

Um Algoritmo para Gerar Grafos Threshold Simplesmente Estruturados

Heber C. Teixeira¹, Leonardo S. de Lima²
 UFPR, Curitiba, PR

Resumo. Seja G um grafo threshold conexo com n vértices. Neste trabalho, caracterizamos todos os grafos G que tem uma autobase simplesmente estruturada para a matriz laplaciana, isto é, uma base de \mathbb{R}^n composta de autovetores com entradas somente em $\{-1, 0, 1\}$. Nosso método consiste em provar que os autoespaços de um grafo threshold devem possuir um número mínimo de vetores, além de desenvolver um algoritmo para gerar todos os grafos threshold que satisfazem essa propriedade.

Palavras-chave. Grafo Threshold, Matriz Laplaciana, Autovetores, Autobase Simplesmente Estruturada

1 Introdução

Considere $G = (V, E)$ um grafo simples com conjunto de vértices $V = \{1, \dots, n\}$ e conjunto de arestas E tal que $|E| = m$. O grau de um vértice i de G é o número arestas incidentes ao vértice i e é denotado por d_i . Escrevemos $L(G) = D(G) - A(G)$ para a matriz laplaciana de G , onde $A(G)$ é a matriz de adjacência e $D(G)$ é a matriz diagonal dos graus dos vértices de G . Os autovalores de $L(G)$ são ordenados como $\mu_1 \geq \dots \geq \mu_{n-1} \geq \mu_n = 0$ e os autovetores correspondentes são $\mathbf{x}^1, \dots, \mathbf{x}^n$. Uma *autobase* de $L(G)$ é uma base de \mathbb{R}^n composta de autovetores de $L(G)$ e uma base é *simplesmente estruturada* se consiste de vetores com entradas somente em $\{-1, 0, 1\}$, [7]. Denotamos o autoespaço associado com um autovalor μ of $L(G)$ por

$$\mathcal{E}_L(\mu) = \{\mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq 0 : L(G)\mathbf{x} = \mu\mathbf{x}\}.$$

Dizemos que $\mathcal{E}_L(\mu)$ é *simplesmente estruturado* se admite uma base simplesmente estruturada e um grafo G é *simplesmente estruturado* se $L(G)$ possui uma autobase simplesmente estruturada, [1, 3, 6]. Nosso interesse aqui é determinar os grafos thresholds que são simplesmente estruturados. Dentre as várias definições e teoremas de caracterização de grafos thresholds (como pode ser visto em [5]), utilizamos a caracterização por uma sequência binária.

Definição 1.1. *Dada uma sequência $\{b_i\}$ de 0's e 1's com n elementos, o grafo threshold associado com a sequência binária $\{b_i\}$ é o grafo G com n vértices construído recursivamente, começando com um grafo vazio, e para $i = 1, \dots, n$*

- (i) *adicione o vértice isolado i se $b_i = 0$,*
- (ii) *adicione o vértice i adjacente a todos os vértices com rótulo menor que i se $b_i = 1$.*

Um vértice i é um vértice isolado se $b_i = 0$ e um vértice dominante se $b_i = 1$. Claramente, G é conexo se, e somente se, $b_n = 1$. Ainda, a sequência binária $b = (0, b_2, \dots, b_{n-1}, 1)$ de um grafo threshold conexo G com pode ser escrita de forma compacta como $b = 0^{s_1} 1^{t_1} \dots 0^{s_k} 1^{t_k}$, onde 0^{s_i} é

¹hebercristina@gmail.com

²leonardo.delima@ufpr.br

uma abreviação para $s_i \geq 1$ zeros consecutivos e 1^{t_i} é uma abreviação para $t_i \geq 1$ uns consecutivos. Como $n = \sum_{i=1}^k (s_i + t_i)$, isso assegura que $1 \leq k \leq \lfloor \frac{n}{2} \rfloor$. Podemos particionar o conjunto de vértices como $V(G) = U_1 \cup V_1 \cup \dots \cup U_k \cup V_k$, onde o conjunto U_i consiste do i -ésimo grupo de vértices isolados consecutivos na construção de G de modo que $|U_i| = s_i$, and similarmente, V_i consiste do i -ésimo grupo de vértices dominantes na construção de G , com $|V_i| = t_i$. Se $s_1 \geq 2$, então a partição de grau de G é dada por $\{U_1, V_1, \dots, U_k, V_k\}$. Caso contrário, se $s_1 = 1$ então a partição de grau torna-se $\{U_1 \cup V_1, U_2, V_2, \dots, U_k, V_k\}$. Em ambos os casos, cada subconjunto U_i é um conjunto independente e cada subconjunto V_i é uma clique [2]. A Figura 1 ilustra a partição de grau de um grafo threshold; uma linha entre U_i e V_j indica que todos os vértices em U_i são adjacentes a todos os vértices em V_j , e o retângulo indica que $V_1 \cup \dots \cup V_k$ é uma clique.

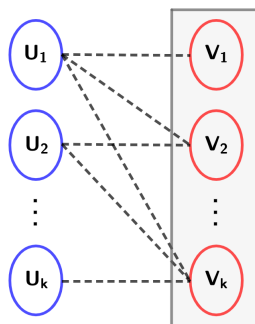


Figura 1: Estrutura global de um grafo threshold com sequência binária $b = 0^{s_1} 1^{t_1} \dots 0^{s_k} 1^{t_k}$. Cada vértice em U_i é adjacente a $V_i \cup \dots \cup V_k$, $V_1 \cup \dots \cup V_k$ é uma clique e $U_i \cup \dots \cup U_k$ é um conjunto independente. Fonte: o autor.

Seja S_n o grafo estrela com n vértices. Para $1 \leq k < n$, é fácil ver que cada \mathbf{x}^k satisfazendo

$$\mathbf{x}_i^k = \begin{cases} 1, & \text{se } i < k + 1, \\ -k, & \text{se } i = k + 1, \\ 0, & \text{se } k + 1 < i \leq n, \end{cases} \quad (1)$$

e $\mathbf{x}^n = (1, 1, \dots, 1)^T$ são autovetores de $L(S_n)$ associados com os autovalores 1, n e 0 de multiplicidades $n - 2$, 1 e 1, respectivamente. Em [4] (Teorema A), os autores provaram que $L(G)$ e $L(S_n)$ compartilham os mesmos autovetores quando G é um grafo threshold com n vértices.

Teorema 1.1. (Teorema A, [4]) *Seja G um grafo conexo com n vértices. Então, G é um grafo threshold se, e somente se, $\{\mathbf{x}^1, \dots, \mathbf{x}^n\}$ é uma autobase para a matriz laplaciana de G .*

2 Resultados Principais

Sob a hipótese de que G é threshold e simplesmente estruturado, provamos que os autoespaços de $L(G)$ devem ter uma dimensão mínima. Na Proposição 2.1 provamos que $\dim(\mathcal{E}_L(n)) \geq \lfloor \frac{n}{2} \rfloor$.

Proposição 2.1. *Seja G um grafo threshold conexo com n vértices definido pela sequência binária $b = (b_1, b_2, \dots, b_k, b_{k+1}, \dots, b_{n-1}, b_n)$ e seja $k \in \{1, \dots, n - 1\}$ tal que $b_{k+1} = \dots = b_n$ e $b_k \neq b_{k+1}$. Então, $\mathcal{E}_L(n)$ é simplesmente estruturado se, e somente se, $1 \leq k \leq \lfloor \frac{n}{2} \rfloor$.*

Da Proposição 2.1, concluímos que o número mínimo de vetores na base de $\mathcal{E}_L(n)$ é $\lceil \frac{n}{2} \rceil$. Assumindo que G é um grafo threshold conexo com sequência binária tal que $b_{k+1} = \dots = b_n = 1$ and $b_k = 0$, para algum $k \in \{2, \dots, n-1\}$, sempre obtemos que $n-k$ é um autovalor de $L(G)$. Na Proposição 2.2, provamos que $\dim(\mathcal{E}_L(n-k)) \geq \lceil \frac{k}{2} \rceil$.

Proposição 2.2. *Seja G um grafo threshold conexo com n vértices definido pela sequência binária $b = (b_1, b_2, b_3, \dots, b_k, b_{k+1}, \dots, b_n)$ tal que $b_k = 0$ e $b_{k+1} = \dots = b_n = 1$, para algum $k \in \{1, \dots, n-1\}$. Seja $l \in \{2, \dots, k-1\}$ tal que $b_{l+1} = \dots = b_k$ e $b_l \neq b_{l+1}$. Então, $\mathcal{E}_L(n-k)$ é simplesmente estruturado se, e somente se, $2 \leq l \leq \lfloor \frac{k}{2} \rfloor$.*

A Proposição 2.2 implica que o número mínimo de vetores na base de $\mathcal{E}_L(n-k)$ é $\lceil \frac{k}{2} \rceil$. Assumindo que G é um grafo threshold conexo com n vértices e sequência binária

$$b = (b_1, b_2, \dots, b_{l-1}, b_l, b_{l+1}, \dots, b_k, b_{k+1}, \dots, b_n)$$

tal que $b_l = 1$, $b_{l+1} = \dots = b_k = 0$ e $b_{k+1} = \dots = b_n = 1$, para algum $k \in \{1, \dots, \lfloor \frac{n}{2} \rfloor\}$ e $l \in \{2, \dots, \lfloor \frac{k}{2} \rfloor\}$, sempre temos que $\mathcal{E}_L(n)$ e $\mathcal{E}_L(n-k)$ são simplesmente estruturados, independentemente do que acontece nas posições b_1, \dots, b_{l-1} da sequência binária. Portanto, as Proposições 2.1 e 2.2 fornecem um importante ponto de partida para determinar os grafos threshold conexos que são simplesmente estruturados. A partir destes resultados temos uma caracterização dos grafos threshold que são simplesmente estruturados. Antes de enunciarmos o teorema precisamos definir conceitos preliminares.

Seja $P = [s_1, t_1, \dots, s_k, t_k]$ uma partição crescente do inteiro n com $2k$ partes, tal que cada parte s_i e t_i , para $i = 1, \dots, k$, foi obtida pela aplicação recursiva das ideias das Proposições 2.1 e 2.2. O maior valor possível que k pode assumir é $\lfloor \frac{\lfloor \log_2(n) \rfloor + 1}{2} \rfloor$, e esse valor é obtido uma vez que podemos escrever

$$n = \lceil \frac{n}{2} \rceil + \lceil \frac{n}{2^2} \rceil + \lceil \frac{n}{2^3} \rceil + \lceil \frac{n}{2^4} \rceil + \dots + \lceil \frac{n}{2^{2^m-1}} \rceil + \lfloor \frac{n}{2^{2^m-1}} \rfloor$$

onde $\lfloor \frac{n}{2^{2^m-1}} \rfloor$ é o menor valor inteiro possível maior ou igual a 1.

O próximo teorema caracteriza quando um grafo threshold conexo de uma dada ordem n é simplesmente estruturado.

Teorema 2.1. *Um grafo threshold G com n vértices é simplesmente estruturado se, e somente se, a sequência binária que define G está associada a uma partição $P = [s_1, t_1, \dots, s_k, t_k]$ tal que $k = \{1, \dots, \lfloor \frac{\lfloor \log_2(n) \rfloor + 1}{2} \rfloor\}$ e cada uma das partes s_i e t_i de P , para $i = 1, \dots, k$, satisfaz*

$$\left\lceil \frac{n - \sum_{l=i+1}^k (s_l + t_l)}{2} \right\rceil \leq t_i \leq n - \sum_{l=i+1}^k (s_l + t_l) - 2^{2i-2}, \tag{2}$$

$$\left\lceil \frac{n - t_i - \sum_{l=i+1}^k (s_l + t_l)}{2} \right\rceil \leq s_i \leq n - t_i - \sum_{l=i+1}^k (s_l + t_l) - 2^{2i-3}, \tag{3}$$

$$s_1 = n - t_1 - \sum_{l=2}^k (s_l + t_l) \tag{4}$$

3 Algoritmo

Os resultados da Seção 2 motivaram o desenvolvimento de um algoritmo, dividido em quatro partes, para determinar todos os grafos thresholds conexos com uma dada ordem n .

O Algoritmo 1 gera todas as partições possíveis de um inteiro n (a ordem do grafo threshold).

Algoritmo 1: Dados os inteiros positivos n e k , seja l uma lista de números inteiros positivos de comprimento j tal que $0 \leq j \leq 2k - 1$. Se $j = 0$, então l é a lista vazia. A soma dos elementos da lista l será denotada por sum . O Algoritmo 1 define as partes s_i ou t_i da partição $P = [s_1, t_1, \dots, s_k, t_k]$ do inteiro n , para algum $i = 1, \dots, k$, da seguinte forma:

- Se $j = 2q$ com $q = 0, \dots, k - 1$, então determinamos uma lista L que começa em $t_{min}^{k-q} = \lceil \frac{n-sum}{2} \rceil$ e termina em $t_{max}^{k-q} = n - sum - 2^{2(k-q)-2}$.
- Se $j = 2q + 1$ com $q = 0, \dots, k - 2$, então determinamos uma lista L que começa em $s_{min}^{k-q} = \lceil \frac{n-sum}{2} \rceil$ e termina em $s_{max}^{k-q} = n - sum - 2^{2(k-q)-3}$.
- Se $j = 2(k - 1) + 1$, então determinamos uma lista L com um único elemento $s^1 = n - sum$.

Algoritmo 1 - Define as partes de P

Entrada: n, k, l (lista)

Saída: Lista L

1. $j \leftarrow len(l)$ // $len(l)$ é o comprimento de l
 2. $sum \leftarrow \sum(l)$ // $\sum(l)$ é a soma dos elementos de l
 3. **se j é par então**
 4. $q \leftarrow \frac{j}{2}$
 5. $t_{min} \leftarrow \lceil \frac{n-sum}{2} \rceil$
 6. $t_{max} \leftarrow n - sum - 2^{2(k-q)-2}$
 7. $L \leftarrow [t_{min} : t_{max}]$ // lista crescente que começa em t_{min}
e termina em t_{max}
 8. **senão se $j < 2k - 1$ então**
 9. $q \leftarrow \frac{j-1}{2}$
 10. $s_{min} \leftarrow \lceil \frac{n-sum}{2} \rceil$
 11. $s_{max} \leftarrow n - sum - 2^{2(k-q)-3}$
 12. $L \leftarrow [s_{min} : s_{max}]$ // lista crescente que começa em s_{min}
e termina em s_{max}
 13. **senão**
 14. $L \leftarrow [n - sum]$
 15. **fim se**
-

Algoritmo 2: Dados os inteiros positivos n e k , seja l uma lista de números inteiros positivos de comprimento j tal que $0 \leq j \leq 2k$ e seja \mathcal{P}_k conjunto. O Algoritmo 2 constrói as partições $P = [s_1, t_1, \dots, s_k, t_k]$ do inteiro n da seguinte forma:

- Se $j = 2k$, então adicionamos a lista l ao conjunto P .
- Se $0 \leq j \leq 2k - 1$, aplique recursivamente os Algoritmos 1 e 2 até obter uma lista l de comprimento $2k$.

Algoritmo 2 - Constrói as partições P de comprimento $2k$ para um dado k

Entrada: n, k, l (lista), $\mathcal{P}_k = \emptyset$

Saída: Conjunto de partições \mathcal{P}_k

1. $j \leftarrow \text{len}(l)$ // $\text{len}(l)$ é o comprimento de l
 2. se $j = 2k$ então
 3. $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup l$ // adicione a reversa de l ao conjunto \mathcal{P}_k
 4. retorne
 5. senão
 6. $J \leftarrow \text{Algoritmo 1}(n, k, l)$
 7. para i em J faça
 8. Algoritmo 2($n, k, l \cup i, \mathcal{P}_k$) // aplique recursivamente o Algoritmo 2
para $n, k, l \cup i, \mathcal{P}_k$, onde $l \cup i$
indica que adicionamos i à lista l
 9. fim para
 10. fim se
-

Algoritmo 3: Dado o inteiro positivo n , seja $m = \lfloor \frac{\lfloor \log_2(n) \rfloor + 1}{2} \rfloor$. O Algoritmo 3 constrói todas as partições $P = [s_1, t_1, \dots, s_k, t_k]$ do inteiro n , para todo $k = 1, \dots, m$, da seguinte forma:

Algoritmo 3 - Constrói todas as partições P , para todo $k = 1, \dots, m$

Entrada: n

Saída: Conjunto de partições \mathcal{P}

1. $m \leftarrow \lfloor \frac{\lfloor \log_2(n) \rfloor + 1}{2} \rfloor$ // número máximo de pares (s_i, t_i)
 2. $\mathcal{P} \leftarrow \emptyset$ // inicializa o conjunto de partições
 3. para cada $k = 1, \dots, m$ faça
 4. $\mathcal{P}_k \leftarrow \emptyset$ // conjunto de partições de comprimento $2k$
 5. $l \leftarrow []$ // lista vazia
 6. Algoritmo 2(n, k, l, \mathcal{P}_k) // gera recursivamente todas as partições
de comprimento $2k$
 7. $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{P}_k$ // adiciona as partições geradas pelo
Algoritmo 2 ao conjunto de partições \mathcal{P}
 8. fim para
-

Algoritmo 4: Consiste em gerar a sequência binária de todos os grafos threshold que são simplesmente estruturados a partir das partições obtidas do Algoritmo 3. Para cada partição $P \in \mathcal{P}$, construímos a sequência binária correspondente b , ou seja, se $P = [s_1, t_1, \dots, s_k, t_k]$, então para cada $i \in \{1, \dots, 2k\}$, temos que $b_{A_0+1} = \dots = b_{A_0+s_{(q+1)}}$ é igual a 0 se i é ímpar e $b_{A_1+1} = \dots = b_{A_1+t_q}$ é igual a 1 se i é par, respectivamente, onde $q = \lfloor \frac{i}{2} \rfloor$, $A_0 = \sum_{j=1}^q (s_j + t_j)$ e $A_1 = \sum_{j=1}^{q-1} (s_j + t_j) + s_q$. Como o número de elementos em cada uma das partes s_i e t_i , para $i \in \{1, \dots, 2k\}$, foi obtido usando o Teorema 2.1, podemos garantir uma base simplesmente estruturada para o autoespaço associado a cada autovalor do grafo. Portanto, os grafos threshold definidos por essas sequências binárias são simplesmente estruturados. O Algoritmo 4 é apresentado abaixo.

Algoritmo 4 - Algoritmo para determinar grafos threshold que são simplesmente estruturados

Entrada: n (ordem do grafo)

Saída: Lista BS de sequências binárias

```

1.  $BS \leftarrow \emptyset$  // inicializa a lista de sequências binárias
2.  $\mathcal{P} \leftarrow$  Algoritmo 3( $n$ ) // obtém as partições de  $n$ 
3. para cada  $P \in \mathcal{P}$  faça
4.      $b \leftarrow \emptyset$  // inicializa sequência binária
5.     para cada  $i = 1, \dots, \text{len}(P)$  faça
6.          $q \leftarrow \lfloor \frac{i}{2} \rfloor$ 
7.          $A_0 \leftarrow \sum_{j=1}^q (s_j + t_j)$ 
8.          $A_1 \leftarrow \sum_{j=1}^{q-1} (s_j + t_j) + s_q$ 
9.         se  $i$  é ímpar então
10.             $b_{A_0+1} = \dots = b_{A_0+s_{(q+1)}} = 0$  // adicione  $s_{q+1}$  zeros a  $b$  começando no
                índice  $A_0 + 1$ 
11.         senão
12.             $b_{A_1+1} = \dots = b_{A_1+t_q} = 1$  // adicione  $t_q$  uns a  $b$  começando no
                índice  $A_1 + 1$ 
13.         fim se
14.     fim para
15.      $BS \leftarrow BS \cup b$  // adicione  $b$  à lista  $BS$ 
16. fim para

```

4 Considerações Finais

Neste trabalho caracterizamos os grafos threshold conexo G com n vértices que são simplesmente estruturados. Além disso, desenvolvemos um algoritmo para gerar todos os grafos threshold conexos que são simplesmente estruturados para dada ordem n . Alguns dos grafos obtidos pelo Algoritmo 4 satisfaz a propriedade de que a base de autovetores da matriz laplaciana admite uma ordenação tal que vetores não consecutivos sejam ortogonais, os quais chamamos de grafos threshold *fracamente Hadamard diagonalizáveis* (WHD). Em [1], os autores levantaram o problema de encontrar quais cografos são WHD. Como qualquer grafo threshold é um cografo, caracterizamos também todos os grafos threshold com n vértices que são WHD. Isso oferece uma resposta parcial ao problema proposto em [1] sobre a determinação de quais cografos são WHD.

Agradecimentos

O autor Leonardo de Lima agradece ao CNPq pelo financiamento parcial do trabalho com o projeto sob número 315739/2021-5.

Referências

- [1] M. Adm, K. Almuhtaseb, S. Fallat, K. Meagher, S. Nasserar, M. N. Shirazi e A. S. Razafimahatratra. “Weakly Hadamard diagonalizable graphs”. Em: **Linear Algebra and its Applications** 610 (2021), pp. 86–119. DOI: 10.1016/j.laa.2020.09.038.
- [2] C. O. Aguilar, M. Ficarra, N. Schurman e B. Sullivan. “The role of the anti-regular graph in the spectral analysis of threshold graphs”. Em: **Linear Algebra and its Applications** 588 (2020), pp. 210–223. DOI: 10.1016/j.laa.2019.12.005.
- [3] N. Johnston e S. Plosker. “Laplacian $\{-1, 0, 1\}$ - and $\{-1, 1\}$ -diagonalizable graphs”. Em: **Linear Algebra and its Applications** 704 (2025), pp. 309–339. DOI: 10.1016/j.laa.2024.10.016.
- [4] R. R. Macharete, R. R. Del-Vecchio, H. Teixeira e L. de Lima. “A Laplacian eigenbasis for threshold graphs”. Em: **Special Matrices** 12.1 (2024), p. 20240029. DOI: 10.1515/spma-2024-0029.
- [5] “Threshold Graphs and Related Topics”. Em: **Threshold Graphs and Related Topics**. Ed. por N.V.R. Mahadev e U.N. Peled. Vol. 56. Annals of Discrete Mathematics. Elsevier, 1995, p. iii. DOI: 10.1016/S0167-5060(13)71063-X.
- [6] D. McLaren, H. Monterde e S. Plosker. “Weakly Hadamard diagonalizable graphs and Quantum State Transfer”. Em: (jul. de 2023). arXiv: 2307.01859 [math.CO].
- [7] T. Sander e J. Sander. “On simply structured kernel bases of unicyclic graphs”. Em: **AKCE International Journal of Graphs and Combinatorics** 4 (jan. de 2007), pp. 61–82.