



NOVOS RESULTADOS SOBRE CONECTIVIDADE EM ÁRVORES

Luciana Lee

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Produção, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Produção.

Orientadores: Nair Maria Maia de Abreu
Lilian Markenzon

Rio de Janeiro
Dezembro de 2012

NOVOS RESULTADOS SOBRE CONECTIVIDADE EM ÁRVORES

Luciana Lee

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE PRODUÇÃO.

Examinada por:

Profa. Nair Maria Maia de Abreu, D.Sc.

Profa. Lilian Markenzon, D.Sc.

Prof. Samuel Jurkiewicz, D.Sc.

Profa. Cláudia Marcela Justel, D.Sc.

Prof. Mitre Costa Dourado, D.Sc.

Profa. Cybele Tavares Maia Vinagre, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
DEZEMBRO DE 2012

Lee, Luciana

Novos resultados sobre conectividade em árvores/Luciana Lee. – Rio de Janeiro: UFRJ/COPPE, 2012.

XII, 90 p.: il.; 29, 7cm.

Orientadores: Nair Maria Maia de Abreu

Lilian Markenzon

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Produção, 2012.

Referências Bibliográficas: p. 85 – 88.

1. Conectividade de Grafos. 2. Grafos Cordais. 3. Conectividade Algébrica. I. Abreu, Nair Maria Maia de *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Produção. III. Título.

Aos meus pais e irmão.

Agradecimentos

Agradeço à minha família, por todo amor, apoio e compreensão em todos os momentos.

Aos meus primos: Ta Shun, Lucia, Michelle, Priscilla e Alessandra, por todo carinho e suporte.

Às professoras Lilian Markenzon e Nair Maria Maia de Abreu, que foram muito mais que orientadoras e sem sua ajuda e apoio eu não teria conseguido chegar até aqui.

Aos meus amigos da COPPE: Aroldo, James, Joelma, José Camata, Leonardo Pessoa, Rodrigo Collazo e Telma Teixeira por todos os bons momentos que compartilhamos. Em especial, a Laura Patuzzi, por ter ajudado em tantos momentos difíceis.

Aos professores do PEP/UFRJ e ao grupo TEG, por todo o apoio.

À secretaria do PEP/UFRJ, em especial à Andreia, por todo o auxílio.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo suporte financeiro.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

NOVOS RESULTADOS SOBRE CONECTIVIDADE EM ÁRVORES

Luciana Lee

Dezembro/2012

Orientadores: Nair Maria Maia de Abreu
Lilian Markenzon

Programa: Engenharia de Produção

A conectividade de grafos e a determinação de invariantes capazes de medi-la é assunto bastante conhecido na literatura. Dentre tais invariantes, o mais conhecido é a conectividade de vértices, que indica o menor número de vértices que devem ser removidos para que o grafo seja desconectado. Apesar de sua ampla utilização, pouca informação é obtida sobre a conectividade dos grafos. Há classes, como, por exemplo, as árvores, em que os elementos não são diferenciados, embora suas estruturas sugiram que eles devam ser distinguidos quanto à sua conectividade. Neste trabalho é proposto um novo invariante para grafos cordais, a sequência de componentes conexas, que permite diferenciar a conectividade entre grafos com mesma conectividade de vértices. Em particular, foram estudados os grafos k -separados, uma subclasse dos cordais, sendo apresentado um algoritmo que gera todas as sequências de componentes conexas possíveis para os elementos desta classe.

Foi realizado um estudo comparativo em árvores entre o invariante proposto e a conectividade algébrica, um invariante de Teoria Espectral de Grafos para o qual existem muitos trabalhos que investigam a ordenação de classes. Nesse estudo, definiu-se uma classe de árvores tal que seus elementos possuem a mesma conectividade algébrica e para a qual o novo invariante apresentou uma ordenação para os elementos. Finalmente, foi provado que a classe definida maximiza a conectividade algébrica das árvores starlike.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

NEW RESULTS ABOUT THE CONNECTIVITY OF TREES

Luciana Lee

December/2012

Advisors: Nair Maria Maia de Abreu

Lilian Markenzon

Department: Production Engineering

The connectivity of graphs and determination of invariants capable of measure it is well known subject in the literature. Among these invariants, the most famous is the vertex connectivity, which indicates the smallest number of vertices that must be removed so that the graph is disconnected. Despite its widespread use, little information is obtained about the connectivity of the graphs. There are classes, such as, trees, wherein the elements are not distinguished, although their structures suggest that they should be distinguished as to their connectivity. In this work a new invariant for chordal graphs is proposed, the sequence of connected components, which distinguish between the connectivity of graphs with the same vertex connectivity. In particular, we studied the k -separated graphs, a subclass of the chordal graphs, and we present an algorithm that generates all possible sequences of connected components related to the elements of this class.

We performed a comparative study applied to the trees between the proposed invariant and the algebraic connectivity, an invariant of the Spectral Graph Theory, for which there are many studies that investigate the ordering of classes. In this study, we defined a class of trees such that their elements have the same algebraic connectivity and to which the invariant presented a new ordering for the elements. Finally, it was proven that the class defined maximizes the algebraic connectivity of the starlike trees.

Sumário

Lista de Figuras	x
Lista de Tabelas	xii
1 Introdução	1
2 Noções básicas	4
2.1 Conceitos básicos e notações	4
2.2 Grafos Cordais	6
2.3 Conectividade Algébrica	11
3 Um novo invariante para o estudo de conectividade	14
3.1 O novo invariante	14
3.2 <i>SCC</i> 's para grafos cordais k -sep	19
3.3 Geração das <i>SCC</i> 's em grafos cordais k -sep	22
4 Conectividade em árvores	28
4.1 Introdução	28
4.2 Características da <i>SGI</i>	30
4.3 Comparação entre <i>SGI</i> e Conectividade Algébrica em algumas classes de árvores	33
4.4 <i>SGI</i> e Conectividade Algébrica em árvores com diâmetro 4	43
5 A classe das medusas	45
5.1 Localização dos autovalores de uma árvore	45
5.1.1 Diagonalização de $\lambda I + L$	52
5.2 Medusas e árvores <i>starlike</i>	57
5.3 Árvores não-medusa	63
5.4 <i>SGI</i> 's para as árvores de \mathcal{T}_2	73
6 Conclusões e trabalhos futuros	76
6.1 Conclusões	76
6.2 Trabalhos futuros	77

Referências Bibliográficas	85
Índice Remissivo	89

Lista de Figuras

2.1	Grafos que realizam a sequência $\pi = [3, 3, 2, 2, 2, 2]$	5
2.2	Hierarquia em árvore enraizada.	6
2.3	Coalescência entre os grafos G_1 e G_2	6
2.4	Um <i>EEP</i> do grafo G_1	8
2.5	Separadores Minimais de Vértices e Separadores Minimais do Grafo	9
2.6	Grafo clique de G_2	10
2.7	Árvores de cliques do grafo G_2 com os <i>SMV</i> 's discriminados.	10
3.1	Grafo $G_1 = (V_1, E_1)$ com $n = 11$, $m = 24$ e $SCC(G_1) = [4, 3, 2, 2, 2]$	15
3.2	Árvores de cliques dos grafos G_1 e G_2 e suas respectivas <i>SCC</i> 's.	17
3.3	Grafos cordais 3-sep, G_3 e G_4 com suas respectivas árvores de cliques.	19
3.4	Grafo G_3 e sua respectiva árvores de cliques.	20
3.5	Obtenção de um grafo 3-sep a partir de $S = [3, 2, 2]$	26
3.6	Grafo G^3 do exemplo anterior e o grafo G^4 não isomorfo a G_3 que também realiza a $SCC = [3, 2, 2]$	27
4.1	<i>SGI</i> 's para casos extremos em árvores.	29
4.2	Exemplos de <i>SGI</i>	29
4.3	Aplicação da nova medida para árvores com 8 vértices	31
4.4	Construção de uma árvore que realiza $SGI = [5, 4, 2]$	32
4.5	Transcrição para números inteiros das <i>SCC</i> 's de árvores com 8 vértices.	33
4.6	Árvores $T_2(k, \ell)$ com $n = 12$ vértices.	34
4.7	Estrutura de uma árvore $T_d(k, \ell)$	35
4.8	Árvores $T_d(k, l)$ com 18 vértices.	35
4.9	Estruturas $K_{1,2}$, $K_{1,1}$ e K_1	35
4.10	Estrutura das árvores pertencentes às classes C_i , $1 \leq i \leq 6$	36
4.11	Árvores das classes C_1 , C_2 e C_4	36
4.12	Árvores com $n = 15$ vértices para as classes C_1 , C_2 e C_4	40
4.13	Árvores de C_3 com 15 vértices.	40
4.14	Árvores de C_5 com 15 vértices.	41
4.15	Árvores de C_6 com 15 vértices.	42

4.16	Estrutura de uma árvore $T(n, k, p_1, \dots, p_k)$	43
5.1	Inicialização das entradas para os vértices de $T = (V, E)$	48
5.2	Localização dos autovalores com relação a 2, ou seja, $\lambda = -2$	49
5.3	Localização dos autovalores com relação ao valor 1 ($\lambda = -1$).	51
5.4	Ilustração da rotulação aplicada a uma árvore medusa $T \in \mathcal{M}(20, 10, 5)$	58
5.5	Rotulação aplicada ao caminho P_{2k+1}	59
5.6	Estrutura de uma árvore <i>starlike</i> que não é medusa.	62
5.7	Estruturas das árvores $P_{l_0+1}^{u_0}$, $S_{l_1+1}^{u_1}$ e $R_{l_2}^{u_2}$ enraizadas em u_0 , u_1 e u_2 , respectivamente.	64
5.8	Estrutura de uma árvore $T \in \mathcal{T}_{n,k}(q, r, s)$	64
5.9	Estrutura de uma árvore $T \in \mathcal{T}_{n,k}(2, 1, 0)$	65
5.10	Árvores $T_i \in \mathcal{T}_{n,4}(0, 1, 0)$, com $n = 15, 16, 17$	66
5.11	Estrutura de uma árvore $T \in \mathcal{T}_1$	66
5.12	Estrutura de uma árvore $T \in \mathcal{T}_k(0, 0, 1)$	67
5.13	Árvores $T_i \in \mathcal{T}_{n,4}(2, 0, 1)$, com $n = 16, 17, 18$	69
5.14	Árvores $T \in \mathcal{T}_2$ com $k = 5$ e $n = 28$	70
5.15	Número máximo de folhas para cada nível do ramo 3 de uma árvore T com $k = 5$	70
5.16	Árvore T com diâmetro $2k$	71
5.17	Árvores $T \in \mathcal{T}_{15,4}(q, r, s)$ com $n = 15$ vértices e $a(T) = 0,12061$	74
5.18	Árvores de $\mathcal{T}_{15,4}(q, r, s)$ que não pertencem a \mathcal{T}_2	75
6.1	Árvores T_1 e T_2 não pertencentes a \mathcal{T}_2 , tais que $a(T_1) = a(T_2) = a(P_{13})$	78
6.2	Análise das estruturas de árvores da classe C_6	79
6.3	Árvores com $a(T_i) = 0,06367$	79
6.4	Exemplo de um grafo bloco G , os blocos de G e sua representação $bc(G)$	81
6.5	Árvores com $a(T) = 0,12061$ e suas respectivas $bc(T)$	82
6.6	Grafos bloco com mesma representação $bc(G_i)$ das árvores da Figura 6.5 e $a(G_i) = 0,09305$	83
6.7	Exemplos de grafos bloco.	84

Lista de Tabelas

4.1	Sequências de componentes conexas para as árvores com 8 vértices.	32
-----	---	----

Capítulo 1

Introdução

Muitos sistemas no mundo real podem ser modelados por grafos. Alguns exemplos são a *internet*, as redes de transporte e telecomunicações, redes de energia, gás e água, circuitos elétricos, entre muitos outros. Em geral, os problemas tratados nesses modelos visam a redução de custos e/ou a maximização de lucros. Independente do problema a ser tratado, o planejamento de um sistema deve levar em conta a construção de um modelo seguro, ou seja, o sistema deve ser modelado por um grafo que não seja facilmente desconectável.

Muitos estudos foram realizados com respeito a conectividade em grafos e a determinação de invariantes capazes de medi-la. Alguns desses invariantes são a conectividade de vértices, a conectividade de arestas, a conectividade condicional, a conectividade algébrica, a integridade, a integridade fraca, a integridade média, a resistência, o número de dispersão e a tenacidade, [41]. Dentre estes, os mais conhecidos são a conectividade de vértices e a conectividade de arestas que indicam, respectivamente, o menor número de vértices e de arestas que devem ser removidos de um grafo para que este se torne desconexo.

Este trabalho se inicia com um estudo sobre a conectividade em grafos cordais. Esta classe tem sido muito estudada na literatura pois sua estrutura baseada em cliques permite soluções mais eficientes para muitos problemas considerados difíceis para grafos em geral [33]. A abordagem aqui utilizada se baseia em propriedades estruturais dos grafos dessa classe, tais como clique maximal, grafo clique, árvore de cliques e separadores minimais de vértices, utilizando resultados conhecidos da literatura que tratam desses aspectos e que podem ser encontrados em [6] e [18].

A Teoria Espectral de Grafos é uma área de pesquisa da Matemática Discreta e Combinatória que estuda a relação entre as estruturas dos grafos e suas matrizes. Tal área se dedica a estudar as estruturas dos grafos via autovalores das matrizes que os descrevem, tais como as matrizes de adjacência, Laplaciana e Laplaciana sem sinal e estes autovalores constituem-se em invariantes espectrais para os grafos. Dentre tais invariantes, o segundo menor autovalor da matriz Laplaciana de um

grafo será do interesse desse trabalho. De fato, Fiedler, em 1973, provou que um grafo é conexo se e somente se tal invariante é diferente de zero, veja [15]. Fiedler chamou tal valor de conectividade algébrica. Existem inúmeros trabalhos dedicados à conectividade algébrica, em geral comparando-a com outros invariantes clássicos de grafos ou estudando-a em grafos pertencentes a classes especiais, [3, 30, 42–45, 49, 50]. Uma das qualidades da conectividade algébrica é que ela é capaz de comparar dois ou mais grafos k -conexos, isto é, com a mesma conectividade de vértices. Há várias subclasses de árvores ou de grafos com poucos ciclos cujos elementos são ordenados pelas suas respectivas conectividades algébricas, veja [21, 30, 31]. Em [2], é dito que “(...) encontrar uma ordenação total para árvores via conectividade algébrica ainda é um problema em aberto” e, em [37], é provado que o problema de adicionar um número específico de arestas em um grafo a fim de maximizar a conectividade algébrica do grafo resultante é um problema NP-completo. As questões acima levantadas visam entender melhor a questão de qual grafo é mais conexo que outro, sobretudo quando as conectividades clássicas de vértices e de arestas são incapazes de fazer tal distinção. Veja que questões como estas estão diretamente associadas à avaliação da segurança da rede que o grafo modela.

Neste trabalho é proposto um novo invariante para grafos, relacionado à conectividade, capaz de distinguir grafos de conectividade de vértices igual a k , como já dito, conhecidos como grafos k -conexos. Este novo invariante será comparado à conectividade algébrica. Para tal comparação, algumas classes de árvores serão analisadas.

Este trabalho é organizado como segue. No Capítulo 2 são apresentados conceitos básicos de Teoria de Grafos que serão utilizados no decorrer do trabalho. Também são estudados conceitos relacionados a grafos cordais, tais como separador minimal de vértices, esquema de eliminação perfeito, grafo clique e árvore de cliques. Por fim, alguns conceitos e resultados de Teoria Espectral de Grafos referentes à conectividade algébrica são introduzidos.

No Capítulo 3 é apresentado um novo invariante para diferenciar a conectividade entre grafos cordais k -conexos. Tal invariante é dado por uma sequência numérica em que cada elemento corresponde ao número de componentes conexas resultante da remoção de um conjunto separador minimal do grafo. Esta sequência é aqui chamada de sequência de componentes conexas e é abreviada por *SCC*. Neste capítulo é também apresentado um algoritmo que gera todas as *SCC*'s possíveis para grafos k -sep, dado o número de cliques maximais. No Capítulo 4 estudam-se características estruturais dos grafos cordais que influenciam em sua conectividade. Para isso, a *SCC* e a conectividade algébrica são aplicadas aos elementos de algumas classes de árvores introduzidas em [12, 19, 49] e seus comportamentos são comparados. Em [49], são apresentadas seis classes de árvores de diâmetro 4 que maximizam

a conectividade algébrica em árvores. Com a análise do comportamento dos dois invariantes estudados, observa-se que algumas das classes têm mais de um elemento e todos possuem mesma conectividade algébrica. Tal observação levou ao estudo das características de tais árvores a fim de determinar condições para se ter árvores com tal propriedade. No Capítulo 5, são estudadas árvores de diâmetro $2k$, $k \geq 2$, cuja conectividade algébrica é igual a do caminho P_{2k+1} . São definidas classes de árvores que possuem tal propriedade e são estudadas as características estruturais que levam tais árvores a ter mesma conectividade algébrica.

Capítulo 2

Noções básicas

Neste capítulo serão apresentados alguns conceitos e resultados de teoria de grafos que podem ser encontrados em [7, 11, 17, 24] e poderão ser úteis para tornar este texto auto-contido.

2.1 Conceitos básicos e notações

Um grafo $G = (V, E)$ é composto por um conjunto não vazio V de elementos chamados *vértices* e um conjunto E de pares não ordenados de pontos distintos $a, b \in V$, $x = \{a, b\}$, chamados de *arestas*. Dados dois vértices $a, b \in V$, diz-se que a e b são *adjacentes* quando estes são extremidades de uma aresta $x = \{a, b\}$. Neste caso, diz-se que a aresta x é incidente tanto no vértice a quanto no vértice b . Grafos com tal definição também são chamados de *grafos simples*.

Um subgrafo de um grafo $G = (V, E)$ é um grafo $H = (W, F)$, tal que $W \subseteq V$ e $F \subseteq E$. Diz-se que H é um *subgrafo próprio* de G se $H \neq G$. Além disso, H é dito ser induzido quando dois vértices de W são adjacentes em H se e somente se eles são adjacentes em G . Neste caso, diz-se que H é um *subgrafo induzido* de G por W , denotado por $H = G[W]$.

A *vizinhança* de $v \in V$ é denotada por $Adj(v) = \{u \in V | \{u, v\} \in E\}$. Um *percurso* é uma sequência alternada de vértices e arestas, $v_1, x_1, v_2, \dots, v_n$, começando e terminando com vértices, onde x_i é a aresta que liga os vértices v_i e v_{i+1} . Assim, um percurso pode ser denotado como uma sequência de vértices, dado que as arestas estão implícitas, ou seja, v_1, v_2, \dots, v_n . Um *caminho* em um grafo G é um percurso no qual todos os vértices aparecem apenas uma vez. Um caminho com n vértices é denotado por P_n . Um *ciclo* é um percurso em que o primeiro e o último vértices são iguais, sendo todos os outros vértices distintos entre si e destes. Um ciclo com n vértices é denotado por C_n .

Um grafo $G = (V, E)$ é *conexo*, se e somente se, para quaisquer dois vértices $a, b \in V$ existe ao menos um caminho que liga a a b . Um *subgrafo conexo maximal*

de um grafo G é chamado de *componente conexa* de G ou simplesmente componente de G . Assim, um grafo desconexo possui ao menos duas componentes conexas.

A *conectividade de vértices* de um grafo conexo G é a cardinalidade do menor conjunto de vértices cuja remoção desconecta G , e será denotada por $\kappa(G)$. Para qualquer $k \leq \kappa(G)$ diz-se que G é *k-conexo*. *Grafos completos* não possuem conjunto de corte, mas por convenção, define-se que a conectividade de vértices de tais grafos é igual a $n - 1$.

Num grafo conexo G , a *distância* entre dois vértices u e v de G é o número de arestas do menor caminho que os liga, denotada por $dist(u, v)$. A *excentricidade* de um vértice v é a maior das distâncias entre v e qualquer outro vértice de G . O *raio* de G é a menor excentricidade dentre aquelas de G . O *diâmetro* de G é a maior das excentricidades. O *centro* de um grafo é um conjunto de vértices, onde cada elemento tem excentricidade igual ao raio. Um vértice é dito ser *central* se ele está contido no centro. Um vértice periférico é aquele que possui excentricidade igual ao diâmetro.

O *grau* de um vértice $v \in V$ é igual ao número de arestas incidentes no mesmo e é denotado por $d(v)$. Para um grafo G com m arestas, tem-se $\sum_{i=1}^n d(v_i) = 2m$. Uma sequência de números inteiros positivos $\pi = [d_1, d_2, \dots, d_n]$ é uma *sequência gráfica* se existe um grafo G tal que π seja igual a sequência de graus de G . O grafo G é chamado de uma *realização* de π , ou ainda, diz-se que G realiza π . Note que uma sequência gráfica π pode ser realizada por mais de um grafo. Um exemplo é ilustrado na Figura 2.1.

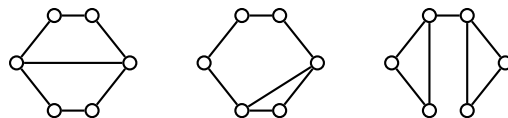


Figura 2.1: Grafos que realizam a sequência $\pi = [3, 3, 2, 2, 2, 2]$.

Um *grafo completo* com n vértices, representado por K_n , é aquele que para qualquer par de vértices existe uma aresta que os une. Uma *clique* em um grafo $G = (V, E)$ é qualquer subconjunto de vértices $V_1 \subseteq V$ onde $G[V_1]$ é completo. Neste caso, diz-se também que $G[V_1]$ é um *subgrafo completo* de G . Um subgrafo é uma *clique maximal* de G se ele não está contido em nenhuma outra clique de G . Um conjunto de vértices é dito ser *independente* se não existem arestas que ligam quaisquer dois pontos desse conjunto.

Dois grafos $G = (V, E)$ e $H = (W, F)$ são *isomorfos* se existe uma bijeção ϕ de V em W tal que, para $x, y \in V$, x é adjacente a y em G se e somente se $\phi(x)$ é adjacente a $\phi(y)$ em H . Diz-se que ϕ é um isomorfismo de G em H . Como ϕ é uma bijeção, ela possui uma inversa, que é o isomorfismo de H em G . Diz-se então que

G e H são isomorfos.

Uma *árvore* é um grafo conexo acíclico. Os vértices de uma árvore com grau maior que 1 são chamados de *vértices internos* e aqueles com grau igual a 1 são chamados de *folhas*. Uma árvore com n vértices, sendo que destes vértices $n - 1$ são folhas, é chamado de *estrela* e denotado por S_n . Uma *floresta* é um grafo em que cada componente conexa é uma árvore.

Uma árvore é enraizada se ela possui um vértice especial, chamado de *raiz*. A partir da raiz de uma árvore é possível definir uma hierarquia entre seus vértices. Suponha a árvore T ilustrada na Figura 2.2. A árvore T é enraizada em r . Dados x , y , z e u vértices de T , se x é um vértice no caminho de r a y , então diz-se que x é ancestral de y , ou também, que y é descendente de x . Se, além disso, x e y são adjacentes, então x é pai de y e y é filho de x . Se existe outro vértice z filho de x , então y é dito ser irmão de z . Se y tem um filho u , então x é avô de u e z é tio de u .

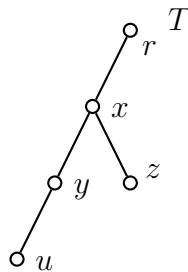


Figura 2.2: Hierarquia em árvore enraizada.

A *coalescência* de dois grafos G_1 e G_2 com vértices distintos v_1 e v_2 , é o grafo G , denotado por $G_1 \circ G_2$ tal que G é formado pela identificação de v_1 com v_2 , isto é, os vértices v_1 e v_2 são substituídos por um único vértice v em G adjacente aos mesmos vértices de G_1 adjacentes a v_1 e aos mesmos vértices de G_2 adjacentes a v_2 . Veja [20]. A Figura 2.3 ilustra a coalescência entre dois grafos.

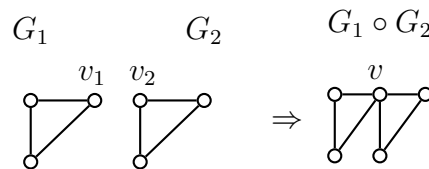


Figura 2.3: Coalescência entre os grafos G_1 e G_2 .

2.2 Grafos Cordais

Um grafo é *cordal* se ele não possui ciclos induzidos de tamanho maior que três. Essa família possui aplicações nas áreas de biologia, computação, estatística e na

computação de matrizes esparsas [34]. Graças à sua estrutura baseada em cliques é possível desenvolver algoritmos eficientes para soluções de problemas que são considerados difíceis para grafos em geral. Em [6], é apresentada uma introdução aos grafos cordais com enfoque em sua estrutura.

É possível representar um grafo cordal através de uma árvore, como é visto adiante. Essa estrutura é chamada de *árvore de cliques*. Podem-se extrair, diretamente desta representação, informações relevantes à conectividade do grafo.

Para melhor compreensão da utilização dessa estrutura na análise da conectividade do grafo, alguns conceitos básicos e resultados relacionados a grafos cordais e árvore de cliques, tratados em [6], são apresentados a seguir.

Definição 2.1. *Um vértice v de um grafo G é simplicial se sua vizinhança induz um subgrafo completo em G .*

Um resultado relacionado aos vértices simpliciais é apresentado no Lema 2.1 que, de acordo com Blair e Peyton [6], foi enunciado por Dirac em 1961.

Lema 2.1 (Dirac, 1961). *Todo grafo cordal G possui um vértice simplicial. Se G não é completo então G possui ao menos dois vértices simpliciais não adjacentes.*

Sabe-se que ao retirar um vértice simplicial de um grafo cordal, o grafo resultante também é cordal. Conhecendo esta propriedade e o resultado do Lema 2.1, pode-se obter uma ordenação dos vértices de um grafo cordal. Esta ordenação considera, a cada passo, um vértice simplicial v_i a ser removido do subgrafo induzido $G[v_i, \dots, v_n]$, para $1 \leq i \leq n$.

Definição 2.2. *Uma bijeção $\sigma : V \rightarrow \{1, 2, \dots, n\}$ é um esquema de eliminação perfeita (EEP) se, para $1 \leq i \leq n$, o vértice v_i é simplicial no grafo induzido $G[v_i, \dots, v_n]$.*

Para melhor entendimento sobre a definição acima, a geração de um EEP para o grafo G_1 é ilustrado na Figura 2.4. Observa-se que o grafo inicial possui quatro vértices simpliciais: a , e , h e i . Então, é possível iniciar a construção do EEP com qualquer um deles. Portanto, o EEP não é único para o grafo. O EEP para G_1 será representado pela sequência σ . Inicia-se a construção de σ escolhendo-se o vértice a e removendo-o do grafo. Com a retirada deste vértice de G_1 , b passa a ser um vértice simplicial do grafo resultante. Escolhe-se o vértice b como próximo elemento de σ e esse procedimento é repetido até que o último vértice seja removido de G_1 . Ao final, tem-se a sequência $\sigma = \langle a, b, e, h, i, c, d, f, g \rangle$.

Um resultado que relaciona o grafo cordal a um EEP é apresentado no Lema 2.2 que, de acordo com Blair e Peyton [6], foi enunciado por Fulkerson e Gross [16].

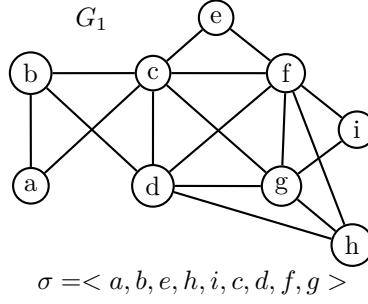


Figura 2.4: Um *EEP* do grafo G_1 .

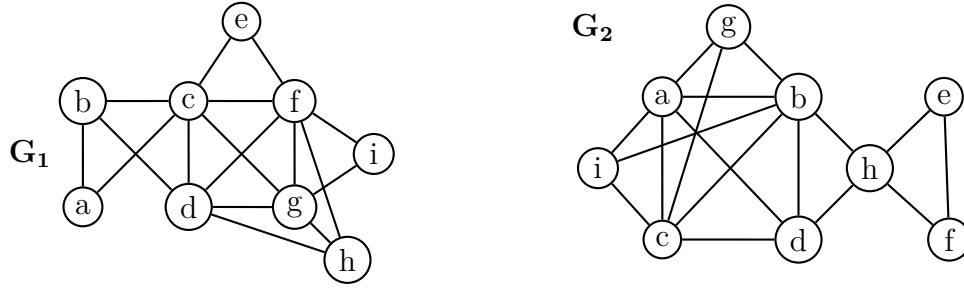
Lema 2.2 (Fulkerson e Gross [16]). *Um grafo G é cordal se e somente se G possui um *EEP*.*

Dois algoritmos muito citados na literatura para encontrar um *EEP* de um grafo cordal são: a *busca em largura lexicográfica* e a *busca por cardinalidade máxima*. A descrição e a prova de correção do primeiro pode ser encontrado em [47] e a do segundo, em [6].

Definição 2.3. *Seja $G = (V, E)$ um grafo e $v \in V$. Um subconjunto $S \subset V$ é um separador de G se ao menos dois vértices na mesma componente conexa de G estão em duas componentes conexas distintas de $G[V - S]$. O conjunto S é um separador minimal do grafo G , *SMG*, se S é um separador e nenhum subconjunto próprio de S separa o grafo. Um subconjunto $S \subset V$ é um separador de vértices para vértices não adjacentes u e v (um *uv-separador*) se a remoção de S do grafo separa u e v em duas componentes conexas distintas. Se nenhum subconjunto próprio de S é um *uv-separador* então S é um *uv-separador minimal*. Se S é um *uv-separador minimal* para algum par de vértices u e v , ele é chamado de um separador minimal de vértices, *SMV*. Um separador minimal é sempre um separador minimal de vértices mas a recíproca não é verdadeira.*

Para exemplificar essas definições, são apresentados os grafos G_1 e G_2 na Figura 2.5. O conjunto cujos elementos são os *SMV*'s distintos será denotado por $\mathcal{S}(G)$. Assim, os conjuntos de *SMV*'s para os grafos G_1 e G_2 ilustrados na Figura 2.5 são, respectivamente, $\mathcal{S}(G_1) = \{\{b, c\}, \{c, d\}, \{c, f\}, \{d, f, g\}, \{f, g\}\}$ e $\mathcal{S}(G_2) = \{\{a, b, c\}, \{b, d\}, \{h\}\}$. No grafo G_1 tem-se que $\{d, f, g\}$ é um *ch-separador minimal*, logo ele é um *SMV*. Como existe o subconjunto $\{f, g\}$ que separa o vértice i do restante do grafo, pode-se concluir que $\{d, f, g\}$ não é um *SMG*. O conjunto $\{f, g\}$ é um *SMG* porque para este não há nenhum subconjunto próprio que separa o grafo G em duas ou mais componentes. No grafo G_2 tem-se que todos os *SMV*'s são *SMG*'s.

Pode-se observar que para G_1 os *SMG*'s são $\{\{b, c\}, \{c, d\}, \{c, f\}, \{f, g\}\}$, e para G_2 os *SMG*'s são $\{\{a, b, c\}, \{b, d\}, \{h\}\}$. Nota-se que em $\mathcal{S}(G_2)$, os separadores



$$\mathcal{S}(G_1) = \{\{b, c\}, \{c, d\}, \{d, f\}, \{d, f, g\}, \{f, g\}\} \quad \mathcal{S}(G_2) = \{\{a, b, c\}, \{b, d\}, \{h\}\}$$

Figura 2.5: Exemplo de Separadores Minimais de Vértices.

possuem cardinalidades diferentes e o conjunto de *SMG*'s de G_2 coincide com seu conjunto de *SMV*'s. Tem-se ainda que $\kappa(G_1) = 2$ e $\kappa(G_2) = 1$.

Nos exemplos da Figura 2.5, todos os *SMG*'s definem a conectividade de vértices de G_1 e, para o grafo G_2 , o único conjunto que define sua conectividade de vértices é $\{h\}$. De acordo com Blair e Peyton [6], o Teorema 2.1 é devido a Dirac e caracteriza os grafos cordais com relação aos separadores minimais de vértices. Merris [36] apresenta uma prova para este teorema. Por este teorema os grafos G_1 e G_2 da Figura 2.5 são cordais.

Teorema 2.1 (Dirac). *Um grafo G é cordal se e somente se todo separador minimal de vértices de G é uma clique em G .*

Suponha $\mathcal{K}_G = \{Q_1, Q_2, \dots, Q_w\}$ o conjunto de todas as w cliques maximais de G . Para qualquer grafo cordal é possível gerar um *grafo clique* como definido a seguir.

Definição 2.4. *Sejam G um grafo cordal e $\mathcal{K}_G = \{Q_1, Q_2, \dots, Q_w\}$ o conjunto de cliques maximais de G . Seja $G' = (\mathcal{K}_G, E')$ um grafo onde os vértices representam as cliques maximais de G e as arestas sejam tais que $\{Q_i, Q_j\} \in E'$, para $Q_i, Q_j \in \mathcal{K}_G$, se $Q_i \cap Q_j \neq \emptyset$. Então G' é chamado de grafo clique de G . Além disso, tais arestas são valoradas de acordo com a cardinalidade da interseção entre as cliques que as compõem.*

A Figura 2.6 ilustra o grafo clique G'_2 a partir do grafo G_2 , dado na Figura 2.5. Os pesos das arestas de G'_2 indicam a cardinalidade da interseção das cliques correspondentes aos vértices das respectivas arestas.

Definição 2.5. *Dados um grafo cordal G e seu respectivo grafo clique G' . Uma árvore de cliques de G é uma árvore geradora de peso máximo em G' . Esta é denotada por $T_G = (\mathcal{K}_G, \mathcal{E}_T)$, onde \mathcal{K}_G é o conjunto de vértices da árvore (as cliques maximais do grafo original) e \mathcal{E}_T representa o conjunto de arestas da árvore (as interseções não-vazias de peso máximo entre as cliques).*

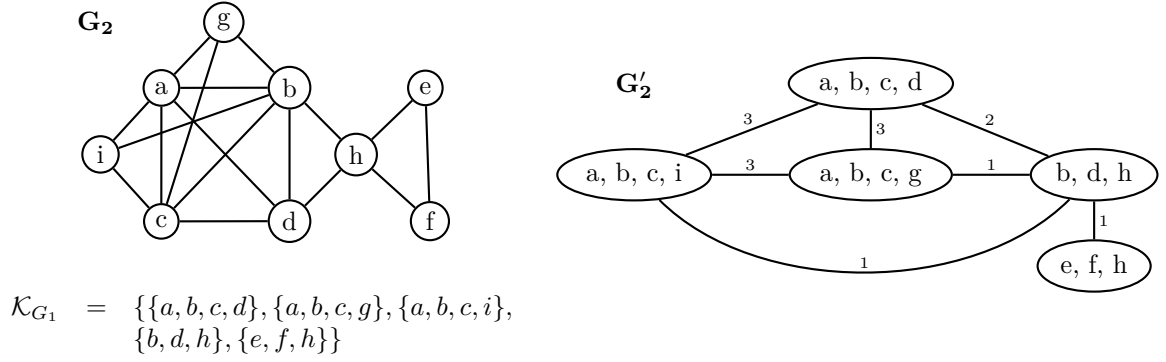


Figura 2.6: Grafo clique de G_2 .

Nota-se que pode existir mais de uma árvore de cliques para um mesmo grafo. Denota-se por \mathcal{T}_G , o conjunto de todas as árvores de cliques de um grafo G . A primeira coluna da Figura 2.7 ilustra as árvores de cliques do grafo G_2 , cujo grafo clique G'_2 está na Figura 2.6. Neste exemplo, tem-se que o conjunto de árvores de cliques de G_2 possui apenas três elementos distintos, $\mathcal{T}_{G_2} = \{T_1, T_2, T_3\}$.

Ao analisar as interseções entre os vértices da árvore de cliques T_1 , tem-se: $\{a, b, c, d\} \cap \{a, b, c, g\} = \{a, b, c\}$, $\{a, b, c, d\} \cap \{a, b, c, i\} = \{a, b, c\}$, $\{a, b, c, d\} \cap \{b, d, h\} = \{b, d\}$ e $\{b, d, h\} \cap \{e, f, h\} = \{h\}$. O valor atribuído a cada aresta é igual à cardinalidade da interseção entre as cliques em suas extremidades, ou seja, se refere à cardinalidade do *SMV* que tal aresta representa. Na segunda coluna da Figura 2.7 a transcrição dos pesos das arestas das árvores de cliques em termos dos respectivos conjuntos de separadores minimais de vértices é apresentado.

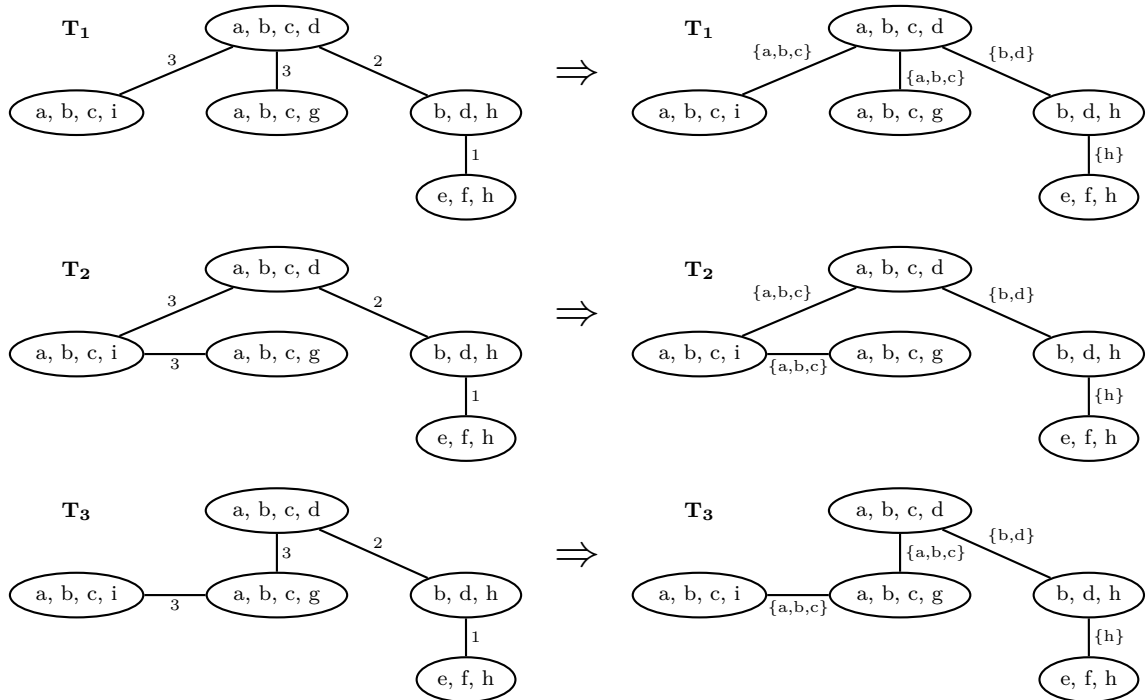


Figura 2.7: Árvores de cliques do grafo G_2 com os *SMV*'s discriminados.

Nota-se que os valores das arestas são iguais nas três árvores e correspondem aos SMV 's do grafo original. Além disso, um SMV pode ocorrer mais de uma vez na árvore de cliques. Em teoria dos conjuntos, o número de vezes que um elemento aparece em um multiconjunto é denominado *multiplicidade*. Assim, o conjunto que representa todas as arestas de uma árvore de cliques é um multiconjunto de SMV 's, denotado por $\mathcal{M}(T)$. Para as árvores da Figura 2.7 o multiconjunto de SMV 's é $\mathcal{M}(T_1) = \mathcal{M}(T_2) = \mathcal{M}(T_3) = \{\{a, b, c\}, \{a, b, c\}, \{b, d\}, \{h\}\}$ e a multiplicidade dos conjuntos $\{a, b, c\}$, $\{b, d\}$ e $\{h\}$ são respectivamente 2, 1 e 1.

Teorema 2.2 (Ho e Lee, 1989 [25], Lundquist, 1990 [32]). *O multiconjunto de separadores minimais de vértices é o mesmo para toda árvore de cliques $T \in \mathcal{T}_G$.*

Um grafo cordal pode ser representado por mais de uma árvore de cliques, mas para trabalhar com as informações necessárias para o estudo de sua conectividade, não é necessário enumerar todas as árvores de cliques do grafo, pois diante do Teorema 2.2, basta encontrar uma árvore de cliques para o grafo e considerar seu multiconjunto de SMV 's. De acordo com [33], Spinrad, em [46], apresenta um algoritmo para obter uma árvore de cliques, através de uma construção indutiva do grafo.

O problema de encontrar os separadores minimais de vértices para um grafo cordal já foi trabalhado em [8, 28, 33]. Kumar e Madhavan [28] propuseram um algoritmo de duas fases baseado na busca por cardinalidade máxima, cuja complexidade é dita ser $O(\omega(G)n \log |\mathcal{S}(G)| + m)$. Markenzon e Pereira [33] propuseram um algoritmo mais simples que [28], composto de apenas uma fase. Esse algoritmo tem como base a árvore de cliques do grafo cordal fornecido. Ele é baseado na busca em largura lexicográfica e em alguns resultados apresentados em [6].

2.3 Conectividade Algébrica

Informalmente pode-se dizer que um invariante relativo à conectividade de um grafo é um parâmetro capaz de informar sobre a sua conectividade, ou seja, sobre o quão é “fácil” ou “difícil” desconectar um grafo. Tais invariantes são muito utilizados em diversos tipos de aplicações que envolvem redes e há na literatura diversos parâmetros utilizados para este fim. Veja, por exemplo, a referência [41].

Como visto anteriormente, a conectividade de vértices de um grafo G , denotada por $\kappa(G)$, e a conectividade de arestas, denotada por $\lambda(G)$, são parâmetros clássicos usados para medir a conectividade de um grafo.

Dentre os invariantes relativos à conectividade de um grafo há um em especial, conhecido por *conectividade algébrica*, muito estudado em Teoria Espectral de Grafos. Esta subseção será dedicada a este invariante.

A Teoria Espectral de Grafos busca obter propriedades estruturais dos grafos a partir das propriedades dos autovalores de matrizes relacionadas a esses grafos. As matrizes utilizadas são quadradas e de ordem n . Considere um grafo $G = (V, E)$, com $|V| = n$ vértices, rotulados por valores de 1 a n , e $|E| = m$ arestas. A *matriz de Adjacências* de G , denotada por $A(G) = [a_{ij}]_{n \times n}$, é tal que $a_{ij} = a_{ji} = 1$ se os vértices i e j são adjacentes em G e $a_{ij} = a_{ji} = 0$, caso contrário. A *matriz de graus* $D(G)$ é a matriz diagonal cujos elementos da diagonal principal são os graus dos vértices em G . A *matriz identidade* de ordem n é denotada por I_n . O *polinômio característico* de uma matriz M quadrada de ordem n é definido por $p_M(x) = \det(M - xI)$, e suas raízes, $\alpha_1, \alpha_2, \dots, \alpha_n$, são chamados de *autovalores* de M . A *matriz laplaciana*, ou *Laplaciano*, de um grafo G é $L(G) = D(G) - A(G)$. O polinômio característico associado a $L(G)$ é denominado *polinômio característico do Laplaciano* de G e definido por $p_{L(G)}(\alpha) = \det(L(G) - \alpha I)$. O espectro de $L(G)$, denotado por $\zeta(G)$, é definido como sendo uma matriz linha cujos elementos são os autovalores de G ordenados em ordem não-crescente. Assim, se $\alpha_1 \geq \dots \geq \alpha_n$ são os autovalores de $L(G)$ então o espectro de $L(G)$ é dado por:

$$\zeta(G) = (\alpha_1, \alpha_2, \dots, \alpha_{n-1}, \alpha_n).$$

A *conectividade algébrica*, que corresponde ao segundo menor autovalor da matriz laplaciana de um grafo, α_{n-1} , é um parâmetro espectral importante que mede a conectividade de redes. Em [2], é apresentado um *survey* sobre conectividade algébrica.

Observa-se que a matriz de adjacências de um grafo G não depende apenas de seus vértices, mas também da ordem como eles estão dispostos na matriz. Entretanto, para grafos isomorfos, suas matrizes de adjacências são semelhantes [5]. O mesmo ocorre com relação aos seus Laplacianos. Ou seja, suponha G e G' dois grafos isomorfos, então existe uma matriz de permutação P onde $L(G) = P^{-1} \cdot L(G') \cdot P$. Assim, os polinômios característicos dos Laplacianos de grafos isomorfos são iguais.

Como o Laplaciano de um grafo é uma matriz real, simétrica, e semidefinida positiva, todos os autovalores de $L(G)$ são maiores ou iguais a zero [35] e, pelo fato de que as colunas de $L(G)$ formam um conjunto linearmente dependente, pelo menos um autovalor de $L(G)$ é 0. Assim, $\alpha_n = 0$ e $\alpha_i \geq 0, \forall i = 0, 1, \dots, n - 1$.

Um resultado importante está na Proposição 2.1 e foi apresentado por Fiedler, em 1973 [15]. Trata-se da relação entre o valor de α_{n-1} e a conectividade do respectivo grafo. Esta diz que um grafo é conexo se e somente se $\alpha_{n-1} \neq 0$. Outra observação importante é feita em [41]: “(...) conjectura-se que quanto mais próximo de zero está a conectividade algébrica, mais vulnerável o grafo parece ser.”

Proposição 2.1 (Fiedler [15]). *Seja $a(G)$ a conectividade algébrica de um grafo G .*

Então G é conexo se e somente se $a(G) \neq 0$.

O Teorema 2.3, provado por Biggs em [5], relaciona a conectividade algébrica de um grafo G , não completo, com sua conectividade de vértices, $\kappa(G)$.

Teorema 2.3 (Biggs [5]). *Seja G um grafo não-completo. Então*

$$a(G) \leq \kappa(G).$$

Capítulo 3

Um novo invariante para o estudo de conectividade

A conectividade de vértices é um invariante importante no estudo de conectividade de grafos. Porém, apenas com esta medida não é possível gerar uma ordenação que possa distinguir grafos k -conexos. Por exemplo, nas famílias das árvores, das k -árvores e dos grafos bloco é necessário a análise de outras características dos grafos para conseguir diferenciá-los quanto à sua conectividade.

Neste capítulo é proposto o uso de um novo invariante de grafos que distingue grafos com a mesma conectividade de vértices. Para defini-lo consideram-se os separadores minimais de vértices (SMV) e o número de componentes conexas resultantes da remoção de cada um desses separadores. Como foi visto no Capítulo 2, existem algoritmos de tempo linear para encontrar os SMV 's de grafos cordais, veja [28, 33].

3.1 O novo invariante

Para o que se segue, seja $G = (V, E)$ um grafo cordal conexo com $|V| = n$ e $|E| = m$. Sejam $T(G)$ uma árvore de cliques desse grafo e $\mathcal{K}(G)$ o conjunto de cliques maximais de G . Sejam $\mathcal{S}(G)$ o conjunto de SMV 's de G e $n_c(G - S_i)$, o número de componentes conexas de $G - S_i$, $S_i \in \mathcal{S}(G)$. Observa-se que quanto maior o número de componentes conexas geradas com a remoção de um separador S_i mais relevante este será em relação à conectividade do grafo. Em uma rede, este conjunto mereceria um reforço maior contra ataques, pois sua destruição poderia causar danos maiores à rede. Por este motivo, a avaliação da conectividade do grafo deve começar pelo separador cuja remoção resulta no maior número de componentes conexas.

Definição 3.1. *Seja $[c_1, c_2, \dots, c_p]$ a sequência onde $c_1 \geq c_2 \geq \dots \geq c_p$, e $c_i = n_c(G - S_i)$, para $S_i \in \mathcal{S}(G)$, $i = 1, \dots, p$. Esta sequência é um invariante de grafos*

chamada de sequência de componentes conexas e é denotada por $SCC(G)$.

Para um dado i , a remoção de $S_i \in \mathcal{S}(G)$ de G causa a desconexão do mesmo. Para se determinar quantas componentes conexas são geradas com a remoção de S_i , deve-se considerar sua multiplicidade no grafo e a ocorrência de outros SMV 's distintos contidos em S_i . Quando o S_i removido contém outros SMV 's, estes também são removidos e o cálculo do número de componentes conexas resultantes deve considerar a multiplicidade de cada SMV distinto envolvido. O exemplo a seguir trata de tal caso. No decorrer do texto a multiplicidade de S_i será denotada por $\mu(S_i)$.

EXEMPLO: A Figura 3.1 ilustra um grafo cordal $G_1 = (V_1, E_1)$ e sua respectiva árvore de cliques $T(G_1)$. O conjunto de SMV 's de G_1 é $\mathcal{S}(G_1) = \{S_1 = \{b, d\}, S_2 = \{e, f\}, S_3 = \{b, d, e, f\}, S_4 = \{f, i\}, S_5 = \{i\}\}$. Cada SMV possui multiplicidade 1. Note que S_1, S_2 e S_5 não contêm nenhum outro SMV e a remoção de cada um gera duas componentes conexas. Já para S_3 e S_4 tem-se que $S_1 \subset S_3, S_2 \subset S_3$ e $S_5 \subset S_4$. Com isso, sabe-se que a remoção de S_3 acarreta na remoção de S_1 e S_2 e o número de componentes conexas resultantes é igual a 4. A remoção de S_4 também remove S_5 e resulta em três componentes conexas. A sequência de componentes conexas de G_1 é, portanto, $SCC(G_1) = [4, 3, 2, 2, 2]$.

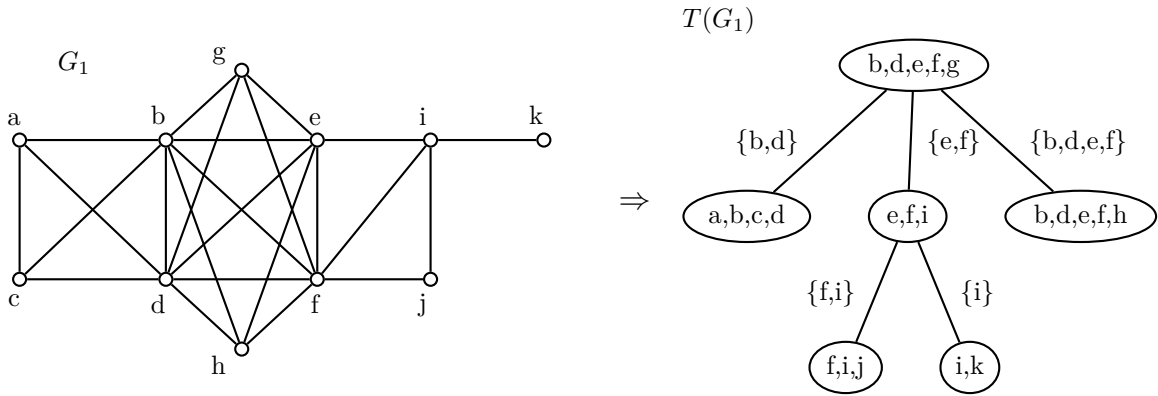


Figura 3.1: Grafo $G_1 = (V_1, E_1)$ com $n = 11$, $m = 24$ e $SCC(G_1) = [4, 3, 2, 2, 2]$.

No decorrer do trabalho será necessário realizar ordenações entre sequências de número inteiros. Para isso deve-se comparar as sequências duas a duas de forma a distinguir quando uma é maior que outra. Tal comparação é chamada de *comparação lexicográfica* e é definida a seguir.

Definição 3.2. Dadas duas cadeias t e s , é conhecido que t é precedida por s em ordem lexicográfica se:

- s é um prefixo de t ;
- se c e d são respectivamente os primeiros caracteres de t e s no qual t e s diferem, então c é precedido por d em ordem de caracteres.

Denote por \succ , a relação “lexicograficamente maior que” e por \prec , a relação “lexicograficamente menor que”. Por exemplo, considere duas sequências: $\psi_1 = [8, 8, 6, 5]$ e $\psi_2 = [8, 7, 6, 6]$, comparando-se elemento a elemento de cada sequência, observa-se que o primeiro elemento de ψ_1 é igual ao primeiro de ψ_2 , mas o segundo elemento de ψ_1 é maior que o segundo de ψ_2 . Logo, diz-se que ψ_1 é lexicograficamente maior que ψ_2 . Para mais um exemplo, considere a sequência $\psi_3 = [8, 8, 6]$, comparando ψ_1 com ψ_3 nota-se que ψ_3 é um prefixo de ψ_1 . Logo, $\psi_3 \prec \psi_1$. Comparando ψ_3 com ψ_2 , tem-se que o primeiro elemento de ambas as sequências é 8 e o segundo elemento de ψ_3 é maior que de ψ_2 , portanto $\psi_3 \succ \psi_2$.

Quando se trabalha com duas sequências de números ordenados pela ordem numérica usual em cada sequência, pode-se compará-las por meio da ordem lexicográfica acima definida e determinar qual delas é maior ou menor. O uso de *SCC*'s pode ser útil para comparar e classificar grafos com mesma conectividade de vértices, utilizando para isso a importância de seus *SMV*'s em sua conectividade.

Assim, dado um conjunto de grafos com mesma conectividade de vértices é possível gerar para cada grafo sua respectiva *SCC* e, em seguida, realizar uma ordenação lexicográfica. A partir deste ponto é possível classificar tais grafos quanto a sua conectividade, informação esta impossível de ser obtida apenas pela aplicação da conectividade de vértices. Cabe acrescentar que nem sempre é possível distinguir dois grafos pelas suas *SCC*'s, pois há grafos distintos com mesma *SCC*.

EXEMPLO: Considere dois grafos, com $n = 11$ vértices e $m = 24$ arestas, ilustrados na Figura 3.2 com suas respectivas árvores de cliques. Note que ambos possuem conectividade de vértices igual a 1. O grafo G_1 já foi tratado no exemplo anterior e sabe-se que $\mathcal{S}(G_1) = \{S_1 = \{b, d\}, S_2 = \{e, f\}, S_3 = \{b, d, e, f\}, S_4 = \{f, i\}, S_5 = \{i\}\}$ e $SCC(G_1) = [4, 3, 2, 2, 2]$. O grafo G_2 possui cinco cliques maximais e o seguinte conjunto de *SMV*'s, $\mathcal{S}(G_2) = \{S_1 = \{b, d\}, S_2 = \{e, f\}, S_3 = \{f, i\}, S_4 = \{f\}\}$. Em G_2 , tem-se $S_4 \subset S_2$ e $S_4 \subset S_3$ e, assim, a remoção de S_2 ou S_3 resulta em três componentes conexas cada. A remoção de S_1 ou S_4 gera duas componentes conexas. Dado que no grafo G_2 a *SCC* é $SCC(G_2) = [3, 3, 2, 2]$, é fácil verificar que $SCC(G_1) \succ SCC(G_2)$.

No exemplo visto, o número de componentes conexas resultantes da remoção de um *SMV* não depende apenas de sua multiplicidade. Ele é igual a soma das

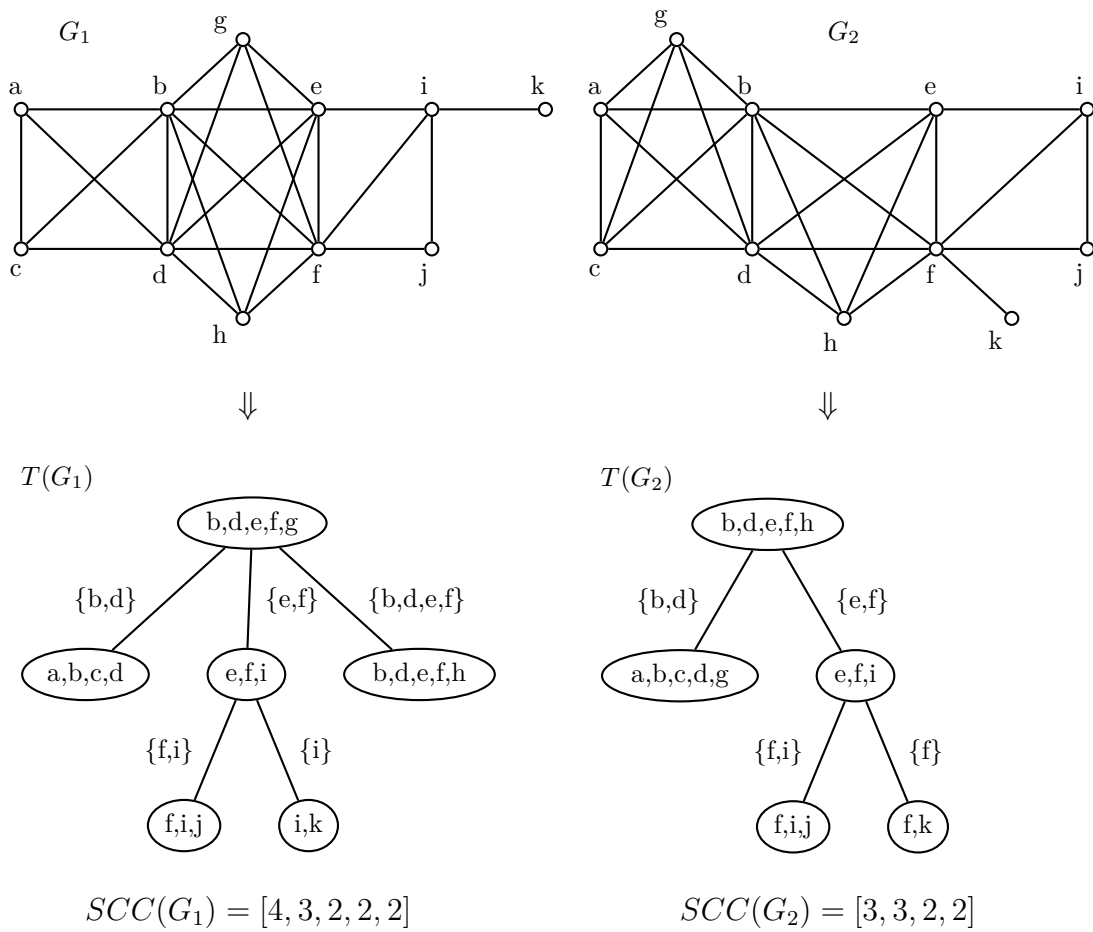


Figura 3.2: Árvores de cliques dos grafos G_1 e G_2 e suas respectivas SCC 's.

multiplicidades dos SMV 's envolvidos acrescido de uma unidade. Por exemplo, no caso de G_1 , com a remoção de S_3 , o número de componentes conexas resulta da soma das multiplicidades de S_1 , S_2 e S_3 mais 1. Quando um SMV não contém nenhum outro, esse cálculo se reduz a apenas o acréscimo de 1 à multiplicidade do SMV considerado, como acontece na remoção de S_1 de G_1 em que tal ação gera apenas duas componentes conexas.

Quando os SMV 's de um grafo possuem cardinalidades diferentes existe a possibilidade de um SMV conter outro. Neste caso, para calcular o número de componentes conexas resultantes da remoção de um SMV , deve-se identificar se há algum outro SMV contido nele. Para algumas classes de grafos cordais, todos os grafos possuem todos os SMV 's com mesma cardinalidade. Uma delas é a classe de *grafos cordais k -separados*. Esta foi definida em [27]. Em [40] e [39], alguns problemas foram resolvidos para o caso $k = 2$.

Definição 3.3. *Um grafo cordal G é chamado de grafo cordal k -separado (grafos cordais k -sep) se todo separador minimal de vértices de G possui, exatamente, cardinalidade igual a k .*

Da definição segue que para grafos cordais k -sep não há nenhum SMV que contenha outro, ou seja, para todo par S_i, S_j de SMV 's, $S_i, S_j \in \mathcal{S}$, $S_i \not\subseteq S_j$ e $S_j \not\subseteq S_i$. Assim, o número de componentes conexas $n_c(G - S_i) = \mu(S_i) + 1$. O exemplo a seguir ilustra a aplicação da SCC para grafos 3-sep.

EXEMPLO:

Sejam os grafos G_3 e G_4 da Figura 3.3. Ambos os grafos tem $q = 6$ cliques maximais. O conjunto de SMV 's do grafo G_3 é $\mathcal{S}(G_3) = \{S_1 = \{f, g, h\}, S_2 = \{d, e, f\}, S_3 = \{b, d, e\}, S_4 = \{a, b, c\}\}$, sendo suas multiplicidades iguais a $\mu(S_1) = 2$, $\mu(S_2) = \mu(S_3) = \mu(S_4) = 1$. Neste caso, como todos os SMV 's possuem a mesma cardinalidade, o número de componentes conexas obtidas após a remoção de S_i é $n_c(G_3 - S_i) = \mu(S_i) + 1$. Então, para se obter a SCC de G_3 , basta considerar o número de componentes conexas da remoção de cada S_i e ordenar tais valores de forma não crescente. Logo, $SCC(G_3) = [3, 2, 2, 2]$. De modo análogo, para o grafo G_4 tem-se $\mathcal{S}(G_4) = \{S_1 = \{f, g, h\}, S_2 = \{d, e, f\}, S_3 = \{b, d, e\}\}$ com as seguintes multiplicidades: $\mu(S_1) = 3$, $\mu(S_2) = \mu(S_3) = 1$. Isso resulta na $SCC(G_4) = [4, 2, 2]$. Ao se comparar lexicograficamente as duas SCC 's chega-se a $SCC(G_4) \succ SCC(G_3)$.

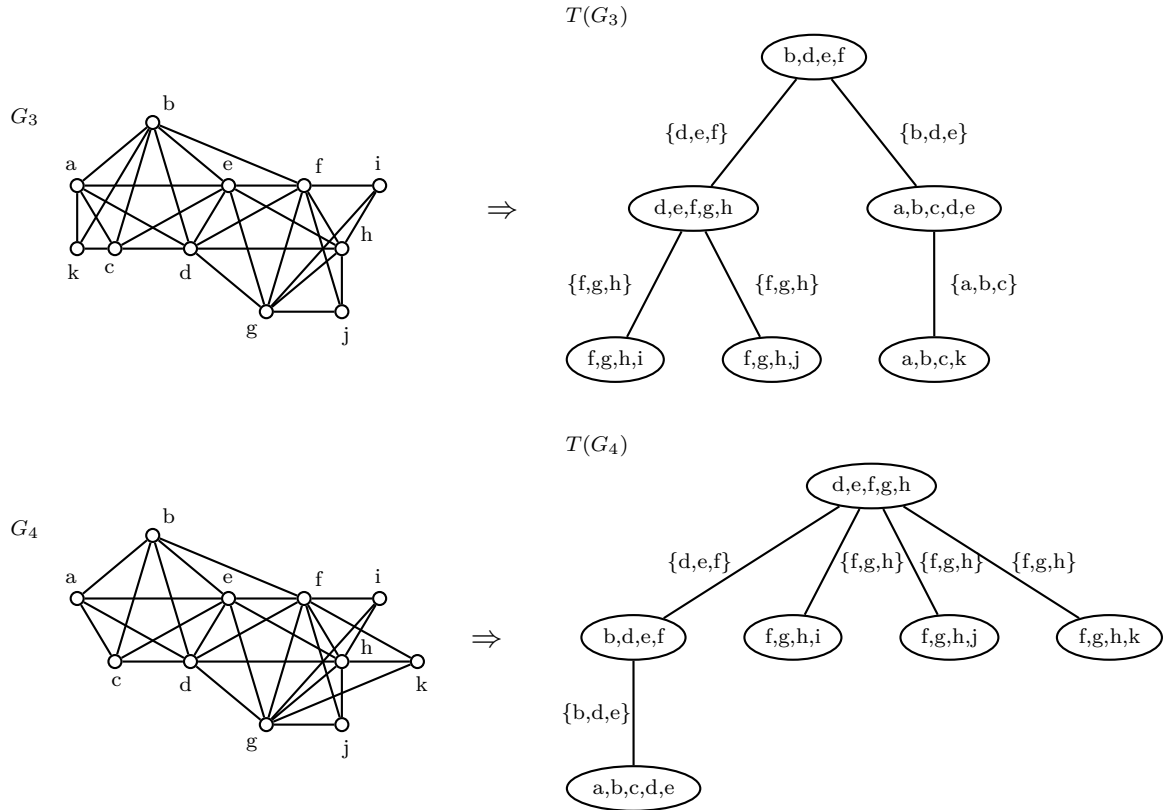


Figura 3.3: Grafos cordais 3-sep, G_3 e G_4 com suas respectivas árvores de cliques.

3.2 SCC 's para grafos cordais k -sep

Como foi visto no Capítulo 2, existe uma representação de árvore de cliques para os grafos cordais o que facilita a visualização das cliques maximais, das SMV 's e de suas respectivas multiplicidades. Há várias subclasses de grafos cordais k -sep muito estudados tais como árvores, k -árvores e grafos blocos.

Nesta seção será apresentado um algoritmo que gera todas as SCC 's possíveis para grafos k -sep, fixado o número de cliques maximais do grafo. Para a geração de todas as SCC 's de k -sep é necessário conhecer apenas o número de cliques maximais dos grafos do conjunto em questão, por exemplo, se G é um grafo k -sep então $q = |\mathcal{K}(G)|$ é o número de cliques maximais de G . As SCC 's serão geradas de acordo com o número de SMV 's distintos dos grafos, $|\mathcal{S}|$. Para um dado \mathcal{S} e um dado q , $1 \leq |\mathcal{S}| \leq q - 1$, o espaço de soluções viáveis é formado por todas as sequências SCC 's válidas possíveis.

Antes de apresentar o algoritmo, alguns pontos relacionados aos SMV 's e às cliques maximais devem ser analisados. Estes são derivados de resultados sobre grafos cordais conhecidos na literatura e provados em [6].

Considere o grafo 3-sep G_3 e sua árvore de cliques $T(G_3)$ apresentados na Figura 3.4. O conjunto de cliques maximais é $\mathcal{K}(G_3) = \{Q_1 = \{a, b, c, d, e\}, Q_2 = \{a, d, e, f\}, Q_3 = \{a, d, e, h\}, Q_4 = \{d, e, g, h\}\}$ e o conjunto de SMV 's é $\mathcal{S}(G_3) =$

$\{S_1 = \{a, d, e\}, S_2 = \{d, e, h\}\}$. Veja que a multiplicidade de S_1 é 2 e de S_2 é 1.

Nesse exemplo todos os SMV 's possuem cardinalidade 3 e, portanto, não há a relação de continência entre SMV 's distintos. O que pode ocorrer é um SMV aparecer mais de uma vez na árvore de cliques, ou seja, sua multiplicidade ser maior que 1, como é o caso de S_1 .

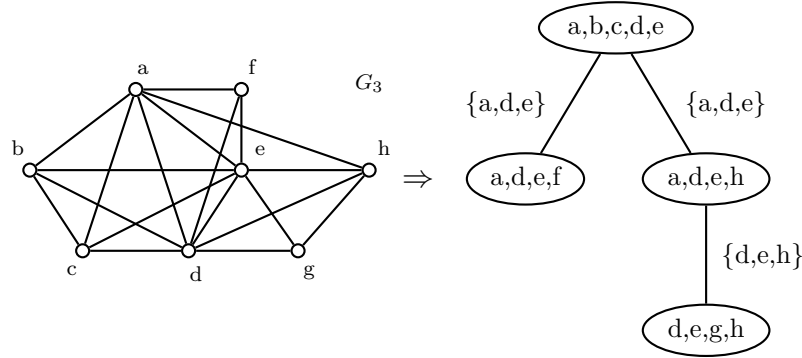


Figura 3.4: Grafo G_3 e sua respectiva árvores de cliques.

Definição 3.4. *Sejam G um grafo k -sep e $[\mu_1, \mu_2, \dots, \mu_p]$ tal que $\mu_1 \geq \mu_2 \geq \dots \geq \mu_p$, a seqüência de multiplicidades dos separadores mínimos de vértices de G . A tal seqüência dá-se o nome de seqüência de multiplicidades, e esta é denotada por $SM(G)$ ou simplesmente SM .*

Note que a cardinalidade de uma SM é igual ao número de SMV 's distintos existentes no grafo. Dado q cliques maximais, a quantidade mínima de SMV 's ocorre quando todas as cliques maximais compartilham o mesmo subconjunto de vértices cuja remoção gera q componentes conexas. Ou seja, o grafo possui apenas um SMV com multiplicidade igual a $q - 1$ e a cardinalidade da SM é 1. Seguindo raciocínio análogo, o número máximo de SMV 's em um grafo com q cliques maximais ocorre quando a multiplicidade de cada SMV é 1. Dessa forma, tem-se que a cardinalidade máxima de uma SM é $q - 1$.

Sabe-se que uma aresta de uma árvore de cliques de um grafo corresponde a um SMV . A multiplicidade de um SMV de um grafo é igual ao número de arestas que o representam na árvore de cliques do grafo. O número de arestas em uma árvore de cliques de um grafo com q cliques maximais é igual a $q - 1$. Assim, a soma das multiplicidades dos SMV 's de um grafo é sempre igual ao número de arestas da árvore de cliques, que é constante. A partir daí, é possível construir uma seqüência de multiplicidades elemento a elemento, de forma que para cada posição da seqüência são calculadas as possíveis multiplicidades do respectivo separador. Como consequência disso, é possível elaborar um algoritmo que gere todas as SM 's, dado o número de cliques maximais.

Como já foi mencionado, em grafos k -sep, o número de componentes conexas resultantes da remoção de um SMV é igual à sua multiplicidade acrescido de 1.

Com isso, uma *SCC* pode ser obtida a partir de uma *SM* válida, incrementando-se cada elemento da *SM* de 1. Desta forma, uma vez que se tem o conjunto de todas as *SM*'s válidas para grafos cordais *k*-sep com *q* cliques maximais é trivial a obtenção do respectivo conjunto de *SCC*'s. Ou seja, um algoritmo que obtenha todas as *SM*'s para grafos cordais *k*-sep com *q* cliques maximais, também obtém todas as *SCC*'s válidas para *q*. Para um dado *q* e um número fixo de *SMV*'s, denotado por $|\mathcal{S}|$, a Proposição 3.1 apresenta os valores máximos e mínimos de cada elemento de uma *SM*.

Proposição 3.1. *Sejam G um grafo cordal k -sep, \mathcal{S} o conjunto de *SMV*'s de G , $SM(G) = [\mu_1, \dots, \mu_{|\mathcal{S}|}]$ a sequência de multiplicidades dos *SMV*'s do grafo e q o número de cliques maximais. Tem-se:*

(i) *O valor mínimo para μ_i é:*

$$\mu_i = \begin{cases} \left\lfloor \frac{q-1}{|\mathcal{S}|} \right\rfloor, & i = 1; \\ \left\lfloor \frac{q-1 - \sum_{j=1}^{i-1} \mu_j}{|\mathcal{S}| - (i-1)} \right\rfloor, & \text{caso contrário.} \end{cases}$$

(ii) *O valor máximo para μ_i é:*

$$\mu_i = \begin{cases} q - |\mathcal{S}|, & i = 1; \\ \min \left\{ \left(q - 1 - \sum_{j=1}^{i-1} \mu_j \right) - (|\mathcal{S}| - i), \mu_{i-1} \right\}, & \text{caso contrário.} \end{cases}$$

PROVA: Dados q e $|\mathcal{S}|$, sabe-se que a soma dos valores dos $|\mathcal{S}|$ elementos de cada *SM* é constante e igual a $q - 1$.

(i) O menor valor possível para μ_1 é $\left\lfloor \frac{q-1}{|\mathcal{S}|} \right\rfloor$. Pela definição da sequência, sabe-se que a condição $\mu_1 \geq \mu_2 \geq \dots \geq \mu_{|\mathcal{S}|}$ deve ser satisfeita para que a sequência seja válida. Tem-se que o inteiro $\left\lfloor \frac{q-1}{|\mathcal{S}|} \right\rfloor$ pode resultar de dois casos:

- (a) quando $q - 1$ é divisível por $|\mathcal{S}|$;
- (b) quando $q - 1$ não é divisível por $|\mathcal{S}|$.

Suponha que $x \in \mathbb{N}$ seja tal que $q - 1 = x|\mathcal{S}| + r$ com $0 \leq r < |\mathcal{S}|$. No primeiro caso, tem-se que todos os elementos possuirão valor x . Se μ_1 tivesse um valor menor, como a soma das multiplicidades é $q - 1$, o valor de algum outro elemento da sequência deveria ser maior. Com isso, a ordenação da sequência não seria satisfeita e esta não seria válida.

No segundo caso, os r primeiros elementos devem ter valor $x + 1$ e os restantes terão valor x , para manter a soma das multiplicidades. Se μ_1 valesse menos, o

valor de outro elemento da sequência seria maior e a ordenação da sequência seria desfeita. Portanto, $\mu_1 = \left\lceil \frac{q-1}{|\mathcal{S}|} \right\rceil$ é o menor valor possível para μ_1 .

A prova de que o menor valor possível para o i -ésimo elemento da sequência é $\mu_i = \left\lceil \frac{q-1 - \sum_{j=1}^{i-1} \mu_j}{|\mathcal{S}| - (i-1)} \right\rceil$ é análogo à do limite inferior de μ_1 determinado em (i).

Neste caso, o total a ser dividido entre os elementos restantes da sequência é $q - 1 - \sum_{j=1}^{i-1} \mu_j$, ou seja, o total $(q - 1)$ subtraído da soma dos elementos já atribuídos na sequência $\left(\sum_{j=1}^{i-1} \mu_j \right)$. Esse valor deve ser distribuído da forma mais uniforme possível entre os $|\mathcal{S}| - (i - 1)$ elementos restantes da sequência.

Portanto, o menor valor possível para μ_i , $1 < i \leq |\mathcal{S}|$, é $\mu_i = \left\lceil \frac{q-1 - \sum_{j=1}^{i-1} \mu_j}{|\mathcal{S}| - (i-1)} \right\rceil$.

- (ii) Como uma SM é ordenada de forma não crescente, o maior valor possível para o primeiro elemento de uma SM ocorre quando todos os outros elementos da sequência possuem valor mínimo, que é igual a 1. Desta forma, $\mu_1 = (q - 1) - (|\mathcal{S}| - 1) = q - |\mathcal{S}|$. Sabe-se que uma sequência pode ser construída elemento a elemento com $i = 1, \dots, |\mathcal{S}|$. Assim, no momento do cálculo de μ_i , valor do i -ésimo elemento de SM , tem-se que todos os elementos que o antecedem na sequência já foram calculados $(\mu_1, \dots, \mu_{i-1})$. Resta então calcular os valores das posições restantes, cuja soma é $q - 1 - \sum_{j=1}^{i-1} \mu_j$. Observe que μ_i é o primeiro elemento da subsequência $(\mu_i, \mu_{i+1}, \dots, \mu_{|\mathcal{S}|})$. Para que μ_i tenha valor máximo, os elementos após μ_i na sequência devem assumir o valor mínimo. Ou seja, a soma dos elementos após μ_i é $|\mathcal{S}| - i$. Dessa forma, temos $\mu_i = (q - 1 - \sum_{j=1}^{i-1} \mu_j) - (|\mathcal{S}| - i)$. Porém, para que a sequência seja válida, o critério de ordenação, $\mu_1 \geq \mu_2 \geq \dots \geq \mu_k$, deve ser satisfeito. Assim, o valor de μ_i não pode ultrapassar o valor de seu antecessor, μ_{i-1} . Portanto, o valor máximo é $\mu_i = \min \left\{ (q - 1 - \sum_{j=1}^{i-1} \mu_j) - (|\mathcal{S}| - i), \mu_{i-1} \right\}$. ■

3.3 Geração das SCC 's em grafos cordais k -sep

Sabe-se que a cardinalidade de uma sequência SM de um grafo G é dada por \mathcal{S} . Da Proposição 3.1, para uma sequência SM , sua respectiva SCC é obtida incrementando-se cada um de seus elementos de uma unidade. Com tal resultado, o algoritmo *geraSequencias*, aplicado a um valor q , gera todas as sequências SM 's possíveis para cada cardinalidade $|\mathcal{S}|$, tal que $|\mathcal{S}| = 1, \dots, q - 1$. Para cada SM gerada é obtida sua respectiva SCC . Esta é guardada em um conjunto Ψ , que inicialmente é vazio.

A Proposição 3.1 é a base para a elaboração do algoritmo *geraSequencias*. Na construção de uma SM , para cada posição i , $i = 1, \dots, |\mathcal{S}|$, são considerados valores c_i que obedecem os limites inferior e superior apresentados pela Proposição 3.1. Assim, para uma posição $1 \leq i \leq |\mathcal{S}|$, são considerados valores inteiros que variam do limite máximo, definido pelo item (ii) da Proposição 3.1, até o limite mínimo, definido pelo item (i) da Proposição 3.1. Após a atribuição de um valor para uma posição i , o processo passa para a análise da próxima posição da sequência e quando a última posição é preenchida, tem-se uma SM completa.

Algoritmo 1: geraSequencias

<p>Entrada: q: número de cliques maximais</p>

<p>Saída: Ψ: conjunto de SCC's</p>

<pre> 1 $\Psi \leftarrow \emptyset$; 2 para $\mathcal{S} = 1, \dots, q - 1$ faça 3 $pos \leftarrow 0$; 4 $SM \leftarrow \emptyset$; 5 $geraseq(SM, pos, q, \mathcal{S})$; </pre>

O procedimento *geraseq* constrói uma SM de forma recursiva. Ele recebe uma SM parcial e, para a posição trabalhada naquela chamada, são calculados os valores máximo e mínimo possíveis de acordo com a Proposição 3.1 (linhas 1 a 6). Em seguida é verificado se a posição corrente é a última da sequência. Caso positivo, o valor obtido em *max* é atribuído a essa posição e tem-se uma SM completa. Caso não seja a última posição, atribui-se à posição corrente da sequência os valores possíveis para tal posição, obedecendo o intervalo de máximo e mínimo calculados no início do procedimento. Para cada valor atribuído, é realizada uma chamada recursiva para trabalhar a próxima posição da SM (linhas 12 a 15).

Quando o Procedimento *geraseq* atribui o valor do elemento $c_{|\mathcal{S}|}$, tem-se uma SM completa e obtém-se a partir desta uma SCC válida, incluindo-a em Ψ (linhas 7 a 11).

Os Lemas 3.2 e 3.1 mostram que para qualquer SCC gerada pelo algoritmo, existe ao menos um grafo k -sep que o realiza e, dado um grafo k -sep com q cliques maximais, sua SCC pertence ao conjunto de SCC 's geradas.

Lema 3.1. *Seja um grafo cordal G com q cliques maximais e sua respectiva $SCC(G)$. Seja Ψ o conjunto com todas as SCC 's geradas pelo Algoritmo *geraSequencias* aplicado a q . Tem-se que existe $S' \in \Psi$ tal que $S' = SCC(G)$.*

PROVA: É sabido que a $SM(G)$ tem seus elementos ordenados de forma não crescente e que a $SCC(G)$ pode ser obtida a partir de sua sequência $SM(G)$. Assim, o valor para cada elemento da $SM(G)$ deve satisfazer os itens da Proposição 3.1.

Procedimento $geraseq(SM, pos, Q, |\mathcal{S}|)$

Entrada: SM : seqüência de multiplicidades parcial, pos : posição de SM a ser trabalhada, q : quantidade de cliques maximais, $|\mathcal{S}|$: quantidade de SMV's

```
/* Caso pos seja 1, somaAnteriores será zero. */
1 somaAnteriores  $\leftarrow \sum_{i=1}^{pos-1} SM[i]$ ;
2  $max \leftarrow q - 1 - somaAnteriores - (|\mathcal{S}| - pos)$ ;
3  $min \leftarrow \left\lceil \frac{q-1-somaAnteriores}{|\mathcal{S}|-(pos-1)} \right\rceil$ ;
4 se  $pos > 1$  então
    /* O valor máximo para multiplicidade do SMV na posição pos deve ser o
       menor entre max e SM[pos - 1] para que a seqüência seja válida. */
5     se  $SM[pos - 1] < max$  então
6     [  $max \leftarrow SM[pos - 1]$ ;
7 se  $pos = |\mathcal{S}|$  então
    /* Se pos for a última posição, então basta atribuir o valor de max
       para esta posição */
8      $SM[pos] \leftarrow max$ ;
    /* Obtenção de uma SCC a partir da SM obtida. */
9     para  $j \leftarrow 1, \dots, |\mathcal{S}|$  faça
10    [  $SM[j] \leftarrow SM[j] + 1$ ;
11     $\Psi \leftarrow \Psi \cup \{SM\}$ ;
12 senão
13    para  $i \leftarrow max, \dots, min$  faça
14    [  $SM[pos] \leftarrow i$ ;
15    [  $geraseq(SM, pos + 1, Q, |\mathcal{S}|)$ ;
```

Como o Algoritmo *geraSequencias* obtém todas as *SM*'s que satisfazem a Proposição 3.1, e a partir de cada *SM* gerada é obtida sua respectiva *SCC* que é guardada em Ψ . Conclui-se que existe algum $S \in \Psi$ tal que $S = SCC(G)$. ■

Lema 3.2. *Sejam k a cardinalidade dos *SMV*'s, q o número de cliques maximais e Ψ o conjunto contendo todas as *SCC*'s geradas pelo Algoritmo *geraSequencias*. Para cada $SCC \in \Psi$, existe ao menos um grafo cordal k -sep que a realiza.*

PROVA: Dado o número de cliques maximais q , para cada cardinalidade $|\mathcal{S}|$, $1 \leq |\mathcal{S}| \leq q - 1$, o algoritmo *geraSequencias* gera todas as *SCC*'s possíveis (que atendem a Propriedade 3.1) com $|\mathcal{S}|$ elementos. Considere k a cardinalidade dos *SMV*'s dos grafos. Dessa forma, as cliques maximais devem ter cardinalidade no mínimo igual a $k + 1$. Seja $S = [c_1, c_2, \dots, c_{|\mathcal{S}|}]$ uma *SCC* gerada por *geraSequencias*. É possível construir um grafo k -sep a partir de S seguindo os passos:

- (1) Considere uma clique maximal Q_1 com $k + 1$ vértices e $c_1 - 1$ vértices isolados. Escolha um subconjunto S_1 de k vértices de Q_1 . Ligue cada vértice independente a todos os vértices de S_1 . Desta forma, obtém-se G^1 o primeiro subgrafo com um *SMV* denotado por S_1 com multiplicidade c_1 .
- (2) Agora, considere o grafo G^1 obtido em (1) e mais $c_2 - 1$ vértices isolados. Escolha uma clique de G^1 com k vértices, diferente de S_1 , denote-o por S_2 . Ligue cada um dos $c_2 - 1$ vértices isolados a todos os vértices de S_2 . Tem-se um subgrafo G^2 com dois *SMV*'s S_1 e S_2 de multiplicidades c_1 e c_2 , respectivamente. Repita o processo para cada c_i da seqüência. Ou seja, considere o grafo G^{i-1} , obtido do passo anterior e $c_i - 1$ vértices isolados. Escolha um subconjunto S_i de G^{i-1} com k vértices, diferente dos S_j já escolhidos, onde $1 \leq j \leq i - 1$, e ligue cada vértice independente a todos os vértices de S_i . Assim, o grafo G^i é obtido. Este processo se repete até que $c_{|\mathcal{S}|}$ seja tratado.

Ao tratar do elemento $c_{|\mathcal{S}|}$, tem-se um grafo cordal que realiza a *SCC* considerada. Com isso, pode-se concluir que, para qualquer *SCC* gerada pelo algoritmo *geraSequencias*, sempre existirá ao menos um grafo cordal que a realiza. Além disso, pelo processo descrito nos itens (1) e (2) acima, sabe-se que o grafo gerado é uma k -árvore. Portanto, ao menos um grafo que realiza a *SCC* é uma k -árvore. ■

Exemplo: Dados $q = 5$, $k = 3$ e $S = [3, 2, 2]$. A Figura 3.5 ilustra o passo a passo do processo de obtenção de um grafo 3-sep com $q = 5$ cliques maximais e uma *SCC* igual a $S = [3, 2, 2]$.

O processo inicia com uma clique de cardinalidade igual a 4 e dois vértices isolados e e f . O conjunto $S_1 = \{b, c, d\}$ é escolhido para ser o *SMV* referente ao

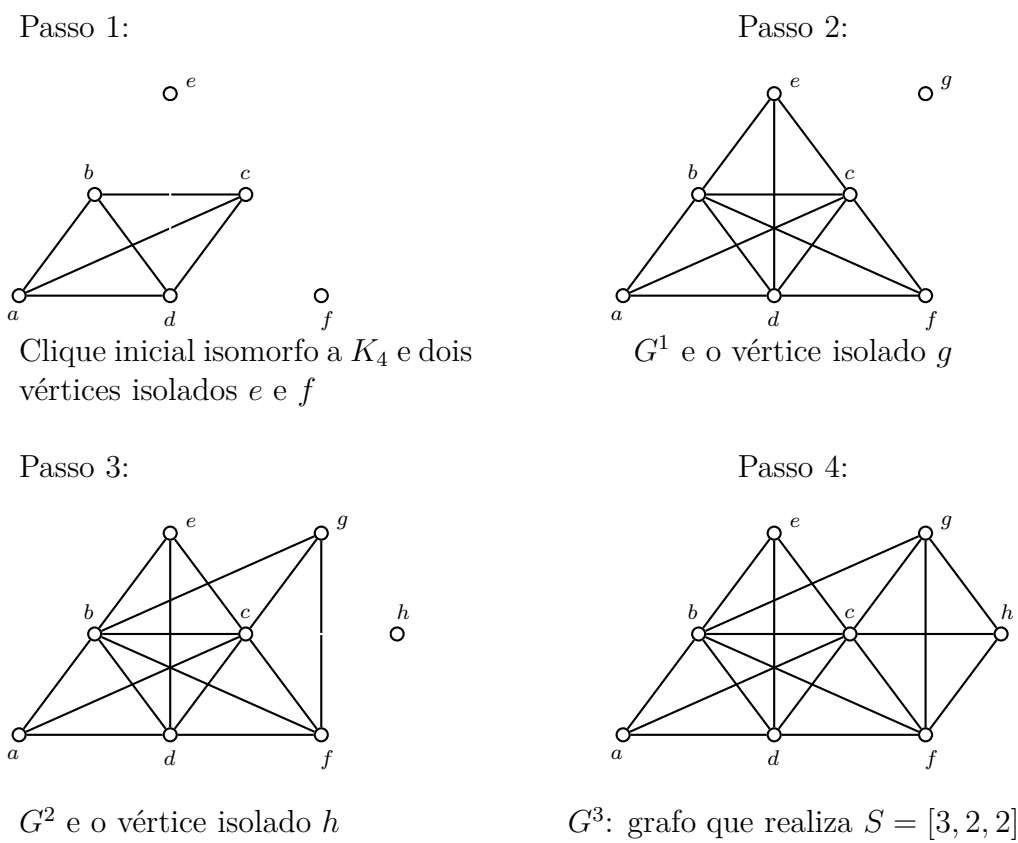


Figura 3.5: Obtenção de um grafo 3-sep a partir de $S = [3, 2, 2]$.

elemento $c_1 = 3$ de S . Os vértices e e f são ligados a todos os vértices de S_1 . Desta forma, obtém-se G^1 . No passo 2, considera-se G^1 o grafo obtido do passo 1 e um vértice isolado g . Escolhe-se o conjunto $S_2 = \{b, c, f\}$ para ser o SMV referente a $c_2 = 2$ em S . Liga-se g a todos os vértices de S_2 . Por fim, considera-se o grafo G^2 obtido do passo 3 e um vértice isolado h , escolhendo-se o conjunto $S_3 = \{c, f, g\}$ para ser o SMV referente a $c_3 = 2$ de S . O vértice h é ligado a todos os vértices de S_3 para se obter o grafo G^3 . Como, todos os elementos de S foram trabalhados, o algoritmo termina e tem-se que G^3 é o grafo final. Observa-se que este realiza S .

Uma SCC pode ser realizada por mais de um grafo. No exemplo, escolheu-se ligar o vértice h à clique $\{c, f, g\}$, formando a clique maximal $\{c, f, g, h\}$. Essa escolha não é única, poder-se-ia escolher ligar o vértice h à clique $\{a, b, d\}$, obtendo-se assim a clique maximal $\{a, b, d, h\}$. O grafo resultante G^4 não é isomorfo ao grafo G^3 obtido no exemplo e possui a mesma SCC . A Figura 3.6 apresenta os dois grafos descritos.

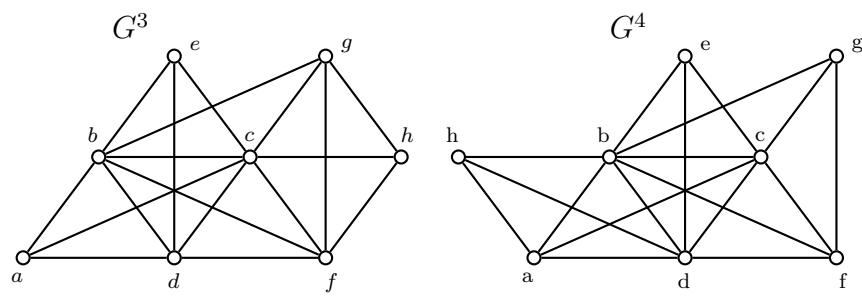


Figura 3.6: Grafo G^3 do exemplo anterior e o grafo G^4 não isomorfo a G_3 que também realiza a $SCC = [3, 2, 2]$.

Capítulo 4

Conectividade em árvores

No capítulo anterior foi apresentado um algoritmo que gera todas as *SCC*'s para grafos cordais k -sep, ou seja, grafos em que todos os *SMV*'s possuem igual cardinalidade. Neste capítulo será estudado um caso particular de grafos k -sep, as árvores, e será feito uma análise do comportamento do novo invariante e da conectividade algébrica aplicadas a árvores com mesmo número de vértices.

4.1 Introdução

Nas árvores todos os *SMV*'s possuem cardinalidade 1 e todas as cliques maximais possuem cardinalidade 2. Consequentemente, o número de componentes conexas geradas pela remoção de um vértice interno é dado pelo seu grau. Com isso, a *SCC* de uma árvore equivale a sequência de graus de seus vértices internos ordenados de forma não crescente. No decorrer deste trabalho, tal sequência será chamada de *sequência de graus internos* e denotada por *SIG*.

Ao se fazer uma análise do ponto de vista de defesa de sistemas modelados por grafos, por exemplo, se duas redes modelam um sistema e estas possuem configurações distintas, aquela que possuir menos pontos vulneráveis representa uma rede mais confiável.

Como já foi visto no Capítulo 3, o número de componentes conexas resultantes da remoção de um separador minimal de vértices num grafo pode indicar a importância do separador em questão sob o ponto de vista da conectividade do grafo. Numa dada árvore T , é possível associar cada situação causada pela remoção de um vértice interno como um “dano” causado à rede modelada por T . Portanto, quanto maior for o dano causado pela remoção de um vértice interno, mais importante este parece ser com relação à conectividade da árvore. Um exemplo que ilustra bem este ponto são os casos extremos de árvores, estrelas e caminhos. No primeiro caso, há apenas um vértice vulnerável enquanto que para os caminhos todos os $n - 2$ vértices internos desconectam o grafo.

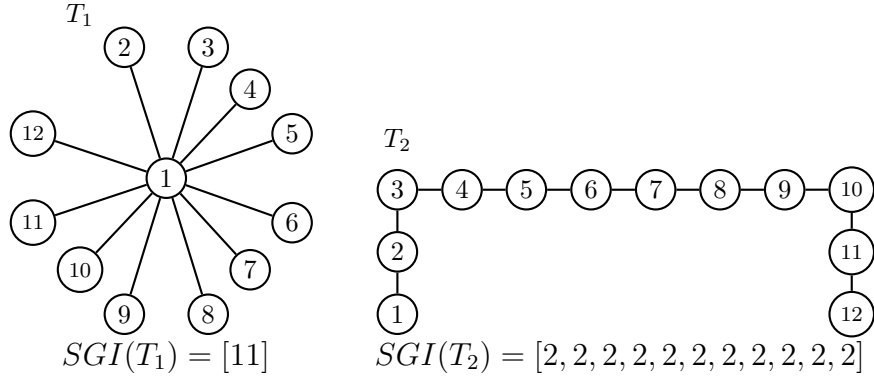


Figura 4.1: SGI 's para casos extremos em árvores.

A Figura 4.1 ilustra os casos extremos de árvores com 12 vértices. Nesse exemplo, observa-se que, segundo a conectividade do grafo, o impacto causado pela remoção de um vértice interno de T_1 (um SMV de T_1) é muito maior que a remoção de qualquer vértice interno (SMV) de T_2 . Assim, sob o ponto de vista de um sistema cuja rede tenha a configuração de estrela é necessário concentrar a vigilância da mesma em apenas um ponto, enquanto que, se a rede tiver a configuração do caminho, deve-se colocar vigilância nos $n - 2$ pontos intermediários do caminho.

EXEMPLO: Considere as árvores T_3 e T_4 apresentadas na Figura 4.2. Os conjuntos de SMV 's de cada árvore são $\mathcal{S}(T_3) = \{S_1 = \{2\}, S_2 = \{3\}, S_3 = \{10\}, S_4 = \{1\}\}$ e $\mathcal{S}(T_4) = \{S_1 = \{3\}, S_2 = \{2\}, S_3 = \{4\}\}$. Para cada S_j , o número de componentes conexas resultante de sua remoção é dado pelo grau $d(S_j)$. Assim,

- para T_3 : $n_c(T_3 - S_1) = d(S_1) = 6$, $n_c(T_3 - S_2) = d(S_2) = 3$, $n_c(T_3 - S_3) = d(S_3) = 3$ e $n_c(T_3 - S_4) = d(S_4) = 2$ e
- para T_4 : $n_c(T_4 - S_1) = d(S_1) = 9$, $n_c(T_4 - S_2) = d(S_2) = 2$ e $n_c(T_4 - S_3) = d(S_3) = 2$.

E as SGI 's para T_3 e T_4 são dadas por $SGI(T_3) = [6, 3, 3, 2]$ e $SGI(T_4) = [9, 2, 2]$, respectivamente.

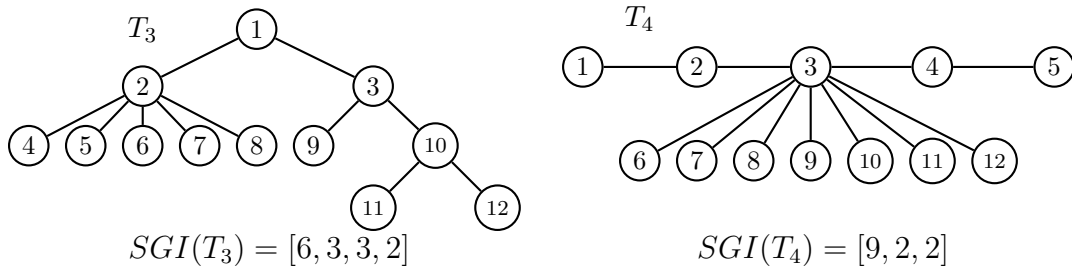


Figura 4.2: Exemplos de SGI .

Nas árvores da Figura 4.2, observa-se que a remoção do vértice 2 em T_3 resulta em seis componentes conexas, sendo que cinco delas são vértices isolados e a outra, uma subárvore induzida de ordem seis. Enquanto que em T_4 , a remoção do vértice 3 resulta em nove componentes conexas, sendo duas de ordem 2 e sete vértices isolados. Com isso, tem-se $SGI(T_4) = [9, 2, 2] \succ SGI(T_3) = [6, 3, 3, 2]$.

4.2 Características da SGI

Para as árvores das Figuras 4.2 e 4.1 tem-se $SGI(T_1) = [11]$, $SGI(T_2) = [2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]$, $SGI(T_3) = [6, 3, 3, 2]$ e $SGI(T_4) = [9, 2, 2]$. Desta forma, verifica-se que $SGI(T_1) \succ SGI(T_4) \succ SGI(T_3) \succ SGI(T_2)$.

Como visto no Capítulo 2, a conectividade algébrica $a(G)$ parece ser um dos invariantes de grafos mais precisos para se medir a conectividade dos mesmos. Há várias ordenações de árvores em classes especiais associadas a este invariante espectral. Veja para isso as referências [1, 19, 45, 49, 50].

Também já é sabido que, para um caminho P_n , $a(P_n) = 2(1 - \cos(\frac{\pi}{n}))$, veja [15]. Ao analisar a conectividades algébrica para as árvores da Figura 4.1 e 4.2 verifica-se que $a(T_1) = 1$, $a(T_2) = 0,06815$, $a(T_3) = 0,13503$, $a(T_4) = 0,38197$. Dessa forma, a ordenação das quatro árvores quanto tal invariante é: $a(T_1) > a(T_4) > a(T_3) > a(T_2)$. Uma observação importante é que a conectividade algébrica e a SGI apresentaram ordenações isomorfas para as árvores dos exemplos. No decorrer do trabalho, a comparação entre esses dois invariantes será aprofundada a fim de verificar se os dois invariantes produzem sempre a mesma ordenação.

O algoritmo *geraSequencias* apresentado no Capítulo 3 pode ser aplicado às árvores já que estas são grafos 1-sep. A Tabela 4.1 apresenta todas as sequências geradas pelo algoritmo para árvores com $n = 8$ vértices, ou seja, $q = 7$ cliques maximais.

A Figura 4.3 apresenta todas as árvores existentes de oito vértices, a menos de isomorfismo. Abaixo de cada árvore tem-se sua respectiva SGI . Os grafos estão ordenados lexicograficamente de forma não crescente de acordo com suas respectivas SGI 's. As árvores foram nomeadas T_i , $1 \leq i \leq 23$, iniciando-se do grafo de maior SGI para o de menor SGI . Como já foi observado no Capítulo 3, uma SCC não representa unicamente um grafo. Da mesma forma, uma SGI também não é realizada por uma única árvore. Por exemplo, a sequência $[5, 2, 2]$ é realizada pelas árvores T_4 e T_5 e estas não são isomorfas. Observa-se que cada uma das sequências $[4, 3, 2]$ e $[3, 3, 2, 2]$ são realizadas por várias árvores diferentes. Por exemplo, a primeira é realizada por T_7 , T_8 e T_9 e a segunda por T_{14} , T_{15} , T_{16} , T_{17} e T_{18} .

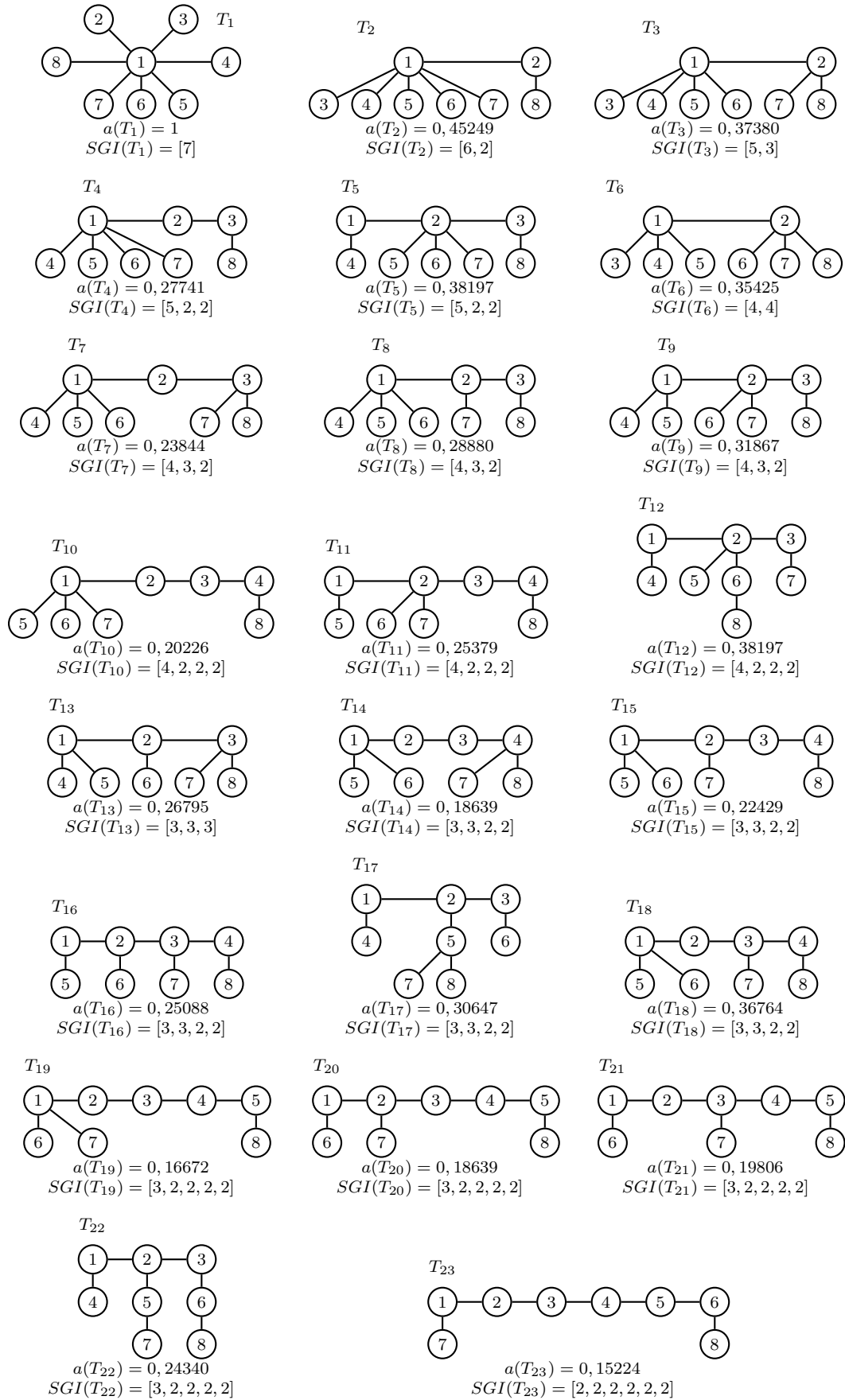


Figura 4.3: Aplicação da nova medida para árvores com 8 vértices.

p	SGI
1	$SGI = [7]$
2	$SGI = [6, 2]$
	$SGI = [5, 3]$
	$SGI = [4, 4]$
3	$SGI = [5, 2, 2]$
	$SGI = [4, 3, 2]$
	$SGI = [3, 3, 3]$
4	$SGI = [4, 2, 2, 2]$
	$SGI = [3, 3, 2, 2]$
5	$SGI = [3, 2, 2, 2, 2]$
6	$SGI = [2, 2, 2, 2, 2, 2]$

Tabela 4.1: Sequências de componentes conexas para as árvores com 8 vértices.

O processo de construção de uma árvore que realiza uma dada SGI é ilustrado na Figura 4.4. No exemplo, é gerada uma árvore com $n = 10$ vértices, a partir de sua sequência de graus internos $SGI = [5, 4, 2]$. Tal processo utiliza a mesma idéia da prova do Lema 3.2.

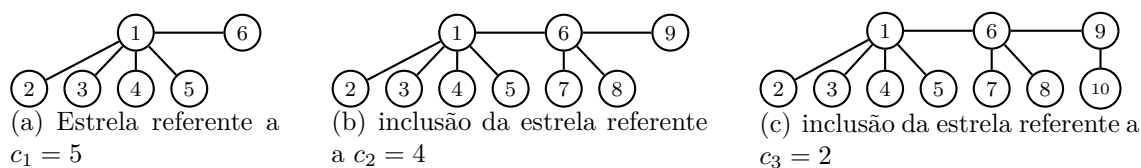


Figura 4.4: Construção de uma árvore que realiza $SGI = [5, 4, 2]$.

Na literatura existe o estudo de ordenação de subclasses de árvores pela conectividade algébrica. Uma dessas classes são os *caterpillars*, veja [43]. Um *caterpillar* é uma árvore em que a remoção de todos os vértices pendentes resulta em um caminho.

Proposição 4.1. *Para toda SGI , pelo menos uma árvore que realiza tal sequência é um caterpillar.*

PROVA: No processo ilustrado pela Figura 4.4, pode-se escolher, em cada passo, uma folha mais recentemente inserida na árvore para ligar às novas folhas. Dessa forma, o diâmetro da árvore resultante é igual à cardinalidade da SGI acrescida de 1 e retirando-se todas as suas folhas, obtém-se um caminho. Logo, para toda SGI , pelo menos uma árvore que realiza tal sequência é um *caterpillar*. ■

Outro ponto interessante a ser observado é a possibilidade de simplificar a representação de uma SGI . Uma SGI pode ter de 1 a $n - 2$ elementos. Computacionalmente, é necessário uma estrutura adequada para armazenar essa informação (um vetor ou uma lista encadeada). O processo dado a seguir atribui a cada SGI um número inteiro, o que facilitará o seu armazenamento. Para isso faça:

- (i) Ordene as sequências SGI 's em ordem lexicográfica crescente;
- (ii) Atribua o valor i ao i -ésimo elemento da sequência ordenada.

Com a nova representação dispensa-se a utilização de uma estrutura de dados mais elaborada para armazenamento de uma SGI no computador, diminuindo o consumo de memória, além de simplificar sua representação em tabelas para comparação de invariantes. Porém, o formato original das SGI 's não pode ser dispensado, pois este possui as informações das árvores que realizam tais sequências.

EXEMPLO: Considere as sequências da Tabela 4.1 geradas pelo algoritmo *geraSequencias*. O processo de transcrição descrito acima é apresentado na Figura 4.5.

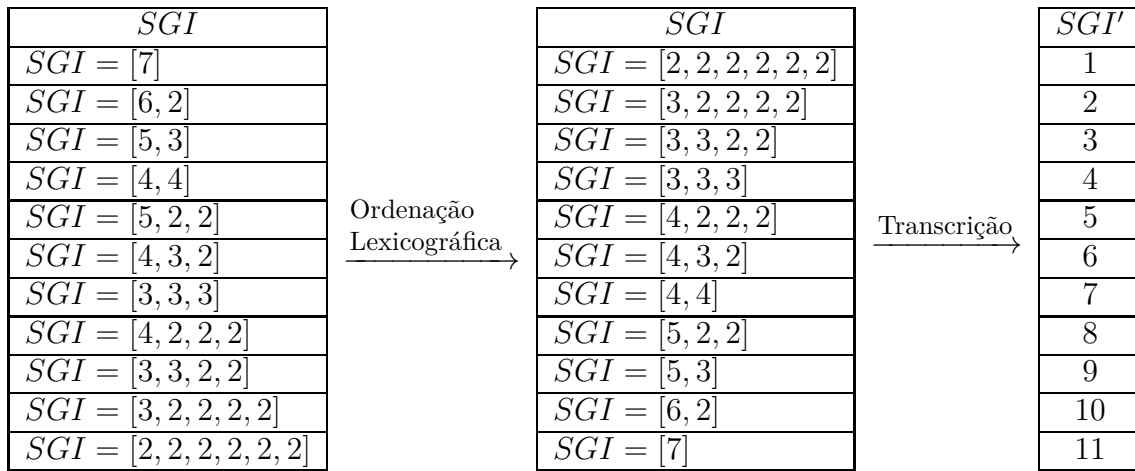


Figura 4.5: Transcrição para números inteiros das SCC 's de árvores com 8 vértices.

4.3 Comparação entre SGI e Conectividade Algébrica em algumas classes de árvores

Nenhum dos invariantes de grafos, mesmo para a classe das árvores, é capaz de sozinho distinguir todos os elementos da classe com relação às suas respectivas conectividades. Entretanto, há artigos referentes à ordenação (não necessariamente estrita) de árvores em classes especiais via conectividade algébrica, que ordenam todas as árvores pertencentes às respectivas classes. Além disso, o cálculo da conectividade algébrica demanda a tarefa de encontrar zeros da função característica do laplaciano do grafo que é uma tarefa não trivial e, na literatura, somente algoritmos aproximativos para resolver este problema são conhecidos, ou seja, na maioria das vezes, somente são encontrados valores não exatos para a conectividade algébrica de um grafo. Como uma SGI de uma árvore é facilmente obtida com precisão,

vale a pena examinar o comportamento desses dois invariantes quando aplicadas às árvores.

Sejam as árvores de diâmetro 3, denotadas por $T_2(k, \ell)$, definidas em [19], onde $k, \ell \in \mathbb{N}$, $1 \leq k \leq \ell$, são os respectivos números de folhas pendentes em cada um dos dois vértices internos. Tais árvores têm $n = k + \ell + 2$ vértices.

Em [19] foi provado que, para uma árvore $T_2(k, \ell)$, seu laplaciano $L(T_2(k, \ell))$ possui exatamente um autovalor menor que 1 que é a sua conectividade algébrica $a_k = a(T_2(k, \ell))$. Além disso, para $1 \leq k \leq (n - 2)/2$, a_k é uma função estritamente decrescente em k e define uma ordem total nas árvores da classe $T_2(k, \ell)$.

A Figura 4.6 apresenta todas as árvores possíveis $T_2(k, \ell)$ com $n = 12$ vértices. Para cada uma delas é apresentado o valor da conectividade algébrica e a sua respectiva SGI . Observa-se que a primeira medida decresce com o aumento de k , satisfazendo a ordenação dada em [19]. Veja que a ordenação das SGI 's apresenta o mesmo comportamento da ordenação dada pelas conectividades algébricas, ou seja, define uma ordenação decrescente em função de k .

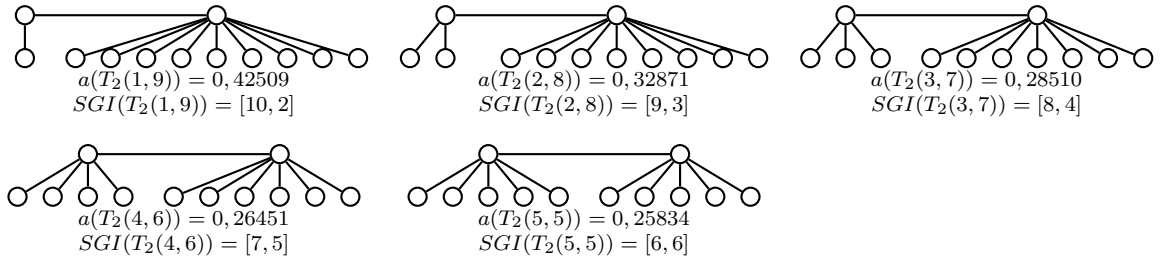


Figura 4.6: Árvores $T_2(k, \ell)$ com $n = 12$ vértices.

Em [12], é apresentada outra família de árvores que abrange mais casos do que a família tratada acima. A família que segue generaliza a anterior para qualquer árvore de diâmetro $d \geq 3$.

Definição 4.1. *Seja $T_d(k, \ell)$, com $1 \leq k \leq \ell$, uma árvore não valorada com n vértices construída a partir de um caminho P_d com d vértices rotulados de 1 até d e adicionando-se k vértices pendentes ao vértice 1 e ℓ vértices pendentes ao vértice d .*

A Figura 4.7 ilustra a estrutura de uma árvore $T_d(k, \ell)$ com diâmetro $d + 1$. O Teorema 4.1 é provado em [12] e exibe a árvore com diâmetro $d + 1$ que minimiza a conectividade algébrica. Similarmente ao caso anterior de árvores $T_2(k, \ell)$, o menor valor da conectividade algébrica é alcançado quando o número de folhas adjacentes aos vértices extremos do caminho P_d são iguais ou um difere do outro de uma unidade.

Teorema 4.1 ([12]). *Dentre todas as árvores não valoradas com n vértices e diâmetro $d + 1$, o menor valor de conectividade algébrica é obtido por $T_d\left(\left\lceil \frac{n-d}{2} \right\rceil, \left\lfloor \frac{n-d}{2} \right\rfloor\right)$.*

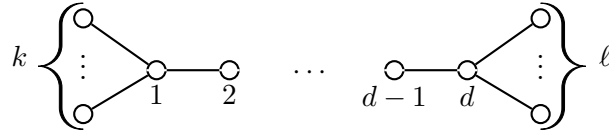


Figura 4.7: Estrutura de uma árvore $T_d(k, \ell)$.

Na Figura 4.8 são ilustradas todas as árvores $T_d(k, \ell)$ com $d = 4$ e $n = 18$ vértices. Para cada árvore são exibidos tanto as sequências SGI 's quanto os respectivos valores das conectividades algébricas.

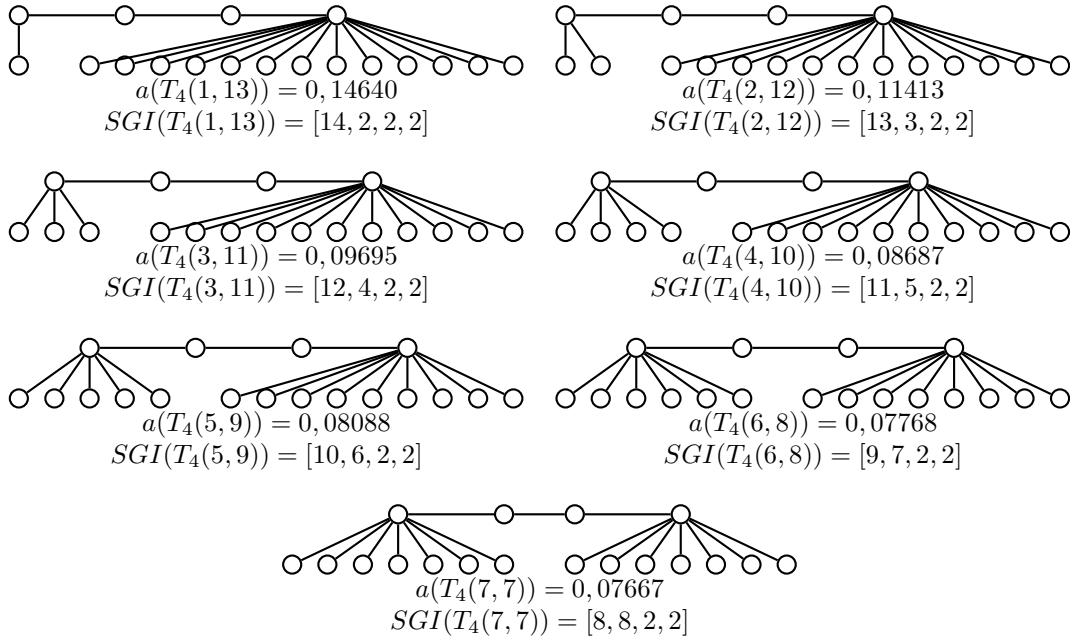


Figura 4.8: Árvores $T_d(k, l)$ com 18 vértices.

Note que a conectividade algébrica decresce com o aumento de k . O mesmo ocorre com a SGI . Ou seja, tanto a SGI quanto a conectividade algébrica determinam a mesma ordenação.

Em [49], são introduzidas 6 novas classes de árvores, denotadas por C_i com $1 \leq i \leq 6$. Os autores provam que $a(T_i) > a(T_j)$ se $T_i \in C_i$, $T_j \in C_j$ e $1 \leq i < j \leq 6$. Lá, são utilizadas as seguintes estruturas de árvores $K_{1,2}$, $K_{1,1}$ e K_1 , apresentadas na Figura 4.9.

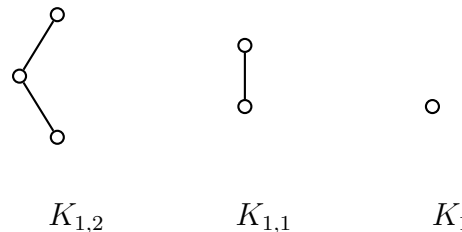


Figura 4.9: Estruturas $K_{1,2}$, $K_{1,1}$ e K_1 .

Uma árvore pertencente a uma classe C_i , $1 \leq i \leq 6$, é formada por um vértice

v adjacente a k estruturas $K_{1,2}$, p estruturas $K_{1,1}$ e q estruturas K_1 . Tal árvore é denotada por $T(k, p, q)$. O valor de i que caracteriza as classes aparece formalmente na definição que segue e a Figura 4.10 apresenta a forma de uma árvore pertencente a alguma de tais classes.

Definição 4.2. *Sejam n, k, p e q inteiros não negativos tais que $3k + 2p + q = n - 1$. Seja $T(k, p, q)$ a árvore de ordem n que contém um vértice v tal que $T(k, p, q) - v = kK_{1,2} \cup pK_{1,1} \cup qK_1$.*

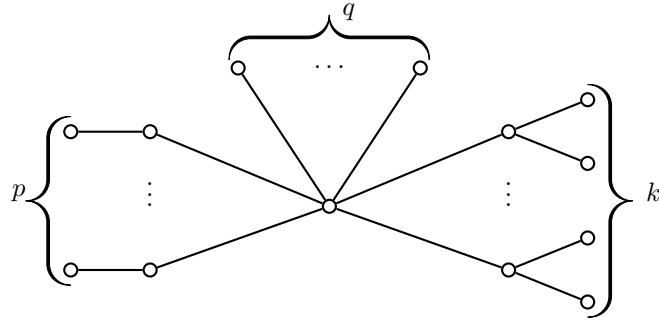


Figura 4.10: Estrutura das árvores pertencentes às classes C_i , $1 \leq i \leq 6$.

As seis classes C_i , $i = 1, \dots, 6$, de árvores de ordem n são dadas por:

- $C_1 = \{T(0, 0, n - 1)\}$;
- $C_2 = \{T(0, 1, n - 3)\}$;
- $C_3 = \{T(0, p, q) | p \geq 2, 2p + q = n - 1\}$;
- $C_4 = \{T(1, 0, n - 4)\}$;
- $C_5 = \{T(1, p, q) | p \geq 1, 2p + q = n - 4\}$;
- $C_6 = \{T(k, p, q) | k \geq 2, 3k + 2p + q = n - 1\}$;

As classes C_1 , C_2 e C_4 possuem apenas uma árvore cada uma e estas são apresentadas na Figura 4.11.

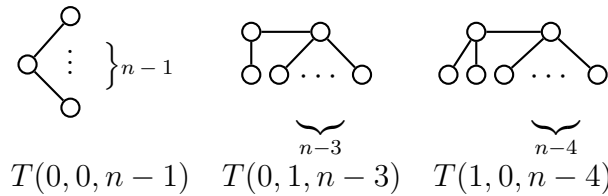


Figura 4.11: Árvores das classes C_1 , C_2 e C_4 .

Em [49], também é mostrado que toda árvore $T \in C_i$, $1 \leq i \leq 6$, possui conectividade algébrica $a(T) \geq 2 - \sqrt{3}$, valendo a igualdade apenas quando $T \in C_6$. Naquele mesmo trabalho também é apresentado o Teorema 4.2.

Teorema 4.2 ([49]). *Se T é uma árvore de ordem $n \geq 15$ com $a(T) \geq 2 - \sqrt{3}$, então $T \in \cup_{i=1}^6 C_i$.*

O Teorema 4.2 nem sempre vale para $n < 15$. Exemplos de árvores com $n = 8$ vértices que possuem $a(T) > 2 - \sqrt{3}$ e não pertencem a nenhuma das seis classes são T_4, T_6, T_8 e T_{18} , ilustradas na Figura 4.3.

O resultado seguinte mostra uma ordenação total e estrita das árvores da classe C_5 quanto à conectividade algébrica dada em função do valor de p em $T(p) = T(1, p, n - 2p - 4)$.

Lema 4.1 ([49]). *Para um dado n , se $T(p) = T(1, p, n - 2p - 4)$ é uma árvore de ordem n cuja conectividade algébrica é denotada por a_p então $a_{p-1} > a_p$, para $1 \leq p \leq \lfloor (n - 4)/2 \rfloor$.*

O Teorema 4.3 faz uma classificação entre as conectividades algébricas das árvores dessas seis classes, resultando numa ordenação total entre elas. Além disso, o teorema exibe o valor da conectividade algébrica de cada árvore em cada classe.

Teorema 4.3 ([49]). *Para $n \geq 8$ e $1 \leq i \leq 6$, seja T_i uma árvore da classe C_i com n vértices. Então, tem-se que:*

$$(1) \ a(T_1) > a(T_2) > a(T_3) > a(T_4) > a(T_5) > a(T_6);$$

$$(2) \ a(T_1) = 1;$$

$$a(T_2) \text{ é a menor raiz da equação: } x^3 - (n + 2)x + (3n - 2)x - n = 0;$$

$$a(T_3) = \frac{3 - \sqrt{5}}{2};$$

$$a(T_4) \text{ é a menor raiz da equação: } x^3 - (n + 2)x + (4n - 7)x - n = 0;$$

$$a(T_5) \text{ é a menor raiz da equação:}$$

$$x^5 - (n - p + 5)x^4 + (7n - 7p)x^3 - (14n - 13p - 19)x^2 + (7n - 3p - 7)x - n = 0 \text{ e}$$

$$a(T_6) = 2 - \sqrt{3}.$$

(3) *Finalmente, se T é uma árvore de ordem $n \geq 15$ e $T \notin \cup_{i=1}^6 C_i$ então $a(T) < a(T_6) = 2 - \sqrt{3}$.*

É possível determinar a SGI de cada $T(k, p, q) \in \cup_{i=1}^6 C_i$ diretamente de sua representação. Numa árvore $T(k, p, q)$ com $n \geq 8$, os elementos k, p e q representam as quantidades de cada uma das estruturas $K_{1,2}, K_{1,1}$ e K_1 , isto é, tal árvore tem k vértices de grau 3, p vértices de grau 2 e q vértices de grau 1, sendo tais estruturas ligadas a um vértice v cujo grau é igual a $k + p + q$. Logo, $T(k, p, q)$ tem $k + p + 1$ vértices internos, onde as q estruturas K_1 são suas folhas. Dessa forma, dados $n \geq 8$ e $T(k, p, q) \in \cup_{i=1}^6 C_i$ tem-se $SGI(T(k, p, q)) = [k + p + q, \underbrace{3, \dots, 3}_k, \underbrace{2, \dots, 2}_p]$.

Segue, explicitamente, que,

$$\begin{aligned}
T(0, 0, n - 1) &= [n - 1]; \\
T(0, 1, n - 3) &= [n - 2, 2]; \\
T(0, p, n - 2p - 1) &= [n - p - 1, \underbrace{2, 2, \dots, 2}_p]; \\
T(1, 0, n - 4) &= [n - 3, 3]; \\
T(1, p, n - 2p - 4) &= [n - p - 3, 3, \underbrace{2, 2, \dots, 2}_p]; \\
T(k, p, n - 3k - 2p - 1) &= [n - 2k - p - 1, \underbrace{3, 3, \dots, 3}_k, \underbrace{2, 2, \dots, 2}_p].
\end{aligned}$$

Para as árvores $T_i \in \cup_{i=1}^6 C_i$, a Proposição 4.2 apresenta uma relação de ordem entre árvores de classes C_i distintas segundo suas respectivas SGI 's. Para isso, é considerado que todas as árvores possuem o mesmo número de vértices n .

Proposição 4.2. *Para $i = 1, \dots, 6$, seja $T_i \in C_i$. Então tem-se que:*

1. $SGI(T_1) \succ SGI(T_2) \succ SGI(T_4)$;
2. $SGI(T_4) \succ SGI(T_3)$;
3. $SGI(T_4) \succ SGI(T_5)$;
4. $SGI(T_4) \succ SGI(T_6)$.

PROVA:

1. Sabe-se que cada classe C_1 , C_2 e C_4 tem apenas um elemento cada uma e é imediato verificar que $SGI(T_1) \succ SGI(T_2) \succ SGI(T_4)$.
2. É fácil ver que para $p = 2$, $SGI(T_3) = [n - 4, 2, 2]$. Como $SGI(T_4) = [n - 3, 3]$, então $SGI(T_4) \succ SGI(T_3)$.
3. De modo análogo é possível comparar os elementos de C_5 e C_6 via SGI 's. A árvore $T_5 \in C_5$ com a maior SGI possui $p = 1$ e $T_6 \in C_6$ com a maior SGI possui $k = 2$ e $p = 0$. Assim, $SGI(T_5) = [n - 4, 3, 2]$ e $SGI(T_6) = [n - 5, 3, 3]$. Conseqüentemente, $SGI(T_4) \succ SGI(T_5)$ e $SGI(T_4) \succ SGI(T_6)$.

■

Observe que entre árvores de classes diferentes o número de estruturas $K_{1,2}$, $K_{1,1}$ e K_1 não se repetem. Em uma mesma classe de árvores a representação $T(k, p, q)$ é realizada por apenas uma árvore, logo não há repetições de SGI 's entre árvores de mesma classe. Portanto, fixando n e uma vez obtidas as SGI 's de todas as

árvores na união das seis classes, é possível obter uma ordenação total estrita para tais árvores, algo que não é possível via conectividade algébrica. Tal resultado é enunciado na seguinte proposição.

Proposição 4.3. *Para n fixo, as sequências de graus internos das árvores de ordem n das classes C_i , $1 \leq i \leq 6$, definem uma ordenação total estrita.*

PROVA: Sabe-se que C_1 , C_2 e C_4 são classes unitárias em que a SGI de cada uma de suas respectivas árvores são distintas das árvores das outras classes. Deste modo, resta provar que as SGI 's dos elementos de C_3 , C_5 e C_6 são também distintas entre si.

Sejam $T_i = T(k_i, p_i, q_i)$ e $T_j = T(k_j, p_j, q_j) \in C_3$. Sabe-se que, para $p \geq 2$, $SGI(T_i) = [n - p_i - 1, \underbrace{2, \dots, 2}_{p_i}]$. Logo, $SGI(T_i) = SGI(T_j) \Rightarrow p_i = p_j$. Daí, como $n = 2p + q + 1$, segue que $q_i = q_j$ e as árvores são idênticas. Portanto, todas as árvores de C_3 possuem SGI 's diferentes. A prova é análoga para $T_i, T_j \in C_5$ e $T_i, T_j \in C_6$.

Agora, basta mostrar que árvores pertencentes a diferentes classes possuem diferentes SGI 's. Sejam $T_i = T(k_i, p_i, q_i) \in C_3$, $T_j = T(k_j, p_j, q_j) \in C_5$ e $T_\ell = T(k_\ell, p_\ell, q_\ell) \in C_6$

Sabe-se que

$$\begin{aligned} SGI(T_i) &= [n - p_i - 1, \underbrace{2, \dots, 2}_{p_i}], \\ SGI(T_j) &= [n - p_j - 3, 3, \underbrace{2, \dots, 2}_{p_j}] \text{ e} \\ SGI(T_\ell) &= [n - 2k_\ell - p_\ell - 1, \underbrace{3, \dots, 3}_{k_\ell}, \underbrace{2, \dots, 2}_{p_\ell}], \quad k_\ell \geq 2. \end{aligned}$$

A árvore T_i não pode ter a mesma SGI que uma árvore pertencente a C_5 ou C_6 , pois em T_i não há nenhum vértice interno de grau 3 que é um vértice obrigatório para as árvores das outras duas classes.

A árvore T_j não pode ter a mesma SGI que uma árvore pertencente a C_6 , pois T_j possui exatamente um vértice interno de grau 3, enquanto que para ser da classe C_6 é obrigatório a árvore ter ao menos dois vértices com tal característica.

Portanto, o invariante SGI fornece uma ordenação estrita para árvores pertencentes às classes C_i , $i = 1, \dots, 6$. ■

Os exemplos a seguir ilustram os resultados apresentados para árvores das classes C_i , $i = 1, \dots, 6$, com 15 vértices. Para tais árvores será analisado o comportamento da SGI e da conectividade algébrica para cada classe C_i .

EXEMPLO: Sejam T_1 , T_2 e T_4 as árvores das classes C_1 , C_2 e C_4 , respectivamente. A ordenação para tais árvores via conectividade algébrica é $a(T_1) > a(T_2) > a(T_4)$.

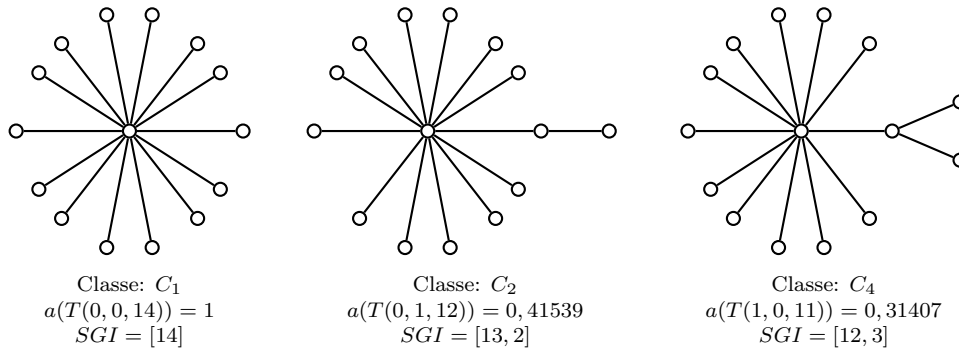


Figura 4.12: Árvores com $n = 15$ vértices para as classes C_1 , C_2 e C_4 .

EXEMPLO: Do Teorema 4.3 tem-se que todas as árvores de C_3 possuem conectividade algébrica igual a $\frac{3-\sqrt{5}}{2}$. Na Figura 4.13 ilustram-se todas as árvores possíveis de C_3 com 15 vértices e para cada uma é apresentada sua respectiva SGI . Da Proposição 4.3, sabe-se que todas as árvores de C_3 possuem SGI 's distintas o que fornece uma ordenação estrita, como é ilustrado pela Figura 4.13.

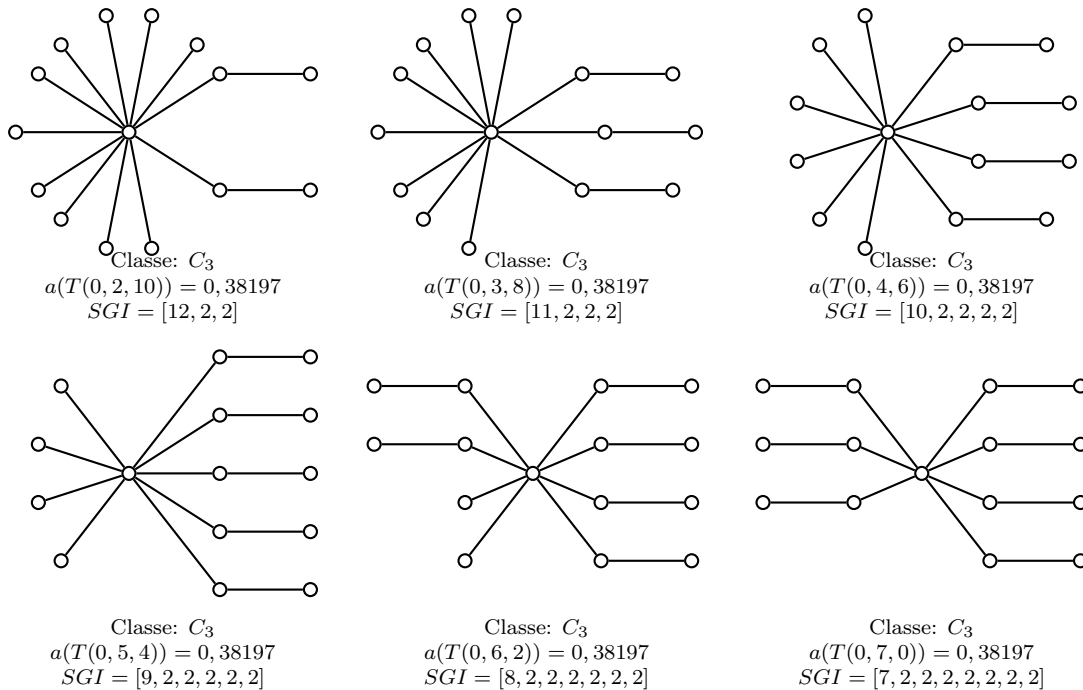


Figura 4.13: Árvores de C_3 com 15 vértices.

EXEMPLO: O Lema 4.1 prova que a ordenação via conectividade algébrica das árvores de C_5 é estrita. Na Figura 4.14 são apresentadas todas as árvores com 15

vértices dessa classe. Tais elementos estão dispostos em ordem decrescente de suas respectivas conectividades algébricas e para cada árvore é apresentada sua *SGI*. Observe que, na Figura 4.14, a ordenação das árvores obtida pela *SGI* é a mesma daquela obtida pela conectividade algébrica.

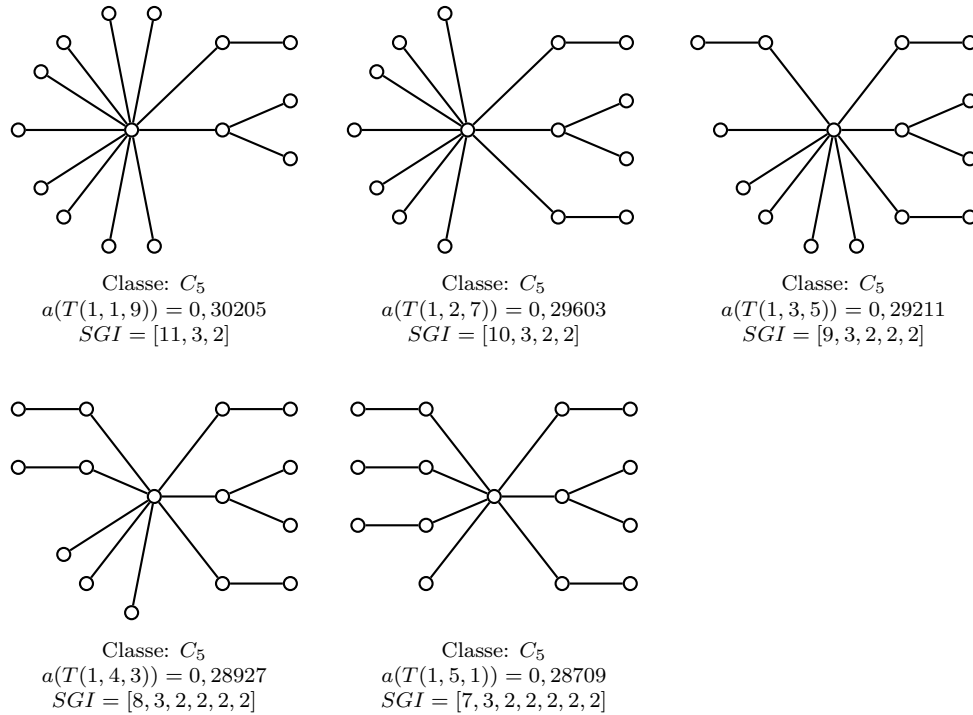


Figura 4.14: Árvores de C_5 com 15 vértices.

EXEMPLO: Veja que, pelo Teorema 4.3, todas árvores de C_6 possuem a mesma conectividade algébrica, $a(T) = 2 - \sqrt{3}$. No entanto, suas respectivas *SGI*'s definem uma ordenação total estrita, veja a Figura 4.15.

Das árvores com 15 vértices ilustradas nas Figuras 4.12, 4.13, 4.14 e 4.15, observe-se que a ordenação obtida pela *SGI* não é isomorfa à obtida pela conectividade algébrica. Veja que a ordenação dada pelas respectivas *SGI*'s das árvores $T'_3 = T(0, 2, 10) \in C_3$, $T''_3 = T(0, 7, 0) \in C_3$, $T'_5 = T(1, 1, 9) \in C_5$, $T''_5 = T(1, 5, 1) \in C_5$, $T'_6 = T(2, 0, 8) \in C_6$ e $T''_6 = T(2, 4, 0) \in C_6$ é a seguinte,

$$SGI(T'_3) \succ SGI(T'_5) \succ SGI(T'_6) \succ SGI(T''_5) \succ SGI(T''_3) \succ SGI(T''_6),$$

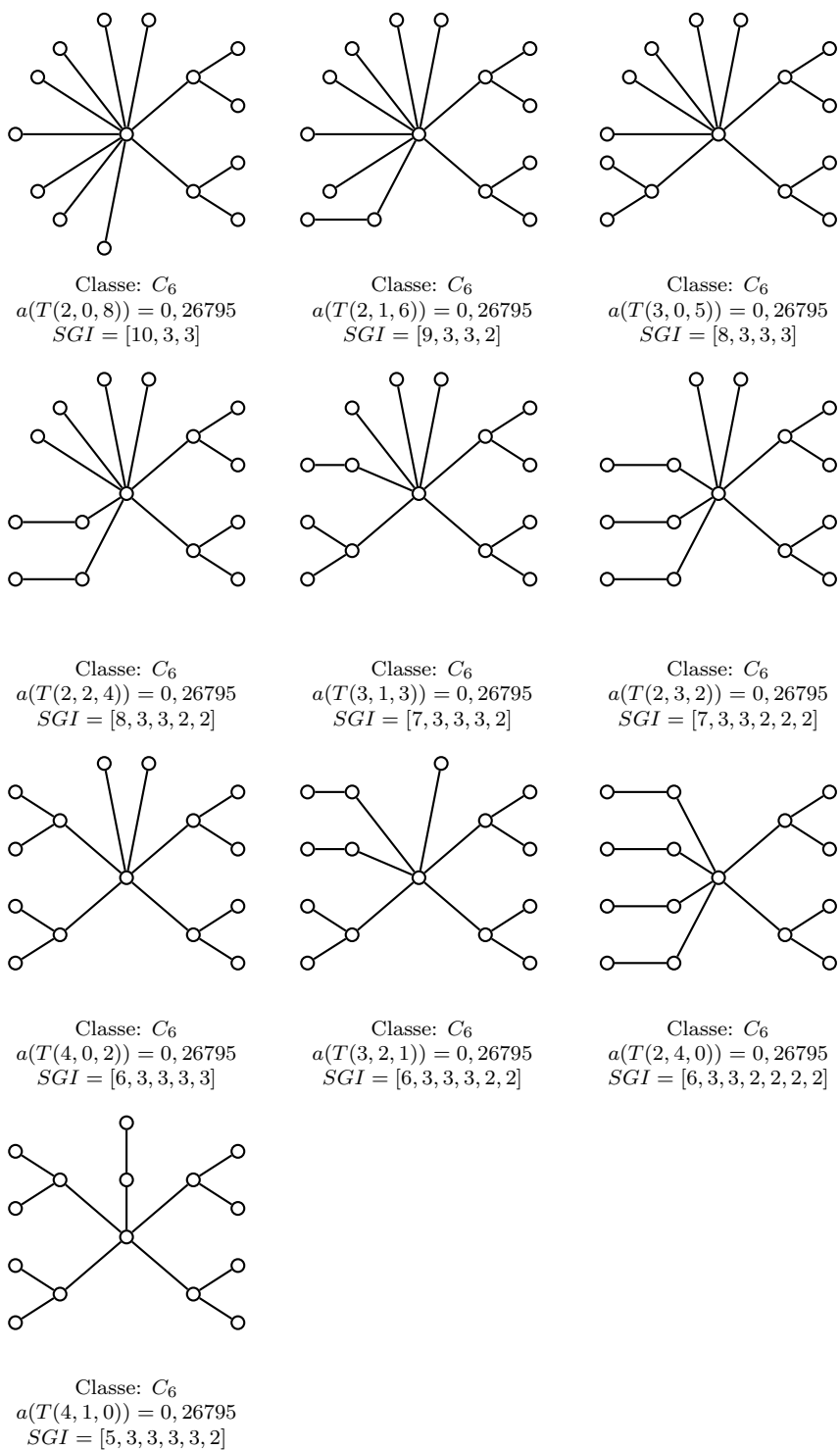


Figura 4.15: Árvore de C_6 com 15 vértices.

e a dada pela conectividade algébrica é igual a

$$a(T'_3) = a(T''_3) > a(T'_5) > a(T''_5) > a(T'_6) = a(T''_6).$$

4.4 SGI e Conectividade Algébrica em árvores com diâmetro 4

Note que as árvores das seis classes apresentadas por [49] possuem diâmetro menor ou igual a 4. Um estudo mais abrangente sobre árvores com diâmetro pequeno que maximizam a conectividade algébrica foi realizado por [48]. Em tal trabalho, os autores continuaram o estudo realizado em [49], apresentando árvores com $n > 45$ vértices e com conectividade algébrica no intervalo $\left[\frac{5-\sqrt{21}}{2}, 2 - \sqrt{3}\right)$.

A subclasse de árvores de diâmetro 4, definida por [50], é apresentada a seguir.

Definição 4.3. *Seja p_1, p_2, \dots, p_k inteiros tais que*

$$p_1 \geq p_2 \geq \dots \geq p_k \geq 0, \quad p_1 \geq p_2 > 0, \quad k \geq 2 \text{ e } k + 1 + p_1 + \dots + p_k = n.$$

Seja $T(n, k, p_1, p_2, \dots, p_k)$ uma árvore com diâmetro 4, c o único vértice central da árvore, $\text{Adj}(c) = \{v_1, v_2, \dots, v_k\}$ e $d(v_1) = p_1 + 1, d(v_2) = p_2 + 1, \dots, d(v_k) = p_k + 1$. Seja \mathbb{T} a classe de todas as árvores $T(n, k, p_1, \dots, p_k)$.

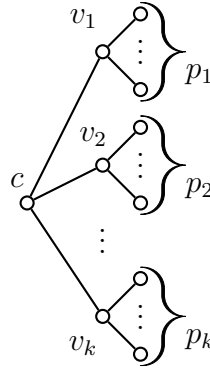


Figura 4.16: Estrutura de uma árvore $T(n, k, p_1, \dots, p_k)$.

Em [48], novas subclasses de árvores foram definidas. São elas:

- $C'_1 = \{S(3, n - 5)\}$, a árvore de ordem n obtida pela ligação do centro de $K_{1,3}$ ao centro de $K_{1,n-5}$ por uma aresta;
- $C'_2 = \{T(n, k, 3, 1, p_3, \dots, p_k)\}$;
- $C'_3 = \{T(n, k, 3, 2, p_3, \dots, p_k)\}$;

- $C'_4 = \{T(n, k, 3, 3, p_3, \dots, p_k)\}$.

Note que C'_1 não é uma subclasse de \mathbb{T} . Wang e Tan [48] provaram alguns resultados interessantes sobre essas classes:

Proposição 4.4 ([48]). *Seja T uma árvore de ordem $n \geq 45$. Se $T \notin \{\cup_{i=1}^6 C_i\} \cup \{\cup_{i=1}^4 C'_i\}$, então $a(T) < \frac{5-\sqrt{21}}{2}$.*

Teorema 4.4 ([48]). *Se T é uma árvore de ordem $n \geq 45$, então $\frac{5-\sqrt{21}}{2} \leq a(T) \leq 2 - \sqrt{3}$ se e somente se $T \in \cup_{i=1}^4 C'_i$.*

Com a *SGI* alguns resultados interessantes podem ser obtidos para as árvores $T \in \cup_{i=1}^4 C'_i$.

Definição 4.4. *Seja \mathbb{T}' uma subclasse de \mathbb{T} , onde $T(n, k, p_1, \dots, p_k) \in \mathbb{T}'$, se e somente se $k > p_1$, isto é, \mathbb{T}' é o conjunto de todas as árvores de diâmetro 4 de \mathbb{T} cujo vértice central da árvore possui o maior grau.*

É possível determinar a *SGI* de uma árvore $T(n, k, p_1, \dots, p_z, \dots, p_k) \in \mathbb{T}'$ diretamente de sua notação. Seja p_z , $2 \leq z \leq k$, o último elemento não nulo da notação. Então

$$SGI(T) = [k, p_1 + 1, p_2 + 1, \dots, p_z + 1].$$

Lema 4.2. *Dada uma sequência $S = [n, k, p_1, p_2, \dots, p_k]$, existe uma única árvore $T(n, k, p_1, p_2, \dots, p_k) \in \mathbb{T}'$ que a realiza.*

PROVA: Sejam $T(n, k, p_1, p_2, \dots, p_k) \in \mathbb{T}'$ e S a sequência $[n, k, p_1, p_2, \dots, p_k]$. É imediato concluir que S é uma sequência não crescente. A sequência $SGI(T)$ é construída a partir de S e seus elementos obedecem exatamente a ordem original. Então, dado S , existe uma única árvore com esta sequência como sua *SGI*. ■

Teorema 4.5. *Para um dado n , as *SGI*'s das árvores da classe \mathbb{T}' com n vértices determinam uma ordem total estrita em \mathbb{T}' .*

PROVA: Do Lema 4.2 sabe-se que não há duas árvores não isomorfas em \mathbb{T}' com mesma *SGI*. Como foi visto no Capítulo 3, é possível obter uma ordenação lexicográfica das *SGI*'s das árvores da classe \mathbb{T}' . Portanto, é possível obter uma ordenação total e estrita das árvores em \mathbb{T}' pela *SGI*. ■

Através do estudo das seis classes apresentadas em [49] percebeu-se algumas características estruturais de árvores que possuem mesmo valor de conectividade algébrica. Por exemplo, as árvores das classes C_3 e C_6 . Tais características serão estudadas no próximo capítulo.

Capítulo 5

A classe das medusas

No Capítulo 4 foi estudado o comportamento da SGI e da conectividade algébrica para seis classes de árvores de diâmetro máximo igual a 4 que maximizam a conectividade algébrica, apresentadas em [49]. Entre as classes estudadas, observou-se que duas delas possuem árvores não isomorfas com mesma conectividade algébrica. As árvores da classe C_3 possuem conectividade algébrica igual a do caminho P_5 e as árvores da classe C_6 têm conectividade algébrica igual a $2 - \sqrt{3}$. Neste capítulo é apresentada uma nova classe de árvores não isomorfas de diâmetro $2k$, $k \geq 2$, que possuem a mesma conectividade algébrica que o caminho P_{2k+1} . Um estudo das características estruturais de tais árvores é apresentado, utilizando o algoritmo proposto em [26] e descrito na Seção 5.1. O capítulo termina com uma análise feita sobre o comportamento das SGI 's aplicadas às árvores dessa classe.

5.1 Localização dos autovalores de uma árvore

Em [26], é apresentado um algoritmo de localização de autovalores da matriz de adjacências A de uma árvore $T = (V, E)$ com relação a um escalar λ . Ou seja, o algoritmo informa quantos autovalores são maiores, quantos são menores e quantos são iguais a λ no espectro de A . O mesmo algoritmo pode ser aplicado para localizar os autovalores da matriz laplaciana. O procedimento tem como base a diagonalização da matriz $\lambda I + L$. As operações realizadas na matriz laplaciana podem ser aplicadas sobre a representação gráfica do grafo, como será visto adiante. No algoritmo, o cálculo desses valores é realizado diretamente nos vértices da árvore, dispensando o processamento da matriz $\lambda I + L$. A explicação sobre seu funcionamento aplicado à matriz $\lambda I + L$ encontra-se na Subseção 5.1.1.

O algoritmo recebe inicialmente uma árvore $T = (V, E)$ e um valor real λ e calcula para cada vértice de T uma valoração. A árvore T deve ser enraizada em algum vértice c (este pode ser escolhido aleatoriamente) e cada vértice deve ser rotulado por v_i , $1 \leq i \leq n$, tal que se v_i é descendente de v_j então $i < j$. Denote

por $b(v_i)$ o valor do vértice v_i e por $filhos(v_i)$, o conjunto de vértices filhos de v_i na árvore enraizada em c .

O algoritmo é composto de duas etapas: a primeira consiste em atribuir valores iniciais, $b(v_i) = \lambda + d(v_i)$, a cada vértice v_i de T . Na segunda etapa, os valores finais de $b(v_i)$ são atualizados, tendo início nas folhas e seguindo para seus ancestrais na árvore até chegar à raiz. O cálculo do novo valor de v_i , notado por $b'(v_i)$ depende do valor anterior $b(v_i)$ e dos valores $b(v_j)$, para todo v_j filho de v_i . Sua fórmula é:

$$b'(v_i) = b(v_i) - \sum_{v_j \in filhos(v_i)} \frac{1}{b(v_j)}. \quad (5.1)$$

Pela rotulação descrita, sabe-se que se v_j é filho de v_i então $j < i$. A segunda etapa do algoritmo realiza os cálculos de $b(v_i)$ variando o índice i de 1 até n . Isso garante que os vértices filhos de v_i já terão suas entradas previamente calculadas antes do cálculo de $b(v_i)$. Note que, para as folhas, o conjunto de filhos é vazio, isto é, se v_i é folha então $filhos(v_i) = \emptyset$. Com isso, o valor de $b'(v_i)$ é igual ao seu valor inicial $b(v_i) = \lambda + 1$.

Uma dificuldade surge no cálculo de $b'(v_i)$ quando, para algum filho $v_j \in filhos(v_i)$, tem-se $b(v_j) = 0$. Este seria um caso de divisão por zero na fórmula (5.1), o que não é possível. Além disso, o vértice v_i pode ter um ou mais filhos v_j com $b(v_j) = 0$. Para contornar este problema, escolhe-se um filho $v_j \in filhos(v_i)$ tal que $b(v_j) = 0$ e atribui-se a $b(v_j)$ e a $b(v_i)$ os valores 2 e $-\frac{1}{2}$, respectivamente, ou seja, $b(v_j) = 2$ e $b(v_i) = -\frac{1}{2}$. Caso v_i tenha um pai e este seja v_k , exclui-se a aresta $\{v_i, v_k\}$ da árvore T . Tais passos serão justificados na Subseção 5.1.1.

Os casos que devem ser observados no processo de cálculo das entradas dos vértices são:

- (i) aquele em que o vértice é uma folha;
- (ii) o caso em que v_i não é folha e $\forall v_j \in filhos(v_i), b(v_j) \neq 0$ e, ainda,
- (iii) aquele em que v_i não é folha e existe algum $v_j \in filhos(v_i)$ tal que $b(v_j) = 0$.

O Algoritmo 2 apresenta o pseudocódigo do processo de localização de autovaleores. A inicialização dos vértices é realizado nas linhas 2 e 3. Quando o vértice v_i é uma folha (caso (i)), o valor de $b(v_i)$ não é alterado (linhas 2 e 3). Para o caso (ii), a fórmula 5.1 é utilizada no cálculo de $b(v_i)$ (linhas 8 e 9). Por último, o caso (iii) ocorre quando o vértice v_i possui algum filho v_j com $b(v_j) = 0$ e, então, são realizados os passos descritos nas linhas de 11 a 15.

Como foi mencionado no início desta seção, o algoritmo se baseia no processo de diagonalização da matriz $\lambda I + L$. A matriz diagonal D , resultante desse processo, é semelhante a $\lambda I + L$ e suas entradas principais equivalem às valorações $b(v_i)$,

Algoritmo 2: *Diagonalizar*(T, λ)

Entrada: árvore T , número real λ
Saída: valoração para os vértices de T .

```
1 rotule os vértices das folhas à raiz;
2 para  $i = 1, \dots, n$  faça
3   |  $b(v_i) = \lambda + d(v_i)$ ;
4 para  $i = 1, \dots, n$  faça
5   | se  $v_i$  é folha então
6   |   | continue;
7   | senão
8   |   | se  $b(v_j) \neq 0, \forall v_j \in \text{filhos}(v_i)$  então
9   |   |   |  $b(v_i) \leftarrow b(v_i) - \sum_{v_j} \frac{1}{b(v_j)}$ ;
10  |   | senão
11  |   |   | Selecione um filho  $v_j$ , tal que  $b(v_j) = 0$ ;
12  |   |   |  $b(v_i) = -\frac{1}{2}$ ;
13  |   |   |  $b(v_j) = 2$ ;
14  |   |   | se  $v_i$  possuir um pai  $v_k$  então
15  |   |   |   | Remova a aresta  $(v_i, v_k)$ ;
```

$i = 1, \dots, n$, obtidas pelo algoritmo *Diagonalizar*. O Teorema 5.1 é apresentado e provado em [26]. Este resultado mostra quais informações podem ser extraídas dos valores das entradas da matriz D e vale tanto para a localização dos autovalores da matriz de adjacências quanto para a localização dos autovalores da matriz laplaciana.

Teorema 5.1 ([26]). *Dada uma árvore T , sejam $\lambda = -a$ e D a matriz diagonal produzida por *Diagonalizar*(T, λ).*

1. *O número de entradas positivas em D é o número de autovalores de T maiores que a ;*
2. *o número de entradas negativas em D é o número de autovalores de T menores que a e*
3. *se existem j entradas nulas em D então a é um autovalor de T com multiplicidade j .*

Outro resultado apresentado por [26] é o Corolário 5.1 que permite determinar o número de autovalores em um dado intervalo de números reais.

Corolário 5.1 ([26]). *O número de autovalores no intervalo (a_1, a_2) é o número de entradas positivas resultantes da chamada *Diagonalizar*($T, -a_1$) menos o número de entradas positivas resultantes da chamada *Diagonalizar*($T, -a_2$).*

EXEMPLO: Considere a árvore T , apresentada na Figura 5.1 (a), enraizada em v_6 . A inicialização dos vértices de T é apresentada na Figura 5.1 (b). Suponha que se deseja descobrir quantos autovalores existem no intervalo $[1, 2]$. Para tal, serão realizadas duas chamadas ao algoritmo *Diagonalizar*. A primeira, será para localizar os autovalores em relação ao valor 2 e a segunda chamada localizará os autovalores em relação ao valor 1. Os passos da chamada $Diagonalizar(T, -2)$ são descritos a seguir.

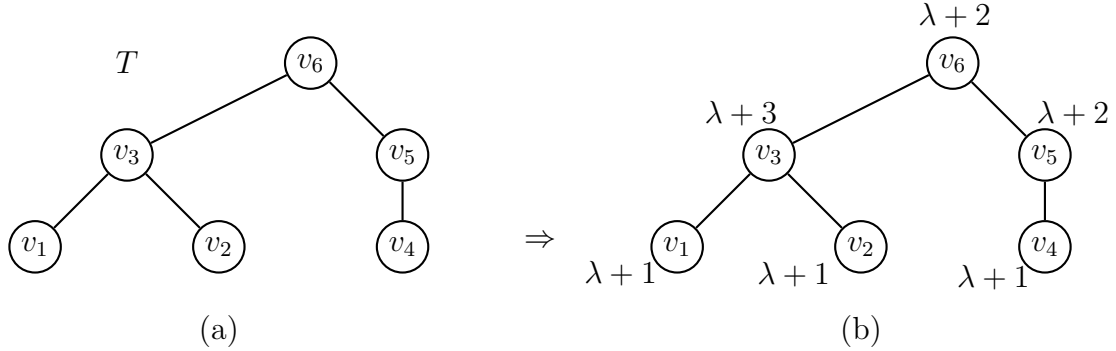


Figura 5.1: Inicialização das entradas para os vértices de $T = (V, E)$.

As entradas dos vértices de T são assim inicializadas:

$$\begin{aligned} b(v_1) &\leftarrow b(v_2) \leftarrow b(v_4) \leftarrow -2 + 1 = -1; \\ b(v_3) &\leftarrow -2 + 3 \leftarrow 1; \\ b(v_5) &\leftarrow b(v_6) \leftarrow -2 + 2 = 0. \end{aligned}$$

As entradas das folhas v_1 e v_2 não são alteradas e a entrada de v_3 é calculada da seguinte forma:

$$b(v_3) \leftarrow b(v_3) - \left(\frac{1}{b(v_1)} + \frac{1}{b(v_2)} \right) = 1 - \left(\frac{1}{-1} + \frac{1}{-1} \right) = 3.$$

Em seguida, o algoritmo calcula a entrada de v_4 , que não é alterada, e segue para o cálculo da entrada de v_5 ,

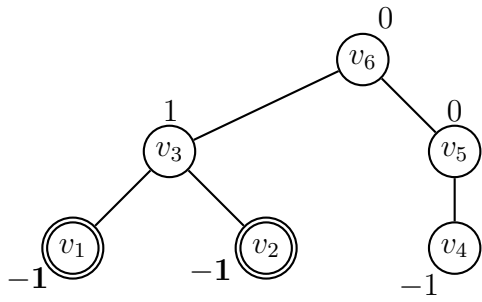
$$b(v_5) \leftarrow b(v_5) - \frac{1}{b(v_4)} = 0 - \frac{1}{-1} = 1.$$

Por fim, a entrada de v_6 é assim calculada,

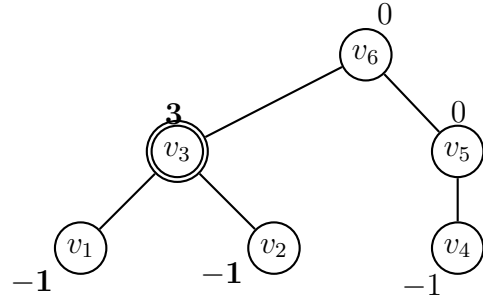
$$b(v_6) \leftarrow b(v_6) - \left(\frac{1}{b(v_3)} + \frac{1}{b(v_5)} \right) = 0 - \left(\frac{1}{3} + \frac{1}{1} \right) = -\frac{4}{3}.$$

Cada passo do processo é ilustrado na Figura 5.2.

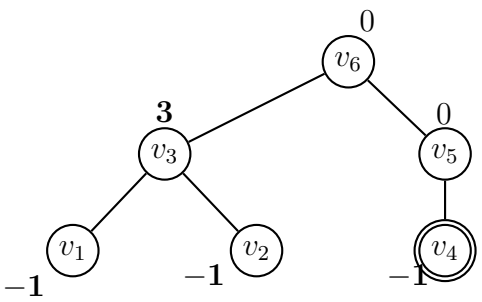
Com as entradas obtidas pelo algoritmo, do Teorema 5.1, conclui-se que existem quatro autovalores menores que 2 e dois autovalores maiores que 2. Além disso,



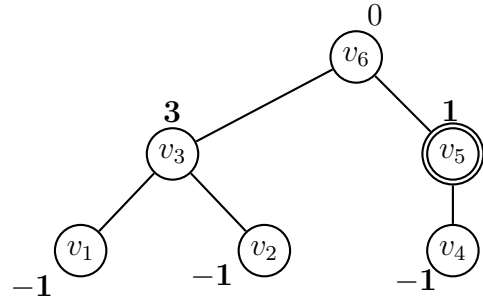
(a) cálculo de $b(v_1)$ e $b(v_2)$



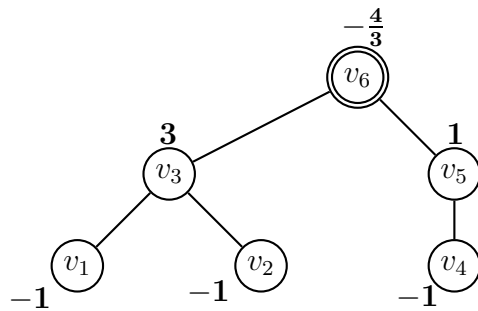
(b) cálculo de $b(v_3)$



(c) cálculo de $b(v_4)$



(d) cálculo de $b(v_5)$



(e) cálculo de $b(v_6)$

Figura 5.2: Localização dos autovalores com relação a 2, ou seja, $\lambda = -2$.

o valor 2 não é autovalor do laplaciano de T , pois não há entradas nulas como resultado do processo.

O próximo passo é realizar a chamada $Diagonalizar(T, -1)$ para a árvore T da Figura 5.1 a fim de localizar os autovalores em relação ao valor 1, que equivale a $\lambda = -1$. As entradas dos vértices de T são inicializadas da seguinte maneira,

$$\begin{aligned} b(v_1) &\leftarrow b(v_2) \leftarrow b(v_4) = -1 + 1 = 0; \\ b(v_3) &\leftarrow -1 + 3 = 2; \\ b(v_5) &\leftarrow b(v_6) \leftarrow -1 + 2 = 1. \end{aligned}$$

O algoritmo inicia calculando as entradas das folhas v_1 e v_2 . Em seguida é calculado o valor de $b(v_3)$. Neste caso, os dois filhos de v_3 possuem entradas nulas. Então, escolhe-se um dos filhos de v_3 com entrada nula, por exemplo v_2 , e faz-se as seguintes atribuições,

$$\begin{aligned} b(v_2) &\leftarrow 2, \\ b(v_3) &\leftarrow -\frac{1}{2}, \end{aligned}$$

para em seguida, remover a aresta $\{v_3, v_6\}$. Este processo é ilustrado na Figura 5.3 (b). Observe que o grafo resultante é desconexo, mas isso não irá interferir no processo de localização dos autovalores. Neste ponto, v_3 deixa de ser filho de v_6 e o processo continua com o cálculo da entrada de v_4 . Como este é uma folha, seu valor de entrada não é modificado, veja que $b(v_4)$ continua igual a 0. No cálculo da entrada de v_5 , análogo ao caso de v_3 , v_5 tem um filho cuja entrada é nula, no exemplo é o v_4 . Então as seguintes operações são realizadas,

$$\begin{aligned} b(v_4) &\leftarrow 2, \\ b(v_5) &\leftarrow -\frac{1}{2} \end{aligned}$$

e a aresta $\{v_5, v_6\}$ é removida. Esses passos são ilustrados na Figura 5.3 (d). Ao calcular a entrada para v_6 (Figura 5.3 (e)), obtém-se o conjunto de filhos de v_6 vazio, ficando a entrada de v_6 igual ao seu valor inicial, $b(v_6) = 1$.

Com as entradas obtidas pelo Algoritmo $Diagonalizar$, conclui-se que o valor 1 é um autovalor do Laplaciano de T e existem dois autovalores menores que 1 e três autovalores maiores que 1, fato esse decorrente do Teorema 5.1. Do Corolário 5.1, vê-se que existe só um autovalor da matriz laplaciana de T entre os valores 1 e 2.

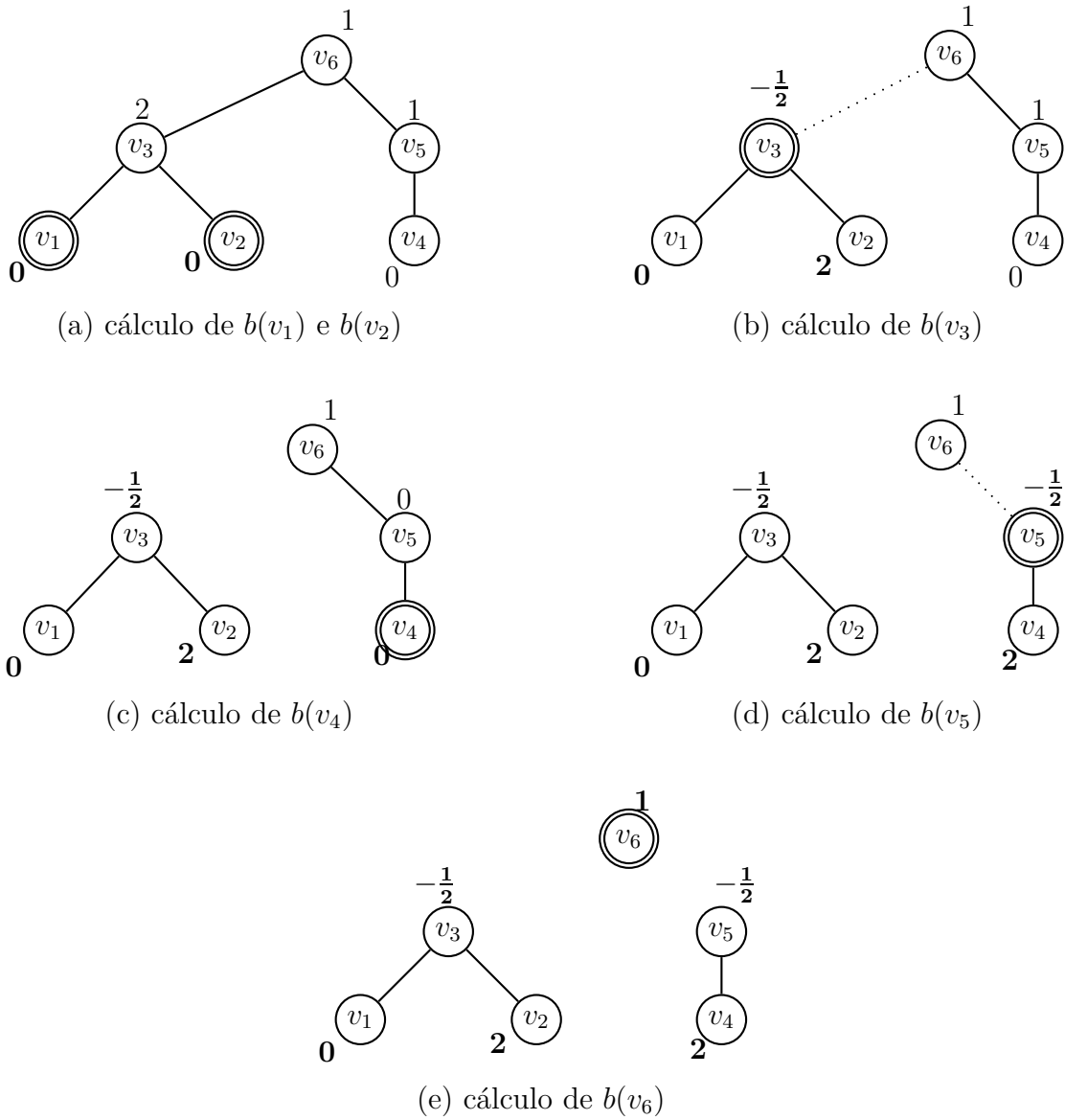


Figura 5.3: Localização dos autovalores com relação ao valor 1 ($\lambda = -1$).

5.1.1 Diagonalização de $\lambda I + L$

Como foi dito no início do capítulo, o algoritmo *Diagonalizar* foi apresentado em [26] com sua aplicação à matriz de adjacências. No mesmo trabalho, os autores afirmam que o algoritmo pode ser utilizado para localizar os autovalores da matriz laplaciana. Para entender a razão das atribuições e dos cálculos realizados pelo algoritmo *Diagonalizar*, será explicado o funcionamento do algoritmo aplicado à matriz laplaciana L de uma árvore T . O processo se baseia na diagonalização da matriz $\lambda I + L$, visto que o polinômio característico do Laplaciano de T é $p(x) = \det(xI - L)$. Se $x = -\lambda$, tem-se $p(-\lambda) = \det(-\lambda I - L) = (-1)^n \det(\lambda I + L)$.

Suponha uma árvore $T = (V, E)$ enraizada em um vértice c , escolhido aleatoriamente entre os n vértices de T . Utilizando-se a mesma rotulação descrita no início desta seção, denote por $B = \lambda I + L$; b_{ij} , o elemento da i -ésima linha e j -ésima coluna de B e por $filhos(v_i)$, o conjunto de vértices filhos de v_i em T . A matriz B tem a seguinte forma,

$$B = \begin{bmatrix} \lambda + d(v_1) & & & -1 & & & & \\ & \ddots & & & & & & \\ -1 & & \lambda + d(v_j) & & & & -1 & \\ & & & \ddots & & & & \\ & & -1 & & \lambda + d(v_i) & & & -1 \\ & & & & & \ddots & & \\ & & & & -1 & & \lambda + d(v_n) & \\ & & & & & & & \end{bmatrix}.$$

Denote por R_i e C_i a i -ésima linha e a i -ésima coluna de B , respectivamente. Se um dado vértice possuir todos os seus filhos v_j com suas respectivas entradas na diagonal principal não nulas, ou seja, $b_{jj} \neq 0$, então pode-se utilizar as entradas dos filhos para zerar os elementos que estão fora da diagonal principal. As operações para zerar os elementos fora da diagonal principal são,

$$R_i = R_i + \frac{1}{b_{jj}} R_j, \quad (5.2)$$

$$C_i = C_i + \frac{1}{b_{jj}} C_j. \quad (5.3)$$

As operações de linha e coluna realizadas para zerar os elementos fora da diagonal de uma dada linha e uma dada coluna i resulta da seguinte expressão,

$$b'_{ii} = b_{ii} - \sum_{v_j \in filhos(v_i)} \frac{1}{b_{jj}}. \quad (5.4)$$

Observe que a entrada da matriz B na linha i e coluna i equivale a entrada do vértice

v_i na árvore T . Por isso, a equação (5.4) coincide com a equação (5.1).

Quando existe ao menos um filho de v_i , digamos v_j , tal que $b_{jj} = 0$, a equação (5.4) não pode ser aplicada. Neste caso, as entradas b_{ij} e b_{ji} são utilizadas para zerar os elementos fora da diagonal na i -ésima linha e na i -ésima coluna referentes aos irmãos de v_j , como será visto a seguir. Denote por B' a matriz nesse ponto do processo. Esta terá a seguinte forma,

$$B' = \begin{matrix} & r & & & \\ & j & & & \\ & i & & & \\ & k & & & \end{matrix} \begin{bmatrix} b_{rr} & & & & -1 \\ & & & & \\ & & 0 & & -1 \\ -1 & & -1 & \lambda + d(v_i) & -1 \\ & & & -1 & \lambda + d(v_k) \end{bmatrix}.$$

Sejam $v_j, v_r \in \text{filhos}(v_i)$, tais que $b'_{jj} = 0$ e $r \neq j$. As operações de linhas e colunas para zerar os elementos não nulos fora da diagonal são,

$$R_r \leftarrow R_r - R_j, \quad (5.5)$$

$$C_r \leftarrow C_r - C_j. \quad (5.6)$$

Se v_k é o pai de v_i , então utiliza-se as entradas b'_{ij} e b'_{ji} para eliminar as entradas b'_{ik} e b'_{ki} , respectivamente. Estas últimas representam a aresta que liga o vértice v_i ao vértice v_k na árvore T , ou seja, as operações (5.7) e (5.8), a seguir, eliminam a aresta $\{v_i, v_k\}$ de T .

$$R_k \leftarrow R_k - R_j, \quad (5.7)$$

$$C_k \leftarrow C_k - C_j. \quad (5.8)$$

A matriz resultante, denotada por B'' , é

$$B'' = \begin{matrix} & r & & & \\ & j & & & \\ & i & & & \\ & k & & & \end{matrix} \begin{bmatrix} b_{rr} & & & & 0 \\ & & & & \\ & & 0 & & -1 \\ 0 & & -1 & \lambda + d(v_i) & 0 \\ & & & 0 & \lambda + d(v_k) \end{bmatrix}.$$

Para que seja possível zerar as entradas b''_{ij} e b''_{ji} de B'' , deve-se zerar a entrada

b''_{ii} com as seguintes operações,

$$R_i \leftarrow R_i + \frac{b_{ji}}{2} b_{ii}; \quad (5.9)$$

$$C_i \leftarrow C_i + \frac{b_{ij}}{2} b_{ii}, \quad (5.10)$$

obtendo-se a matriz B''' ,

$$B''' = \begin{matrix} & r & & & \\ & \left[\begin{array}{cccc} b_{rr} & 0 & & \\ & 0 & -1 & \\ & 0 & -1 & 0 & 0 \\ & & 0 & \lambda + d(v_k) \end{array} \right] & & \\ & j & & & \\ & i & & & \\ & k & & & \end{matrix}.$$

Em seguida, para que os elementos fora da diagonal principal sejam anulados, deve-se executar as operações,

$$R_j \leftarrow R_j - R_i;$$

$$C_j \leftarrow C_j - C_i;$$

$$R_i \leftarrow R_i + \frac{1}{2} R_j e$$

$$C_i \leftarrow C_i + \frac{1}{2} C_j.$$

A matriz resultante, denotada por B^{iv} , é a seguinte,

$$B^{iv} = \begin{matrix} & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{matrix} \left[\begin{array}{cccc} b_{rr} & 0 & & \\ & 2 & 0 & \\ 0 & 0 & -\frac{1}{2} & 0 \\ & & 0 & \lambda + d(v_k) \end{array} \right].$$

Note que, de acordo com B^{iv} , a entrada referente ao vértice v_i recebe valor $-\frac{1}{2}$ e a entrada do filho escolhido (um filho $v_j \in \text{filhos}(v_i)$, cuja entrada em B era $b'''_{jj} = 0$) passa a ser $b^{iv}_{jj} = 2$. As operações (5.7) e (5.8) zeram as entradas b_{ik} e b_{ki} de B' . Isso equivale a remover a aresta que liga o vértice v_i a seu pai v_k na árvore. Estes passos compreendem às linhas 11 a 15 do algoritmo *Diagonalizar*.

EXEMPLO: Considere a árvore T , da Figura 5.1 (a), enraizada em v_6 . A matriz $B = \lambda I + L$ é:

$$B = \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{matrix} \begin{bmatrix} \lambda + 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & \lambda + 1 & -1 & 0 & 0 & 0 \\ -1 & -1 & \lambda + 3 & 0 & 0 & -1 \\ 0 & 0 & 0 & \lambda + 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & \lambda + 2 & -1 \\ 0 & 0 & -1 & 0 & -1 & \lambda + 2 \end{bmatrix}.$$

Suponha que se deseja localizar os autovalores de B com relação ao valor 1. Deve-se fazer $\lambda = -1$. A matriz B' , resultante da substituição do valor de λ , é,

$$B' = \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{matrix} \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & -1 \\ 0 & 0 & -1 & 0 & -1 & 1 \end{bmatrix}.$$

No processo de diagonalização da matriz, tem-se que as entradas b'_{11} e b'_{22} , referentes aos vértices v_1 e v_2 da árvore, respectivamente, não são alteradas, pois tais vértices são folhas em T e, portanto, não possuem filhos. Já v_3 é um vértice interno da árvore e possui v_1 e v_2 como seus filhos, por isso b'_{33} deverá ser alterado pelo processo de anulação dos elementos b'_{31} , b'_{32} , b'_{13} e b'_{23} , mas as entradas b'_{11} e b'_{22} são nulas. Neste caso, escolhe-se um dos filhos, por exemplo v_2 , e realiza-se as operações,

$$\begin{aligned} R_1 &\leftarrow R_1 - R_2, \\ C_1 &\leftarrow C_1 - C_2. \end{aligned}$$

Como v_3 possui pai (v_6 em T), deve-se anular os elementos b'_{63} e b'_{36} com as seguintes operações:

$$\begin{aligned} R_6 &\leftarrow R_6 - R_2, \\ C_6 &\leftarrow C_6 - C_2. \end{aligned}$$

A matriz resultante B'' é

$$B'' = \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{matrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

Agora deve-se zerar a entrada b''_{33} , que em B'' é igual a 2, com as seguintes operações,

$$R_3 \leftarrow R_3 + \frac{b''_{23}}{2} b''_{33}; \quad (5.11)$$

$$C_3 \leftarrow C_3 + \frac{b''_{32}}{2} b''_{33}. \quad (5.12)$$

Neste passo do processo, chega-se a seguinte matriz,

$$B''' = \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{matrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

Em seguida, para que os elementos fora da diagonal principal sejam anulados, é preciso executar as operações,

$$R_2 \leftarrow R_2 - R_3;$$

$$C_2 \leftarrow C_2 - C_3;$$

$$R_3 \leftarrow R_3 + \frac{1}{2} R_2;$$

$$C_3 \leftarrow C_3 + \frac{1}{2} C_2.$$

A matriz resultante é

$$B^{iv} = \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{matrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

Como v_4 é uma folha na árvore, em B^{iv} , não há elementos da linha 4 e da coluna 4, fora da diagonal principal, que sejam não nulos. Logo, a entrada b_{44}^{iv} não será alterada. Já para a entrada b_{55}^{iv} tem-se que $b_{45}^{iv} = b_{54}^{iv} = -1$ e $b_{44}^{iv} = 0$. Então o valor de b_{55}^{iv} será modificado de forma a zerar os elementos b_{45}^{iv} e b_{54}^{iv} . Como $b_{44}^{iv} = 0$, realiza-se operações semelhantes às aplicadas anteriormente para tratar o caso da entrada (3, 3) em B''' . A matriz B^v resultante de tais operações é

$$B^v = \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{matrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Observe que as entradas em B que representavam as arestas que ligavam v_6 a seus filhos na árvore foram anuladas no processo de diagonalização. Assim, desde que $b_{45}^v = b_{54}^v = 0$, a entrada b_{55}^v não será alterada. Neste ponto, a matriz B^v tem todos os elementos fora da diagonal principal iguais a zero e é a matriz diagonal procurada. Do Teorema 5.1, tem-se que o valor 1 é um autovalor do laplaciano de T com multiplicidade 1, há três autovalores maiores que 1 e dois autovalores menores que 1. Veja que as entradas obtidas são iguais às encontradas para os vértices da árvore T da Figura 5.3 resultantes do processamento executado pelo algoritmo *Diagonalizar*.

5.2 Medusas e árvores *starlike*

Uma árvore *starlike* é aquela que contém um único vértice com grau maior que 2. Há vários artigos dedicados a tais árvores [13, 22, 29]. Embora não existam duas árvores *starlike* com o mesmo espectro, [13], podem existir árvores *starlike* não isomorfas com a mesma conectividade algébrica. A seguinte definição será utilizada no decorrer deste texto.

Definição 5.1. *Seja uma árvore starlike $T = (V, E)$ e c um vértice qualquer de T . Cada subárvore de T obtida pela remoção de c é chamada de ramo de T em c .*

Definição 5.2. *Dados n e k , com $n \geq 2k$, seja T uma árvore com n vértices e diâmetro $2k$. Seja P_{2k+1} um caminho cujo comprimento é igual ao diâmetro de T com vértice central c . Uma árvore T é chamada uma árvore medusa se qualquer ramo de T em c for isomorfo a um caminho de comprimento máximo igual a k .*

Para o que se segue, será adotada a seguinte rotulação para os vértices de uma árvore medusa:

Definição 5.3. Para $k > 0$, escolha $n \geq 2k$. Seja T uma árvore medusa com n vértices e diâmetro $2k$, e seja c o vértice central de um caminho P_{2k+1} (subárvore induzida de T) tal que $d(c) = g$. Faça T enraizada em c . Sejam v_i^r e v_j^r rótulos atribuídos a vértices de um mesmo ramo r em c com $1 \leq r \leq g$. As seguintes condições devem ser satisfeitas:

- (i) Se v_i^r é uma folha então $i = 1$;
- (ii) Se v_j^r é pai de v_i^r em T e $v_j^r \neq c$ então $j = i + 1$.

No decorrer do trabalho, um conjunto de árvores medusas com n vértices, diâmetro d , vértice central c e $d(c) = g$ será denotada por $\mathcal{M}(n, d, g)$.

A Figura 5.4 apresenta um exemplo de uma árvore medusa $T \in \mathcal{M}(20, 10, 5)$ com a rotulação proposta. Note que como o diâmetro da árvore é igual a 10, então $k = 5$. Dessa forma, os ramos 1 e 2 são isomorfos a P_5 e compõem um caminho cujo comprimento é igual ao diâmetro da árvore.

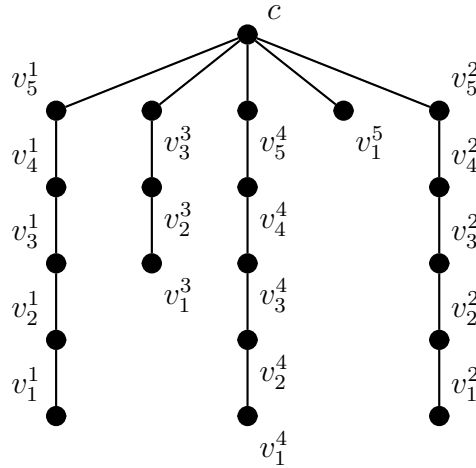


Figura 5.4: Ilustração da rotulação aplicada a uma árvore medusa $T \in \mathcal{M}(20, 10, 5)$.

Dado $k > 1$, observe que o caminho P_{2k+1} é a única árvore do conjunto $\mathcal{M}(2k + 1, 2k, 2)$. Deste modo, considerando c o seu vértice central, P_{2k+1} tem dois ramos isomorfos a P_k em c e cuja rotulação está na Figura 5.5.

O Lema 5.1 mostra o resultado da aplicação do algoritmo *Diagonalizar* sobre P_{2k+1} com um valor λ , isto é, $Diagonalizar(P_{2k+1}, \lambda)$.

Lema 5.1. Dados $k \geq 1$ e λ um número real qualquer, seja T o caminho P_{2k+1} . Na execução da chamada ao algoritmo *Diagonalizar* passando como parâmetros T e λ , $Diagonalizar(T, \lambda)$, tem-se que $\forall i, 1 \leq i \leq k, b(v_i^1) = b(v_i^2)$.

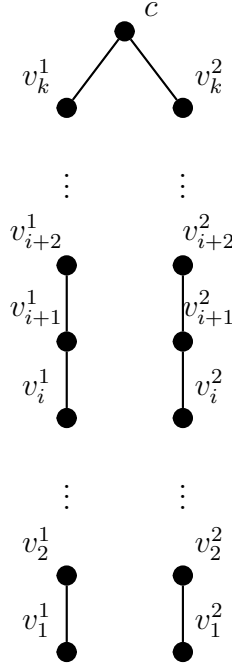


Figura 5.5: Rotulação aplicada ao caminho P_{2k+1} .

PROVA: Sabe-se que os dois ramos que partem de c são isomorfos a P_k e que a execução do algoritmo procede das folhas até a raiz. Discrimine os ramos por $r = 1$ e $r = 2$. O valor de $b(v_i^r)$ calculado para cada vértice v_i^r depende somente de seus filhos. Como cada ramo, $r = 1$ ou $r = 2$, é isomorfo ao caminho P_k , para cada i , $1 \leq i \leq k$, tem-se

$$b(v_i^r) = \begin{cases} 2 + \lambda - \frac{1}{b(v_{i-1}^r)} & , \text{ se } v_i^r \text{ é um vértice interno de } P_k, \\ 1 + \lambda & , \text{ se } v_i^r \text{ é uma folha de } P_k. \end{cases}$$

Dos passos do Algoritmo $Diagonalizar(T, \lambda)$, tem-se que:

$$\begin{cases} b(v_1^1) = b(v_1^2) = 1 + \lambda \\ b(v_j^1) = b(v_j^2) = 2 + \lambda - \frac{1}{b(v_{j-1}^r)}. \end{cases}$$

Portanto, $\forall i, 1 \leq i \leq k, b(v_i^1) = b(v_i^2)$. ■

A conectividade algébrica de P_n é conhecida e é dada por $a(P_n) = 2(1 - \cos(\frac{\pi}{n}))$, veja [15]. Como a conectividade algébrica é o segundo menor autovalor do laplaciano de um grafo, a chamada ao algoritmo $Diagonalizar$ aplicado a P_{2k+1} e $a_k = a(P_{2k+1})$, $Diagonalizar(P_{2k+1}, a_k)$, com raiz no vértice central c , deve gerar apenas um valor negativo na árvore, indicando que existe um único autovalor do laplaciano de P_{2k+1} menor que $a(P_{2k+1})$. Tal valor é negativo e deve ser associado a um dos filhos de c . O Teorema 5.2 prova tal resultado.

Teorema 5.2. *Dado $k > 1$, seja o caminho P_{2k+1} e denote por a_k a sua conhecida*

conectividade algébrica. Ao aplicar o algoritmo $\text{Diagonalizar}(P_{2k+1}, -a_k)$ ao caminho enraizado no vértice central c , começando por um de seus dois ramos, as seguintes afirmações são verdadeiras:

- (i) O único valor nulo é aquele associado ao vértice v_k^1 (ou v_k^2) adjacente à raiz;
- (ii) O único valor negativo gerado pelo algoritmo é aquele atribuído ao vértice central c ;
- (iii) Para todos os outros vértices de T o algoritmo atribui valores positivos.

PROVA: item (i): Do Lema 5.1, tem-se $b(v_i^1) = b(v_i^2)$, $1 \leq i \leq k$ e quando um vértice v_i possui ao menos um filho cujo valor é nulo, o algoritmo escolhe um filho v_j de v_i , tal que $b(v_j) = 0$, realiza as operações: $b(v_j) = 2$, $b(v_i) = -\frac{1}{2}$ e remove a aresta $\{v_i, v_p\}$, caso v_i tenha pai v_p . Observe que se v_i possui outros filhos v_h , seus valores $b(v_h)$ não são alterados.

Suponha que na execução do algoritmo ocorra, para algum $i < k$, $b(v_i^r) = 0$. Lembrar que $b(v_i^1) = b(v_i^2) = 0$. A partir daí, as seguintes operações serão realizadas no ramo 1: $b(v_i^1) = 2$, $b(v_{i+1}^1) = -\frac{1}{2}$ e a aresta que liga v_{i+1}^1 a v_{i+2}^1 é removida. Operações análogas são realizadas no ramo 2, para os vértices v_i^2 e v_{i+1}^2 . Neste ponto, têm-se dois vértices com valores negativos, $b(v_{i+1}^1) = b(v_{i+1}^2) = -\frac{1}{2}$ que indicam existir dois autovalores não-nulos menores que a conectividade algébrica, o que é um absurdo. Portanto, na execução do algoritmo, os valores nulos são atribuídos apenas aos filhos da raiz, ou seja, $b(v_k^1) = b(v_k^2) = 0$. Após tais operações, um dos filhos da raiz possuirá valor nulo, o outro terá valor igual a 2 e a raiz c receberá o valor $b(c) = -\frac{1}{2}$.

itens (ii) e (iii): Do item (i), sabe-se que $b(c) < 0$, $b(v_k^1) = 0$ e $b(v_k^2) = 2$, ou $b(v_k^2) = 0$ e $b(v_k^1) = 2$. Suponha que a chamada $\text{Diagonalizar}(T, -a_k)$ tenha gerado uma rotulação tal que exista algum v_i^1 , $1 < i < k$, com $b(v_i^1) < 0$. Do Lema 5.1, tem-se $b(v_i^2) = b(v_i^1) < 0$. Com isso, há a ocorrência de três vértices com valores negativos, indicando que existem três autovalores menores que a conectividade algébrica, o que é um absurdo. Portanto, existe apenas um valor negativo, aquele referente ao vértice central c e um valor nulo atribuído a um dos filhos de c . Todas as outras entradas devem ser positivas. Ou seja, $b(v_i^1) = b(v_i^2) > 0$, $\forall i < k$. ■

Teorema 5.3. Para $k > 1$ e $n \geq 2k + 1$, toda árvore medusa $T \in \mathcal{M}(n, 2k, g)$ tem conectividade algébrica igual a do caminho P_{2k+1} , ou seja, $a(T) = a(P_{2k+1})$.

PROVA: Para dados $k > 1$ e $n \geq 2k + 1$, seja $a_k = a(P_{2k+1})$ e $T \in \mathcal{M}(n, 2k, g)$ uma árvore medusa com vértice central c . Da Definição 5.2, todos os vértices a exceção do vértice c possuem grau máximo igual a 2. Considere o vértice central c como a raiz da árvore. Ao menos dois ramos em c possuem comprimento k (ramos do caminho

que compõem o diâmetro). Do Teorema 5.2, sabe-se que $b(v_k^r) = 2 - a_k - \frac{1}{b(v_{k-1}^r)} = 0$, para todo ramo r de comprimento k em c . Além disso, tem-se que $b(v_i^r) > 0$, quando $1 \leq i < k$. Considere dois ramos distintos r e s em c . Do Lema 5.1, $b(v_i^r) = b(v_i^s)$. Dado que $i \leq k$, tem-se $b(v_i^r) = 0$, quando $i = k$, e $b(v_i^r) > 0$, quando $i < k$.

Do algoritmo *Diagonalizar*, um dos filhos v_k^r de c recebe valor $b(v_k^r) = 2$ e $b(c) = -\frac{1}{2}$, enquanto os outros vértices adjacentes a c ficam com seus valores inalterados. Desse modo, pelo menos um vértice v_k^r adjacente a c possui $b(v_k^r) = 0$, o valor de c é negativo e todos os outros vértices v_i^r possuem valores $b(v_i^r) > 0$. Isto indica que a_k é o segundo menor autovalor do polinômio característico do laplaciano da árvore. Portanto, $a(T) = a_k$. ■

Os resultados até o momento mostram que todas as árvores medusas de diâmetro $2k$ possuem conectividade algébrica igual a do caminho P_{2k+1} , isto é, igual a $a(P_{2k+1})$. O resultado a seguir mostra que na família das árvores *starlike* com diâmetro fixo e igual a $2k$, as medusas são as que maximizam a conectividade algébrica.

Observe que as árvores *starlike* de diâmetro par que não são medusas são aquelas em que o vértice de grau maior que dois não é o vértice central do caminho de comprimento igual ao diâmetro. Para esses casos vale o seguinte resultado.

Teorema 5.4. *Seja T uma árvore starlike com $n \geq 2k + 1$ vértices e diâmetro $2k$ e seja x o único vértice de T tal que $d(x) > 2$. Tem-se que $a(T) \leq a(P_{2k+1})$ com a igualdade verdadeira se e somente se $T \in \mathcal{M}(n, 2k, d(x))$.*

PROVA: Sejam n e k dados, tal que $n \geq 2k + 1$. Considere T uma *starlike* com diâmetro $2k$ e n vértices. Para mostrar que apenas as medusas possuem conectividade algébrica igual a $a(P_{2k+1})$ e este é o valor máximo para a conectividade algébrica de qualquer árvore desta família, é suficiente mostrar que as *starlikes* que não são medusas possuem conectividade algébrica estritamente menor que $a(P_{2k+1})$, já que a igualdade é diretamente verificada pelas medusas de diâmetro par, como provado no Teorema 5.3.

Seja T uma árvore *starlike* não medusa com diâmetro $2k$ e c seja o vértice central do caminho que determina o diâmetro. Suponha que T é enraizada em c e $x \neq c$ é o único vértice de T com $d(x) > 2$. Seja $a_k = a(P_{2k+1})$. Como T não é medusa, o grau de c é igual a 2. Deste modo, há somente dois ramos partindo de c . Um deles é isomorfo ao caminho P_k e o outro possui $n - k - 1$ vértices tendo exatamente um vértice com grau maior que 2, que é x . A Figura 5.6 ilustra a estrutura de T .

Seja $r = 1$ o ramo em c isomorfo a P_k e $r = 2$ o outro ramo em c . Rotule os vértices do ramo 1 por v_i^1 , de forma que se v_i^1 é filho de v_j^1 , então $j = i + 1$. Rotule os vértices do ramo 2 por v_j^2 , de modo que se v_i^2 é descendente de v_j^2 , então $i < j$.

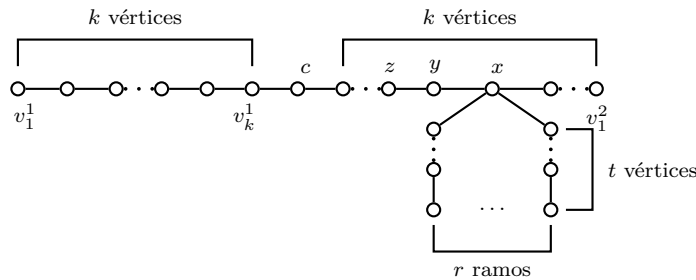


Figura 5.6: Estrutura de uma árvore *starlike* que não é medusa.

Dado que o ramo 1 é isomorfo a P_k , do Teorema 5.2, a chamada *Diagonalizar*($T, -a_k$) faz $b(v_k^1) = 0$ e todos os outros vértices v_j^1 com valores positivos, ou seja, $j < i$, recebem $b(v_j^1) > 0$. Para o ramo 2, seja $dist(x, v_1^2)$ a distância entre x e v_1^2 e seja x o único vértice de T com $d(x) > 2$. Em x há $d(x) - 1$ ramos isomorfos a caminhos de comprimento máximo igual a $dist(x, v_1^2)$. Note que, como o diâmetro de T é igual a $2k$, a distância máxima de x a qualquer uma das folhas descendentes de x é igual a $dist(x, v_1^2)$.

Faça $p = dist(x, v_1^2)$. Como os ramos pendentes em x são isomorfos a caminhos P_t , $1 \leq t \leq p$, se v_j^2 e v_ℓ^2 são descendentes de x em um mesmo ramo de x , os seguintes casos podem ocorrer:

- (i) ou v_j^2 é folha e assim, $b(v_j^2) = 1 - a_k$;
- (ii) ou v_ℓ^2 é pai de v_j^2 , e então, $b(v_\ell^2) = 2 - a_k - \frac{1}{b(v_j^2)}$.

Do Teorema 5.2, $b(v_j^2) > 0$, para qualquer v_j^2 descendente de x em T . Desta forma, o cálculo de $b(x)$ é dado por:

$$b(x) = d(x) - a_k - \sum_{v_j^2 \in \text{filhos}(x)} \frac{1}{b(v_j^2)}.$$

Considerando a rotulação da árvore apresentada na Figura 5.6, a maior distância de x a uma folha do ramo 2 é $p = dist(x, v_1^2)$. Então a distância de x ao vértice c é $dist(x, c) = k - p$. O vértice do ramo 1 com mesma distância ao vértice c é rotulado por v_{p+1}^1 , cuja distância à folha do mesmo ramo é igual a p . De forma análoga à análise realizada no Teorema 5.2, os valores dos vértices v_{p+1}^1 até v_k^1 serão comparados, um a um, com os valores dos respectivos vértices entre x até o filho de c no ramo 2.

Para cada $v_j^2 \in \text{filhos}(x)$, tem-se $\frac{1}{b(v_j^2)} > 0$. Isso conduz a $b(x) < b(v_{p+1}^1)$. A partir daí, os três seguintes casos podem ocorrer a $b(x)$ e a seus ancestrais em T . Sejam y e z , respectivamente, o pai e o avô de x em T .

- (a) $b(x) > 0$. Neste caso,

$$b(x) < b(v_{p+1}^1) \Rightarrow \frac{1}{b(x)} > \frac{1}{b(v_{p+1}^1)} \Rightarrow$$

$$\Rightarrow b(y) = 2 - a_k - \frac{1}{b(x)} < b(v_{p+2}^1) = 2 - a_k - \frac{1}{b(v_{p+1}^1)}.$$

Calcula-se então o valor de $b(y)$ e verifica-se em qual dos três casos ele se encaixa.

- (b) $b(x) = 0$. Quando isso ocorre, faz-se $b(x) = 2$, $b(y) = -\frac{1}{2}$ e remove-se a aresta que liga y ao seu pai z , avô de x . Observe que

$$b(z) = 2 - a_k > b(v_{p+3}^1) \Rightarrow \frac{1}{b(z)} < \frac{1}{b(v_{p+3}^1)}.$$

Se w é pai de z , então $b(w) > b(v_{p+4}^1)$. A desigualdade segue verdadeira até os filhos da raiz c . Desta forma, sabe-se que se ocorrer um valor nulo para algum vértice entre x e o filho de c no ramo 2, incluindo o próprio x , tem-se que seus ancestrais naquele ramo terão valor positivo.

- (c) $b(x) < 0$. Neste caso, sabe-se que

$$\begin{aligned} b(y) = 2 - a_k - \frac{1}{b(x)} > b(v_{p+2}^1) = 2 - a_k - \frac{1}{b(v_{p+1}^1)} &\Rightarrow \\ \Rightarrow \frac{1}{b(y)} < \frac{1}{b(v_{p+2}^1)} &\Rightarrow b(z) > b(v_{p+3}^1). \end{aligned}$$

A desigualdade segue verdadeira para os ancestrais de y e v_{p+3}^1 até os filhos de c . Isso significa que há exatamente um valor negativo no ramo 2. Ou seja, a chamada *Diagonalizar*($T, -a_k$) gera exatamente dois valores negativos para T . Isto indica que quando x não é o vértice central do caminho P_{2k+1} que determina o diâmetro de T , $a(T) < a(P_{2k+1})$. Com isso, pode-se concluir que as medusas maximizam a conectividade algébrica das árvores *starlike*.

Com isso, pode-se concluir que qualquer árvore *starlike* de diâmetro $2k$, que não seja uma medusa, possui conectividade algébrica menor que $a(P_{2k+1})$. ■

5.3 Árvores não-medusa

Para um dado k , $k \geq 1$, a classe de árvores medusa de diâmetro par e igual a $2k$, apresentada na seção anterior, não abrange todas as árvores de mesmo diâmetro e valor de conectividade algébrica igual a do caminho P_{2k+1} . Nesta seção serão estudadas árvores não-medusas com as propriedades acima referidas.

Seja $P_{l_0+1}^{u_0}$, $1 \leq l_0 \leq k$, o caminho com $l_0 + 1$ vértices enraizado numa de suas folhas u_0 ; seja $S_{l_1+1}^{u_1}$, a estrela com $l_1 + 1$ folhas, $l_1 \geq 2$, enraizada em uma de suas folhas u_1 e $R_{l_2}^{u_2}$, a árvore, enraizada em u_2 , resultante da coalescência do caminho P_3

com a estrela S_{l_2} , $l_2 \geq 2$, pela identificação de umas das folhas de P_3 com o vértice central de S_{l_2} , como é ilustrado na Figura 5.7.

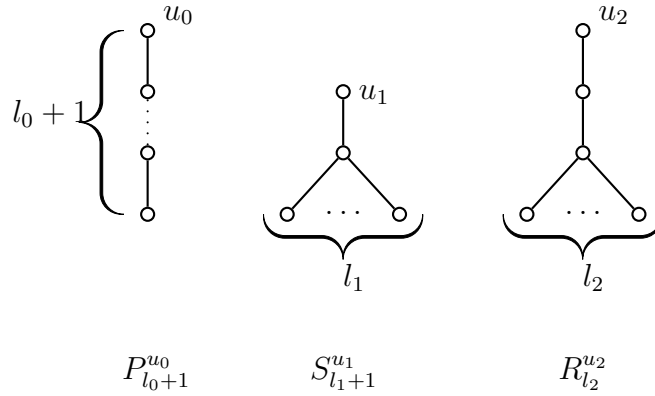


Figura 5.7: Estruturas das árvores $P_{l_0+1}^{u_0}$, $S_{l_1+1}^{u_1}$ e $R_{l_2}^{u_2}$ enraizadas em u_0 , u_1 e u_2 , respectivamente.

Definição 5.4. Dados $k \geq 3$, $n \geq 2k + 1$, $q \geq 0$, $r \geq 0$ e $s \geq 0$, seja $\mathcal{T}_{n,k}(q, r, s)$, um conjunto de árvores com n vértices e diâmetro $2k$ resultante da coalescência do caminho P_{2k+1} com q árvores isomorfas a $P_{l_0+1}^{u_0}$, r árvores isomorfas a $S_{l_1+1}^{u_1}$ e s árvores isomorfas a $R_{l_2}^{u_2}$, por identificação do vértice central c de P_{2k+1} com as respectivas raízes u_0 , u_1 e u_2 das árvores $P_{l_0+1}^{u_0}$, $S_{l_1+1}^{u_1}$ e $R_{l_2}^{u_2}$.

A Figura 5.8 ilustra a estrutura de uma árvore pertencente ao conjunto $\mathcal{T}_{n,k}(q, r, s)$.

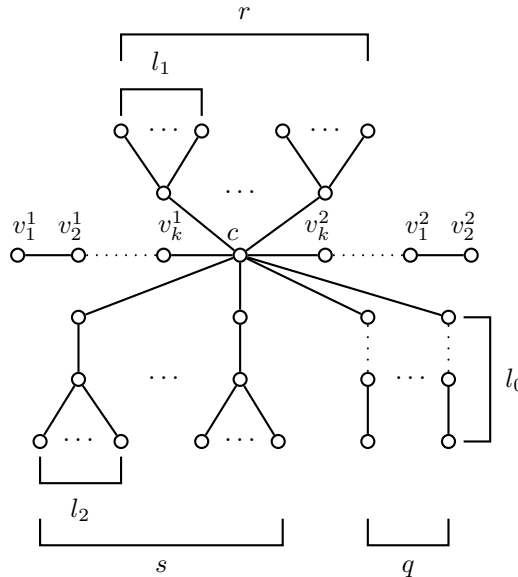


Figura 5.8: Estrutura de uma árvore $T \in \mathcal{T}_{n,k}(q, r, s)$.

Quando $r = 0$, $s = 0$ e $1 \leq l_0 \leq k$, o conjunto $\mathcal{T}_{n,k}(q, 0, 0)$ é exatamente o conjunto de árvores medusas com n vértices, diâmetro $2k$ e $q + 2$ ramos isomorfos a caminhos de no máximo k vértices, ou seja, $\mathcal{M}(n, 2k, q+2)$. Como foi visto na Seção

5.2, a conectividade algébrica de uma medusa de n vértices e diâmetro $2k$ é igual a do caminho P_{2k+1} , ou seja, para $q \geq 0$, se $T \in \mathcal{T}_{n,k}(q, 0, 0)$ então $a(T) = a(P_{2k+1})$.

Os Teoremas 5.5 e 5.6 determinam limites superiores para o número de folhas l_1 e l_2 das estruturas $S_{l_1+1}^{u_1}$ e $R_{l_2}^{u_2}$ para que a conectividade algébrica da árvore $T \in \mathcal{T}_{n,k}(q, r, s)$ seja igual a do caminho P_{2k+1} , isto é, $a(T) = a(P_{2k+1})$.

Definição 5.5. Dados $k \geq 3$, $n \geq 2k + 1$, sejam as árvores P_{2k+1} e $S_{l_1+1}^{u_1}$ com $l_1 \geq 2$. O conjunto $\mathcal{T}_{n,k}(0, 1, 0)$ é exatamente o conjunto de árvores T constituídas pela coalescência $P_{2k+1} \circ S_{l_1+1}^{u_1}$ dada pela identificação do vértice central c de P_{2k+1} com a raiz u_1 de $S_{l_1+1}^{u_1}$. Assim, $T = P_{2k+1} \circ S_{l_1+1}^{u_1}$. O novo vértice resultante da identificação será rotulado por c' (ver Figura 5.9).

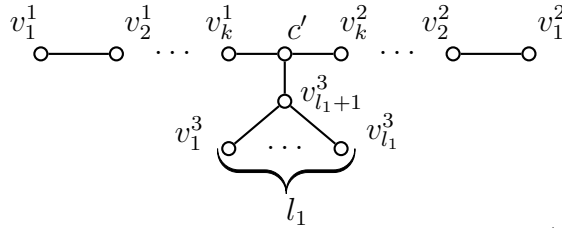


Figura 5.9: Estrutura de uma árvore $T \in \mathcal{T}_{n,k}(2, 1, 0)$.

Teorema 5.5. Dados $k \geq 3$ e $n > 2k + 1$, seja $a_k = a(P_{2k+1})$ e l_1 o número de folhas do ramo resultante da coalescência de P_{2k+1} com $S_{l_1+1}^{u_1}$. Se $T \in \mathcal{T}_{n,k}(0, 1, 0)$ então $a(T) = a_k$ se e somente se $l_1 \leq \left\lfloor \frac{(1-a_k)^2}{a_k} \right\rfloor$.

PROVA: Seja T uma árvore com a estrutura daquela da Figura 5.9, c' sua raiz e $v_{l_1+1}^3$ o vértice adjacente a c' . Pelo algoritmo *Diagonalizar* chega-se a

$$b(v_{l_1+1}^3) = (l_1 + 1 + \lambda) - l_1 \left(\frac{1}{1 + \lambda} \right) e$$

consequentemente,

$$b(v_{l_1+1}^3) = \frac{(1 + \lambda)^2 + l_1 \lambda}{1 + \lambda}.$$

O algoritmo *Diagonalizar* com $\lambda = -a_k$, resulta em,

$$\frac{(1 - a_k)^2 + l_1(-a_k)}{1 - a_k} \geq 0 e$$

como $0 < a_k < 1$, tem-se $1 - a_k > 0$. A desigualdade é verdadeira se e somente se

$$(1 - a_k)^2 + l_1(-a_k) \geq 0 e$$

$$l_1 \leq \frac{(1 - a_k)^2}{a_k}. \quad (5.13)$$

Tem-se que l_1 é inteiro, então $l_1 \leq \lfloor \frac{(1-a_k)^2}{a_k} \rfloor$. Logo, $d(v_{l_1+1}^3) \leq \lfloor \frac{(1-a_k)^2}{a_k} \rfloor + 1$. ■

EXEMPLO: Se $k = 4$, o diâmetro de uma árvore $T \in \mathcal{T}_{n,4}(0, 1, 0)$ é igual a 8. Para que $a(T) = a(P_9)$, o número máximo de folhas l_1 em T é igual a 6. A Figura 5.10 ilustra exemplos para os casos em que $a(T) = a(P_9)$ e $a(T) \neq a(P_9)$.

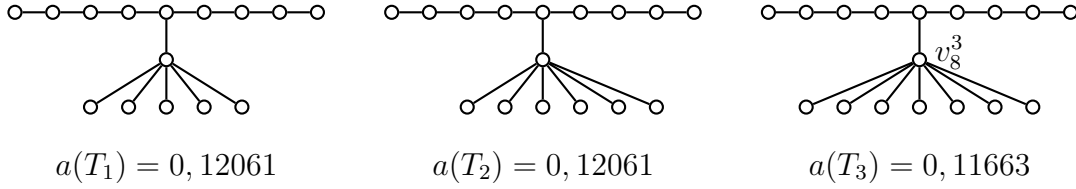


Figura 5.10: Árvores $T_i \in \mathcal{T}_{n,4}(0, 1, 0)$, com $n = 15, 16, 17$.

As árvores T_1 e T_2 na Figura 5.10 satisfazem o Teorema 5.5, mas veja que a árvore T_3 é tal que o vértice v_3^8 tem grau 8 e tem 7 folhas, não atendendo, portanto, ao Teorema 5.5. Neste caso, tem-se $a(T_3) = 0, 11663 < a(P_9)$.

O Corolário 5.2, apresentado mais adiante, decorrente do Teorema 5.5, estende o resultado para qualquer árvore $T \in \mathcal{T}_{n,k}(q, r, 0)$, com $q \geq 0$, $r \geq 0$, $1 \leq l_0 \leq k$ e $2 \leq l_1 \leq \lfloor \frac{(1-a_k)^2}{a_k} \rfloor$. Ou seja, se T satisfizer a tais condições, então $a(T) = a(P_{2k+1})$. Para isso será definido o conjunto de árvores com a estrutura igual a ilustrada na Figura 5.11.

Definição 5.6. Para dados $k \geq 3$, $n \geq 2k + 1$ e $a_k = a(P_{2k+1})$, sejam as árvores P_{2k+1} , $P_{l_0+1}^{u_0}$, com $1 \leq l_0 \leq k$, e $S_{l_1+1}^{u_1}$, com $2 \leq l_1 \leq \lfloor \frac{(1-a_k)^2}{a_k} \rfloor$. Seja \mathcal{T}_1 o conjunto de árvores $T \in \mathcal{T}_{n,k}(q, r, 0)$, constituídas pela coalescência $P_{2k+1} \circ P_{l_0+1}^{u_0} \circ S_{l_1+1}^{u_1}$, dada pela identificação do vértice central c de P_{2k+1} com a raiz u_0 de $P_{l_0+1}^{u_0}$ e com a raiz u_1 de $S_{l_1+1}^{u_1}$. Assim, $T = P_{2k+1} \circ P_{l_0+1}^{u_0} \circ S_{l_1+1}^{u_1}$. O novo vértice resultante da identificação será denotado por c'' (ver Figura 5.11).

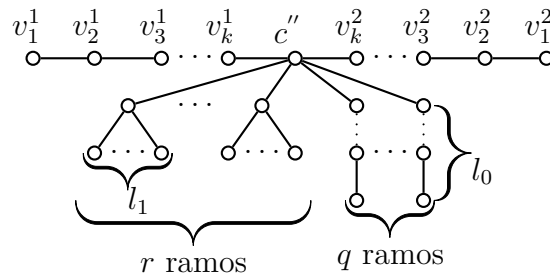


Figura 5.11: Estrutura de uma árvore $T \in \mathcal{T}_1$.

Corolário 5.2. Se $T \in \mathcal{T}_1$, então $a(T) = a(P_{2k+1})$.

PROVA: Seja $T \in \mathcal{T}_1$ e c'' , o vértice central de um caminho P_{2k+1} que compõe o diâmetro de T . Escolha dois ramos em c'' isomorfos a P_k e os rotule por 1 e 2. Os demais ramos em c'' têm rótulo r , $2 \leq r \leq d(c'')$. Do Teorema 5.2, tem-se que

durante a execução do algoritmo $Diagonalizar(T, -a_k)$, obtém-se $b(v_k^1) = b(v_k^2) = 0$. Então, independente dos valores dos irmãos de v_k^1 e v_k^2 , o algoritmo realiza as seguintes atribuições: $b(v_k^1) = 0$, $b(v_k^2) = 2$ e $b(c) = -\frac{1}{2}$. Tem-se ainda que qualquer ramo em c'' ou é isomorfo a um caminho de comprimento máximo k , ou é isomorfo a uma estrela S_{l_1} , com $l_1 \leq \left\lfloor \frac{(1-a_k)^2}{a_k} \right\rfloor$, resultante da coalescência de $S_{l_1+1}^{u_1}$ com P_{2k+1} , pela identificação do vértice central c de P_{2k+1} com a raiz u_1 de $S_{l_1+1}^{u_1}$. Então, seja $v_{l_1+1}^r$ o vértice de um ramo r em c'' , tal que o ramo r é isomorfo à estrela S_{l_1} . Do Teorema 5.2, sabe-se que para os vértices dos ramos isomorfos a caminhos de comprimento máximo k , o algoritmo $Diagonalizar$ obtém valores positivos. Logo, vértices de tais ramos não influenciam no valor da conectividade algébrica da árvore. Do Teorema 5.5, sabe-se que a chamada ao algoritmo $Diagonalizar(T, -a_k)$ gera um valor positivo para $v_{l_1+1}^r$, isto é, $b(v_{l_1+1}^r) \geq 0$. Assim, tal valor não interfere na quantidade de valores $b(v_i^r)$ negativos da árvore. Portanto, a_k continua sendo o segundo menor autovalor do laplaciano de T , independente do número de ramos isomorfos S_{l_1} ligadas a c . ■

O Teorema 5.5 dá um limite para o grau máximo do vértice $v_{l_1+1}^r$, $2 < r \leq d(c'')$, numa árvore de estrutura igual àquela da Figura 5.11, para que esta árvore possa ter a conectividade algébrica igual a do caminho P_{2k+1} .

O Teorema 5.6, a seguir, garante o número máximo para l_2 de uma árvore T com estrutura semelhante à ilustrada na Figura 5.12, de forma que $a(T) = a(P_{2k+1})$.

Definição 5.7. Para dados $k \geq 3$, $n \geq 2k+1$, sejam as árvores P_{2k+1} e $R_{l_2}^{u_2}$ com $l_2 \geq 2$. O conjunto $\mathcal{T}_{n,k}(0,0,1)$ é exatamente o conjunto de árvores T constituídas pela coalescência de $P_{2k+1} \circ R_{l_2}^{u_2}$ dada pela identificação do vértice central c de P_{2k+1} com a raiz u_2 de $R_{l_2}^{u_2}$. Assim, $T = P_{2k+1} \circ R_{l_2}^{u_2}$. O novo vértice resultante da identificação será rotulado por c''' (ver Figura 5.12).

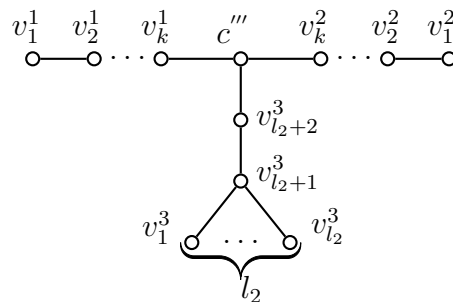


Figura 5.12: Estrutura de uma árvore $T \in \mathcal{T}_k(0,0,1)$.

Teorema 5.6. Dados $k \geq 3$, $n \geq 2k + 1$. Sejam $a_k = a(P_{2k+1})$ e l_2 o número de folhas do ramo resultante da coalescência de P_{2k+1} com $R_{l_2}^{u_2}$. Se $T \in \mathcal{T}_{n,k}(0,0,1)$ então $a(T) = a_k$ se e somente se $l_2 \leq \left\lfloor \frac{(1-a_k)^3 - a_k(1-a_k)}{a_k(2-a_k)} \right\rfloor$.

PROVA: Seja $T \in \mathcal{T}_{n,k}(0,0,1)$ com a estrutura e a rotulação ilustrada na Figura 5.12. Da execução do algoritmo *Diagonalizar*, tem-se que $b(v_{l_2+2}^3) = (2+\lambda) - \frac{1}{b(v_{l_2+1}^3)}$. Substituindo o valor de $b(v_{l_2+1}^3)$ no cálculo de $b(v_{l_2+2}^3)$,

$$b(v_{l_2+2}^3) = (2+\lambda) - \left(\frac{1+\lambda}{(1+\lambda)^2 + l_2\lambda} \right),$$

consequentemente,

$$b(v_{l_2+2}^3) = \frac{(2+\lambda)(1+\lambda)^2 + l_2\lambda(2+\lambda) - (1+\lambda)}{(1+\lambda)^2 + l_2\lambda}.$$

O algoritmo *Diagonalizar* com $\lambda = -a_k$, resulta em,

$$b(v_{l_2+2}^3) = \frac{(2-a_k)(1-a_k)^2 - l_2a_k(2-a_k) - (1-a_k)}{(1-a_k)^2 - l_2a_k}. \quad (5.14)$$

Do Teorema 5.5, para que $b(v_{l_2+1}^3)$ seja positivo, o número máximo de folhas ligadas a $v_{l_2+1}^3$ é $\lfloor \frac{(1-a_k)^2}{a_k} \rfloor$. Com $l_2 < \lfloor \frac{(1-a_k)^2}{a_k} \rfloor$, tem-se que $(1-a_k)^2 - l_2a_k > 0$. Então, basta analisar o limite para l_2 de forma que o numerador de (5.14) seja positivo, ou seja,

$$\begin{aligned} (2-a_k)(1-a_k)^2 - l_2a_k(2-a_k) - (1-a_k) &\geq 0 \\ (1-a_k)[(2-a_k)(1-a_k) - 1] - l_2a_k(2-a_k) &\geq 0 \\ (1-a_k)[(1-a_k)^2 - a_k] - l_2a_k(2-a_k) &\geq 0 \\ l_2a_k(2-a_k) &\leq (1-a_k)[(1-a_k)^2 - a_k] \end{aligned}$$

$$l_2 \leq \frac{(1-a_k)^3 - a_k(1-a_k)}{a_k(2-a_k)}. \quad (5.15)$$

Como $\frac{(1-a_k)^3 - a_k(1-a_k)}{a_k(2-a_k)} < \frac{(1-a_k)^2}{a_k}$, então uma vez que l_2 satisfaz o limite (5.15), automaticamente, l_2 satisfaz o limite (5.13) e o valor calculado para $v_{l_2+1}^3$ em T é positivo.

Tem-se que l_2 é inteiro, logo $l_2 \leq \lfloor \frac{(1-a_k)^3 - a_k(1-a_k)}{a_k(2-a_k)} \rfloor$. ■

EXEMPLO: Se $k = 5$, o diâmetro de uma árvore $T \in \mathcal{T}_{n,5}(0,0,1)$ é igual a 10. Considere a rotulação dos vértices de T como ilustrado na Figura 5.12. Do Teorema 5.6, para que $a(T) = a(P_{11})$, o número máximo de folhas para o ramo 3 de T é $l_2 \leq \lfloor \frac{(1-a_k)^3 - a_k(1-a_k)}{a_k(2-a_k)} \rfloor$. Ou seja, $l_2 \leq 4$. A Figura 5.13 ilustra exemplos para os casos em que $a(T) = a(P_{11})$ e $a(T) \neq a(P_{11})$.

Do Teorema 5.6, seja T uma árvore resultante da coalescência de um caminho P_{2k+1} com s árvores $R_{l_2}^{u_2}$, pela identificação do vértice central c de P_{2k+1} com as raízes

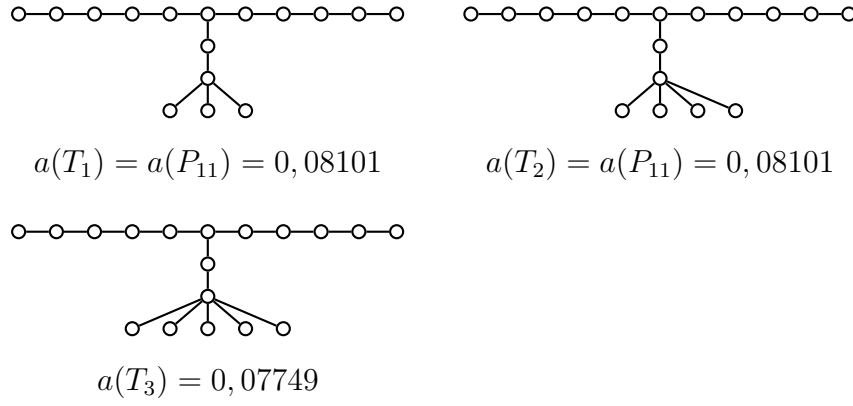


Figura 5.13: Árvores $T_i \in \mathcal{T}_{n,4}(2, 0, 1)$, com $n = 16, 17, 18$.

u_2 de cada árvore $R_{l_2}^{u_2}$. Assim, desde que $l_2 \leq \left\lfloor \frac{(1-a_k)^3 - a_k(1-a_k)}{a_k(2-a_k)} \right\rfloor$, a quantidade de ramos resultantes dessa coalescência não interfere no valor da conectividade algébrica de T .

O Corolário 5.3 estende o Teorema 5.6 para o conjunto de árvores definido a seguir.

Definição 5.8. *Dados $k \geq 3$, $n \geq 2k + 1$ e $a_k = a(P_{2k+1})$. Seja \mathcal{T}_2 o conjunto de árvores $T \in \mathcal{T}_{n,k}(q, r, s)$, com $q \geq 0$, $r \geq 0$ e $s \geq 0$, tal que $1 \leq l_0 \leq k$, $2 \leq l_1 \leq \left\lfloor \frac{(1-a_k)^2}{a_k} \right\rfloor$ e $2 \leq l_2 \leq \left\lfloor \frac{(1-a_k)^3 - a_k(1-a_k)}{a_k(2-a_k)} \right\rfloor$.*

Foi apresentado, na Figura 5.8, a estrutura de uma árvore da classe $\mathcal{T}_{n,k}(q, r, s)$. Veja que \mathcal{T}_2 é uma subclasse de $\mathcal{T}_{n,k}(q, r, s)$ onde os valores de l_0 , l_1 e l_2 satisfazem as restrições da Definição 5.8.

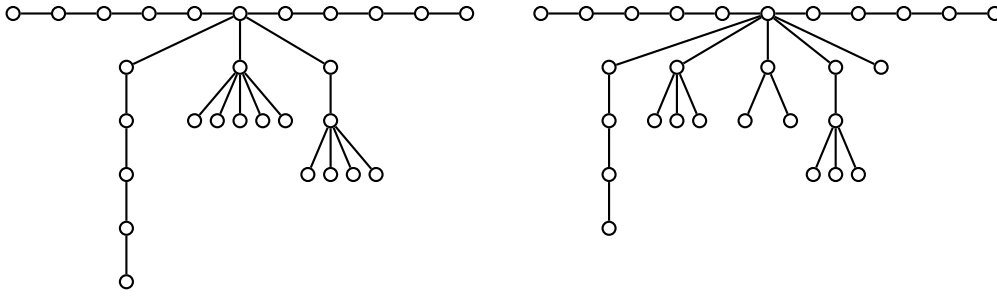
Corolário 5.3. *Dados $k \geq 3$ e $n \leq 2k + 1$. Seja $T \in \mathcal{T}_2$, então $a(T) = a(P_{2k+1})$.*

PROVA: A prova é similar àquela dada para o Corolário 5.2 e decorre deste e dos Teoremas 5.5 e 5.6. ■

EXEMPLO: Para $k = 5$ e $n = 28$, a Figura 5.14 apresenta duas árvores do conjunto \mathcal{T}_2 . A primeira é $T_3 \in \mathcal{T}_{28,5}(1, 1, 1)$ e a segunda é $T_4 \in \mathcal{T}_{28,5}(2, 2, 1)$, ambas possuem conectividade algébrica iguais ao do caminho P_{11} , isto é, $a(T_3) = a(T_4) = a(P_{11}) = 0,08101$.

Note que, de acordo com o Teorema 5.6, o número de folhas ligadas ao vértice $v_{l_2+1}^3$ da estrutura ilustrada na Figura 5.12, é menor que o limite de folhas para a estrutura da Figura 5.10 (veja Teorema 5.5), sendo que, no ramo 3, a distância de uma folha ao vértice c da árvore ilustrada na Figura 5.12 é maior do que a distância de uma folha no ramo 3 ao vértice c da árvore ilustrada na Figura 5.10.

Sejam as árvores T_5 , T_6 , T_7 e T_8 , dadas na Figura 5.10. Todas possuem diâmetro 11 ($k = 5$) e têm 3 ramos, dos quais, 2 são ramos de P_{2k+1} . O terceiro, é o único



$$T_3 \in \mathcal{T}_{28,5}(3, 1, 1)$$

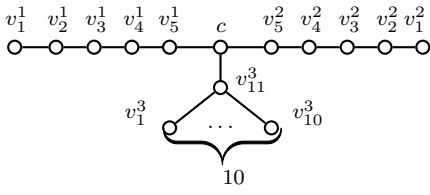
$$a(T_3) = 0,08101$$

$$T_4 \in \mathcal{T}_{28,5}(4, 2, 1)$$

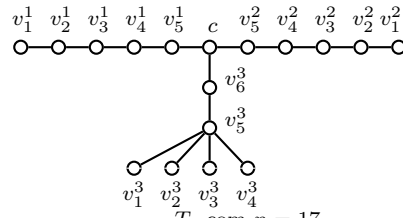
$$a(T_4) = 0,08101$$

Figura 5.14: Árvores $T \in \mathcal{T}_2$ com $k = 5$ e $n = 28$.

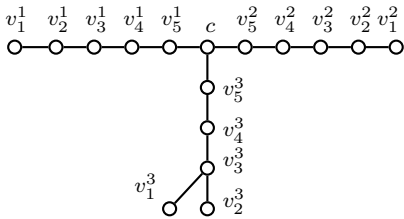
que não necessariamente é um caminho. A raiz c de cada árvore é o vértice central de P_{2k+1} . Dado que um *vértice quase pendente* é aquele adjacente a uma folha da árvore, é possível observar se o grau máximo do vértice quase pendente do terceiro ramo aumenta ou não, à medida que a distância de cada folha deste ramo ao centro c aumenta. Além disso, quantas folhas o vértice quase pendente suporta a fim de que a conectividade algébrica de cada uma das árvores seja a mesma do caminho P_{11} ?



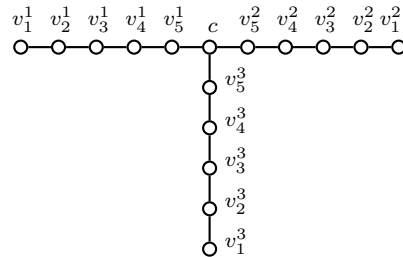
T_5 com $n = 22$
número máximo de folhas do ramo 3 é 10



T_6 com $n = 17$
número máximo de folhas do ramo 3 é 4



T_7 com $n = 16$
número máximo de folhas do ramo 3 é 2



T_8 com $n = 16$
o ramo 3 é isomorfo a P_5

Figura 5.15: Número máximo de folhas para cada nível do ramo 3 de uma árvore T com $k = 5$.

A árvore T_5 pertence ao conjunto $\mathcal{T}_{n,k}(0, 1, 0)$ e, do Teorema 5.5, sabe-se que o número máximo de folhas em v_{11}^3 é 10, para que a conectividade algébrica seja igual a de P_{11} . Para a árvore T_6 , para que valha a mesma propriedade, usando o Teorema 5.6, se tem que v_5^3 é 4. Já a árvore T_7 não pertence ao conjunto $\mathcal{T}_{n,k}(q, r, s)$

e a distância do vértice v_3^3 até c é 3. Neste caso, foi observado, por teste, que esta árvore tem conectividade algébrica igual a de P_{11} e 2 é o número máximo de folhas de v_3^3 para que esta propriedade aconteça, ou seja, ao se inserir mais uma folha a v_3^3 , a árvore resultante tem conectividade algébrica distinta daquela dada por P_{11} . Para $k = 5$, veja T_8 , o terceiro ramo é também isomorfo a P_5 e esta é uma árvore medusa. Neste exemplo, nota-se que à medida que a distância de c até uma folha v_i^3 aumenta, o número máximo de folhas de tal ramo diminui. Para o caso ilustrado por T_8 , tem-se o resultado provado no Teorema 5.7.

Sejam T uma árvore com estrutura igual a da Figura 5.16, x o número de folhas do ramo 3 e v_{x+1}^3 o seu único vértice que tem grau maior que 2.

Definição 5.9. Para $k > 1$ e $n \geq 2k + 1$, fixados, sejam os caminhos P_{2k+1} , P_k e a estrela $S_{1,x}$, $x \geq 1$, dados. O conjunto \mathcal{T}_0 é um conjunto de árvores T constituídas pela coalescência de P_{2k+1} com a árvore P_k e, da árvore resultante com $S_{1,x}$, pela identificação do vértice central c de P_{2k+1} com uma das folhas de P_k e a identificação da outra folha de P_k com o vértice central de $S_{1,x}$ (veja Figura 5.16).

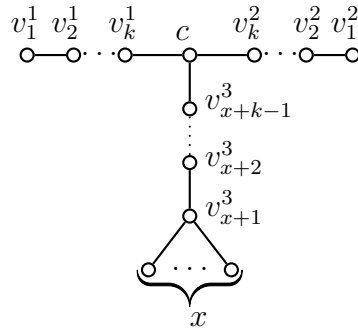


Figura 5.16: Árvore T com diâmetro $2k$.

Teorema 5.7. Dado $k > 0$. Seja T uma árvore enraizada em c da classe \mathcal{T}_0 e seja x o número de folhas do ramo 3. Sendo k a distância de c até a folha do ramo 3, tem-se $a(T) = a(P_{2k+1})$ se e somente se $x = 1$.

PROVA: Seja $a_k = a(P_{2k+1})$. Os valores para os vértices gerados pela chamada $Diagonalizar(T, -a_k)$ são tais que

$$b(v_i^1) = b(v_i^2) = \begin{cases} 1 - a_k, & \text{se } i = 1; \\ (2 - a_k) - \frac{1}{b(v_{i-1}^1)}, & \text{caso contrário.} \end{cases}$$

$$b(v_i^3) = \begin{cases} 1 - a_k, & \text{se } 1 \leq i \leq x; \\ (x + 1 - a_k) - \frac{x}{1 - a_k}, & \text{se } i = x + 1; \\ (2 - a_k) - \frac{1}{b(v_{i-1}^3)}, & \text{caso contrário.} \end{cases}$$

Como $0 < a_k < 1$, então $1 - a_k > 0$. Se a distância de v_{x+1}^3 a c é igual a $k - 1$, então o número de vértices entre v_{x+1}^3 e c é igual ao número de vértices de v_2^1 a c . Do Teorema 5.2, sabe-se que $b(v_k^1) = b(v_k^2) = 0$ e $b(v_i^1) = b(v_i^2) > 0$, para $1 \leq i \leq k - 1$.

Analisando os valores dos vértices do ramo 3, tem-se $b(v_i^3) = 1 - a_k > 0$, quando $1 \leq i \leq x$. No caso em que $x = 1$, este é isomorfo a P_k e, do Teorema 5.3, $a(T) = a(P_{2k+1})$. Supondo $x > 1$,

$$\begin{aligned} b(v_{x+1}^3) &= (x + 1) - a_k - \frac{x}{1 - a_k} \\ &= \frac{((x + 1) - a_k)(1 - a_k) - x}{1 - a_k} \\ &= \frac{(1 - a_k)^2 - xa_k}{1 - a_k}. \end{aligned}$$

Comparando $b(v_{x+1}^3)$ com $b(v_2^1)$,

$$b(v_{x+1}^3) = \frac{(1 - a_k)^2 - xa_k}{1 - a_k} < \frac{(1 - a_k)^2 - a_k}{1 - a_k} = b(v_2^1). \quad (5.16)$$

Para os outros vértices dos ramos 1 e 3, respectivamente, se tem,

$$\begin{aligned} b(v_i^1) &= (2 - a_k) - \frac{1}{b(v_{i-1}^1)}, \quad 3 \leq i \leq k; e \\ b(v_j^3) &= (2 - a_k) - \frac{1}{b(v_{j-1}^3)}, \quad x + 2 \leq j \leq x + k - 1. \end{aligned}$$

Note que, para $2 \leq i \leq k$, os vértices v_i^1 e v_{x+i-1}^3 estão no mesmo nível em T . Da comparação 5.16, tem-se que $b(v_{x+1}^3) < b(v_2^1)$. Então,

$$\begin{aligned} b(v_{x+2}^3) &= (2 - a_k) - \frac{1}{b(v_{x+1}^3)}; e \\ b(v_3^1) &= (2 - a_k) - \frac{1}{b(v_2^1)}. \end{aligned}$$

Como $b(v_{x+1}^3) < b(v_2^1)$, então $\frac{1}{b(v_{x+1}^3)} > \frac{1}{b(v_2^1)} \Rightarrow b(v_{x+2}^3) < b(v_3^1)$.

Seguindo raciocínio análogo para os vértices restantes, ancestrais de v_3^1 e de v_{x+2}^3 , tem-se que para j , $x + 1 \leq j < x + k - 1$, $v_j^3 = 0$ ou $v_j^3 < 0$. No primeiro caso, o algoritmo *Diagonalizar* faz $b(v_j^3) = 2$ e $b(v_{j+1}^3) = -\frac{1}{2}$. Desta forma, tem-se um valor negativo atribuído a um vértice diferente de c , indicando que a_k não é o segundo menor autovalor do Laplaciano de T . No segundo caso, o valor de v_j^3 indica que a_k não é a conectividade algébrica de T . Em ambos os casos, sabe-se que $a(T) < a(P_{2k+1})$.

Portanto, para que $a(T) = a(P_{2k+1})$, o valor máximo para x é 1. Ou seja, quando a distância de uma folha do ramo 3 até c é igual a k , o ramo deve ser isomorfo a P_k .

■

5.4 SGI's para as árvores de \mathcal{T}_2

Nesta seção, será analisado o comportamento das *SGI's* aplicadas às árvores do conjunto \mathcal{T}_2 . Como foi visto, cada árvore desse conjunto tem conectividade algébrica igual a de P_{2k+1} . A análise será realizada para todas as árvores $T \in \{\mathcal{T}_{15,4}(q, r, s) \cap \mathcal{T}_2\}$. Tais árvores possuem conectividade algébrica igual a $a(T) = 0, 12061$ e, na Figura 5.17, são apresentadas em ordem não crescente de *SGI*.

Como visto na seção anterior, para valores nulos de r e s , o conjunto de árvores $T_{n,k}(q, r, s)$ se resume às árvores medusas. Das árvores apresentadas na Figura 5.17, nove são medusas: $T_{15,4}(6, 0, 0)$, $T_{15,4}(5, 0, 0)$, $T'_{15,4}(4, 0, 0)$, $T''_{15,4}(4, 0, 0)$, $T'_{15,4}(3, 0, 0)$, $T''_{15,4}(3, 0, 0)$, $T'''_{15,4}(3, 0, 0)$, $T'_{15,4}(2, 0, 0)$ e $T''_{15,4}(2, 0, 0)$. Quando se trata dessas árvores, sua *SGI* pode ser extraída diretamente de sua representação. No exemplo, cada árvore tem $n = 13$ vértices, diâmetro $d = 8$, comprimento máximo de um ramo em c é $k = 4$ e o número de ramos em c é $4 \leq r \leq 8$. Desta forma, a *SGI* de uma medusa $\mathcal{M}(n, d, r)$ é:

$$SGI(\mathcal{M}(n, d, r)) = [r, \underbrace{2, \dots, 2}_{n-r-1}].$$

Então:

$$SGI(T_{15,4}(6, 0, 0)) = [8, 2, 2, 2, 2, 2, 2, 2];$$

$$SGI(T_{15,4}(5, 0, 0)) = [7, 2, 2, 2, 2, 2, 2, 2];$$

$$SGI(T'_{15,4}(4, 0, 0)) = SGI(T''_{15,4}(4, 0, 0)) = [6, 2, 2, 2, 2, 2, 2, 2];$$

$$SGI(T'_{15,4}(3, 0, 0)) = SGI(T''_{15,4}(3, 0, 0)) = SGI(T'''_{15,4}(3, 0, 0)) = [5, 2, 2, 2, 2, 2, 2, 2];$$

$$SGI(T'_{15,4}(2, 0, 0)) = SGI(T''_{15,4}(2, 0, 0)) = [4, 2, 2, 2, 2, 2, 2, 2].$$

Proposição 5.1. *As árvores do conjunto \mathcal{T}_2 maximizam a conectividade algébrica das árvores $T_{n,k}(q, r, s)$.*

PROVA: É fácil ver que $\mathcal{T}_2 \subsetneq \mathcal{T}_{n,k}(q, r, s)$. Faça $a_k = a(P_{2k+1})$. Do Teorema 5.6, tem-se que o número de folhas de um ramo resultante da coalescência de P_{2k+1} com $R_{l_2}^{u_2}$ é limitado por $l_2 \leq \left\lfloor \frac{(1-a_k)^3 - a_k(1-a_k)}{a_k(2-a_k)} \right\rfloor$. Substituindo o valor de a_k por $a(P_9) = 0, 12061$ e calculando o limite de l_2 tem-se $l_2 \leq 2$. Do Teorema 5.6, quando T possui um ramo r resultante da coalescência de P_{2k+1} com $R_{l_2}^{u_2}$ com $l_2 > \left\lfloor \frac{(1-a_k)^3 - a_k(1-a_k)}{a_k(2-a_k)} \right\rfloor$, o algoritmo atribui um valor negativo para algum vértice do ramo r . Logo, existe algum autovalor da matriz laplaciana de T menor que $a(P_{2k+1})$. Ou seja, $a(T) < a(P_{2k+1})$. ■

A Figura 5.18 apresenta duas árvores de $\mathcal{T}_{15,4}(q, r, s)$ que não pertencem a \mathcal{T}_2 e suas respectivas conectividades algébricas. São elas: $T_{15,4}(3, 0, 1)$ e $T_{15,4}(2, 0, 1)$. Em ambas, tem-se um ramo resultante da coalescência de P_{2k+1} com $R_{l_2}^{u_2}$, porém o número de folhas l_2 são, respectivamente, 3 e 4, o que viola o limite $l_2 \leq 2$.

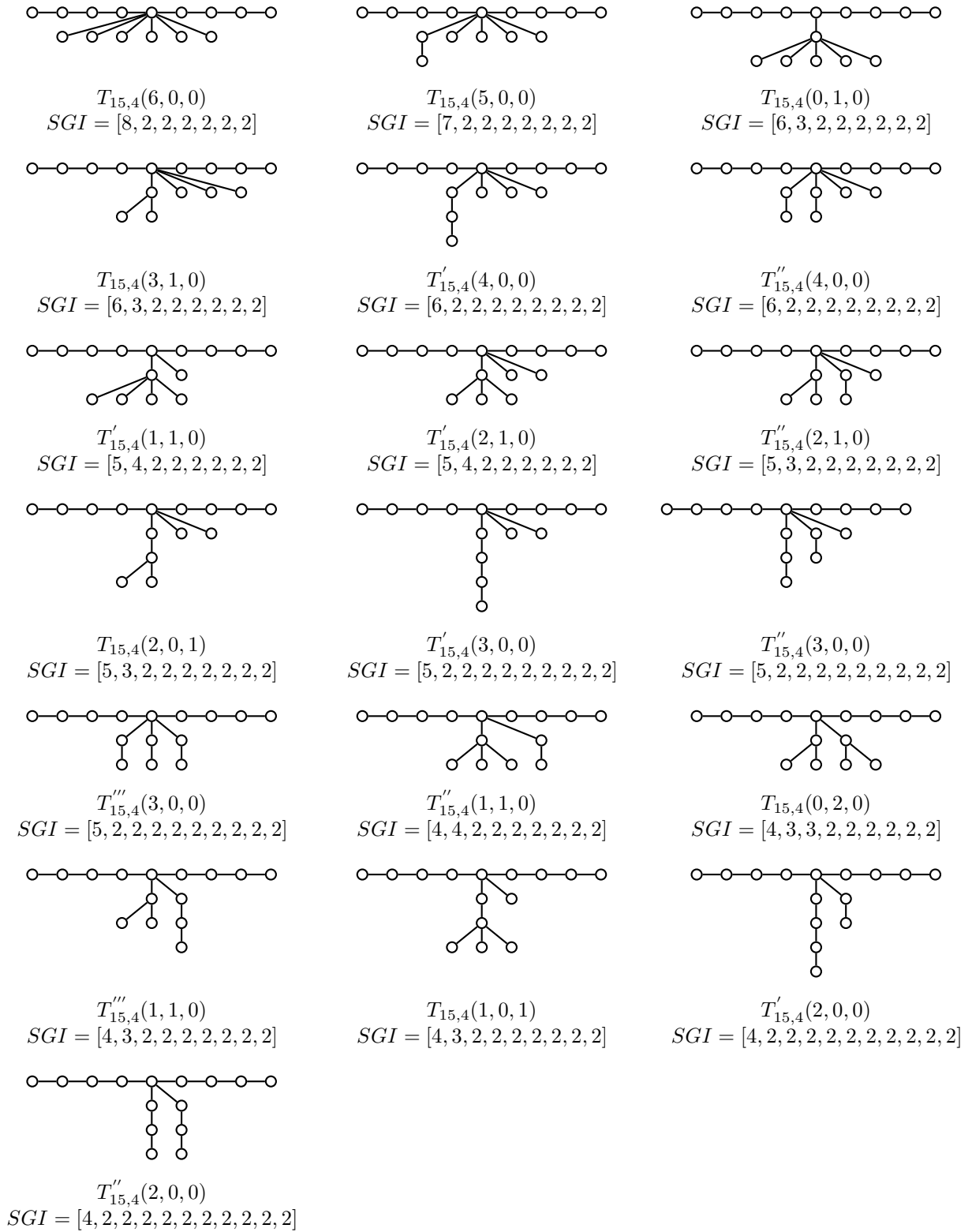
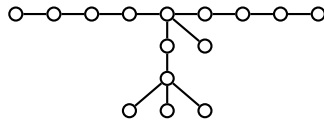
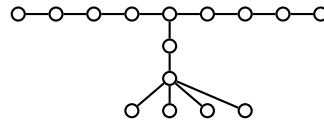


Figura 5.17: Árvore $T \in \mathcal{T}_{15,4}(q, r, s)$ com $n = 15$ vértices e $a(T) = 0, 12061$.



$$a(T_{15,4}(3, 0, 1)) = 0, 11264$$



$$a(T_{15,4}(2, 0, 1)) = 0, 10002$$

Figura 5.18: Árvores de $\mathcal{T}_{15,4}(q, r, s)$ que não pertencem a \mathcal{T}_2 .

Para finalizar este capítulo é conveniente ressaltar que tanto a conectividade algébrica quanto a *SGI* apresentam uma ordenação não estrita das árvores quanto à conectividade e, para algumas classes em que a primeira medida não consegue diferenciar seus elementos quanto a esta característica, a *SGI* pode ser uma alternativa para este feito, dado que em alguns exemplos, as *SGI*'s induziram uma ordenação total das árvores envolvidas.

Capítulo 6

Conclusões e trabalhos futuros

6.1 Conclusões

Este trabalho teve como objetivo inicial estudar a conectividade de grafos cordais, propondo um novo invariante que pudesse diferenciar a conectividade entre grafos cordais k -conexos, nos casos em que os invariantes clássicos não fossem capazes de fazer tal distinção. Tal invariante foi definido como uma n -upla de números naturais, chamada de *sequência de componentes conexas* e abreviada por SCC , onde cada ordenada é igual ao número de componentes conexas resultantes da remoção de cada separador minimal de vértices do grafo, abreviado por SMV . Sabe-se que os grafos cordais k -sep determinam uma importante classe dos grafos cordais. Assim, dado o número de cliques maximais de um grafo k -sep, foi apresentado um algoritmo que gera todas as SCC 's possíveis para tais grafos. As árvores, por sua vez, são casos particulares dos grafos k -sep. Utilizou-se então as árvores para o estudo detalhado do novo invariante. Neste caso, os graus dos vértices internos de cada árvore são suficientes para caracterizar sua SCC e este invariante passou a ser chamado de *sequência de graus internos*, abreviada por SIG .

A conectividade algébrica é um invariante espectral muito mais sensível para mensurar a conectividade de uma árvore que os invariantes clássicos. Por exemplo, a conectividade de vértices e de arestas são incapazes de medir esta propriedade para as árvores, uma vez que estes têm os mesmos valores para esta classe. Então, foi decidido comparar as SIG 's com as conectividades algébricas de classes de árvores que já possuem resultados conhecidos para o segundo invariante.

Estudou-se o comportamento da SIG e da conectividade algébrica nas seis classes de árvores C_i , $i = 1, \dots, 6$, que maximizam o segundo invariante para árvores com $n \geq 15$ vértices. Para os quais sabe-se que todas as árvores possuem diâmetro máximo igual a 4 e se conhece uma ordenação pela conectividade algébrica. Para tais grafos, observou-se que os dois invariantes apresentaram ordenações não isomorfas.

Além disso, para as árvores dessas classes com mesmo número de vértices, as *SGI*'s definiram uma ordenação total e estrita entre seus elementos, isto é, não há duas árvores não isomorfas pertencentes a tais classes com mesma *SGI*.

Dentre as classes estudadas, duas apresentaram árvores não isomorfas com mesma conectividade algébrica, são elas C_3 e C_6 . Para as árvores de C_3 e C_6 , a *SGI* conseguiu diferenciar todas elas quanto a sua conectividade. O comportamento deste invariante para tais classes levanta a questão de quais características estruturais levam as árvores a ter a mesma conectividade algébrica. A partir daí, iniciou-se o estudo de árvores com tal propriedade e diâmetro $2k$, $k > 3$, a fim de identificar características estruturais que não interferem em sua conectividade algébrica. Foi considerada a classe das *medusas*, casos particulares de *starlikes*, e foi provado que as medusas de diâmetro $2k$ possuem conectividade algébrica igual ao do caminho P_{2k+1} . Além disso, elas maximizam o valor de tal invariante para as *starlikes* de mesmo diâmetro.

A partir do estudo das medusas, definiu-se uma classe mais abrangente de árvores, $T_{n,k}(q, r, s)$, onde todas elas possuem diâmetro $2k$ e não são necessariamente medusas. Dela identificou-se o conjunto \mathcal{T}_2 , onde todas as árvores possuem conectividade algébrica igual a do caminho P_{2k+1} . Foi provado que as árvores de \mathcal{T}_2 maximizam a conectividade algébrica das árvores de $T_{n,k}(q, r, s)$. Além das características estruturais analisadas neste trabalho, observou-se que as árvores com diâmetro par fixo e mesma conectividade algébrica não se restringem ao conjunto \mathcal{T}_2 .

Como o estudo em árvores foi se aprofundando e gerando resultados interessantes, o desenvolvimento do trabalho não conseguiu chegar ao estudo de outras subclasses de grafos cordais. Ainda no estudo de classes de árvores, observou-se que as árvores com diâmetro fixo e mesma conectividade algébrica não se restringem ao conjunto \mathcal{T}_2 , por exemplo, as árvores da classe C_6 , apresentada no Capítulo 5, não pertencem a tal conjunto.

A proposta para trabalhos futuros segue duas vertentes, uma indica o aprofundamento do estudo da conectividade algébrica das árvores em classes mais abrangentes que aquelas já estudadas e a outra aponta para estender a comparação entre *SCC*'s e a conectividade algébrica para outras subclasses de grafos cordais.

6.2 Trabalhos futuros

No Capítulo 5 foi apresentado o conjunto de árvores \mathcal{T}_2 onde, para um dado k , todas as árvores possuem diâmetro $2k$ e conectividade algébrica igual ao do caminho P_{2k+1} . Observou-se ainda que existem mais árvores além daquelas em \mathcal{T}_2 que possuem tais características. Alguns exemplos são apresentados na Figura 6.1.

A partir deste trabalho, pretende-se continuar o estudo da estrutura de árvores

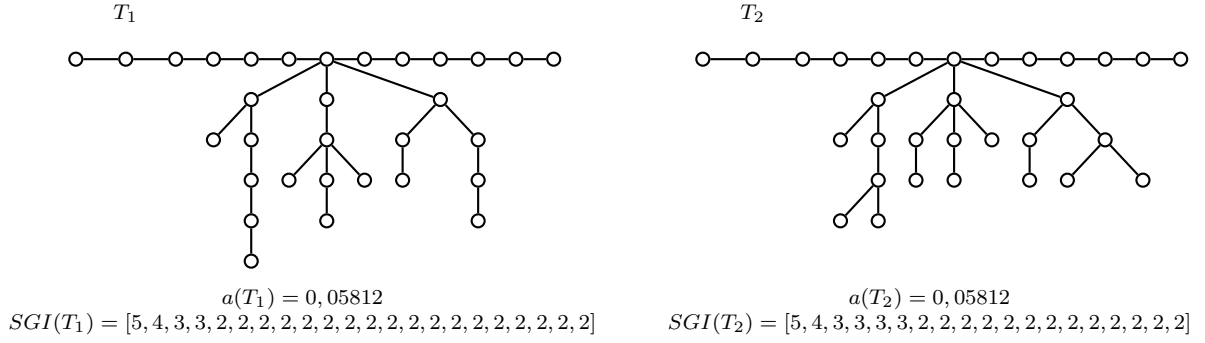


Figura 6.1: Árvores T_1 e T_2 não pertencentes a \mathcal{T}_2 , tais que $a(T_1) = a(T_2) = a(P_{13})$.

que possuem a mesma conectividade algébrica e analisar o comportamento da SGI para tais árvores. Alguns fatos já foram observados. Por exemplo, as árvores da classe C_6 possuem a mesma conectividade algébrica e esta difere da conectividade algébrica do caminho P_5 . Neste caso, se $T(k, p, q) \in C_6$ então $a(T(k, p, q)) = 2 - \sqrt{3}$. A restrição para que $T(k, p, q)$ seja da classe C_6 é $k \geq 2$. Ou seja, a menor árvore da classe C_6 é $T(2, 0, 0)$ com $n = 7$ vértices. A Figura 6.2 apresenta alguns exemplos de árvores da classe C_6 . Em tal amostra, têm-se quantidades diferentes de ramos isomorfos a $K_{1,2}$, $K_{1,1}$ e K_1 . A primeira árvore da Figura 6.2 é a $T(2, 0, 0)$ com vértice central c . Observe que a remoção de qualquer folha de $T(2, 0, 0)$ resulta em uma árvore da classe C_5 (que possui conectividade algébrica maior que $2 - \sqrt{3}$). Tem-se que $T(2, 0, 6)$ é resultante da coalescência de $T(2, 0, 0)$ com seis caminhos isomorfos a P_2 , pela identificação do vértice c de $T(2, 0, 0)$ com um vértice de cada caminho P_2 e o vértice resultante da operação é rotulada por c' . Observe que os ramos isomorfos a K_1 não interferem na conectividade algébrica da árvore, isto é, $a(T(2, 0, 6)) = a(T(2, 0, 0))$. De forma análoga, $T(2, 3, 0)$ é obtida pela coalescência de $T(2, 0, 0)$ com três caminhos P_3 , pela identificação de c com uma folha de cada caminho, e $T(4, 0, 0)$ é resultado da coalescência de $T(2, 0, 0)$ com duas estrelas $K_{1,3}$ pela identificação do vértice c com uma folha de cada estrela. Por fim, a árvore $T(3, 1, 1)$ é resultado da coalescência de $T(2, 0, 0)$ com as estruturas $K_{1,2}$, $K_{1,1}$ e P_2 , pela identificação do vértice c de $T(2, 0, 0)$ com uma folha de cada estrutura. Em todas essas árvores, o número de ramos isomorfos a $K_{1,2}$, $K_{1,1}$ e K_1 não interfere na conectividade algébrica de qualquer árvore $T \in C_6$, ou seja, $a(T) = a(T(2, 0, 0))$.

Note que em $T(2, 0, 0)$ existe uma simetria entre os ramos em c , isto é, os ramos em c são isomorfos entre si. Com tal observação, resolveu-se analisar a conectividade algébrica de árvores que possuem tal característica. Alguns testes foram realizados para árvores T com diâmetro $2k > 4$ e com dois ramos isomorfos entre si. Para toda folha u das árvores T analisadas, a distância máxima entre u e o vértice c deve ser no máximo igual a k , ou seja, $dist(c, u) \leq k$. Algumas das árvores analisadas são ilustradas na Figura 6.3.

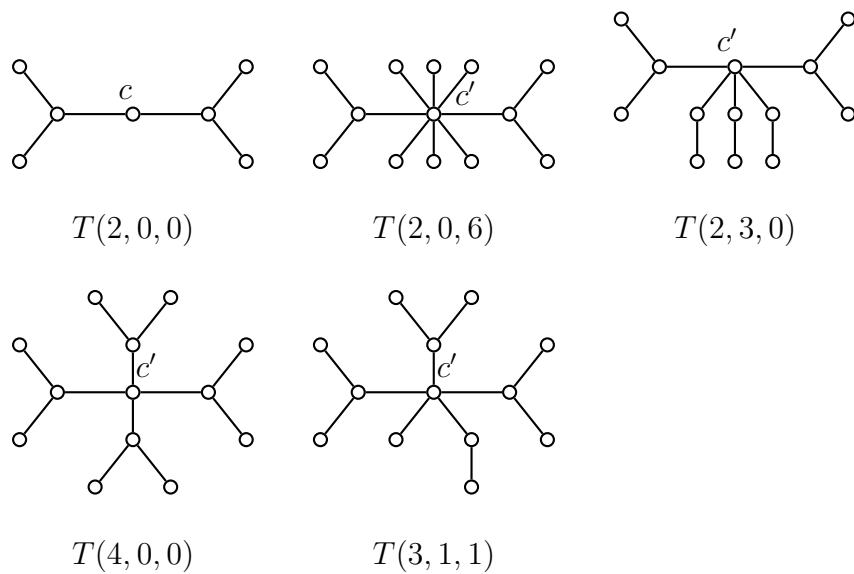


Figura 6.2: Análise das estruturas de árvores da classe C_6 .

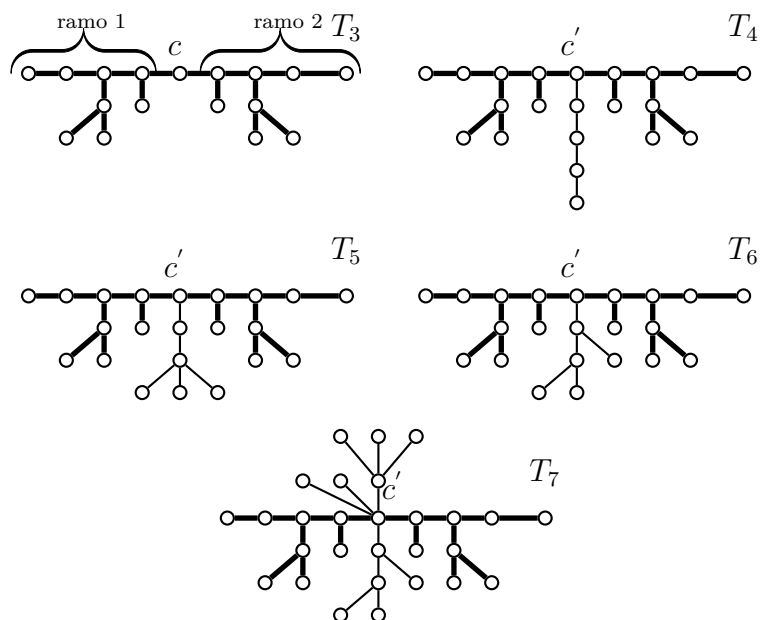


Figura 6.3: Árvores com $a(T_i) = 0,06367$.

A Figura 6.3 apresenta um conjunto de árvores com conectividade algébrica igual a $a(T_i) = 0,06367$, dentre as quais aquela que possui menor número de vértices é a árvore T_3 . Esta possui diâmetro 8 e dois ramos em c isomorfos entre si, o que caracteriza uma simetria em T_3 . Na figura, tais ramos foram rotulados por ramo 1 e ramo 2. As árvores T_4 , T_5 , T_6 e T_7 possuem, cada uma delas, uma subárvore induzida isomorfa a T_3 com vértice central c' . Nessas árvores, além dos ramos que compõem a subárvore isomorfa a T_3 , têm-se outros ramos em c' não isomorfos ao ramo 1 (ou 2). Em tais ramos, a distância máxima de qualquer folha u ao vértice c' é igual a 4, ou seja, $dist(u, c') \leq 4$. Nos casos analisados, parece que os ramos em c' que não são isomorfos àqueles de T_3 não interferem na conectividade algébrica da árvore, desde que não mudem seu diâmetro.

O estudo da conectividade de grafos iniciou-se com a classe de árvores, para os quais vários resultados foram obtidos e outros já se encontram em andamento. Como foi visto, as árvores possuem todos os SMV 's com cardinalidade 1 e todas as cliques maximais possuem cardinalidade 2. Neste caso, tem-se que o número de componentes conexas resultantes da remoção de um SMV de uma árvore é igual ao grau do vértice interno referente ao SMV . Por isso, uma SCC em árvores equivale a uma SGL . Mas para outras classes de grafos cordais, este fato não ocorre. Pretende-se estender o estudo comparativo do invariante SCC e da conectividade algébrica quando estes são aplicados a grafos bloco, para qual algumas observações já foram feitas e serão apresentadas mais adiante.

Os grafos bloco definidos por Harary [23] compõem uma classe de grafos cordais k -sep. As definições já existentes na literatura serão adaptadas para a notação já utilizada no decorrer do trabalho. As árvores são um caso particular de grafos bloco, onde todas as cliques maximais têm cardinalidade 2. Várias caracterizações e aplicações de grafos bloco foram estudadas em [4, 9, 10, 38].

Em [14], Harary e Prins apresentaram a definição de *block-cut-vertex graph* que, neste trabalho, será denotada por $bc(G)$. Sua definição, apresentada a seguir, foi extraída de [24] e adaptada para grafos bloco.

Definição 6.1. *Para um grafo bloco conexo G , sejam $\{B_i\}$ o conjunto formado pelas cliques maximais de G e $\{c_j\}$ o conjunto de SMV 's de G . $bc(G)$ é definido como o grafo com conjunto de vértices igual a $\{B_i\} \cup \{c_j\}$, com dois vértices adjacentes se um deles corresponde a uma clique maximal B_i e o outro a um SMV c_j , tal que c_j é um vértice de B_i .*

Um resultado importante para grafos $bc(G)$ conhecido na literatura é apresentado no Teorema 6.1. O enunciado a seguir foi extraído de [24].

Teorema 6.1. *Um grafo G é um grafo $bc(G)$ de um grafo H se e somente se ele é uma árvore onde a distância entre quaisquer duas folhas é par.*

Com o Teorema 6.1, sabe-se que todo grafo $bc(G)$ é uma árvore. A Figura 6.4 ilustra um grafo $bc(G)$ de um grafo G . Os vértices referentes aos blocos são valorados com a cardinalidade do bloco que cada um representa.

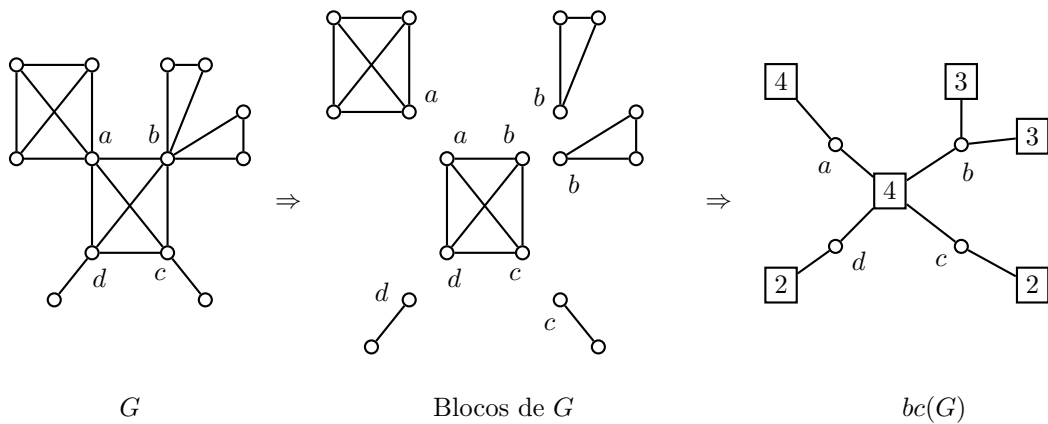


Figura 6.4: Exemplo de um grafo bloco G , os blocos de G e sua representação $bc(G)$.

Pode-se considerar que um grafo $bc(G)$ é uma representação de um grafo G em um árvore. Com um grafo $bc(G)$ é possível analisar o comportamento da conectividade algébrica de um grafo G quanto a forma da árvore gerada. Como as árvores também são grafos bloco tem-se que para cada árvore T existe um grafo $bc(T)$. Observe que um grafo $bc(T)$ possui mais vértices e mais arestas que a árvore T que ele representa. Mas para a análise de alguns casos de grafos bloco, realizada mais adiante, serão utilizadas as mesmas $bc(T)$ do exemplo seguinte. A Figura 6.5 apresenta uma amostra de árvores de mesma conectividade algébrica e seus respectivos $bc(T)$.

Como o número de vértices de um bloco em G é atribuído ao seu respectivo vértice em $bc(G)$, tem-se que mais de um grafo bloco pode ter a mesma representação $bc(G)$, diferenciando-se apenas dos valores nos vértices referentes aos blocos. Por exemplo, considere as representações ilustradas para as árvores da Figura 6.5. Existem grafos bloco em que todos os blocos são isomorfos a K_4 e possuem a mesma representação $bc(G)$ que as árvores apresentadas na Figura 6.5. Neste caso, cada vértice referente a um bloco tem valor 4. Tais grafos são ilustrados na Figura 6.6.

Note que todos os grafos da Figura 6.6 possuem mesma conectividade algébrica, ou seja, $a(G_1) = a(G_2) = a(G_3) = 0,09305$. Com isso, observa-se que existem grafos bloco não isomorfos com mesma conectividade algébrica. Com a representação $bc(G)$ é possível fazer uma análise estrutural de tais grafos de forma a identificar características que os levam a ter mesma conectividade algébrica. Uma análise semelhante foi realizada para as árvores. Porém, nas árvores a SCC equivale a SGI enquanto que em grafos bloco o grau de um vértice de corte não indica o número de componentes conexas resultantes de sua remoção. Para grafos cordais em geral, tal informação é facilmente obtida por algoritmos de tempo linear apresentados em [28, 33].

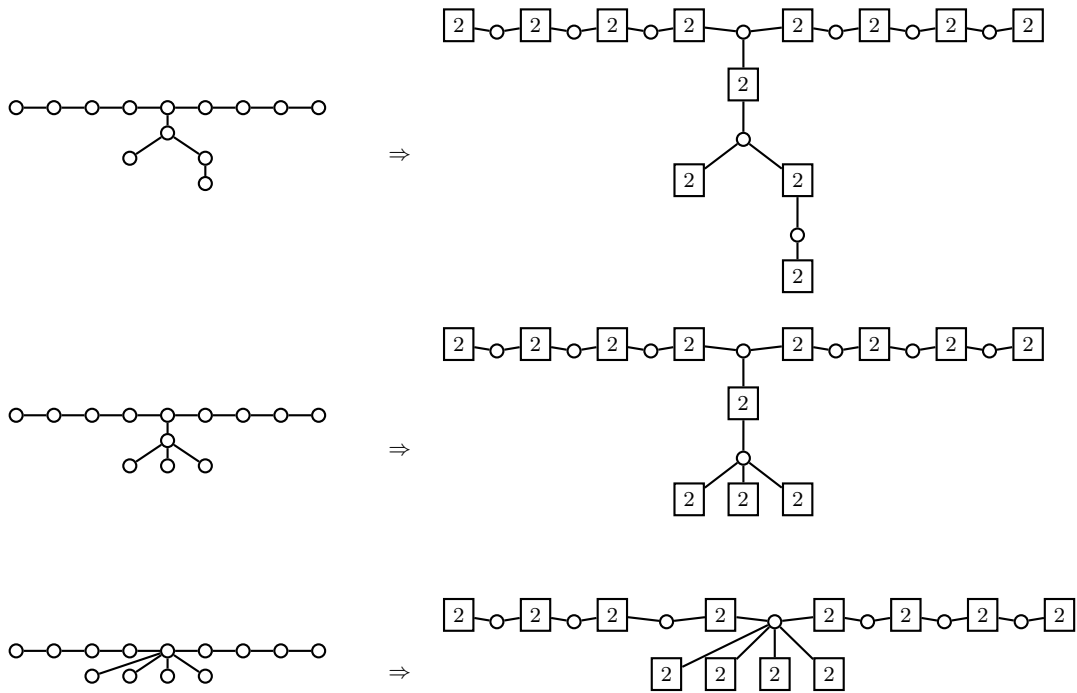


Figura 6.5: Árvores com $a(T) = 0, 12061$ e suas respectivas $bc(T)$.

A Figura 6.7 ilustra grafos bloco com seus respectivos valores de conectividade algébrica e SCC 's. Todos os grafos da Figura 6.7 possuem sete cliques maximais isomorfos a K_4 , $n = 22$ vértices e $m = 42$ arestas. Os grafos estão ordenados de forma não decrescente de conectividade algébrica. Observa-se que o sentido de crescimento das duas medidas são semelhantes. Tanto pela conectividade algébrica quanto pela SCC , o grafo G_4 é o menos conexo e o grafo G_{10} é o mais conexo dentre os grafos da figura. Tem-se ainda que os grafos G_4 , G_5 e G_6 possuem mesma SCC e conectividade algébrica diferentes, enquanto que os grafos G_6 e G_7 possuem mesma conectividade algébrica e SCC 's diferentes. O mesmo ocorre com G_8 e G_9 . Graças à representação $bc(G)$ foi possível observar tal comportamento em grafos bloco. Note que isso também foi verificado no estudo das árvores. Com tais observações, propõe-se a continuar o estudo comparativo das duas medidas para a classe de grafos bloco. Além disso, pretende-se estudar características estruturais a partir dos quais é possível identificar se a conectividade algébrica de dois grafos bloco não isomorfos são iguais.

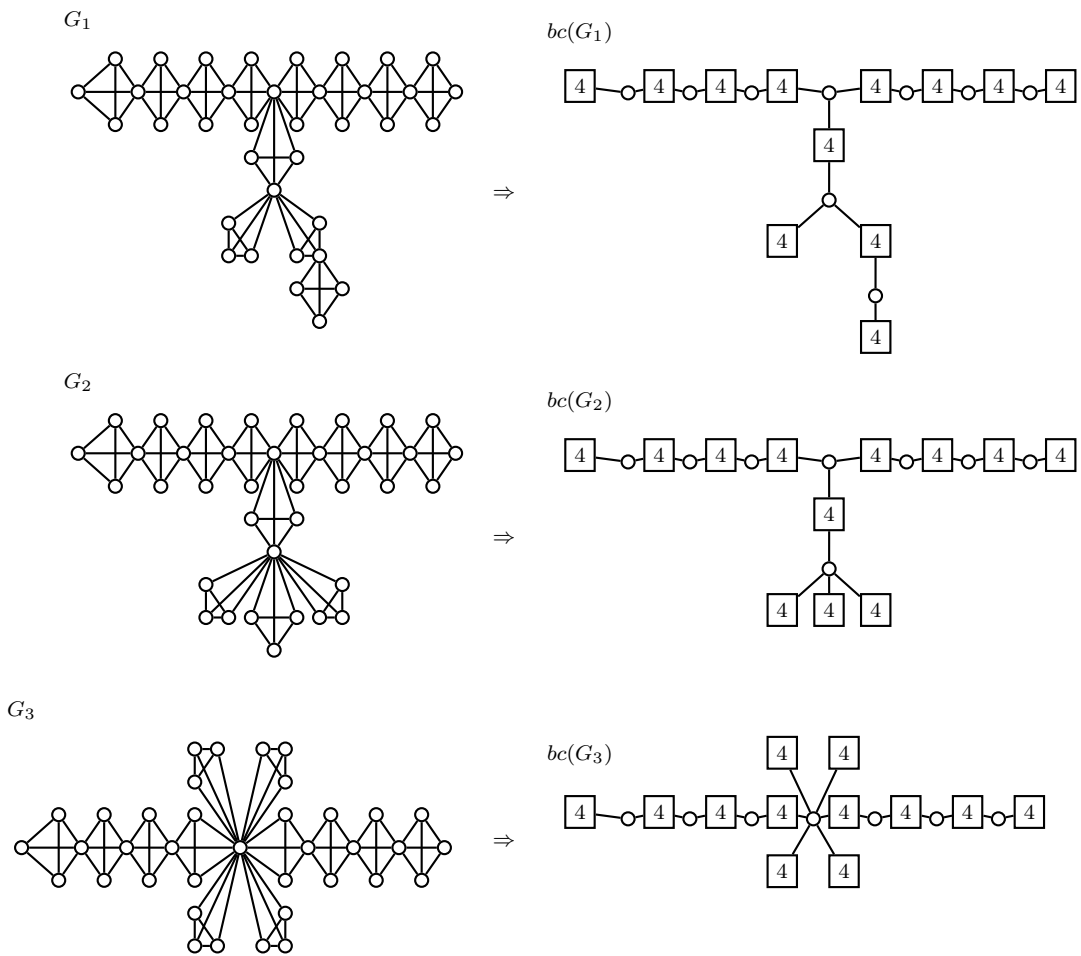


Figura 6.6: Grafos bloco com mesma representação $bc(G_i)$ das árvores da Figura 6.5 e $a(G_i) = 0,09305$.

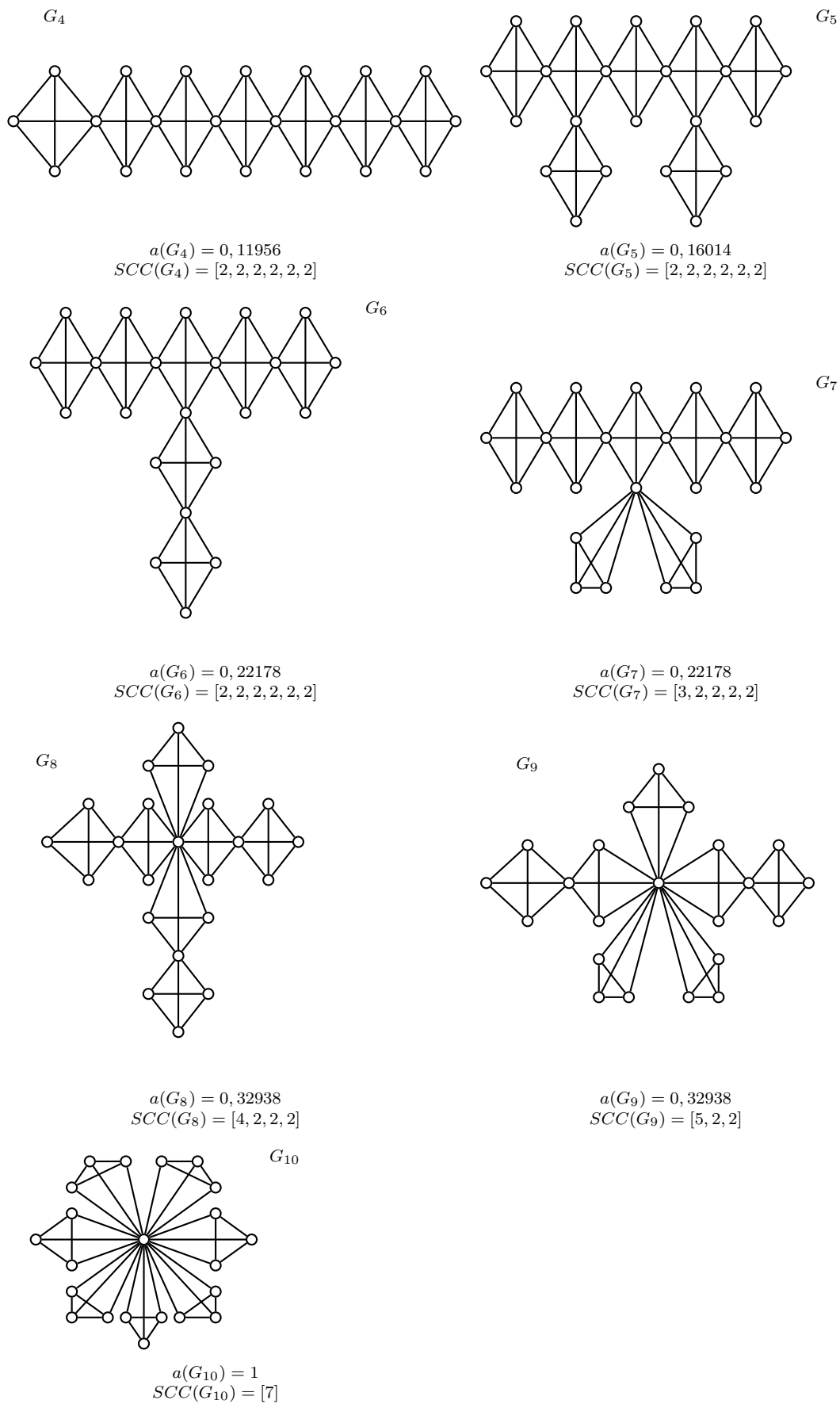


Figura 6.7: Exemplos de grafos bloco.

Referências Bibliográficas

- [1] ABREU, N., ROJO, O., JUSTEL, C., 2010, “Algebraic Connectivity for Subclasses of Caterpillars”, *Applicable Analysis and Discrete Mathematics*, v. 4, n. 1, pp. 181–196.
- [2] ABREU, N. M. M. D., 2007, “Old and new results on algebraic connectivity of graphs”, *Linear Algebra and its Applications*, v. 423, n. 1, pp. 53 – 73.
- [3] BARRIÉRE, L., HUEMER, C., MITSCHKE, D., et al., 2011, “On the Fiedler value of large planar graphs (Extended abstract)”, *Electronic Notes in Discrete Mathematics*, v. 38, n. 0, pp. 111 – 116.
- [4] BEHTOEI, A., JANNESARI, M., TAERI, B., 2010, “A characterization of block graphs”, *Discrete Applied Mathematics*, v. 158, n. 3, pp. 219 – 221.
- [5] BIGGS, N., 1993, *Algebraic Graph Theory*. Cambridge.
- [6] BLAIR, J. R. S., PEYTON, B., 1993, “An introduction to chordal graphs and clique trees”, *Institute for Mathematics and Its Applications*, v. 56, pp. 1–29.
- [7] BOAVENTURA, P. O. N., 2006, *Grafos: Teoria, Modelos, Algoritmos*. 4 ed. , Edgard Blucher.
- [8] CHANDRAN, L. S., GRANDONI, F., 2006, “A linear time algorithm to list the minimal separators of chordal graphs”, *Discrete Mathematics*, v. 306, n. 3, pp. 351–358.
- [9] CHANG, G. J., HWANG, F. K., YAO, Y. C., 1998, “Localizing Combinatorial Properties for Partitions on Block Graphs”, *Journal of Combinatorial Optimization*, v. 2, pp. 429–441.
- [10] CHOU, H.-H., KO, M.-T., HO, C.-W., et al., 2008, “Node-searching problem on block graphs”, *Discrete Applied Mathematics*, v. 156, pp. 55–75.
- [11] DIESTEL, R., 2005, *Graph Theory*, v. 173, *Graduate Texts in Mathematics*. Third ed. , Springer-Verlag, Heidelberg. Disponível em:

<<http://www.math.uni-hamburg.de/home/diestel/books/graph.theory/GraphTheoryIII.pdf>>.

- [12] FALLAT, S., KIRKLAND, S., 1998, “Extremizing algebraic connectivity subject to graph theoretic constraints”, *The Electronic Journal of Linear Algebra*, v. 3 (April), pp. 48–74. A publication of the International Linear Algebra Society.
- [13] FENG, LIHUA; YU, G., 2007, “No starlike trees are Laplacian cospectral.” *Publ. Elektroteh. Fak., Univ. Beogr., Ser. Mat.*, v. 18, pp. 46–51.
- [14] F.HARARY, G.PRINS, 1966, “The block-cutpoint-tree of a graph”, *Publ. Math. Debrecen*, v. 13, pp. 103–107.
- [15] FIEDLER, M., 1973, “Algebraic connectivity of graphs”, *Czechoslovak Mathematical Journal*, v. 23, n. 98, pp. 298–305.
- [16] FULKERSON, D. R., GROSS, O. A., 1965, “Incidence matrices and interval graphs”, *Pacific Journal of Mathematics*, v. 15, n. 3, pp. 835–855.
- [17] GODSIL, C. D., ROYLE, G. F., 2001, *Algebraic Graph Theory*. New York, Springer-Verlag.
- [18] GOLUMBIC, M. C., 2004, *Algorithmic Graph Theory and Perfect Graphs*. 2nd ed. New York, Academic Press.
- [19] GRONE, R., MERRIS, R., 1990, “Ordering trees by algebraic connectivity”, *Graphs Combinatorics*, v. 6, pp. 229–237.
- [20] GROSS, J. L., YELLEN, J., 2003, *Handbook of Graph Theory (Discrete Mathematics and Its Applications)*. 1 ed. , CRC Press.
- [21] GUO, J.-M., SHIU, W. C., LI, J., 2011, “The algebraic connectivity of lollipop graphs”, *Linear Algebra and its Applications*, v. 434, n. 10, pp. 2204 – 2210.
- [22] GUTMAN, I., ARAUJO, O., RADA, J., 2001, “Matchings in starlike trees”, *Appl. Math. Lett.*, v. 14, n. 7, pp. 843–848.
- [23] HARARY, F., 1963, “A characterization of block-graphs”, *Bulletin canadien de mathématiques = Canadian Mathematical Bulletin*, v. 6, n. 1 (January), pp. 1–6.
- [24] HARARY, F., 1969, *Graph Theory*. Addison-Wesley.

- [25] HO, C. W., LEE, R. C. T., 1989, “Counting clique trees and computing perfect elimination schemes in parallel”, *Inf. Process. Lett.*, v. 31, n. 2, pp. 61–68.
- [26] JACOBS, D. P., TREVISAN, V., 2011, “Locating the Eigenvalues of Trees”, *Linear Algebra and its Applications*, v. 434, n. 1, pp. 81 – 88.
- [27] KUMAR, P. S., MADHAVAN, C. E. V., 2002, “Clique tree generalization and new subclasses of chordal graphs”, *Discrete Applied Mathematics*, v. 117, n. 1-3, pp. 109–131.
- [28] KUMAR, P. S., MADHAVAN, C. E. V., 1998, “Vertex Separators of Chordal Graphs”, *Discrete Applied Mathematics*, v. 89, n. 1-3, pp. 155–168.
- [29] LEPOVIC, M., GUTMAN, I., 2002, “No starlike trees are cospectral”, *Discrete Mathematics*, v. 242, n. 1-3, pp. 291–295.
- [30] LI, J., GUO, J.-M., SHIU, W. C., 2010, “The smallest values of algebraic connectivity for unicyclic graphs”, *Discrete Applied Mathematics*, v. 158, n. 15, pp. 1633 – 1643.
- [31] LIU, Y., LIU, Y., 2009, “The ordering of unicyclic graphs with the smallest algebraic connectivity”, *Discrete Mathematics*, v. 309, n. 13, pp. 4315 – 4325.
- [32] LUNDQUIST, M., 1990, *Zero patterns, chordal graphs, and matrix completions*. Ph.d. thesis, Department of Mathematical Sciences, Clemson University.
- [33] MARKENZON, L., PEREIRA, P. R. D. C., 2010, “One-Phase Algorithm for the Determination of Minimal Vertex Separators of Chordal Graphs”, *International Transactions in Operational Research*, v. 17, n. 6 (November), pp. 683–690.
- [34] MCKEE, T., MCMORRIS, F., 1999, *Topics in intersection graph theory*. Philadelphia, PA, SIAM Monograph on Discrete Math. and Appl.
- [35] MERRIS, R., 1994, “Laplacian matrices of graphs: a survey”, *Linear Algebra and its Applications*, v. 197-198 (January-February), pp. 143–176.
- [36] MERRIS, R., 2001, *Graph Theory*. New York, Wiley Interscience.
- [37] MOSK-AOYAMA, D., 2008, “Maximum algebraic connectivity augmentation is NP-hard”, *Operations Research Letters*, v. 36, n. 6, pp. 677 – 679.
- [38] PAK-KEN, W., 1999, “Optimal path cover problem on block graphs”, *Theoretical Computer Science*, v. 225, n. 1-2, pp. 163–169.

- [39] PANDA, B. S., DAS, A., 2010, “Tree 3-spanners in 2-sep chordal graphs: Characterization and algorithms”, *Discrete Applied Mathematics*, v. 158, n. 17, pp. 1913–1935.
- [40] PANDA, B. S., NATARAJAN, V., DAS, S. K., 2002, “Parallel Algorithms for Hamiltonian 2-Separator Chordal Graphs”, *Parallel Processing Letters*, v. 12, n. 1, pp. 51–64.
- [41] REIS, B. M. N., 2005, *Parâmetros de Grafos em Vulnerabilidade de Redes*. Dissertação de mestrado, PEP/COPPE/UFRJ, Setembro.
- [42] ROJO, O., MEDINA, L., 2006, “Tight bounds on the algebraic connectivity of Bethe trees”, *Linear Algebra and its Applications*, v. 418, n. 2-3, pp. 840 – 853.
- [43] ROJO, O., MEDINA, L., ABREU, N., et al., 2010, “On the algebraic connectivity of some caterpillars: A sharp upper bound and a total ordering”, *Linear Algebra and its Applications*, v. 432, n. 2-3, pp. 586 – 605.
- [44] ROJO, O., ROCHA, I., TREVISAN, V., 2011, “Algebraic connectivity on a subclass of caterpillars”, *Electronic Notes in Discrete Mathematics*, v. 37, n. 0, pp. 153 – 158. LAGOS’11 - VI Latin-American Algorithms, Graphs and Optimization Symposium.
- [45] SHAO, J.-Y., GUO, J.-M., SHAN, H.-Y., 2008, “The ordering of trees and connected graphs by algebraic connectivity”, *Linear Algebra and its Applications*, v. 428 (April), pp. 1421–1438.
- [46] SPINRAD, J., 2003, *Efficient Graph Representations*, *Fields Institute Monographs*, v. 19. Providence, AMS.
- [47] SZWARCFITER, J. L., 1986, *Grafos e Algoritmos Computacionais*. 2 ed. , Campus.
- [48] WANG, X.-K., TAN, S.-W., 2012, “Ordering trees by algebraic connectivity”, *Linear Algebra and its Applications*, v. 436, n. 9, pp. 3684 – 3691.
- [49] YUAN, X.-Y., SHAO, J.-Y., ZHANG, L., 2008, “The six classes of trees with the largest algebraic connectivity”, *Discrete Appl. Math.*, v. 156 (March), pp. 757–769.
- [50] ZHANG, X.-D., 2007, “Ordering trees with algebraic connectivity and diameter”, *Linear Algebra and its Applications*, v. 427, n. 2-3, pp. 301 – 312.

Índice Remissivo

- árvore, 6
 - de cliques, 7, 9
 - enraizada, 6
 - medusa, 57
 - starlike, 57
- árvores
 - \mathcal{T}_1 , 66
 - \mathcal{T}_2 , 69
 - $\mathcal{T}_{n,k}(q, r, s)$, 64
- ancestral, 6
- aresta, 4
- autovalores, 12
- caminho, 4
- caterpillars, 31
- centro, 5
- ciclo, 4
- clique, 5
 - maximal, 5
- coalescência, 6
- componente conexa, 5
- conectividade
 - algébrica, 12
 - de arestas, 11
 - de vértices, 5, 11
- conjunto
 - independente de vértices, 5
- descendente, 6
- diâmetro, 5
- distância, 5
- espectro, 12
- esquema de eliminação perfeita, 7
- estrela, 6
- excentricidade, 5
- floresta, 6
- folhas, 6
- grafo
 - k -conexo, 5
 - k -sep, 18
 - de cliques, 9
 - block-cut-vertex, 80
 - completo, 5
 - conexo, 4
 - cordal, 6
 - de cliques, 9
 - isomorfos, 5
 - simples, 4
- grau
 - de vértice, 5
- laplaciano, 12
- matriz
 - diagonal, 12
 - identidade, 12
 - laplaciana, 12
- multiplicidade, 11, 20
- ordem lexicográfica, 15
- percurso, 4
- polinômio
 - característico, 12
 - do laplaciano, 12
- raio, 5

- raiz, 6
- realização de uma sequência, 5
- separador, 8
 - de vértices, 8
 - minimal
 - de vértices, 8
 - do grafo, 8
 - minimal de vértices, 9
- sequência
 - de componentes conexas, 15
 - de graus internos, 28
 - de multiplicidades, 20, 21
 - gráfica, 5
- subgrafo
 - conexo maximal, 4
 - induzido, 4
 - próprio, 4
- teoria espectral de grafos, 12
- vértice, 4
 - adjacente, 4
 - central, 5
 - interno, 6
 - periférico, 5
 - simplicial, 7
- vizinhança, 4