $\theta^+(G) = 4.80016011291$.

6 Diag(T_G,-M)

We start the algorithm Diag$(T_G, -M)$, for a fixed number $M < -1$, processing the leaves with identical value using Lemmas 5 and 6 as follows.

Lemma 9 is applied at each $\otimes$-node at depth i. We perform $a_i - 1$ iterations that leave positive permanent diagonal values $\frac{j+1}{j}(-M-1)$ for $j = 1 \ldots a_i - 1$ by (1). And, by (2), at the last iteration, a remaining assignment

$$d_l \leftarrow \frac{-M + a_i - 1}{a_i} = 1 - \frac{M+1}{a_i}$$

as represented in Figure 20.

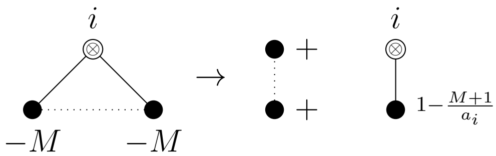

Figure 20: Join.

And, at each $\cup$-node at depth i we use Lemma 8 as follows. We perform $a_i - 1$ iterations that leave positive permanent diagonal values $\frac{j+1}{j}(-M)$ for $j = 1 \ldots a_i - 1$ by (3). And a remaining value

$$d_l \leftarrow \frac{-M}{a_i}$$

at the last iteration by (4) as represented in Figure 21.

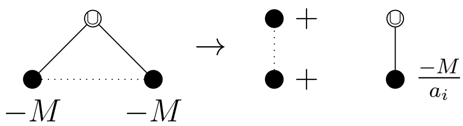

Figure 21: Union.

Therefore, after the specialization in the leaves with identical value we already have $\sum_{i=1}^{r}(a_i - 1) = n - r$ positive permanent diagonal values. Hence, following our strategy we will work with the cotree represented in Figure 22 and we will show that each $\cup$-node gives a positive permanent diagonal value and each $\otimes$-node returns a negative permanent diagonal value with a positive remaining assignment. It will imply that $M_+ = (n-r) + (|\cup| + 1)$ if r is odd and $M_+ = (n-r) + |\cup|$ if r is even.

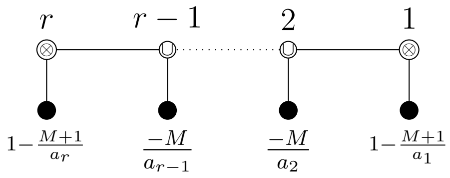

Figure 22: Specialized cotree.

Next, we prove that each $\otimes$-node leaves a permanent negative value and a remaining value positive. Whereas, each $\cup$-node leaves a permanent positive value. And, these results will prove that we obtain $|\cup|$ positive permanent values. In the last iteration, at the final $\otimes$-node, we will have a positive remaining diagonal assignment. It will imply that $M_+ = (n-r) + (|\cup| + 1)$ if r is odd and $M_+ = (n-r) + |\cup|$ if r is even.

Consider the sequence of remaining assignments $\left\{1 - \frac{M+1}{a_r}, \frac{-M}{a_{r-1}}, \ldots, \frac{-M}{a_2}, 1 - \frac{M+1}{a_1}\right\}$ at the specialized cotree in Figure 22. If we choose the initial value $s_{r+1} = 1 - \frac{M+1}{a_r}$ and for each iteration we choose the function (6) for a union and (5) for a join, then we obtain recursively $s_r = g\left(s_{r+1}, \frac{-M}{a_{r-1}}\right)$, $s_{r-1} = f\left(s_r, 1 - \frac{N+1}{a_{r-2}}\right)$ and so on, until we reach s_1. These are the assignments we obtain by applying the **subcase 1a** or **subcase 2a** from $\mathrm{Diag}(T_G, -M)$, which is performed from left to right. That is why we produce s_i in the inverse order, from s_{r+1} to s_1.

In the next two lemmas we show that by choosing the number of terminal children we can control the sign of the permanent and remaining assignments.

Lemma 16. *Suppose we have a $|\cup|$-node with leaves having values $s_{i+1} > 1$ and $\frac{-M}{a_i}$ with $a_i > -M + \frac{M}{s_{i+1}}$ as illustrated in Figure 23. Then we obtain a positive permanent value p_i and a remaining value $0 < s_i < 1$.*

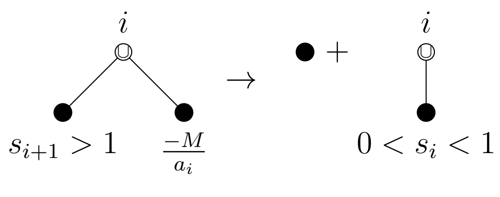

Figure 23: Union step.

Proof. Since $\alpha = s_{i+1} > 1$ and $\beta = -\frac{M}{a_i}$ then $\alpha + \beta > 1$ and we can use **subcase 2a**. It returns a permanent diagonal assignment

$$p_i = d_k \leftarrow s_{i+1} - \frac{M}{a_i} > 1$$

and a remaining one

$$s_i = d_l \leftarrow g\left(s_{i+1}, -\frac{M}{a_i}\right) = \frac{(s_{i+1})\left(\frac{-M}{a_i}\right)}{p_i} > 0.$$

Now, we want to show that $s_i < 1$, so

$$s_i = \frac{(s_{i+1})\left(\frac{-M}{a_i}\right)}{s_{i+1} - \frac{M}{a_i}} < 1$$

if and only if

$$a_i > -M + \frac{M}{s_{i+1}},$$

as our hypothesis holds. $\square$

Lemma 17. *Suppose we have a $\otimes$-node with leaves having assignments $0 < s_{i+1} < 1$ and $1 - \frac{(M+1)}{a_i}$ with $a_i > -\frac{M+1}{1-s_{i+1}}$ as illustrated in Figure 24. Then we obtain a negative permanent value p_i and a positive remaining value $s_i > 1$.*

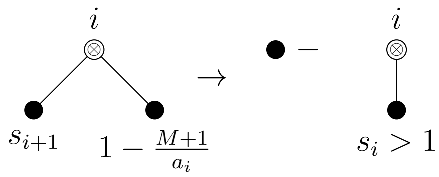

Figure 24: Join step.

Proof. Since $\alpha = s_{i+1}$ and $\beta = 1 - \frac{M+1}{a_i}$ then $\alpha + \beta - 2 = s_{i+1} + 1 - \frac{M+1}{a_i} - 2 = s_{i+1} - 1 - \frac{M+1}{a_i} < 0$ if and only if $a_i > -\frac{M+1}{1-s_{i+1}}$. Therefore, we can use **subcase 1a** and the permanent diagonal assignment will be

$$p_i = d_k \leftarrow s_{i+1} - 1 - \frac{M+1}{a_i} < 0.$$

And the remaining one

$$s_i = d_l \leftarrow f\left(s_{i+1}, 1 - \frac{M+1}{a_i}\right) = \frac{(s_{i+1})(1 - \frac{M+1}{a_i}) - 1}{s_{i+1} - 1 - \frac{M+1}{a_i}} > 1$$

if and only if

$$s_{i+1}(-(M+1)) < -(M+1),$$

which is satisfied if and only if $s_{i+1} < 1$. $\square$

As in Section 5, the two lemmas above used together produce an algorithm to construct threshold graphs that are $[M, -1)$-eigenvalue free.

Initial step: Considering r odd, after the specialization of the leaves with identical value, the $\otimes$-node at depth r has a leaf with assignment

$$s_r = 1 - \frac{M+1}{a_r} > 1$$

if and only if $\frac{-(M+1)}{a_r} > 0$, which is satisfied for all $a_r \geqslant 2$. Then, using $a_r \geqslant 2$ and Lemmas 16 and 17 we can construct a threshold graph of depth r that is $[M, -1)$-eigenvalue free. And, for r even, after the specialization, the $\cup$-node at depth r has a leaf with assignment

$$s_r = \frac{-M}{a_r}.$$

$s_r > 0$ is trivially satisfied and $s_r < 1$ iff $a_r > -M$. Then, using $a_r > -M$ and Lemmas 16 and 17 we can construct a threshold graph of depth r that is $[M, -1)$-eigenvalue free.

Remark 18. In the left case we have a_r free if r is odd and in the right case we have a_r free if r is even.

The next two theorems sum up the above results and their proofs are similar to the proofs of Theorems 10 and 11.

Theorem 19. *Let $M < -1$ be a fixed number and r an odd number, if we choose natural numbers $a_r \geqslant 2$, $a_{r-1} > -M\left(1 - \frac{1}{s_r}\right)$, $a_{r-2} > -\frac{M+1}{1-s_{r-1}}$, etc., then $T_G(a_1, a_2, \ldots, a_r)$ is $[M, -1)$-eigenvalue free.*

Theorem 20. *Let $M < -1$ be a fixed number and r an even number, if we choose natural numbers $a_r > -M$, $a_{r-1} > -\frac{M+1}{1-s_r}$, $a_{r-2} > -M\left(1 - \frac{1}{s_{r-1}}\right)$, etc., then $T_G(a_1, a_2, \ldots, a_r)$ is $[M, -1)$-eigenvalue free.*

Now we can define an algorithm to produce threshold graphs $T_G(a_1, a_2, \ldots, a_r)$ satisfying the Theorem 19 or 20:

Algorithm 3 The Left Free Interval Algorithm: LFI(N, r)

Input: a negative real number $M < -1$ and a positive integer r
Output: cotree $T_G(a_1, a_2, \ldots, a_r)$
  if r is odd **then**
    Choose $a_r \geqslant 2$
    $s_r \leftarrow 1 - \frac{1+M}{a_r}$
  else if r is even **then**
    Choose $a_r \geqslant 1 + \lfloor -M \rfloor$
    $s_r \leftarrow -\frac{M}{a_r}$
  end if
  for $i = r - 1$ to 1 **do**
    if i is odd **then**
      Choose $a_i \geqslant 1 + \left\lfloor \frac{-M-1}{1-s_{i+1}} \right\rfloor$
      $p_i \leftarrow s_{i+1} - 1 - \left(\frac{M+1}{a_i}\right)$
      $s_i \leftarrow f\left(s_{i+1}, 1 - \left(\frac{M+1}{a_i}\right)\right)$
    else if i is even **then**
      Choose $a_i \geqslant 1 + \left\lfloor -M + \frac{M}{s_{i+1}} \right\rfloor$
      $p_i \leftarrow s_{i+1} - \frac{M}{a_i}$
      $s_i \leftarrow g\left(s_{i+1}, -\frac{M}{a_i}\right)$
    end if
  end for

As before, the initial threshold is obtained by making the smallest choice for each a_i in LFI(M, r).

Next, we denote the first negative eigenvalue of a graph G smaller than -1 by a $-$ sign in the exponent, as $\theta^-(G)$.

Example 21. Given $M = -3.3$ and $r = 7$ then LFI($-3.3, 7$) generates the initial threshold graph with cotree $T_G(11, 4, 46, 3, 35, 2, 2)$ which is $[-3.3, -1)$-eigenvalue free. Indeed, a direct computation shows that the maximum negative eigenvalue is $\theta^-(G) = -3.30000464177 < M$.

Curiously, the left initial threshold graph and the right initial threshold graph coincides in LFI(M, 7)=RFI(N, 7) $= T_G(1, 1, 1, 1, 1, 1, 2)$ for $M = \frac{-1-\sqrt{2}}{2} < -1$ and $N = -1 - M = \frac{-1+\sqrt{2}}{2} > 0$. A direct computation shows that $\theta^-(G) = -1.24338010982 < M = -1.20710678118$ and $\theta^+(G) = 0.231890667597 > N = 0.20710678118$. Notice that $T_G(1, 1, 1, 1, 1, 1, 2)$ is the cotree associated to an anti-chain graph of order $n = 8$. However, as we exemplify below, this correspondence between N and $M = -N - 1$ does not hold for any threshold.

Using Example 21, where G is the threshold graph with associated cotree $T_G(11, 4, 46, 3, 35, 2, 2)$, we compute $\theta^+(G) = 0.558865493736 < N = -1 - M = 2.3$. Actually, the right initial threshold graph obtained from RFI$(-3.3,\ 7)$ is $T_G(3,14,3,16,3,14,4)$ which satisfies $\theta^+(G) = 2.30004052499 > 2.3$.

7 Infinite families

In this section, we show that the initial threshold graph generated by Algorithms 2 or 3 is an initial threshold graph for a family of threshold graphs that have the same I-eigenvalue free property.

We recall below the known interlacing property that can be found in [5].

Theorem 22. *Let G be a graph and $H = G - v$, where v is a vertex of G. If $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n$ and $\theta_1 \geqslant \theta_2 \geqslant \cdots \geqslant \theta_{n-1}$ are the eigenvalues of $A(G)$ and $A(H)$, respectively, then*

$$\lambda_i \geqslant \theta_i \geqslant \lambda_{i+1} \text{ for each } i = 1, 2, \ldots, n-1.$$

Given a threshold graph G with eigenvalues $\lambda_1 \geqslant \lambda_2 \cdots \geqslant \lambda_n$ and $H = G - v$ with eigenvalues $\theta_1 \geqslant \theta_2 \cdots \geqslant \theta_{n-1}$.

Consider the cotree $T_H(a_1, \ldots, a_r)$ of depth r associated to the threshold graph H with $a_i \geqslant 1$ for $1 \leqslant i \leqslant r-1$ and $a_r \geqslant 2$. Denote $m(0,H) = p$ and $m(-1,H) = q$ the respective multiplicities of 0 and -1 in H. We recall that

$$\theta^+(H) = \text{ first positive eigenvalue of } H \text{ greater than } 0,$$

and

$$\theta^-(H) = \text{ first negative eigenvalue of } H \text{ smaller than } -1.$$

Suppose that $M < \theta^-(H)$ and $\theta^+(H) > N$. Using the notation above we have the following:

$$\theta_{n-1} \leqslant \cdots \leqslant \theta_{i+p+q+2} \leqslant \theta_{i+p+q+1} < M < -1 = \underbrace{\theta_{i+p+q} = \cdots = \theta_{i+p+2} = \theta_{i+p+1}}_{q=m(-1,H)} < 0$$

$$0 = \underbrace{\theta_{i+p} = \cdots = \theta_{i+2} = \theta_{i+1}}_{p=m(0,H)} < N < \theta_i \leqslant \cdots \theta_2 \leqslant \theta_1.$$

Hence, $\theta_{i+p+q+1} = \theta^- < M$ and $\theta_i = \theta^+ > N$.

Notice that, when we add a vertex v to the threshold H then we have two possibilities:

1. we add a vertex v to a $\cup$-node at depth l that has a_l terminal vertices in H. Then G will have a $\cup$-node at depth l that has $a_l + 1$ terminal vertices;

2. we add a vertex v to a $\otimes$-node at depth l that has a_l terminal vertices in H. Then G will have a $\otimes$-node at depth l that has $a_l + 1$ terminal vertices.

Using Theorems 3 and 4 we have the following:

1. $m(0,G) = m(0,H) + 1$ and $m(-1,G) = m(-1,H)$;
2. $m(0,G) = m(0,H)$ and $m(-1,G) = m(-1,H) + 1$;

Now, we apply the interlacing Theorem 22.

$$\lambda_i \geqslant \underbrace{\theta_i}_{\theta^+} \geqslant \underbrace{\lambda_{i+1}}_{?} \geqslant \theta_{i+1} = \lambda_{i+2} = \cdots = \lambda_{i+p} = \theta_{i+p} = 0 \geqslant \underbrace{\lambda_{i+p+1}}_{-1 \text{ or } 0} \geqslant$$

$$\geqslant \theta_{i+p+1} = -1 = \lambda_{i+p+2} = \cdots = \lambda_{i+p+q} = \theta_{i+p+q} = -1 \geqslant \underbrace{\lambda_{i+p+q+1}}_{?} \geqslant \underbrace{\theta_{i+p+q+1}}_{\theta^-} \geqslant \lambda_{i+p+q+2}.$$

Considering the two possible cases we conclude the following.

1. $m(0,G) = p+1$ and $m(-1,G) = q$. It implies that $\lambda_{i+1} = \lambda_{i+p+1} = 0$ and $\lambda_{i+p+q+1} = -1$.
2. $m(0,G) = p$ and $m(-1,G) = q+1$. It implies that $\lambda_{i+1} = 0$ and $\lambda_{i+p+1} = \lambda_{i+p+q+1} = -1$.

And, it implies that

$$\lambda_i = \lambda^+(G) \geqslant \theta_i = \theta^+ > N \text{ and } \lambda_{i+p+q+2} = \lambda^-(G) \leqslant \theta_{i+p+q+1} = \theta^- < M.$$

The above computations motivate the following definition.

Definition 23. Let $T_G(a_1, \ldots, a_r)$ with $a_i \geqslant 1$ for $1 \leqslant i \leqslant r-1$ and $a_r \geqslant 2$, and $T_{G'}(b_1, b_2, \ldots, b_r)$ be two threshold graphs. We say that $G \preceq G^{'}$ if $a_i \leqslant b_i$ for $i = 1, 2, \ldots, r$ i.e., if $G^{'}$ is generated from G by adding any amount of leaves to $\cup$-nodes or $\otimes$-nodes (of course, $G \prec G^{'}$ if $G \preceq G^{'}$ and $G \neq G^{'}$). Notice that G and $G^{'}$ have the same depth.

Then, we have proved the next result.

Corollary 24. *For $I = (0, N]$ or $I = [M, -1)$ we have that, if $G \preceq G^{'}$ and $T_G(a_1, \ldots, a_r)$ is I-eigenvalue free, then $T_{G'}(b_1, \ldots, b_r)$ is I-eigenvalue free.*

In Example 15, given $N = 4.8$ and $r = 6$, RFI$(4.8, 6)$ generated the initial threshold graph G with associated cotree $T_G(6, 44, 6, 36, 5, 2)$ which is $(0, 4.8]$-eigenvalue free. By Corollary 24, any threshold graph represented by the cotree $T_{G'}(6+i_1, 44+i_2, 6+i_3, 36+i_4, 5+i_5, 2+i_6)$ for any integers $i_j \geqslant 0$, $1 \leqslant j \leqslant 6$, is $(0, 4.8]$-eigenvalue free.

8 Revisiting Ghorbani's work

In this section we use our approach to revisit the conjecture proposed by Aguilar et al. in [1] proved by Ghorbani in [4].

Theorem 25. *Given* $N = \frac{-1+\sqrt{2}}{2} = 0.20710678+$ *and* r *any odd natural number (the even case is identical) then the algorithm RFI*(N, r) *generates the initial threshold graph with cotree* $T_G(1, \ldots, 1, 2)$ *(with* $r - 1$ *leafs 1's) that is* $(0, N]$*-eigenvalue free.*

Proof. A direct computation shows that $a_r \geqslant 1 + \lfloor N + 1 \rfloor = 2$ so the initial choice is $a_r = 2$. From there our proof is by induction. Also by direct computation, we can compute $s_r = 3/4 - 1/4\sqrt{2} \simeq 0.3964 > 0$ and $a_{r-1} \geqslant 1 + \lfloor \frac{N}{s_r} \rfloor = 1 + \lfloor 0.5224 \rfloor = 1$, so the initial choice is $a_{r-1} = 1$ and compute $s_{r-1} = g\left(s_r, \frac{-N}{a_{r-1}}\right) = g(s_r, -N) = \frac{\left(-3+\sqrt{2}\right)\left(-1+\sqrt{2}\right)}{10-6\sqrt{2}} \simeq -0.4336 < 0$. Analogously, $a_{r-2} \geqslant 1 + \left\lfloor \frac{N+1}{1-\left(\frac{1}{s_{r-1}}\right)} \right\rfloor = 1 + \lfloor 0.3651 \rfloor = 1$, is the initial choice, and compute $s_{r-2} = f\left(s_{r-1}, 1 - \frac{N+1}{a_{r-2}}\right) = f(g(s_r, -N), -N)$. In what follows we obtain the initial choice equal to 1 and therefore $s_{2k} = f\left(g\left(s_{2k+2}, -N\right), -N\right)$, $s_{2k+1} = g\left(f\left(s_{2k+3}, -N\right), -N\right)$. Now we will consider $k + 1$. Both recurrences are explicit and easy to solve. For even indices we get $s_{2k} = f(g(s_{2k+2}, -N), -N) = \varphi(s_{2k+2})$ where

$$\varphi(t) = \frac{\left(2\sqrt{2}+1\right)t + 2 - 2\sqrt{2}}{4\left(1+\sqrt{2}\right)t - 2\sqrt{2} + 1}.$$

It is easy to see that $\varphi(t) = t$ has a unique solution $\mu = \frac{\sqrt{2}}{2(1+\sqrt{2})} \simeq 0.2928$. Since $\varphi'(t) < 1$ for $t > \mu = 0.2928$(see Figure 25, left) and $s_0 = 0.3964$ we get that if $\mu < s_{2k} < s_0$ then $s_{2k} = \varphi(s_{2k+2}) \in (\mu, s_r)$ meaning that $0 = \left\lfloor \frac{N}{\mu} \right\rfloor > \left\lfloor \frac{N}{s_{2k+2}} \right\rfloor > \left\lfloor \frac{N}{s_r} \right\rfloor = 0$, that is the initial choice for $2k + 2$ is 1.
For the odd indices we get $s_{2k+1} = g(f(s_{2k+3}, -N), -N) = \psi(s_{2k+3})$ where

$$\psi(t) = \frac{\left(1-\sqrt{2}\right)\left(2+\left(-1+\sqrt{2}\right)t\right)}{\left(4\sqrt{2}-4\right)t + 5 - 2\sqrt{2}}.$$

It is easy to see that $\psi(t) = t$ has a unique solution $\mu' = \frac{\sqrt{2}-2}{2\sqrt{2}-2} \sim -0.7071$. Since $\psi'(t) < 1$ for $t > \mu' = -0.7071$ (see Figure 25, right) and $s_{r-1} = -0.4336 < 0$ we get that, if $\mu' < s_{2k+3} < s_{r-1}$ then $s_{2k+1} = \psi(s_{2k+3}) \in (\mu', s_{r-1})$ meaning that $0 = \left\lfloor \frac{N+1}{1-\left(\frac{1}{\mu}\right)} \right\rfloor > \left\lfloor \frac{N+1}{1-\left(\frac{1}{s_{2k+1}}\right)} \right\rfloor > \left\lfloor \frac{N+1}{1-\left(\frac{1}{s_{r-1}}\right)} \right\rfloor = 0$, which is the initial choice for $2k + 1$ is also 1.

In both cases we will always have $a_i = 1$ as the initial choice. Given the decreasing convergence towards to the fixed points μ and μ' we deduce that our claim is true for a arbitrary large r, concluding our proof. ∎

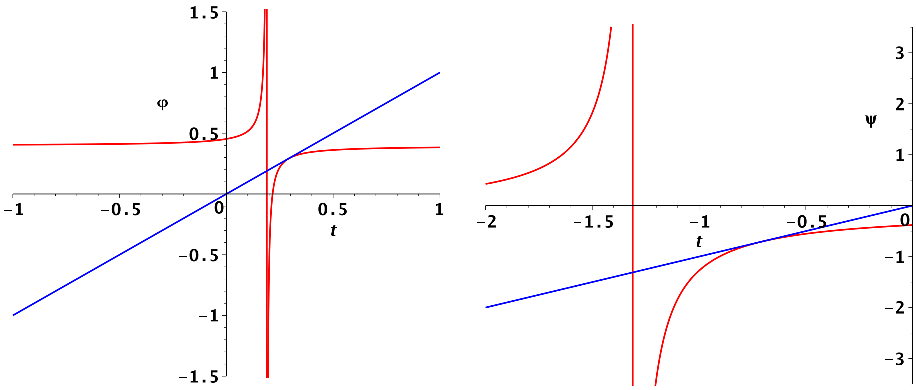

Figure 25: From the left, $\varphi(t)$ and $\psi(t)$

Corollary 26. *Given the threshold graph G with associated cotree $T_G(1,\ldots,1,2)$ of depth r then, any threshold graph $T_{G'}(a_1,\, a_2,\, \ldots,\, a_r)$ with $a_r \geqslant 2$, $a_i \geqslant 1$ for $1 \leqslant i \leqslant r-1$ is $(0, \frac{-1+\sqrt{2}}{2}]$-eigenvalue free.*

Proof. Our hypothesis implies that $G \preceq G'$ and, by Corollary 24 and Theorem 25, we conclude that there is no threshold graph with eigenvalues in the interval $(0, \frac{-1+\sqrt{2}}{2}]$. $\square$

The proof of the next two results are similar to the ones above.

Theorem 27. *Given $M = \frac{-1-\sqrt{2}}{2} = -1.207106781$ and r any odd natural number (the even case is identical) then the algorithm LFI(M,r) generates the initial threshold graph with cotree $T_G(1,\ldots,1,2)$ (with $r-1$ leafs 1's) that is $(M,-1]$-eigenvalue free.*

Corollary 28. *Given the threshold graph G with associated cotree $T_G(1,\ldots,1,2)$ of depth r then, any threshold graph $T_{G'}(a_1,\, a_2,\, \ldots,\, a_r)$ with $a_r \geqslant 2$, $a_i \geqslant 1$ for $1 \leqslant i \leqslant r-1$ is $(\frac{-1-\sqrt{2}}{2}, -1]$-eigenvalue free.*

9 Future work

As a sequel of this work, we would like to investigate additional properties of the initial threshold graphs generated by Algorithms RFI(N,r) and LFI(N,r), such as the existence of minimal ones, w.r.t. the I-eigenvalue free property.

By Corollary 24, the relation $G \preceq G'$ preserves the I-eigenvalue free property. In this way, given for instance $I = (0, N]$ and the initial threshold graph $T_G(a_1,\ldots,a_r)$, obtained from RFI(N,r), we already know that, if $G \preceq G'$ then G' is also $(0,N]$-eigenvalue free. However, if $G'' \preceq G$ the situation is not clear. By a simple counting procedure we can see that there are exactly $K = a_1 \cdot \ldots \cdot a_r$ of such graphs. For a fairly small number $N = \sqrt{5} = 2.26+$ the initial threshold graph, obtained from RFI$(\sqrt{5},7)$, is $T_G(4,10,4,55,3,12,4)$, producing $K = 4 \cdot 10 \cdot 4 \cdot 55 \cdot 3 \cdot 12 \cdot 4 = 1,267,200$ threshold graphs smaller than itself to compare. We can perform an exhaustive search for a $G'' \preceq G$ such that G'' is also $(0,N]$-eigenvalue free, by directly computing the spectrum of each graph.

This task could be extremely hard from a computational point of view but we can make it easy by using the Diagonalize algorithm (it took about 5 minutes for $N = \sqrt{5}$ using a naive implementation). Comparing the number of positive outputs for $\mathrm{Diag}\left(T_{G''}, \frac{-1+\sqrt{2}}{2}\right)$ (which has no zero outputs, because $\frac{-1+\sqrt{2}}{2}$ is not eigenvalue for any threshold graph) and $\mathrm{Diag}(T_{G''}, N)$ (which should not have zero outputs if $N < \theta^+(G'')$) we can see that G'' is also $(0, N]$-eigenvalue free if, and only if, this number of positive outputs remains unchanged. Remembering that, after the specialization, $n - r$ outputs will be negative, we only need compute r values in $\mathrm{Diag}\left(T_{G''}, \frac{-1+\sqrt{2}}{2}\right)$ and $\mathrm{Diag}(T_{G''}, N)$. We made this computation for several values $N > 0$ never founding any $G'' \prec G$ such that G'' is also $(0, N]$-eigenvalue free. Despite the efficiency of our method, it still prohibitive, for $N = 3.5$ we must to perform about 22 million tests. In other words, we conjecture that the initial graph is *minimal* in the sense that any *smaller* threshold graph is not $(0, N]$-eigenvalue free.

Conjecture 29. Let $N > 0$ and $r \geqslant 2$ be fixed numbers defining the interval $I = (0, N]$, and $T_G(a_1, \ldots, a_r)$, the initial threshold graph obtained from $\mathrm{RFI}(N, r)$, then G is minimal, i.e., if $\tilde{G} \preceq G$ is such that $\tilde{G}$ is also I-eigenvalue free then $\tilde{G} = G$.

The conjecture is obviously true for $N = \frac{-1+\sqrt{2}}{2}$ because the initial threshold graph, $T_G(2, 1, 1, \ldots, 1)$, is the small one. In a future work we expect to investigate this conjecture trying to prove the minimality of the initial graph for each fixed r. We believe that probably does not exist a global minimum, except if we fix r, because we already know that the sequence

$$\theta^+(T_G(1, 2)), \theta^+(T_G(1, 1, 2)), \theta^+(T_G(1, 1, 1, 2)), \ldots$$

converges to $N = \frac{-1+\sqrt{2}}{2}$.

Acknowledgments
We thank the anonymous referees for their careful reading of our manuscript and their many insightful comments and suggestions.

References

[1] Aguilar, C. O., Lee, J.-y., Piato, E., and Schweitzer, B. J. Spectral characterizations of anti-regular graphs. *Linear Algebra and its Applications 557* (2018), 84–104.

[2] Biyikoglu, T., Simic, S. K., and Stanic, Z. Some notes on spectra of cographs. *Ars Combin 100* (2011), 421–434.

[3] Chvátal, V., and Hammer, P. Aggregation of inequalities in integer programming. *Ann. Discrete Math 1* (1977), 145–162.

[4] Ghorbani, E. Eigenvalue-free interval for threshold graphs. *Linear Algebra and its Applications 583* (2019), 300–305.

[5] Hall, F., Patel, K., and Stewart, M. Interlacing results on matrices associated with graphs. *J. Combin. Math. Combin. Comput 68* (2009), 113–127.

[6] Henderson, P. B., and Zalcstein, Y. A graph-theoretic characterization of the pv_chunk class of synchronizing primitives. *SIAM Journal on Computing 6*, 1 (1977), 88–108.

[7] Hoffman, A. J. On limit points of spectral radii of non-negative symmetric integral matrices. In *Graph theory and applications*. Springer, 1972, pp. 165–172.

[8] Jacobs, D. P., Trevisan, V., and Tura, F. Eigenvalues and energy in threshold graphs. *Linear Algebra and its Applications 465* (2015), 412–425.

[9] Jacobs, D. P., Trevisan, V., and Tura, F. C. Eigenvalue location in cographs. *Discrete Applied Mathematics 245* (2018), 220–235.

[10] Mahadev, N. V., and Peled, U. N. *Threshold graphs and related topics*. Elsevier, 1995.

[11] Sciriha, I., and Farrugia, S. On the spectrum of threshold graphs. *International Scholarly Research Notices 2011* (2011).

[12] Shearer, J. B. On the distribution of the maximum eigenvalue of graphs. *Linear Algebra and its Applications 114* (1989), 17–20.

[13] Zhang, F., and Chen, Z. Limit points of eigenvalues of (di) graphs. *Czechoslovak Mathematical Journal 56*, 3 (2006), 895–902.