

UNIVERSIDADE FEDERAL FLUMINENSE

EDMUNDO PINTO NETO

**Minimizando Distâncias entre Vértices e Arestas  
através de Árvores t-Geradoras em Grafos**

NITERÓI

2023

EDMUNDO PINTO NETO

# **Minimizando Distâncias entre Vértices e Arestas através de Árvores t-Geradoras em Grafos**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Ciência da Computação

Orientador:

Luís Felipe Ignácio Cunha

NITERÓI

2023

Ficha catalográfica automática - SDC/BEE  
Gerada com informações fornecidas pelo autor

P659m Pinto Neto, Edmundo  
Minimizando Distâncias entre Vértices e Arestas através  
de Árvores t-Geradoras em Grafos / Edmundo Pinto Neto. - 2023.  
67 f.

Orientador: Luís Felipe Ignácio Cunha.  
Dissertação (mestrado)-Universidade Federal Fluminense,  
Instituto de Computação, Niterói, 2023.

1. Complexidade computacional. 2. Teoria dos grafos. 3.  
Teoria da computação. 4. Produção intelectual. I. Cunha,  
Luís Felipe Ignácio, orientador. II. Universidade Federal  
Fluminense. Instituto de Computação.III. Título.

CDD - XXX

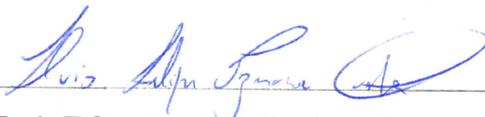
EDMUNDO PINTO NETO

MINIMIZANDO DISTÂNCIAS ENTRE VÉRTICES E ARESTAS ATRAVÉS DE  
ÁRVORES T-GERADORAS EM GRAFOS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Ciência da Computação

Aprovada em Dezembro de 2023.

BANCA EXAMINADORA



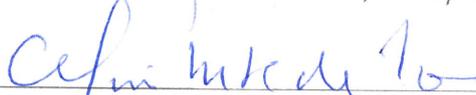
Prof. Dr. Luís Felipe Ignácio Cunha - Orientador, TCC/UFF



Prof. Dr. Uéverton dos Santos Souza, TCC/UFF



Prof. Dr. Fábio Protti, TCC/UFF



Profa. Dra. Celina Miraglia Herrera de Figueiredo, PESC/COPPE/UFRJ

Niterói

2023

*Dedico este trabalho aos meus filhos, Eduardo e Cecília, amores eternos e incondicionais, e aos meus pais, Cesar e Lana, minhas referências de vida.*

# Agradecimentos

Ao meu orientador, que me instruiu com entusiasmo, trazendo diversas ideias, reflexões e contribuições para este trabalho e que, de forma criativa e inovadora, me fez aprender a produzir conhecimento científico na área de complexidade computacional. Aos meus filhos, pais, irmãs e amigos, que foram muito compreensivos nesse período de dedicação à pesquisa e ausência do convívio social. À banca examinadora pelas palavras e observações assertivas colocadas. E a todos os funcionários e professores da UFF, Universidade que me deu toda a formação acadêmica obtida até aqui, da graduação ao mestrado.

# Resumo

Uma árvore  $t$ -geradora de um grafo  $G$  é uma árvore geradora  $T$  de  $G$  na qual quaisquer dois vértices adjacentes de  $G$  têm distância no máximo  $t$  em  $T$ . Dizemos que  $G$  é  $t$ -admissível se admite uma árvore  $t$ -geradora, e  $\sigma_G$  é o menor  $t$  tal que  $G$  seja  $t$ -admissível, chamado de índice de extensão de  $G$ . Decidir se  $G$  tem uma árvore  $t$ -geradora está em  $\mathbf{P}$  para  $t \leq 2$ , é  $\mathbf{NP}$ -completo para  $t \geq 4$ , e decidir a 3-admissibilidade é um problema em aberto há mais de 25 anos. Couto, Cunha e Posner definiram a árvore aresta  $t$ -geradora de um grafo, que é uma árvore  $T$  do grafo linha de  $G$  em que duas arestas adjacentes de  $G$  têm distância no máximo  $t$  em  $T$ . Quando se pretende minimizar as distâncias entre arestas, eles provaram que a 3-admissibilidade de arestas é um problema solucionável em tempo polinomial, enquanto é  $\mathbf{NP}$ -completo para  $t \geq 8$ , mesmo para grafos bipartidos. Esta dissertação investiga a complexidade de lidar com distâncias minimizadas entre vértices e arestas ao mesmo tempo. Isso motiva o estudo do problema da ÁRVORE TOTAL ADMISSIBILIDADE, definido como a determinação de uma árvore  $t$ -geradora para um grafo total  $\text{Tot}(G)$ , onde  $\text{Tot}(G)$  é o grafo de interseção dos vértices e arestas de  $G$ . Como contribuições, são apresentadas relações entre os índices de extensão de  $G$ ,  $L(G)$  e  $\text{Tot}(G)$ . Além disso, é provado que ÁRVORE TOTAL ADMISSIBILIDADE é  $\mathbf{NP}$ -completo, mesmo para as classes de grafos bipartidos ou planares. Ademais, alguns subgrafos de grafos totais podem ser obtidos da seguinte forma: grafo-meio (*middle graph*)  $M(G)$  de  $G$ , obtido a partir do grafo linha de  $G$ ,  $L(G)$ , unindo aos vértices de  $G$  e tornando cada vértice recém inserido vizinho aos vértices de  $L(G)$  que compartilham rótulo; grafo-quase-total  $QT(G)$  de  $G$ , obtido subdividindo cada aresta de  $G$  exatamente uma vez e adicionando as arestas originais de  $G$  novamente. Essas duas classes anteriores, como várias outras classes clássicas de grafos, podem ser vistas como pertencentes a classe de grafos *clique-aumentantes*, obtidas de um grafo  $G$  escolhendo cliques arbitrários de  $G$  e para cada clique, adicionando um vértice simplicial. Este trabalho mostra uma série de propriedades relativas a construções especiais, casos tratáveis e provas de  $\mathbf{NP}$ -completude da ADMISSIBILIDADE para grafos clique-aumentantes.

**Palavras-chave:** árvores geradoras, grafos linha, grafos totais, grafos-meio, grafos-quase-totais, grafos clique-aumentantes.

# Abstract

A tree  $t$ -spanner of a graph  $G$  is a spanning tree  $T$  of  $G$  in which any two adjacent vertices of  $G$  have distance at most  $t$  in  $T$ . We say that  $G$  is  $t$ -admissible if it admits a tree  $t$ -spanner, and the  $\sigma_G$  is the lowest  $t$  such that  $G$  is  $t$ -admissible, called the stretch index of  $G$ . Deciding whether  $G$  has a tree  $t$ -spanner is in **P** for  $t \leq 2$ , is **NP**-complete for  $t \geq 4$ , and deciding 3-admissibility has been an open problem for more than 25 years. Couto, Cunha and Posner defined the edge tree  $t$ -spanner of a graph, which is a spanner tree  $T$  of the line graph of  $G$  in which two adjacent edges of  $G$  have distance at most  $t$  in  $T$ . When it is wondering to minimize distances between edges, they proved that the edge 3-admissibility is a polynomial-time solvable problem, while it is **NP**-complete for  $t \geq 8$ , even for bipartite graphs. This dissertation investigates the complexity of dealing with minimized distances between vertices and edges at the same time. This motivates the study of the **TOTAL TREE ADMISSIBILITY** problem, defined as determining a tree  $t$ -spanner with the lowest  $t$  for a total graph  $\text{Tot}(G)$ , where  $\text{Tot}(G)$  is the intersection graph of the vertices and edges of  $G$ . As contributions, relationships are presented between the stretch indexes of  $G$ ,  $L(G)$  and  $\text{Tot}(G)$ . Furthermore, it is proved that **TOTAL TREE ADMISSIBILITY** is **NP**-complete, even for the classes of bipartite or planar graphs. Moreover, some subgraphs of total graphs can be obtained as follows: middle graph  $M(G)$  of  $G$ , obtained from the line graph of  $G$ ,  $L(G)$ , joining the vertices of  $G$  and making each newly inserted vertex neighboring the vertices of  $L(G)$  that share label; almost-total graph  $AT(G)$  of  $G$ , obtained by subdividing each edge of  $G$  exactly once and adding the original edges of  $G$  again. These two previous classes, like several other classical classes of graphs, can be seen as belonging to the class of clique-augmenting graphs, obtained from a graph  $G$  by choosing arbitrary cliques of  $G$  and for each clique, adding a simplicial vertex. This work shows a series of properties relating to special constructions, tractable cases and **NP**-completeness proofs of **ADMISSIBILITY** for clique-augmenting graphs.

**Keywords:** tree spanners, line graphs, total graphs, middle graphs, almost-total graphs, clique-augmenting graphs.

# Lista de Figuras

1.1	Um grafo $G$ , uma árvore 3-geradora de $L(G)$ e $G$ com sua árvore aresta 3-geradora em vermelho. . . . .	15
1.2	À esquerda: Um grafo total $\text{Tot}(G)$ , vértices e arestas pretas são de $G$ , arestas azuis estão entre $G$ e $L(G)$ , arestas pretas em negrito são de $L(G)$ e vértices vermelhos são de $L(G)$ ; no meio: outra representação de $\text{Tot}(G)$ , bordas vermelhas definem uma árvore total 4-geradora de $G$ ; à direita: $G$ com a árvore total 4-geradora relacionada em vermelho. . . . .	16
1.3	À esquerda $G$ , no centro o grafo-meio de $G$ , $M(G)$ e à direita o grafo-quase-total de $G$ , $QT(G)$ . . . . .	16
2.1	Relações entre as classes de problemas. . . . .	21
2.2	Exemplo de aresta $e$ forçada e caminhos $P1$ e $P2$ forçados. . . . .	25
2.3	a) Construção de $H_i$ para $t = 4$ b) Ligação dos $H_i$ 's com o vértice $x$ e os vértices de cláusula $c_j$ . . . . .	26
2.4	Grafo obtido da <a href="#">Construção 9</a> da instância $I = (\{u,v,w\}, \{(u,v,w), (\bar{u},v,\bar{w})\})$ e uma árvore aresta 8-geradora do grafo em vermelho. . . . .	29
3.1	$\text{Tot}(C_6)$ e sua árvore 6-geradora. . . . .	32
3.2	Classe de grafo com $\sigma'$ e $\sigma''$ menor do que $\sigma$ . (a) Grafo usando $C_6$ 's. (b) árvore geradora com $\sigma = 9$ . (c) árvore aresta geradora com $\sigma' = 7$ . (d) árvore total geradora com $\sigma'' = 8$ . . . . .	34
3.3	Árvore Total geradora com $\sigma'' \leq \sigma + 2$ . . . . .	37
3.4	Caminhos entre $u$ e $v$ , $u$ e $w$ , e $v$ e $w$ in $T''$ . . . . .	38
3.5	(a) Grafo $G$ e sua árvore 4-geradora em verde. (b) Grafo $H$ obtido da <a href="#">Construção 14</a> sobre $G$ . (c) Grafo total $\text{Tot}(H)$ e sua árvore de solução 18-geradora (arestas marcadas em verde) com base na árvore 4-geradora de $G$ provada no <a href="#">Teorema 15</a> . . . . .	39

- 
- 3.6 (a) Caso i)  $18 = 3 \times 6$  não é possível. (b) Caso ii)  $18 = 3 \times 5 + 3$ , um exemplo onde existe outro vértice com distância maior que 18 de  $y$ . . . . 42
- 3.7 (a) Árvore Total geradora do [Lema 19](#). (b) Árvore Total geradora do [Lema 20](#).  
43
- 4.1 (a) Árvore 17-geradora de  $M(H)$  (b) Árvore 17-geradora de  $Tot(H)$ . . . . 58

# Lista de Tabelas

- 3.1 Coluna  $\sigma''(G)$  e grafos  $C_n$ -multi-ciclos são alguns dos resultados apresentados nesta dissertação.  $\sigma''(G) > \sigma(G)$  para grafos completos, ciclos, caminhos e árvores. Enquanto  $\sigma'(G) = \sigma(G)$  para grafos ciclos e caminhos e  $\sigma'(G) > \sigma(G)$  para grafos completos e árvores. Os três parâmetros são diferentes para grafos  $C_n$ -multi-ciclos e, contraintuitivamente,  $\sigma(G) > \sigma''(G) > \sigma'(G)$ . Enquanto  $\sigma(G)$  está limitado superiormente a 3 para grafos *split* e junção,  $\sigma'(G)$  e  $\sigma''(G)$  estão limitados a 4. Decidir os 3 parâmetros para grafos planares e bipartidos é NP-Completo. Uma questão que permanece em aberto é conhecer alguma classe de grafos cuja complexidade de determinar o índice de extensão seja distinta da complexidade de determinar o seu índice de aresta ou total. . . . . 47
- 5.1 Resultados da coluna  $G$  obtidos por Cai e Corneil (CAI; CORNEIL, 1995). Resultados da coluna  $L(G)$  obtidos por Couto, Cunha e Posner (COUTO; CUNHA; POSNER, 2021). As colunas seguintes são de alguns dos resultados obtidos nesta dissertação. Na linha NP-Completo, as colunas  $Tot(G)$ ,  $M(G)$ ,  $QT(G)$  e  $CA(G)$  são dos Corolário 50, Corolário 49, Corolário 35 e Teorema 53, respectivamente. . . . . 62

# Sumário

<b>1</b>	<b>Introdução</b>	<b>12</b>
1.1	Problema de Decisão de Árvores $t$ -Geradoras . . . . .	12
1.2	Trabalhos relacionados . . . . .	13
1.3	Contribuições . . . . .	15
1.4	Organização do trabalho . . . . .	17
<b>2</b>	<b>Preliminares sobre Complexidade Computacional e Árvore <math>t</math>-Geradora</b>	<b>18</b>
2.1	Complexidade computacional . . . . .	18
2.1.1	NP-completude . . . . .	20
2.1.2	Questão P versus NP . . . . .	21
2.2	Definições básicas em grafos . . . . .	23
2.3	NP-completude da 4-admissibilidade . . . . .	24
2.4	Admissibilidade em operações de grafos . . . . .	27
2.4.1	NP-completude da aresta 8-admissibilidade . . . . .	29
<b>3</b>	<b>Árvores <math>t</math>-Geradoras em Grafos Totais</b>	<b>32</b>
3.1	Limites nas aresta e total admissibilidades . . . . .	32
3.2	Total admissibilidade . . . . .	35
3.2.1	Limite superior no índice de extensão total . . . . .	35
3.2.2	Total $k$ -admissibilidade é NP-completo . . . . .	37
3.2.3	Grafos total 4-admissíveis . . . . .	42
<b>4</b>	<b>Grafos Clique-Aumentantes</b>	<b>48</b>

---

4.1	Grafos Clique-Aumentantes . . . . .	48
4.2	Alternando tamanhos de cliques . . . . .	49
4.2.1	Grafos Clique-aumentantes escolhendo cliques de tamanho 2 . . . . .	50
4.2.2	Grafos Planares Clique-aumentantes escolhendo cliques de tamanho 3 . . . . .	55
4.2.3	Grafos Linhas Clique-aumentantes escolhendo cliques de tamanhos arbitrários . . . . .	56
<b>5</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>61</b>
	<b>REFERÊNCIAS</b>	<b>63</b>

# 1 Introdução

## 1.1 Problema de Decisão de Árvores $t$ -Geradoras

A determinação de árvores geradoras é um problema clássico em grafos com muitas aplicações e abordagens, como, por exemplo, na determinação de protocolos de roteamento em redes de telecomunicações (BHATT et al., 1986; PELEG; ULLMAN, 1989). Uma árvore  $T$  é chamada de árvore geradora de um grafo  $G$  se  $T$  é um subgrafo de  $G$  que possui todos os vértices de  $G$ . Uma *árvore  $t$ -geradora* de um grafo  $G$  é uma árvore geradora  $T$  de  $G$  na qual quaisquer dois vértices adjacentes de  $G$  têm distância no máximo  $t$  em  $T$ . Um grafo  $G$  com uma árvore  $t$ -geradora é chamado de grafo  *$t$ -admissível*. O menor  $t$  para o qual um grafo  $G$  é  $t$ -admissível é o *índice de extensão de  $G$*  e é denotado por  $\sigma_T(G)$  (ou simplesmente  $\sigma(G)$ ). O problema da  *$t$ -admissibilidade* visa decidir se um determinado grafo  $G$  possui  $\sigma(G) \leq t$ . O problema de determinar o índice de extensão das árvores, ou seja, *the minimum stretch spanning tree problem* (MSST) foi estudado estabelecendo limites em  $\sigma(G)$  ou desenvolvendo a complexidade computacional da versão de decisão de MSST para diversas classes de grafos (CAI; CORNEIL, 1995; COUTO; CUNHA, Luís Felipe I., 2018, 2019). O Problema MSST é um dos interessantes problemas de otimização, que são estudados não apenas em grafos, mas em vários outros problemas combinatórios, de tal forma que limites, algoritmos e estudos de complexidade computacional são amplamente desenvolvidos (CUNHA et al., 2013). De agora em diante, quando refere-se ao MSST, trata-se da versão de decisão deste problema. Além disso, como os grafos desconexos não têm árvores geradoras e árvores são os únicos grafos 1-admissíveis, considera-se apenas grafos conexos distintos de árvores.

Cai e Corneil (CAI; CORNEIL, 1995) provaram que o problema de  $t$ -admissibilidade é NP-completo, para  $t \geq 4$ , enquanto grafos 2-admissíveis podem ser reconhecidos em tempo polinomial. A caracterização de grafos 2-admissíveis, declarada no Teorema 1, trata de componentes triconexas de um grafo conexo, definido como qualquer subgrafo máximo que não contém dois vértices cuja remoção desconecta o grafo (os autores também consideram

grafos completos  $K_2$  (arestas) e  $K_3$  (triângulos) como componentes triconexas). Um grafo *biconexo* é um grafo sem um *vértice de corte*, ou seja, um vértice cuja remoção desconecta o grafo. Uma *estrela* com  $n + 1$  vértices é o grafo bipartido completo  $K_{1,n}$ . Uma estrela  $v$ -centrada é uma estrela centrada no vértice universal  $v$ . Da mesma forma, uma bi-estrela é um grafo tal que existe uma aresta  $uv$  e cada aresta do conjunto de arestas do grafo compartilha um ponto final com  $uv$ . Portanto,  $uv$  é uma *aresta universal* da bi-estrela. Uma bi-estrela  $uv$ -centrada é uma bi-estrela centrada em uma aresta universal  $uv$ .

**Teorema 1.** (CAI; CORNEIL, 1995) *Um grafo biconexo  $G$  é 2-admissível se e somente se  $G$  contiver uma árvore geradora  $T$  tal que para cada componente triconexa  $H$  de  $G$ ,  $T \cap H$  seja uma estrela geradora de  $H$ .*

A  $t$ -admissibilidade tem sido amplamente estudada especialmente porque determinar se um grafo  $G$  é 3-admissível ainda é um problema em aberto desde a sua proposição (CAI; CORNEIL, 1995). Apesar disso, muitos avanços foram desenvolvidos para caracterizar a 3-admissibilidade para classes de grafos (BRANDSTADT et al., 2007; COUTO; CUNHA, Luís Felipe I, 2020; FEKETE; KREMER, 2001). Em muitas dessas caracterizações, as operações em grafos foram extensivamente estudadas.

O principal objetivo deste trabalho é estabelecer relações entre o índice de extensão dos grafos antes e depois de tais operações. A seguir, alguns resultados interessantes considerando operações em grafos são apresentados.

## 1.2 Trabalhos relacionados

Couto e Cunha (COUTO; CUNHA, Luís Felipe I, 2021) provaram que qualquer grafo  $G$  distinto do grafo completo pode ser transformado em um grafo 4-admissível pela união de  $G$  e seu grafo complementar  $\overline{G}$ , e a adição de um emparelhamento perfeito entre os vértices correspondentes de  $G$  e  $\overline{G}$ . O grafo resultante é o prisma complementar de  $G$ , denotado como  $G\overline{G}$ . Além disso, eles provaram que os índices de extensão dos grafos  $G\overline{G}$  são iguais a 4, e como esses grafos têm diâmetro de no máximo 5, esta é uma classe de grafo para a qual a  $t$ -admissibilidade é resolvida em tempo polinomial, enquanto se sabe que decidir grafos  $t$ -admissíveis para grafos com diâmetro no máximo  $t + 1$  é NP-completo (PAPOUTSAKIS, 2018).

Outra operação bem conhecida em grafos é fazer a união de duas cópias de  $G$  e adicionar um emparelhamento perfeito entre os vértices correspondentes dos dois grafos  $G$ .

O grafo resultante é o prisma de  $G$  (também definido como o produto cartesiano  $G \times K_2$ ). Couto e Cunha (COUTO; CUNHA, Luís Felipe I, 2021) provaram que para grafos  $GG$ , cuja definição é muito semelhante à de  $G\overline{G}$ 's, a  $t$ -admissibilidade é NP-completa, para  $t \geq 4$ .

Gomez, Miyazawa e Wakabayashi (GÓMEZ; MIYAZAWA; WAKABAYASHI, 2022) investigaram árvores geradoras restritas a prismas de grafos, e caracterizaram aquelas que admitem uma árvore 3-geradora. Como resultado, eles obtiveram um algoritmo de tempo linear para grafos prisma 3-admissíveis.

Couto, Cunha e Posner (COUTO; CUNHA; POSNER, 2021) introduziram o problema da ÁRVORE ARESTA ADMISSIBILIDADE para investigar distâncias entre arestas de um grafo, em vez da distância entre vértices, e obtiveram resultados mais gerais sobre NP-Dificuldade para aresta 8-admissibilidade e um algoritmo de tempo polinomial para aresta 3-admissibilidade.

Dado um grafo  $G$ , seu *grafo linha*  $L(G)$  é obtido da seguinte forma:  $V(L(G)) = E(G)$ ;  $E(L(G)) = \{\{uv, uw\} \mid uv, uw \in E(G)\}$ . Ou seja, cada aresta de  $G$  é um vértice de  $L(G)$  e se duas arestas compartilham um vértice, então seus vértices correspondentes são adjacentes em  $L(G)$ . A *distância entre duas arestas*  $e_1$  e  $e_2$  de  $G$ , para  $e_1, e_2 \in E(G)$ , é a distância entre seus vértices correspondentes em  $L(G)$ .

A seguir, é apresentado em mais detalhes o problema da ÁRVORE ARESTA ADMISSIBILIDADE, introduzido em (COUTO; CUNHA; POSNER, 2021).

**Árvores aresta geradoras** A *árvore aresta  $t$ -geradora* de um grafo  $G$  é definida como uma árvore geradora  $T$  de  $L(G)$  tal que, para quaisquer duas arestas adjacentes de  $G$ , sua distância é no máximo  $t$  em  $T$ . Portanto, uma árvore aresta  $t$ -geradora de  $G$  é uma árvore  $t$ -geradora de  $L(G)$ . Um grafo  $G$  que possui uma árvore aresta  $t$ -geradora é chamado *aresta  $t$ -admissível*. O menor  $t$  para o qual  $G$  é um grafo aresta  $t$ -admissível é o *índice de extensão de aresta* de  $G$ , e é denotado por  $\sigma'_T(G)$  (ou simplesmente  $\sigma'(G)$ ). O problema *aresta  $t$ -admissibilidade* visa decidir se um determinado grafo  $G$  possui  $\sigma'(G) \leq t$ . A Figura 1.1 descreve a relação entre a árvore aresta geradora de um grafo e a árvore geradora de seu grafo linha.

Couto, Cunha e Posner (COUTO; CUNHA; POSNER, 2021) estabeleceram que o problema da aresta 8-admissibilidade é NP-completo, mesmo para grafos bipartidos, determinaram o índice de extensão para diversas classes de grafos, como grafos *split*, seus grafos generalizados e grafos com poucos  $P_4$ 's.

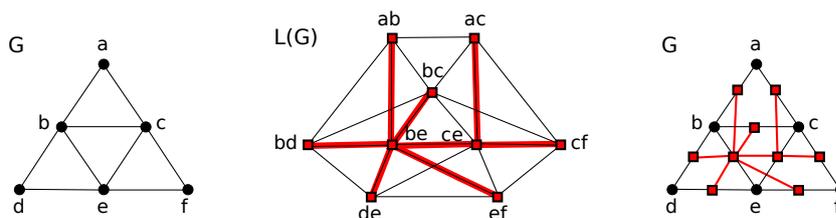


Figura 1.1: Um grafo  $G$ , uma árvore 3-geradora de  $L(G)$  e  $G$  com sua árvore aresta 3-geradora em vermelho.

Ainda em relação a operações em grafos e qual relação existe quanto a admissibilidade, temos os grafos total, meio e quase-total. A seguir definiremos os grafos total e grafos meio, enquanto que os grafos quase-total (propostos nesta dissertação) serão definidos na [Seção 1.3](#)

Dado um grafo  $G$ , seu *grafo total* tem  $V(Tot(G)) = V(G) \cup E(G)$  e  $E(Tot(G)) = E(G) \cup E(L(G)) \cup \{u uv \mid u \in V(G) \text{ and } uv \in E(G)\}$ . Observe que  $Tot(G)[V(G)] = G$  e  $Tot(G)[E(G)] = L(G)$ . Uma maneira equivalente de definir o grafo total de um grafo  $G$  é subdividir cada aresta de  $G$ , ou seja, cada aresta  $uv$  de  $G$  é substituída por um caminho  $u, uv, v$  onde  $uv$  é um novo vértice do grafo, e para cada par de vértices a uma distância 2 neste novo grafo, adicionamos uma aresta entre eles. Hamada e Yoshimura ([HAMADA; YOSHIMURA, 1976](#)) definiram o *middle graph* (grafo-meio). O *grafo-meio*  $M(G)$  de um grafo  $G$  é definido como segue. O conjunto de vértices  $V(M(G))$  é  $V(G) \cup E(G)$ . Dois vértices  $v, w \in V(M(G))$  são adjacentes em  $M(G)$  se (i)  $v, w \in E(G)$  e  $v$  e  $w$  associar a arestas incidentes a um mesmo vértice em  $G$  ou  $v \in V(G)$  e  $w \in E(G)$  e  $w$  corresponde a uma aresta é incidente a  $v$  em  $G$ . Outra maneira de construir um grafo-meio de  $G$  é removendo as arestas de  $G$  de  $Tot(G)$ .

## 1.3 Contribuições

Este trabalho está interessado em uma investigação mais geral, minimizando distâncias entre vértices e arestas em grafos, que é considerar distâncias em grafos totais. Além disso, também investiga em alguns subgrafos de grafos totais e outras classes de grafos que satisfaçam também a operação de grafo clique-aumentante proposta.

**Árvores total geradoras** A *árvore total  $t$ -geradora* de um grafo  $G$  é uma árvore geradora de  $T$  de  $Tot(G)$  tal que, para quaisquer dois vértices e arestas adjacentes de  $G$ , a distância deles é no máximo  $t$  em  $T$ . Portanto, uma árvore total  $t$ -geradora de  $G$  é uma árvore

$t$ -geradora de  $\text{Tot}(G)$ . Um grafo  $G$  que possui uma árvore total  $t$ -geradora é chamado *total  $t$ -admissível*. O menor  $t$  para o qual  $G$  é um grafo total  $t$ -admissível é o *índice de extensão total de  $G$* , e é denotado por  $\sigma_T''(G)$  (ou simplesmente  $\sigma''(G)$  ou  $\sigma''$ ). O problema da *total  $t$ -admissibilidade* visa decidir se um determinado grafo  $G$  possui  $\sigma''(G) \leq t$ . A [Figura 1.2](#) descreve a relação entre a árvore total geradora de um grafo e a árvore geradora de seu grafo total.

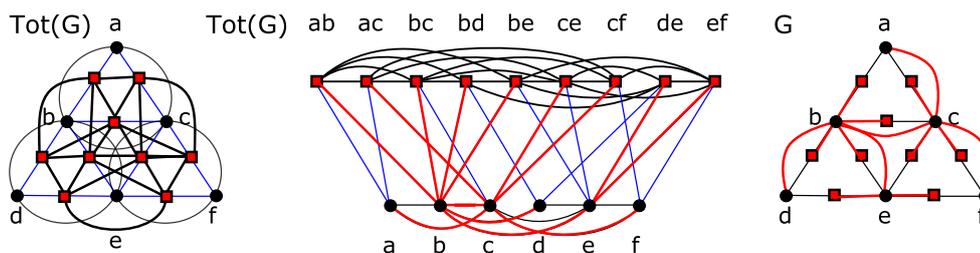


Figura 1.2: À esquerda: Um grafo total  $\text{Tot}(G)$ , vértices e arestas pretas são de  $G$ , arestas azuis estão entre  $G$  e  $L(G)$ , arestas pretas em negrito são de  $L(G)$  e vértices vermelhos são de  $L(G)$ ; no meio: outra representação de  $\text{Tot}(G)$ , bordas vermelhas definem uma árvore total 4-geradora de  $G$ ; à direita:  $G$  com a árvore total 4-geradora relacionada em vermelho.

Este trabalho define o *grafo-quase-total*,  $QT(G)$  de um grafo  $G$  da seguinte forma: O conjunto de vértices  $V(QT(G))$  é  $V(G) \cup E(G)$ . Dois vértices  $v, w \in V(QT(G))$  são adjacentes em  $QT(G)$  se forem adjacentes em  $G$  ou  $v \in V(G)$ ,  $w \in E(G)$  e  $w$  correspondem a uma aresta incidente a  $v$  em  $G$ . Outra maneira de construir um grafo-quase-total de  $G$  é remover de  $\text{Tot}(G)$  as arestas de  $L(G)$ . A [Figura 1.3](#) apresenta o grafo  $G$ , o grafo-meio  $M(G)$  e o grafo-quase-total  $QT(G)$ .

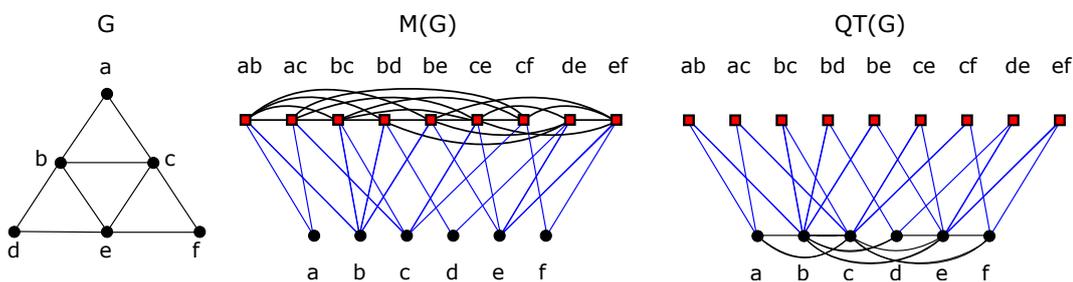


Figura 1.3: À esquerda  $G$ , no centro o grafo-meio de  $G$ ,  $M(G)$  e à direita o grafo-quase-total de  $G$ ,  $QT(G)$ .

Nesta dissertação, apresentamos as seguintes contribuições:

- Relações entre os parâmetros  $\sigma(G)$ ,  $\sigma'(G)$  e  $\sigma''(G)$ .
- Provas de NP-completude para total-admissibilidade.

- Operações em grafos que admitam árvore total 4-geradora, induzindo a algoritmos polinomiais para a obtenção do índice de extensão.
- Definição de grafos clique-aumentantes  $CA(G)$  e relações entre  $\sigma(G)$  e  $\sigma(CA(G))$ , cujas relações implicam em algoritmos eficientes ou provas de NP-completude para obtenção dos índices de extensão.

## 1.4 Organização do trabalho

O [Capítulo 2](#) apresenta preliminares sobre: complexidade computacional, aprofundamento em relação ao problema de decisão de árvores  $t$ -geradoras, a NP-Completeness para o problema da  $t$ -admissibilidade (CAI; CORNEIL, 1995) para  $t \geq 4$  e detalhamento de outros trabalhos relacionados na literatura.

O [Capítulo 3](#) e o [Capítulo 4](#) apresentam as contribuições desta dissertação.

No [Capítulo 3](#), a [Seção 3.1](#) apresenta uma relação entre os três parâmetros sobre árvores geradoras de um grafo  $G$ , que são  $\sigma(G)$ ,  $\sigma'(G)$  e  $\sigma''(G)$ . A [Seção 3.2](#) é dedicada ao problema de árvore total admissibilidade e prova que  $\sigma''(G)$  é limitado superiormente por  $\sigma(G) + 2$ , prova também que ÁRVORE TOTAL 18-ADMISSIBILIDADE é NP-completo para grafos arbitrários ou se os grafos de instância são restritos a grafos bipartidos ou grafos planares, e como o problema é difícil em geral, isso motiva a apresentar condições suficientes para que os grafos sejam árvore total 4-admissíveis e como subproduto determina algoritmos solucionáveis em tempo polinomial para definir o índice de extensão total para grafos de junção (como cografos) e grafos *split*.

O [Capítulo 4](#) é dedicado a grafos clique-aumentantes, que é uma operação que generaliza a construção de grafos-meio, grafos-quase-totais e relaciona os índices de extensão de grafos clique-aumentantes e seus grafos base. Apresentamos relações de admissibilidade de grafos antes e após as operações serem aplicadas. Para isso, lidamos com alguns possíveis tamanhos de cliques para gerar grafos cliques-aumentantes.

No [Capítulo 5](#) são apresentadas as considerações finais e trabalhos futuros.

# 2 Preliminares sobre Complexidade Computacional e Árvore $t$ -Geradora

Este capítulo apresenta discussões sobre complexidade computacional, NP-completude e classes de problemas computacionais. Apresentamos definições básicas sobre classes de grafos referenciadas nesta dissertação, detalhamos a prova de NP-completude de árvores 4-geradoras (CAI; CORNEIL, 1995) e descrevemos outros trabalhos relacionados na literatura acerca da determinação do índice de extensão em grafos. Para o leitor que quiser ir direto às preliminares acerca da complexidade computacional de determinar uma árvore  $t$ -geradora, recomendamos a [Seção 2.3](#).

## 2.1 Complexidade computacional

É possível implementar máquinas que em algum nível fazem computação, como autômatos, por exemplo, e esse modelo evolui para modelos mais complexos e poderosos capazes de resolver um conjunto maior de problemas. As máquinas de Turing, descritas pela primeira vez por Alan Turing (TURING, 1936), são dispositivos computacionais abstratos simples destinados a ajudar a investigar a extensão e as limitações do que pode ser computado. Elas possuem o mesmo poder de expressão dos algoritmos, pois tudo que você é capaz de resolver utilizando um computador é capaz de visualizar como um algoritmo. Se você é capaz de fazer um algoritmo para resolver um problema, você é capaz de projetar uma Máquina de Turing para resolvê-lo.

A Tese de Church-Turing (CHURCH, 1936) enuncia que qualquer forma de computação do mundo real pode ser simulada através de uma Máquina de Turing. É uma tese que não se tem uma prova irrefutável, pois não sabemos o que pode vir a ser no futuro “qualquer forma de computação do mundo real”. Logo, a tese é um axioma da Ciência da Computação. Como a Tese de Church-Turing é mais geral e não necessariamente a forma que fazemos algoritmo hoje é a única que existe, o que conseguimos provar é que a noção habitual de algoritmo é equivalente à computação via Máquina de Turing.

Um algoritmo pode possuir um tempo determinístico, ou seja, consigo determinar exatamente um limite superior para o tempo que o algoritmo consome. Preciso distinguir um código que entra em *loop* infinito de um código que demora por causa de esforço computacional muito grande. Se eu tenho o limite superior do algoritmo, consigo identificar após determinado tempo se entrou em *loop* ou não. Um algoritmo deve possuir correteude para garantir que funcione em todos os casos.

Para medir eficiência, devemos contar o número de passos que um algoritmo executa em função de uma determinada entrada de tamanho definido. Essa contagem nos dá uma função em relação a esse tamanho da entrada. Como entradas diferentes de tamanho iguais podem ter números de passos de execução diferentes, para realizar a análise, consideramos o pior caso. A análise de pior caso dá um limite superior para o algoritmo independente da instância de entrada  $e$ , com esse limite, podemos determinar se um algoritmo é eficiente ou não. Essa função obtida pela análise do pior caso nos dá a complexidade do algoritmo.

Um algoritmo eficiente possui complexidade definida por uma função polinomial em relação ao tamanho da entrada. Analogamente, em um algoritmo ineficiente, essa função é exponencial. Saber distinguir problemas que não admitem soluções eficientes de problemas que podem admitir, mesmo que a solução não seja conhecida, é importante, por exemplo, para evitar gastar esforço resolvendo problemas de forma mais eficiente que não podem ser resolvidos.

As funções polinomiais são aquelas em que o expoente não cresce em função da entrada. Por exemplo  $n$ ,  $n \log n$  e  $n^3$  são polinomiais e  $2^n$ ,  $n^{\log n}$  e  $n!$  não. Funções não polinomiais crescem muito em função da entrada  $e$ , portanto, algoritmos com essa complexidade apresentam tempo de execução que cresce bastante a medida que a entrada cresce. Quando determinamos um limite superior para um algoritmo, determinamos que não precisamos de mais tempo do que esse limite para resolver o problema usando esse algoritmo, porém, não significa que não existam outros algoritmos com outras complexidades para resolver o mesmo problema. Quando definimos o limite inferior do problema, determinamos que não existem algoritmos mais eficientes que esse limite inferior para resolver o problema. Quando encontramos um algoritmo com complexidade equivalente à complexidade (limite inferior) do problema, concluímos que temos um algoritmo ótimo para o problema.

Demonstração de limites inferiores é algo que ainda não se sabe fazer muito bem. Há limites inferiores para determinados problemas que são naturais, para outros não. Por exemplo, em multiplicação de matrizes podemos considerar um limite inferior natural o

número de células da matriz, ou seja,  $n^2$ , pois precisaremos no mínimo visitar pelo uma vez cada posição.

Quando analisamos o problema e não o algoritmo, estamos interessados em entender a sua natureza para determinar o limite inferior para qualquer algoritmo. O limite inferior de um problema é o número de passos do pior caso entre qualquer algoritmo que resolve esse problema. Se provarmos um limite inferior de um problema, podemos provar que um problema não é polinomial, ou seja, que não existe algoritmo com complexidade polinomial que o resolve. Porém, para vários problemas, não somos capazes de fazer uma demonstração matemática que dê um limite inferior, principalmente para aqueles com limites não polinomiais.

Para calcular esses limites, precisamos nos basear em hipóteses e a principal hipótese utilizada no estudo da NP-completude é a que a classe dos problemas resolvidos em tempo polinomial de forma determinística (classe P) é diferente da classe dos problemas resolvidos em tempo polinomial de forma não determinística (classe NP).

**Definição 2.** *A classe P é o conjunto dos problemas de decisão que podem ser resolvidos por um algoritmo polinomial.*

**Definição 3.** *A classe NP é o conjunto dos problemas de decisão verificáveis em tempo polinomial. Ou seja, dada uma instância do problema para o qual a resposta é “sim”, existe um algoritmo de validação que verifica a corretude dessa resposta em tempo polinomial.*

### 2.1.1 NP-completude

Na década de 70, os cientistas Stephen Cook e Leonid Levin desenvolveram independentemente estudos que introduziram o conceito de NP-completude (COOK, 1971; LEVIN, 1973). A contribuição deles é conhecida como Teorema Cook-Levin e define a classe NP-completa como um subconjunto da classe NP composta pelos problemas aos quais todos os elementos da classe NP podem ser reduzidos em tempo polinomial. Os problemas dessa classe são denominados NP-completos. Quando não há necessidade de um problema  $A$  estar em NP mas além disso todos os problemas de NP se reduzem a  $A$ , então  $A$  é NP-difícil.

A Figura 2.1 ilustra a relação entre as classes P, NP, NP-completa e NP-difícil.

O processo de redução polinomial, ou redução de Karp, tem como objetivo determinar afinidades entre problemas computacionais (KARP, 1972).

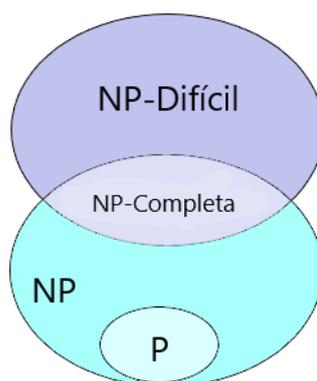


Figura 2.1: Relações entre as classes de problemas.

**Definição 4.** *Considerando dois problemas de decisão  $P1$  e  $P2$ , a redução de Karp consiste em transformar, em tempo polinomial, uma instância de  $P1$  em uma instância de  $P2$  tal que o retorno de  $P2$  para a instância transformada é afirmativo se e somente se o retorno de  $P1$  para a instância original também for.*

A existência de uma redução em tempo polinomial de  $P1$  para  $P2$  implica que  $P2$  é pelo menos tão difícil quanto  $P1$ . Uma consequência importante dessa definição para a classe NP-completa é que caso seja encontrado um algoritmo polinomial que soluciona qualquer problema a ela pertencente, então todos os problemas da classe NP poderiam ser resolvidos em tempo polinomial, o que seria suficiente para provar que  $P = NP$ .

### 2.1.2 Questão P versus NP

A hipótese  $P \neq NP$  é considerada forte e relevante, mas quando nos baseamos nela, apesar de termos uma prova matemática baseada em uma boa hipótese, ela ainda é mais fraca que uma prova matemática irrefutável.

Problemas polinomiais são aqueles que admitem um algoritmo polinomial. Se quero provar que um problema não é polinomial basta demonstrar um limite inferior não polinomial para o problema. Até hoje, ainda não temos um ferramental matemático muito bom para essa demonstração. Já para provar que um problema é polinomial basta projetar um algoritmo que resolve o problema em tempo polinomial. Quando provamos que um problema é polinomial implica que muito provavelmente existem bons algoritmos para resolver o problema e mesmo que se projete um algoritmo polinomial de grau alto, isso “abre as portas” para desenvolvermos algoritmos mais eficientes, ou seja, de polinômios cada vez menores.

Problemas com complexidade computacional em aberto são aqueles que não consigo

projetar um algoritmo polinomial e não consigo provar um limite inferior não polinomial. Ou seja, não consigo provar que esse problema está em  $P$  e também que não está em  $P$ .

A classe  $NP$  é a classe dos problemas que admitem certificados positivos que podem ser verificados em tempo polinomial. Ou seja, significa que uma possível solução positiva para o problema é verificada/validada em tempo polinomial. Uma forma de visualizar essa computação é considerar que a descoberta de um certificado é feita por um Oráculo e não sabemos como ele faz isso. O Oráculo fornece uma solução positiva e o problema é apenas verificar essa solução em tempo polinomial.

Os problemas  $P$  e  $NP$  são decidíveis, ou seja, sabemos que a máquina vai parar sempre, porém, não garantimos que essa decisão vai ser feita em tempo polinomial quando usamos determinismo para problemas  $NP$ . Os problemas que conseguimos decidir em tempo polinomial via Máquina de Turing não determinística (MTND) são aqueles que estão em  $NP \cap coNP$ . O que não é provado é se a classe  $NP$  é uma superclasse da classe  $P$  ou é igual. Portanto, a questão  $P$  versus  $NP$  é uma questão teórica sobre o poder de computação dessas classes quando se adiciona a restrição de tempo polinomial. Se provarmos que  $P = NP$  então todo problema que admite certificado positivo via MTND em tempo polinomial pode ser solucionado em tempo polinomial. Se provarmos que um problema que admite certificado positivo verificado em tempo polinomial não pode ser solucionado em tempo polinomial através da demonstração de um limite inferior não polinomial para o problema, provamos  $P \neq NP$ .

Supondo que existe uma prova que  $P \neq NP$ , não podemos afirmar que os problemas que têm certificado verificado em tempo polinomial, mas não conhecemos um algoritmo polinomial que resolva, estão em  $NP \setminus P$ . A  $NP$ -completude tenta identificar limites inferiores não polinomiais se  $P \neq NP$ , ou seja, se alguém provar essa condição de forma irrefutável, certamente esses problemas não serão resolvíveis em tempo polinomial. É uma prova de intratabilidade assumindo a hipótese  $P \neq NP$ . Isso significaria que não determinismo realmente é capaz de aumentar o poder de computação de uma Máquina de Turing. Outro benefício de estudar esses problemas é que se caso alguém descubra um algoritmo polinomial para qualquer problema nessa condição, então  $P$  será igual a  $NP$ , contradizendo a hipótese.

## 2.2 Definições básicas em grafos

Seja  $G = (V, E)$  um grafo. Seja  $N(u) = \{v \mid uv \in E(G)\}$  a vizinhança de um vértice  $v \in V(G)$ . O grau de um vértice  $v \in V(G)$  é  $d_G(v) = |N(v)|$ . Um *pendente*  $v$  é um vértice de grau  $d_G(v) = 1$ . O grau máximo de  $G$ , denotado por  $\Delta(G)$ , é  $\max_{v \in V} \{d_G(v)\}$ .  $N_E(u) = \{uv \mid uv \in E(G)\}$ ,  $N_E(uv) = N_E(u) \cup N_E(v)$  para uma aresta  $uv \in E(G)$ . A distância  $d_G(u, v)$  entre dois vértices  $u$  e  $v$  de  $G$  é o número mínimo de arestas em um caminho entre  $u$  e  $v$  em  $G$ . Um *subgrafo*  $H$  de  $G$  tem  $V(H) \subseteq V(G)$  e  $E(H) \subseteq E(G)$ . Um *subgrafo induzido*  $H[S]$  em um conjunto de vértices  $S$  de  $G$  tem  $V(H[S]) = S$  e  $E(H[S]) = \{uv \mid u \in S, v \in S, \text{ and } uv \in E(G)\}$ . Um subconjunto  $S \subseteq V(G)$  é um *conjunto independente* se não houver aresta entre seus pares de vértices e é uma *clique* se houver todas as arestas possíveis entre seus pares. Um grafo  $G$  é dito *H-livre* se não contém  $H$  como subgrafo induzido de  $G$ . Um *vértice simplicial* é aquele cujos vizinhos formam uma *clique*, ou seja, cada dois vizinhos são adjacentes. Uma *ponte* é uma aresta de corte do grafo, ou seja, é uma aresta que aumenta o número de componentes conexas do grafo após sua remoção. Um grafo  $G = G_1 \vee G_2$  é uma *união* entre dois grafos  $G_1$  e  $G_2$  se  $V(G) = V(G_1) \cup V(G_2)$  e  $E(G) = E(G_1) \cup E(G_2)$ . Um grafo  $G = G_1 \wedge G_2$  é uma *junção* entre dois grafos  $G_1$  e  $G_2$  se  $V(G) = V(G_1) \cup V(G_2)$  e  $E(G) = E(G_1) \cup E(G_2) \cup \{uv \mid u \in V(G_1) \text{ e } v \in V(G_2)\}$ .

Um grafo de interseção  $G = \Omega(I)$  de um modelo  $I$  tem os elementos de  $I$  como seus vértices e há uma aresta entre dois vértices se os elementos tiverem alguma interseção no modelo  $I$ . Um *grafo linha* é o grafo de interseção das arestas de um grafo e um *grafo total* é o grafo de interseção dos vértices e arestas de um grafo. Uma *subdivisão*  $S_1(G)$  de um grafo  $G$  é obtida de  $G$  substituindo cada aresta  $uv \in E(G)$  por um caminho de tamanho três  $u, uv, v$ . Ou seja,  $V(S_1(G)) = V(G) \cup E(G)$  e existem arestas  $E(S_1(G)) = \{u uv, uv v \mid uv \in E(G)\}$ . O *quadrado*  $G^2$  de um grafo  $G$  possui  $V(G^2) = V(G)$  e  $E(G^2) = E(G) \cup \{uv \mid d_G(u, v) = 2\}$ .

Um grafo é *bipartido* se seus vértices podem ser particionados em dois conjuntos independentes. Um *grafo bipartido completo* é um grafo bipartido tal que cada vértice de um conjunto independente é adjacente a todos os vértices do outro conjunto independente.  $K_{n,m}$  é a denotação de grafo bipartido completo onde um conjunto independente possui  $n$  vértices e o outro possui  $m$  vértices. O *grafo garra* é o grafo  $K_{1,3}$ . Um grafo é *grafo cordal* se todo ciclo de tamanho pelo menos quatro tem uma *corda*, que é uma aresta que não está no conjunto de arestas de um ciclo  $C$  cujas extremidades estão no conjunto de vértices de  $C$ . Um grafo é *grafo split* se seus vértices podem ser particionados em

um conjunto independente e uma clique. Um grafo é *cografo* se for  $P_4$ -livre, ou seja, não existe caminho induzido com 4 vértices. Outra forma de definir cografos é por operações sucessivas de união ou junção de cografos, ou seja, união (e junção) de cografos é um grafo. Um *grafo planar* é um grafo desenhado de forma que nenhuma aresta se cruze. Cada região formada por um grafo planar é chamada de *face*. Dado um grafo planar  $G$ , seu *grafo dual*  $G^*$  é construído colocando um vértice em cada face de  $G$  e, se duas faces de  $G$  tiverem uma aresta em comum, os vértices correspondentes são conectados por apenas um cruzamento de arestas em  $G^*$ . O *grafo gema* pode ser obtido a partir de um ciclo  $C_5$  incluindo duas arestas com extremos em um mesmo vértice e dois outros vértices não adjacentes a esse vértice no ciclo. Um grafo é livre de gema quando não contém o grafo gema como subgrafo induzido.

Para mais definições em grafos, recomendamos (BONDY; MURTY et al., 1976).

## 2.3 NP-completude da 4-admissibilidade

Cai e Corneil (CAI; CORNEIL, 1995) provaram que a construção de uma árvore 2-geradora em um grafo pode ser feita em tempo linear e que o problema da  $t$ -admissibilidade é NP-completo para  $t \geq 4$ . Por outro lado, encontrar uma árvore 3-geradora, apesar de parecer um problema difícil de resolver, ainda está em aberto.

A seguir, é apresentada uma revisão e um maior detalhamento do teorema de Cai e Corneil (CAI; CORNEIL, 1995) para a NP-completude de árvores 4-geradoras.

**Teorema 5.** *Para qualquer  $t \geq 4$ , o problema da árvore  $t$ -geradora é NP-Completo.*

*Demonstração.* Como o problema está em NP, pois é possível verificar em tempo polinomial uma solução dada, a estratégia utilizada é realizar uma redução polinomial de 3-SAT para o problema da árvore  $t$ -geradora, provando que se trata de um problema NP-Difícil, logo NP-Completo. Seja  $t \geq 4$  e  $(U, C)$  uma instância arbitrária de 3-SAT. Onde  $U$  é o conjunto de variáveis e  $C$  o conjunto de cláusulas, constrói-se um grafo  $G$  pela [Construção 7](#) tal que  $C$  é admissível se somente se  $G$  tem uma árvore  $t$ -geradora.

Para completar a prova do [Teorema 5](#), precisamos ainda provar que  $C$  é admissível se somente se  $G$  tem uma árvore  $t$ -geradora. Mas antes disso, precisamos trazer o [Lema 6](#) que será necessário para a [Construção 7](#).

**Lema 6.** *Seja  $G$  um grafo e  $e$  uma aresta de  $G$ . Seja  $G'$  um grafo formado de  $G$  adicionando dois  $t$ -caminhos (chama-se de  $t$ -caminho um caminho com  $t$  arestas) distintos  $P_1$ ,*

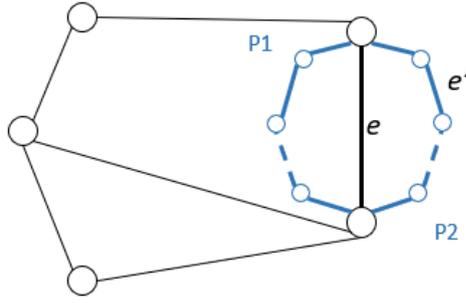


Figura 2.2: Exemplo de aresta  $e$  forçada e caminhos  $P1$  e  $P2$  forçados.

$P2$  (todos os vértices internos de  $P1$  e  $P2$  são novos vértices) entre as duas extremidades de  $e$  e seja  $T$  uma árvore  $t$ -geradora de  $G'$ . Então a aresta  $e \in E(T)$ . Chamamos  $e$  de aresta forçada e os caminhos  $P1$  e  $P2$  de caminhos forçados como na [Figura 2.2](#).

*Demonstração.* Para toda aresta  $e'$  de  $P1$  e  $P2$  há apenas um caminho em  $G' - e'$  de comprimento  $\leq t$  entre as duas extremidades de  $e'$ . Além disso, este único caminho contém a aresta  $e$ . Segue-se que se  $e$  não está em  $T$ , então todas as arestas de  $P1$  e  $P2$  teriam que estar em  $T$ . Porém,  $P1$  e  $P2$  formam um ciclo, o que contradiz que  $T$  seja uma árvore.  $\square$

**Construção 7.** Seja  $|U| = n$  e  $|C| = m$ . Para todo  $u_i \in U, 1 \leq i \leq n$ , vamos construir um grafo  $H_i$  da seguinte forma: 1) Criar 5 vértices  $x_i, u_i, \bar{u}_i, y_i$  e  $z_i$ ; 2) Adicionar arestas  $x_i y_i, x_i u_i, x_i \bar{u}_i, z_i u_i$  e  $z_i \bar{u}_i$ ; 3) Juntar  $y_i$  com  $z_i$  por um  $(t-2)$ -caminho e forçar as arestas; 4) Juntar  $u_i$  com  $\bar{u}_i$  por um  $(t-3)$ -caminho e forçar as arestas. A [Figura 2.3\(a\)](#) ilustra a construção de um grafo  $H_i$ . Em seguida, ligar todos  $H_i$ 's através da união dos vértices  $x_i$  em um único vértice  $x$  e para todo  $c_j \in C, 1 \leq j \leq m$ , criar seu vértice correspondente e ligar aos vértices literais da cláusula. [Figura 2.3\(b\)](#) ilustra a ligação dos  $H_i$ 's com os vértices  $c_j$ 's.

É fácil verificar que essa construção pode ser feita em tempo polinomial, então basta mostrar que  $C$  é satisfeito se e somente se  $G$  possui uma árvore  $t$ -geradora. Antes de descrever a prova, o [Lema 8](#) a seguir permite definir uma propriedade de atribuição para  $C$  a partir de uma árvore  $t$ -geradora de  $G$ .

**Lema 8.** Qualquer árvore  $t$ -geradora  $T$  de  $G$  contém exatamente uma das duas arestas  $xu_i$  e  $x\bar{u}_i$  para cada  $1 \leq i \leq n$ .

*Demonstração.* Apenas uma das duas arestas pode estar em  $T$ , caso contrário haveria um ciclo  $(xu_i, x\bar{u}_i, u_i \bar{u}_i)$ . Suponha que nem  $xu_i$  nem  $x\bar{u}_i$  estejam em  $T$ , logo o  $(x, u_i)$ -caminho

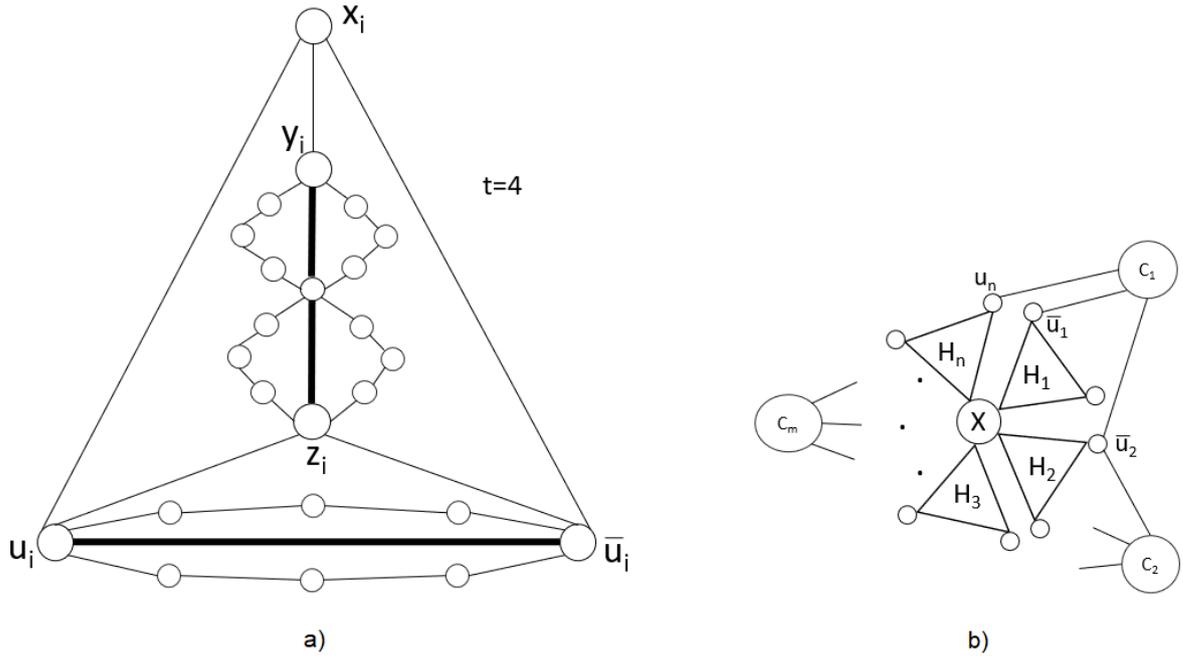


Figura 2.3: a) Construção de  $H_i$  para  $t = 4$  b) Ligação dos  $H_i$ 's com o vértice  $x$  e os vértices de cláusula  $c_j$ .

em  $T$  tem que passar por  $F = G - E(H_i)$  e tem tamanho maior ou igual a 3 (de  $x$  para  $u_k|\bar{u}_k$ , depois para  $c_j$  e depois para  $u_i$ ). Temos então 2 casos possíveis.

Caso i)  $xy_i \in E(T)$ . Se  $z_i u_i, z_i \bar{u}_i \notin E(T)$  então não existe um  $(x, u_i \bar{u}_i)$ -caminho em  $H_i$ , logo  $d_T(x, u_i) \geq d_F(x, u_i) \geq 3$ , então  $d_T(z_i, u_i) = d_T(z_i, x) + d_T(x, u_i) \geq (t-1) + d_F(x, u_i) \geq t+2$ , que é uma contradição. Se  $z_i u_i \in E(T)$  (sem perda de generalidade, note que  $z_i \bar{u}_i \notin E(T)$ ) então  $d_T(x, \bar{u}_i) = d_T(x, z_i) + d_T(z_i, u_i) + d_T(u_i, \bar{u}_i) = (t-1) + 1 + (t-3) = 2t-3 > t$  para  $t \geq 4$ , que é outra contradição.

Caso ii)  $xy_i \notin E(T)$  então não existe um  $(x, u_i|\bar{u}_i)$ -caminho dentro de  $H_i$ , logo  $d_T(x, u_i) \geq d_F(x, u_i) \geq 3$  e  $d_T(x, \bar{u}_i) \geq d_F(x, \bar{u}_i) \geq 3$ , então  $d_T(x, z_i) = \min\{d_T(x, u_i) + d_T(u_i, z_i), d_T(x, \bar{u}_i) + d_T(\bar{u}_i, z_i)\} \geq 4$ , logo  $d_T(x, y_i) = d_T(x, z_i) + d_T(z_i, y_i) \geq t+2$ , que é uma contradição para  $T$  ser uma árvore  $t$ -geradora.  $\square$

Para completar a prova do Teorema 5, falta provar que dada a Construção 7  $C$  é admissível se e somente se  $G$  tiver uma árvore  $t$ -geradora  $T$ . Suponha que  $\varepsilon$  seja uma atribuição que torna  $C$  admissível, vamos construir  $T$ . Para cada aresta forçada  $e$ , coloque-a em  $T$ . Para cada caminho forçado, coloque todas as arestas  $e'$  do caminho em  $T$ , exceto uma aresta arbitrária. Para todo  $u_i \in U$ ,  $1 \leq i \leq n$ , se  $\varepsilon(u_i) = 1$ , coloque  $xu_i$  e  $z_i u_i$  em  $T$ , senão,  $\varepsilon(u_i) = 0$ , coloque  $xu_i$  e  $z_i \bar{u}_i$  em  $T$ . Para todo  $c_j \in C$ ,  $1 \leq j \leq m$ , arbitrariamente pegue um literal verdadeiro  $l_j$  de  $c_j$  e coloque a aresta  $c_j l_j$  em  $T$ .

Suponha que  $T$  seja uma árvore  $t$ -geradora de  $G$ . Apresentamos  $\varepsilon_T$  uma atribuição para  $C$ . Pelo [Lema 8](#),  $T$  contém exatamente uma das duas arestas  $xu_i$  e  $x\bar{u}_i$ . Definimos  $\varepsilon_T(u_i) = 1$  se  $xu_i$  está em  $T$ , caso contrário, atribuímos  $\varepsilon_T(u_i) = 0$ . Suponha que uma cláusula  $c_j$  só possua literais falsos. Então para uma aresta  $c_j l_j \notin E(T)$ ,  $d_T(c_j, l_j) = (t-3) + 1 + 1 + (t-3) + 1 = 2t - 3 \geq t + 1$  para  $t \geq 4$ , enquanto a distância de quaisquer literais falsos deve ser  $2t - 4$  em  $T$ , o que é uma contradição. Portanto, cada cláusula  $c_j$  deve conter ao menos um literal verdadeiro em  $\varepsilon_T$  e então  $C$  é admissível. Com isso, concluimos a demonstração do [Teorema 5](#).  $\square$

## 2.4 Admissibilidade em operações de grafos

A determinação da dicotomia P versus NP-completa da  $t$ -admissibilidade é o maior desafio até agora, uma vez que é conhecido que  $t$ -admissibilidade, para  $t \geq 4$  é NP-completo, 2-admissibilidade é um problema solucionável em tempo polinomial, enquanto 3-admissibilidade é um problema aberto há mais de 28 anos ([CAI; CORNEIL, 1995](#)). Portanto, vários artigos ([BRANDSTÄDT; DRAGAN; LE et al., 2004](#); [COUTO; CUNHA, Luís Felipe I., 2018, 2020](#); [PANDA; DAS, 2018](#)) desenvolveram condições necessárias ou estabeleceram grafos 3-admissíveis. Devido à dificuldade em determinar o índice de extensão dos grafos, muitos estudos baseiam-se também na determinação de limites para o problema.

Um dos parâmetros usados para limites inferiores é a cintura do grafo. A cintura de um grafo  $G$ , denotada por  $g(G)$ , é o comprimento do menor ciclo de  $G$ . Portanto, como nenhuma árvore contém um ciclo, então, para qualquer grafo  $t$ -admissível,  $t \geq g(G) - 1$ . O problema da árvore geradora de diâmetro mínimo é o problema de determinar uma árvore geradora de  $G$  com diâmetro mínimo, denotado por  $D_T(G)$ . Tal parâmetro pode ser determinado em tempo polinomial ([HASSIN; TAMIR, 1995](#)). ([COUTO; CUNHA, Luís Felipe I., 2021](#)) apresentaram limites relacionando parâmetros de cintura e diâmetro para o problema da  $t$ -admissibilidade, onde, dado uma cintura  $g(G)$  de um grafo  $G$ , então  $g(G) - 1 \leq \sigma_T(G) \leq D_T(G)$ .

**$t$ -admissibilidade para grafos prismas complementares e prisma** Couto e Cunha realizaram estudo para determinar o índice de extensão do grafo prisma complementar de  $G$ , denotado como  $G\bar{G}$  ([COUTO; CUNHA, Luís Felipe I., 2020](#)). Considerando esta abordagem, obtiveram árvores 4-geradoras para grafos prismas complementares  $G\bar{G}$ , e também provaram que a  $t$ -admissibilidade é um problema solucionável em tempo polinomial para

grafos  $G\overline{G}$ . Por último, provaram que para grafos bipartidos  $G$  com emparelhamento perfeito, decidir se  $G$  é  $t$ -admissível é NP-completo, assim como para grafos prismas  $GG$ .

Gómez, Miyazawa e Wakabayashi (GÓMEZ; MIYAZAWA; WAKABAYASHI, 2022) investigaram o problema de árvores  $t$ -geradoras em grafos prisma, e caracterizaram aqueles que admitem uma árvore 3-geradora. Como resultado, obtiveram um algoritmo de tempo linear para o problema. Estudaram também uma partição das arestas de um grafo relacionada à condição de distância imposta por uma árvore  $t$ -geradora, e derivaram uma condição necessária verificável em tempo polinomial para a existência de uma árvore  $t$ -geradora em um grafo arbitrário. Como consequência, mostraram que o problema de árvore  $t$ -geradora pode ser resolvido em tempo polinomial na classe de prismas.

**$t$ -admissibilidade para grafos linha** Além do problema da  $t$ -admissibilidade, existem diversas outras variações, como a versão para grafos direcionados, grafos com arestas ponderadas (CAI; CORNEIL, 1995) ou problemas de árvores geradoras em subgrafos (PELEG; ULLMAN, 1989). Outra variação é a proposta recente do problema da aresta  $t$ -admissibilidade, onde o objetivo é minimizar distâncias entre arestas de um grafo (COUTO; CUNHA; POSNER, 2021). Dado um grafo  $G$ , seu grafo linha  $L(G)$  é obtido da seguinte forma:  $V(L(G)) = E(G)$ ;  $E(L(G)) = \{\{uv, uw\} | uv, uw \in E(G)\}$ . Ou seja, cada aresta de  $G$  é um vértice de  $L(G)$  e se duas arestas compartilham um vértice, então seus vértices correspondentes são adjacentes em  $L(G)$ . A distância entre duas arestas  $e_1$  e  $e_2$  de  $G$ , para  $e_1, e_2 \in E(G)$  é a distância entre seus vértices correspondentes em  $L(G)$ . Em (COUTO; CUNHA; POSNER, 2021), o estudo da  $t$ -admissibilidade em grafos linha é amplamente explorado, foi definida a árvore aresta  $t$ -geradora de um grafo  $G$  como uma árvore geradora  $T$  de  $L(G)$  tal que, para quaisquer duas arestas adjacentes de  $G$ , sua distância é no máximo  $t$  em  $T$ . Portanto, uma árvore aresta  $t$ -geradora de  $G$  é uma árvore  $t$ -geradora de  $L(G)$ . O grafo  $G$  que possui uma árvore aresta  $t$ -geradora é chamado de aresta  $t$ -admissível. Embora a decisão de 3-admissibilidade esteja aberta há mais de 2 décadas, foi provado que decidir se um grafo é aresta 3-admissível é um problema solucionável em tempo polinomial, e também aresta  $2k$ -admissibilidade, para  $k \geq 4$  é NP-completa, mesmo para grafos bipartidos. Na Subseção 2.4.1, descrevemos em mais detalhes a NP-completude da aresta 8-admissibilidade.

### 2.4.1 NP-completude da aresta 8-admissibilidade

A seguir, é apresentada a transformação de tempo polinomial descrita em (COUTO; CUNHA; POSNER, 2021) de 3-SAT (GAREY; JOHNSON, 1979) para a aresta 8-admissibilidade de grafos bipartidos.

**Construção 9.** Dada uma instância  $I = (U, C)$  de 3-SAT construímos um grafo  $G$  como segue. Adicionamos um  $P_2$  com os rótulos  $x$  e  $x'$  a  $G$ . Para cada variável  $u \in U$  adicionamos  $C_8$  a  $G$  com três vértices consecutivos rotulados como  $u$ ,  $m_u$  e  $\bar{u}$  e os outros cinco vértices consecutivos rotulados como  $u_1$  a  $u_5$ . Para cada  $u_i, i = 1, \dots, 5$ ,  $u$  e  $\bar{u}$  adicionamos um vértice pendente. Para cada variável  $u \in U$  adicionamos a aresta  $xm_u$  a  $G$ . Para cada cláusula  $c_1 = (u, v, w) \in C$ , adicionamos dois vértices  $c_1$  e  $c_1'$  a  $G$  e as arestas  $c_1c_1'$ ,  $c_1u$ ,  $c_1v$  e  $c_1w$ . Para cada variável  $u \in U$  adicionamos um  $P_4$  a  $G$  com extremos rotulados  $p_{u1}$  e  $p_{u4}$  e as arestas  $p_{u1}x$  e  $p_{u4}m_u$ .

A Figura 2.4 mostra um exemplo de grafo obtido a partir de uma instância de 3-SAT.

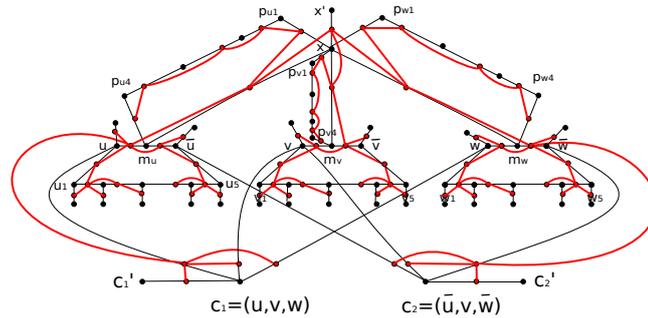


Figura 2.4: Grafo obtido da Construção 9 da instância  $I = (\{u, v, w\}, \{(u, v, w), (\bar{u}, v, \bar{w})\})$  e uma árvore aresta 8-geradora do grafo em vermelho.

A ideia central da prova do Teorema 10 é que, para cada variável  $u \in U$ , temos exatamente uma aresta na árvore aresta 8-geradora  $T$  que está próxima de  $x$  e  $u$  ou  $\bar{u}$ . Relacionamos esta proximidade com uma atribuição verdadeira desse literal. A seguir, exigimos que pelo menos uma aresta incidente em cada cláusula esteja conectada a um literal verdadeiro. Caso contrário, se todos forem literais falsos, terminaremos com duas das arestas incidentes nessa cláusula sendo vértices de  $L(G)$  com distância de pelo menos 9 em  $T$ .

**Teorema 10.** *Aresta 8-admissibilidade é NP-completo para grafos bipartidos.*

*Demonstração.* Por construção,  $G$  é bipartido. Além disso, não só o problema está em NP, mas também o tamanho do grafo  $G$ , obtido de Construção 9 em uma instância  $I = (U, C)$

de 3-SAT, é polinomialmente limitado pelo tamanho de  $I$ . Provamos que  $G$  é aresta 8-admissível se e somente se houver uma atribuição de verdade para  $I$ . Considere uma atribuição verdade de  $I = (U, C)$ . Obtemos uma árvore aresta 8-geradora  $T$  de  $G$  como segue (veja Figura 2.4).

Adicione a  $T$  as arestas:  $\{x'x, xm_u \mid u \in U\}$ ;  $\{xm_u, m_u u \mid u \in U \text{ e } u \text{ é verdadeiro}\}$  ou  $\{xm_u, m_u \bar{u} \mid u \in U \text{ e } \bar{u} \text{ é verdadeiro}\}$ ;  $\{um_u, \bar{u}m_u \mid u \in U\}$ ; Para cada cláusula selecione um literal verdadeiro e adicione a  $T$ :

$\{c'c, uc \mid c \text{ é uma cláusula com o literal verdadeiro selecionado } u\}$ ;

$\{uc, um_u \mid c \text{ é uma cláusula com o literal verdadeiro selecionado } u\}$ ;

$\{\bar{u}c, \bar{u}m_u \mid c \text{ é uma cláusula com o literal verdadeiro selecionado } \bar{u}\}$ ;

$\{uc, vc \mid c \text{ é uma cláusula com o literal verdadeiro } u \text{ e } v \text{ é outro literal de } c\}$ ;

Para cada variável  $u \in U$  adicione a  $T$  as arestas:  $\{m_u p_{u_4}, p_{u_4} p_{u_3}\}$ ;  $\{p_{u_4} p_{u_3}, p_{u_3} p_{u_2}\}$ ;  $\{p_{u_3} p_{u_2}, p_{u_2} p_{u_1}\}$ ;  $\{p_{u_2} p_{u_1}, p_{u_1} x\}$ ;  $\{p_{u_1} x, xm_u\}$ ;  $\{um_u, uu_1\}$ ;  $\{\bar{u}m_u, \bar{u}u_5\}$ ;  $\{uu_1, u_1 u_2\}$ ;  $\{u_3 u_4, u_4 u_5\}$ ;  $\{u_4 u_5, \bar{u}u_5\}$ ; e cada pendente de  $G$  é adicionado a uma árvore de soluções.

Considere uma árvore aresta 8-geradora  $T$  de  $G$  (resp. árvore 8-geradora  $L(G)$ ), apresentamos uma atribuição verdadeira  $I = (U, C)$ . Primeiro afirmamos que para cada variável  $u \in U$ , existe exatamente uma destas duas arestas em  $T$ :  $\{xm_u, um_u\}$  e  $\{xm_u, \bar{u}m_u\}$ . Suponha que ambas as arestas estejam em  $T$ . Existem em  $L(G)$  dois vértices adjacentes  $u_i u_{i+1}$  e  $u_{i+1} u_{i+2}$  do ciclo  $C_9$  da variável  $u$  com distância 9 em  $T$ , uma contradição. Agora, suponha que ambas as arestas não estejam em  $T$ . Consideramos dois casos. Se não houver arestas  $p_{u_4} m_u, um_u$  ou  $p_{u_4} m_u, \bar{u}m_u$ , então existem em  $L(G)$  dois vértices adjacentes  $p_{u_4} m_u$  e  $um_u$  (ou  $\bar{u}m_u$ ) com distância de pelo menos 9 em  $T$ , pois é necessário fazer um caminho passando por  $xx'$ , uma contradição. Caso contrário, existe uma aresta  $p_{u_4} m_u, um_u$  ou  $p_{u_4} m_u, \bar{u}m_u$ . Em ambos os casos, seja  $c_1 = (u, v, w)$  uma cláusula que contém  $u$ , existem em  $L(G)$  dois vértices adjacentes  $c_1 v, vv_1$  que possuem distância pelo menos 9 em  $T$ , uma contradição.

Agora, relacione a aresta  $\{xm_u, um_u\}$  ou  $\{xm_u, m_u \bar{u}\}$  em  $T$  para cada variável  $u \in U$  a uma atribuição verdadeira do literal  $u$  ou  $\bar{u}$ . Suponha que exista uma cláusula com três literais falsos  $c_3 = (x, y, z)$ . Não importa como conectamos os vértices  $c'_3 c_3, c_3 x, c_3 y$  e  $c_3 z$  em  $T$ , dois deles têm distância de pelo menos 9 em  $T$ , uma contradição. Portanto, cada cláusula tem pelo menos um literal verdadeiro, e esta é uma atribuição verdadeira de  $I$ .  $\square$

---

A [Construção 9](#) pode ser adaptada para provar que a aresta  $2k$ -admissibilidade é NP-completa, para  $k \geq 5$ . Isso pode ser obtido subdividindo a aresta  $m_u x$  e os ciclos correspondentes a cada variável  $u$ .

# 3 Árvores $t$ -Geradoras em Grafos Totais

Este capítulo realiza um estudo detalhado sobre o problema da árvore total  $t$ -admissibilidade: apresenta relações e limites da aresta e total admissibilidades para algumas classes de grafos, desenvolve um estudo detalhado do índice de extensão total, seus limites, NP-completude e determina casos polinomiais para determinados valores do parâmetro.

## 3.1 Limites nas aresta e total admissibilidades

Ao lidar com  $\sigma(G)$ ,  $\sigma'(G)$  e  $\sigma''(G)$ , uma tentativa natural é determinar uma relação entre eles. Como os ciclos induzidos em um grafo  $G$  correspondem a ciclos de mesmo comprimento em  $L(G)$ , temos que  $\sigma'(C_n) = \sigma(C_n) = n - 1$ , pois uma árvore geradora de um ciclo é um caminho de comprimento  $n - 1$  ao remover uma das arestas do ciclo. Considerando agora  $\text{Tot}(C_n)$ , temos que este grafo é isomorfo ao grafo *potência de ciclo 2*  $C_{2n}^2$ , que é obtido a partir de um  $C_{2n}$  adicionando arestas entre todos os vértices com distância de no máximo 2 em  $C_{2n}$ . Couto e Cunha (COUTO; CUNHA, Luís Felipe I, 2020) determinaram o índice de extensão para qualquer grafo *potência de ciclo*  $p$ . Como consequência de (COUTO; CUNHA, Luís Felipe I, 2020), temos que  $\sigma(C_{2n}^2) = n$ , portanto,  $\sigma''(C_n) = n$ . A Figura 3.1 apresenta um  $\text{Tot}(C_6)$  e sua árvore de solução 6-geradora.  $\sigma(C_{12}^2) = n$

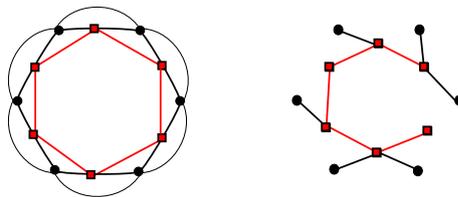


Figura 3.1:  $\text{Tot}(C_6)$  e sua árvore 6-geradora.

Árvores são os únicos grafos que possuem índice de extensão igual a 1, e os únicos grafos aresta 1-admissíveis são aqueles tais que seus grafos linha são árvores. Como os grafos linha não têm garras, os caminhos são os únicos grafos aresta 1-admissíveis.

Na [Proposição 11](#) determinamos o índice de extensão de aresta das árvores.

**Proposição 11.** *Seja  $G$  uma árvore. Se  $G$  for um caminho então  $\sigma'(G) = 1$ , caso contrário  $\sigma'(G) = 2$ .*

*Demonstração.* Observe que se  $G$  é um caminho, então  $L(G)$  é um caminho e  $\sigma'(G) = 1$ . Para qualquer outra árvore existe um vértice de grau pelo menos 3, implicando um subgrafo completo de tamanho pelo menos 3 em  $L(G)$ . Cada nó interno  $u$  de  $G$  corresponde a um subgrafo completo máximo de  $L(G)$  de tamanho  $d_G(u)$  e dois desses subgrafos completos máximos compartilham no máximo um vértice em  $L(G)$ . Portanto, qualquer componente triconexa de  $L(G)$  é um subgrafo completo e satisfaz o [Teorema 1](#).  $\square$

Um grafo total  $\text{Tot}(G)$  é isomorfo a uma árvore se e somente se  $G$  for um grafo trivial, porque qualquer aresta  $uv$  de  $G$  forma um  $C_3$  em  $\text{Tot}(G)$  pelos vértices  $u, v, uv$ , onde  $u, v \in V(G)$  e  $uv \in L(G)$ . Portanto, temos que  $K_1$  é o único grafo total 1-admissível. Dada uma árvore arbitrária, temos a [Proposição 12](#).

**Proposição 12.** *Seja  $G$  uma árvore distinta de um grafo trivial. Portanto,  $\sigma''(G) = 2$ .*

*Demonstração.* Dado o grafo total de uma árvore  $G$ ,  $\text{Tot}(G)$ , construímos uma árvore 2-geradora  $T$  de  $\text{Tot}(G)$  adicionando todos os vértices de  $\text{Tot}(G)$  e todas as arestas entre os vértices de  $G$  e  $L(G)$ .

Agora, provamos que a distância em  $T$  entre qualquer par de vértices adjacentes de  $\text{Tot}(G)$  e não adjacentes em  $T$  é 2. Como todas as arestas entre os vértices de  $G$  e  $L(G)$  pertencem a  $T$ , temos dois casos de pares de vértices adjacentes de  $\text{Tot}(G)$  para determinar sua distância em  $T$ : caso i)  $u$  e  $v$  pertencem a  $G$ . Como  $uv \in E(G)$ , então  $uv \in L(G)$ . Portanto,  $uuv$  e  $vuv$  são arestas de  $\text{Tot}(G)$ , e pertencem a  $T$ , por construção. Portanto,  $d_T(u, v) = 2$ , pelo caminho  $u, uv, v$ ; caso ii)  $uv$  e  $vw$  pertencem a  $L(G)$ . De forma semelhante ao caso i),  $uvv$  e  $vwv$  são arestas de  $\text{Tot}(G)$ , portanto  $d_T(uv, vw) = 2$  pelo caminho  $uv, v, vw$  em  $T$ .

Resta provar que  $T$  é uma árvore geradora de  $\text{Tot}(G)$ . Suponha que  $G$  seja uma árvore com  $n$  vértices, portanto, existem  $n - 1$  vértices em  $L(G)$  e, portanto, existem  $2n - 1$  vértices em  $\text{Tot}(G)$ . Como qualquer vértice  $uv$  de  $L(G)$  é adjacente exatamente a  $u$  e  $v$  em  $G$ , então existem  $2(n - 1)$  arestas entre  $L(G)$  e  $G$  em  $\text{Tot}(G)$ . Portanto, existem  $2n - 2$  arestas em  $T$ , que possuem  $2n - 1$  vértices. Observe, também, que não existe um ciclo em  $T$ , pois qualquer vértice de  $L(G)$  tem grau 2 em  $T$ , seus vizinhos pertencem a  $G$  mas não são adjacentes em  $T$ , por construção.  $\square$

Descrevemos agora uma classe de grafos para a qual  $\sigma(G)$  está tão longe quanto queremos de  $\sigma'(G)$  e  $\sigma''(G)$ . Isso mostra que ambos os parâmetros  $\sigma'(G)$  e  $\sigma''(G)$  não podem vincular  $\sigma(G)$  para um fator constante. Um grafo  $G$  pertence à classe  $C_n$ -grafo multi-ciclo se  $G$  é obtido da união de um ciclo  $C_n$  com vértices  $\{v_1, \dots, v_n\}$  e  $n$  caminhos disjuntos  $P_{n-2}$  com vértices  $\{p_{1,1}, \dots, p_{1,n-2}\}$ ,  $\{p_{2,1}, \dots, p_{2,n-2}\}$ ,  $\dots$ ,  $\{p_{n,1}, \dots, p_{n,n-2}\}$ , e adicionando as arestas  $\{p_{1,1}, v_1\}$ ,  $\{p_{1,n}, v_2\}$ ,  $\{p_{2,1}, v_2\}$ ,  $\{p_{2,n}, v_3\}$ ,  $\dots$ ,  $\{p_{n,1}, v_n\}$ ,  $\{p_{n,n}, v_1\}$ . Portanto, podemos construir um grafo  $C_n$ -multi-ciclo  $G$  identificando vértices de  $n$  ciclos de tamanho  $n$  onde sua parte do meio se torna um ciclo de tamanho  $n$ . Figura 3.2(a) representa um grafo multi-ciclo  $G$  obtido de um  $C_6$ .

Para obter uma árvore geradora  $T$  de  $G$  (descrito na Figura 3.2(b)) devemos remover uma aresta por cada ciclo de  $G$  de tal forma que o grafo resultante seja conexo. Uma árvore geradora ideal de grafos  $C_n$ -multi-ciclo é obtida removendo uma aresta entre os vértices de grau dois para cada um dos ciclos externos e uma das arestas do ciclo interno. Tal remoção implica dois vértices adjacentes em  $G$  a uma distância  $2n - 3$  em  $T$ , ou seja,  $\sigma(G) = 2n - 3$ .

Podemos obter uma árvore aresta geradora de  $G$  com  $\sigma'(G) = n + 1$  conectando as arestas do ciclo interno e usando essas arestas para conectar as arestas dos ciclos externos incidentes a elas e as outras arestas dos ciclos externos podem ser conectadas de qualquer maneira (descrito na Figura 3.2(c)). A distância máxima entre dois vértices de aresta que compartilham um vértice em  $G$  é  $n + 1$ .

Finalmente, podemos obter uma árvore total geradora de  $G$  com  $\sigma''(G) = n + 2$  incluindo à árvore aresta geradora de  $G$  com  $\sigma'(G) = n + 1$  os vértices de  $G$  de modo que cada vértice seja adjacente a um de seus extremos em  $G$ , conforme descrito na Figura 3.2(d). Aumentando o valor dos ciclos da classe de grafos multi-ciclo, obtemos exemplos de grafos com  $\sigma(G)$  muito maiores que  $\sigma'(G)$  e  $\sigma''(G)$ .

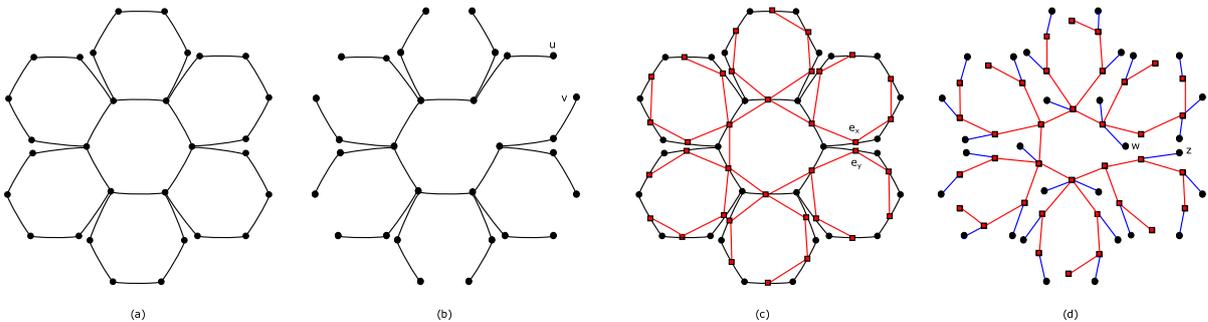


Figura 3.2: Classe de grafo com  $\sigma'$  e  $\sigma''$  menor do que  $\sigma$ . (a) Grafo usando  $C_6$ 's. (b) árvore geradora com  $\sigma = 9$ . (c) árvore aresta geradora com  $\sigma' = 7$ . (d) árvore total geradora com  $\sigma'' = 8$ .

## 3.2 Total admissibilidade

Na [Seção 3.1](#) apresentamos os grafos  $C_n$ -multi-ciclo, para os quais  $\sigma''(G)$  é muito menor que  $\sigma(G)$ . No [Corolário 24](#) verificamos que grafos completos com quatro ou mais vértices possuem  $\sigma''(G) = 4$ . Portanto, grafos ciclos, grafos completos e árvores distintas de grafos triviais são exemplos de classes de grafos para os quais  $\sigma''(G)$  é maior que  $\sigma(G)$ . No [Teorema 13](#), apresentamos um limite superior de  $\sigma''$  em função de  $\sigma(G)$ .

### 3.2.1 Limite superior no índice de extensão total

No teorema a seguir usamos uma árvore  $t$ -geradora de um grafo  $G$  com  $t = \sigma(G)$  para construir uma árvore total  $t''$ -geradora de  $G$  com  $t'' \leq \sigma(G) + 2$ . Portanto, obtemos um limite superior para o valor de  $\sigma''(G)$  dado pelo parâmetro  $\sigma(G)$ .

**Teorema 13.** *Dado um grafo  $G$ ,  $\sigma''(G) \leq \sigma(G) + 2$ .*

*Demonstração.* Considere uma árvore geradora enraizada  $T$  de um grafo  $G$  com índice de extensão  $\sigma(G)$ . Construa uma árvore total geradora  $T''$  de  $G$  como segue. Inclua todos os vértices e arestas de  $T$  a  $T''$ . Considere uma orientação das arestas de  $T$  partindo do nó inferior até o nó superior. Além disso, adicione uma orientação das arestas  $uv \in (E(G) \setminus E(T))$  em  $T$  como segue: se  $u$  e  $v$  estão na mesma profundidade e  $u$  está à esquerda de  $v$ , então usamos a orientação  $\vec{uv}$ ; se  $u$  estiver em uma profundidade maior que  $v$ , então usamos a orientação  $\vec{uv}$ . Agora, para cada aresta  $uv$  de  $E(G)$  com orientação  $\vec{uv}$ , adicione um vértice  $uv$  a  $T''$  e uma aresta  $\{uv, v\}$ . [Figura 3.3](#) (a) representa uma árvore geradora  $T$ , (b) a orientação das arestas de  $T$  (em preto) e a orientação das arestas de  $E(G) \setminus E(T)$  (em azul), e (c) uma árvore total geradora construída usando  $T$  e esta orientação.

Afirmamos que  $T''$  é uma árvore total  $t''$ -geradora de  $G$  com  $t'' \leq \sigma(G) + 2$ .

A distância entre dois vértices adjacentes  $u$  e  $v$  de  $G$  em  $T$  é a mesma em  $T''$ . Visto que,  $uv \in E(G)$ , a distância entre  $u$  e  $v$  em  $T$  é considerada para determinar  $\sigma(G)$ , portanto, a distância entre  $u$  e  $v$  em  $T''$  é menor ou igual a  $\sigma(G)$  (por exemplo, a distância entre os vértices  $h$  e  $i$  em  $T$  é seis ([Figura 3.3\(a\)](#)), que é a mesma distância entre  $h$  e  $i$  em  $T''$  ([Figura 3.3\(c\)](#))).

Um vértice-aresta  $uv$  é conectado a  $u$  (com orientação  $\vec{vu}$ ) ou  $v$  (com orientação  $\vec{uv}$ ) em  $T''$ , portanto, a distância entre  $uv$  e um de seus extremos  $v$  em  $T''$  é menor ou igual à distância entre  $u$  e  $v$  em  $T''$  mais um, que é a mesma distância entre  $u$  e  $v$  em  $T$  mais

um. Visto que,  $uv \in E(G)$ , a distância entre  $u$  e  $v$  em  $T$  é considerada para determinar  $\sigma(G)$ . Portanto, a distância entre um vértice-aresta e uma de suas extremidades é menor ou igual a  $\sigma(G) + 1$  (por exemplo, a distância entre o vértice-aresta  $hi$  e o vértice  $h$  em  $T''$  é a distância entre os vértices  $h$  e  $i$  em  $T$  mais um).

Consideramos agora os seguintes casos para as possíveis orientações de duas arestas  $uv$  e  $vw$  de  $G$ : (i)  $\overrightarrow{uv}$  e  $\overrightarrow{vw}$ , ou seja, ambos  $u$  e  $w$  são direcionados para  $v$ ; (ii)  $\overrightarrow{uv}$  e  $\overrightarrow{vw}$  (resp.  $\overrightarrow{wv}$  e  $\overrightarrow{vu}$ ), ou seja, temos um caminho de  $u$  para  $w$  (resp.  $w$  a  $u$ ) passando por  $v$  ou; e (iii)  $\overrightarrow{vu}$  e  $\overrightarrow{vw}$  ou seja,  $v$  é direcionado tanto para  $u$  quanto para  $w$ .

O caso (i) implica que os vértices-aresta  $uv$  e  $vw$  estão conectados a  $v$  em  $T''$ , ou seja, a distância entre eles em  $T''$  é dois. Por exemplo, na [Figura 3.3\(c\)](#) os vértices-aresta  $ef$  e  $if$  estão conectados a  $f$  e os vértices-aresta  $eg$  e  $hg$  estão conectados a  $g$  em  $T'$ .

O caso (ii) implica que, sem perda de generalidade, o vértice-aresta  $uv$  está conectado a  $v$  e o vértice-aresta  $vw$  está conectado a  $w$  em  $T''$ . Visto que,  $vw \in E(G)$ , a distância entre  $v$  e  $w$  em  $T$  é considerada para determinar  $\sigma(G)$ . Portanto, a distância entre  $uv$  e  $vw$  em  $T''$  é menor ou igual a  $\sigma(G) + 2$ . Por exemplo, na [Figura 3.3\(c\)](#) o vértice-aresta  $hi$  está conectado a  $i$  e o vértice-aresta  $ij$  está conectado a  $i$  em  $T''$  e a distância entre  $hi$  e  $ij$  em  $T''$  é menor ou igual à distância entre  $i$  e  $j$  em  $T''$  mais dois.

O caso (iii) implica que o vértice-aresta  $uv$  está conectado a  $u$  e o vértice-aresta  $vw$  está conectado a  $w$  em  $T''$ . Ao contrário dos casos anteriores, como  $uw$  pode não ser uma aresta de  $G$ , não podemos afirmar que a distância entre  $u$  e  $w$  em  $T$  seja considerada para determinar  $\sigma(G)$ . Afirmamos, portanto, que o máximo da distância entre  $u$  e  $v$  e a distância entre  $v$  e  $w$  (para a qual  $uv$  e  $vw$  são arestas de  $G$  e, portanto, são consideradas no cálculo de  $\sigma(G)$ ) é maior ou igual à distância de  $u$  e  $w$  in  $T$ .

Como orientamos as arestas de  $E(T)$  e  $E(G) \setminus E(T)$  de baixo para cima, precisamos que  $v$  esteja em uma profundidade igual ou maior que  $u$  e  $w$ . Além disso, como orientamos as arestas da esquerda para a direita, se  $u$ ,  $v$  e  $w$  estão todos na mesma profundidade, então  $v$  deve estar à esquerda de  $u$  e  $w$ .

Considere que a distância entre dois vértices  $u$  e  $w$  em  $T''$  é dada pelo caminho  $P_{u,w}$  com vértices  $(u, u_2, \dots, u_p, m_{u,w}, w_q, w_{q-1}, \dots, w_2, w)$ , onde  $m_{u,w}$  é o vértice deste caminho com menor profundidade. Por exemplo, em [Figura 3.3\(c\)](#) o caminho entre  $i$  e  $j$  é  $(i, f, c, g, j)$  onde  $c$  é o vértice com menor profundidade.

Sem perda de generalidade, considere que a distância entre  $v$  e  $u$  é maior ou igual à distância entre  $v$  e  $w$  em  $T''$ . Considere agora o caminho  $P_{v,u}$  com vértices  $(v, v_2, \dots, v_k,$

$m_{v,u}, u'_1, u'_{l-1}, \dots, u'_2, u)$  com vértice de menor profundidade  $m_{v,u}$ . Ou (a)  $m_{v,w} \in \{m_{u,w}, w_q, \dots, w_2, w\}$  ou (b)  $m_{u,w} \in \{u'_1, \dots, u'_2\}$ .

Na situação (a), um nó do caminho entre  $u$  e  $w$  é pai de um nó do caminho entre  $v$  e  $u$ . Assim, neste caso temos  $m_{v,u} = m_{u,w}$ . Na situação (b), um nó do caminho entre  $v$  e  $u$  é pai de um nó do caminho entre  $u$  e  $w$ . Este nó é exatamente  $m_{u,w}$  pois é o único vértice sem pai neste caminho. Ambos os casos são representados na Figura 3.4.

Resta provar nossa afirmação de que a distância entre  $u$  e  $w$  em  $T''$  é menor ou igual ao máximo da distância entre  $v$  e  $w$  e a distância entre  $v$  e  $u$ . Na situação (a), como a profundidade de  $v$  é maior ou igual às profundidades de  $w$  e  $u$ , o caminho  $P_{v,m_{v,w}}$  é maior ou igual que o caminho  $P_{w,m_{v,w}}$  em  $T''$ . Portanto o caminho  $P_{v,u}$  é maior ou igual ao caminho  $P_{u,w}$ . Na situação (b), como a profundidade de  $v$  é maior ou igual às profundidades de  $w$  e  $u$ , o caminho  $P_{v,m_{v,w}}$  é maior ou igual ao caminho  $P_{w,m_{u,w}}$ , portanto o caminho  $P_{v,u}$  é maior ou igual ao caminho  $P_{u,w}$ .

Como a distância entre  $w$  e  $u$  é menor ou igual à distância entre  $v$  e  $u$  em  $T$ , que é considerada para calcular  $\sigma(G)$ , a distância entre os vértices-aresta  $uv$  e  $vw$  em  $T''$  é menor ou igual a  $\sigma(G) + 2$ . Por exemplo, na Figura 3.3(c) a distância entre os vértices-aresta  $dh$  e  $hg$  é menor ou igual à distância de  $d$  a  $g$  mais dois, que é menor ou igual à distância entre  $h$  e  $g$  em  $T''$  mais dois.  $\square$

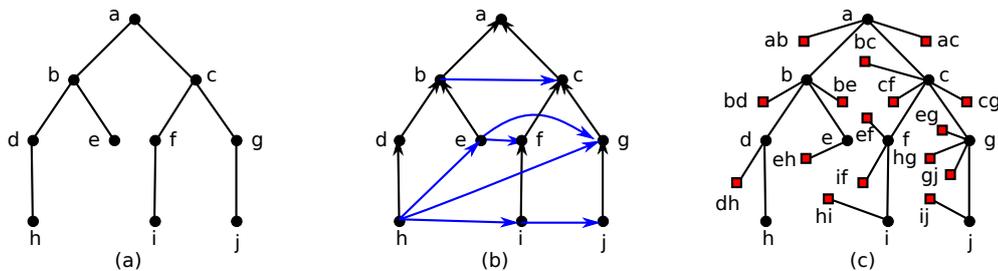


Figura 3.3: Árvore Total geradora com  $\sigma'' \leq \sigma + 2$ .

### 3.2.2 Total $k$ -admissibilidade é NP-completo

A seguir apresentamos uma transformação em tempo polinomial da determinação de uma árvore 4-geradora para qualquer grafo arbitrário  $G$  (CAI; CORNEIL, 1995) para a determinação de uma árvore total 18-geradora para um grafo transformado  $H$  de  $G$ .

**Construção 14.** *Seja  $H$  um grafo obtido do grafo  $G = (V, E)$  pela substituição de cada aresta  $uv \in E(G)$  por dois caminhos  $u, c_{uv1}, c_{vu1}, v$  e  $u, c_{uv2}, c_{uv3}, c_{uv4}, c_{uv5}, c_{uv6}, c_{uv7}, c_{mv}, c_{vu7}, c_{vu6}, c_{vu5}, c_{vu4}, c_{vu3}, c_{vu2}, v$ .*

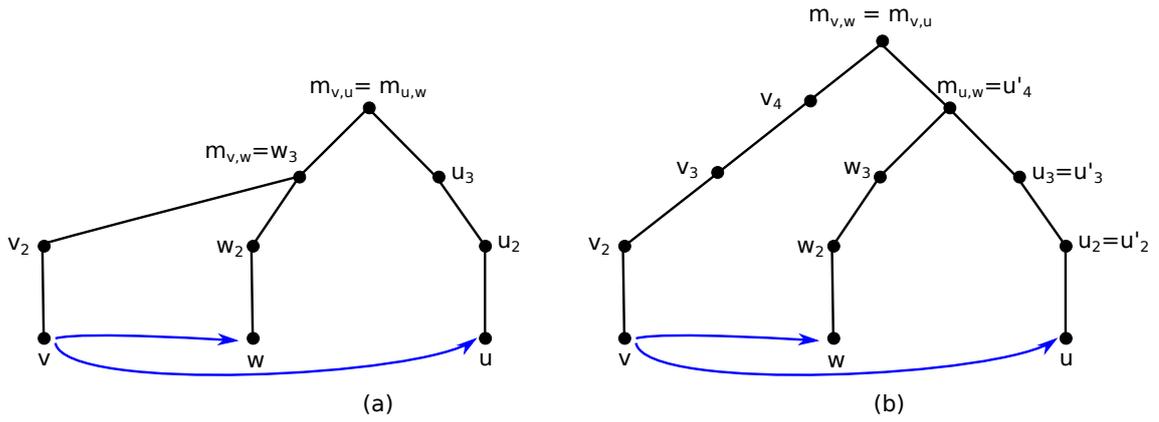


Figura 3.4: Caminhos entre  $u$  e  $v$ ,  $u$  e  $w$ , e  $v$  e  $w$  in  $T''$ .

Figura 3.5 representa uma transformação de um grafo  $G$  em um grafo  $H$  descrito na Construção 14.

A ideia central do próximo teorema é que, substituindo cada aresta de  $G$  por um ciclo de tamanho 17 em  $H$ , forçamos cada um desses ciclos a ser conectados como um caminho em uma árvore total 18-geradora de  $H$ . A seguir verificamos que as conexões restantes entre esses caminhos para construir a árvore total 18-geradora de  $H$  estão diretamente relacionadas às conexões de uma árvore 4-geradora de  $G$ .

**Teorema 15.** *Um grafo  $G$  tem uma árvore 4-geradora se e somente se o grafo  $H$  obtido de  $G$  na Construção 14 tem uma árvore total 18-geradora.*

*Demonstração.* Considere um grafo  $G$  e  $T_G$  sua árvore 4-geradora. Construímos uma árvore total 18-geradora  $T''$  de  $H$  da seguinte forma: i) Para cada aresta  $uv$  que pertence a  $T_G$ , existe um caminho forçado como subárvore em  $T''$  com um par de vértices a uma distância 17, pois grafos totais de ciclos  $C_{17}$  são grafos de potência de ciclo 2 com 34 vértices e têm índice de extensão igual a 17, conforme discutido na Seção 3.1. Como  $d_H(u,v) = 3$ , pela Construção 14, suponha que o caminho não induzido em  $\text{Tot}(H)$  entre eles é dado por  $u, uc_{uv1}, c_{uv1}, c_{uv1}c_{vu1}, c_{vu1}, c_{vu1}v, v$ . Portanto, considere em  $T''$  o caminho  $u, c_{uv1}, c_{vu1}, v$ . Para os demais vértices de  $C_{17}$  relacionados à aresta  $uv$  de  $G$ , adicione-os a  $T''$  conforme apresentado na Seção 3.1; ii) Para cada aresta  $xy$  que pertence a  $G \setminus T_G$ , o caminho entre  $x$  e  $y$  em  $T_G$  tem tamanho no máximo 4, e portanto, entre  $x$  e  $y$  em  $T''$  existe um caminho de tamanho no máximo 12, pelos caminhos de tamanho 3 por cada aresta de  $G$  em um caminho considerado no caso i).

Como a aresta  $xy$  que pertence a  $G \setminus T_G$  está relacionada a um ciclo  $X = C_{17}$  em  $H$ , então a subárvore de  $X$  em  $T''$  não deve conter um caminho entre  $x$  e  $y$  usando exclusivamente vértices de  $X$ , caso contrário existiria um ciclo em  $T''$ . Portanto, do

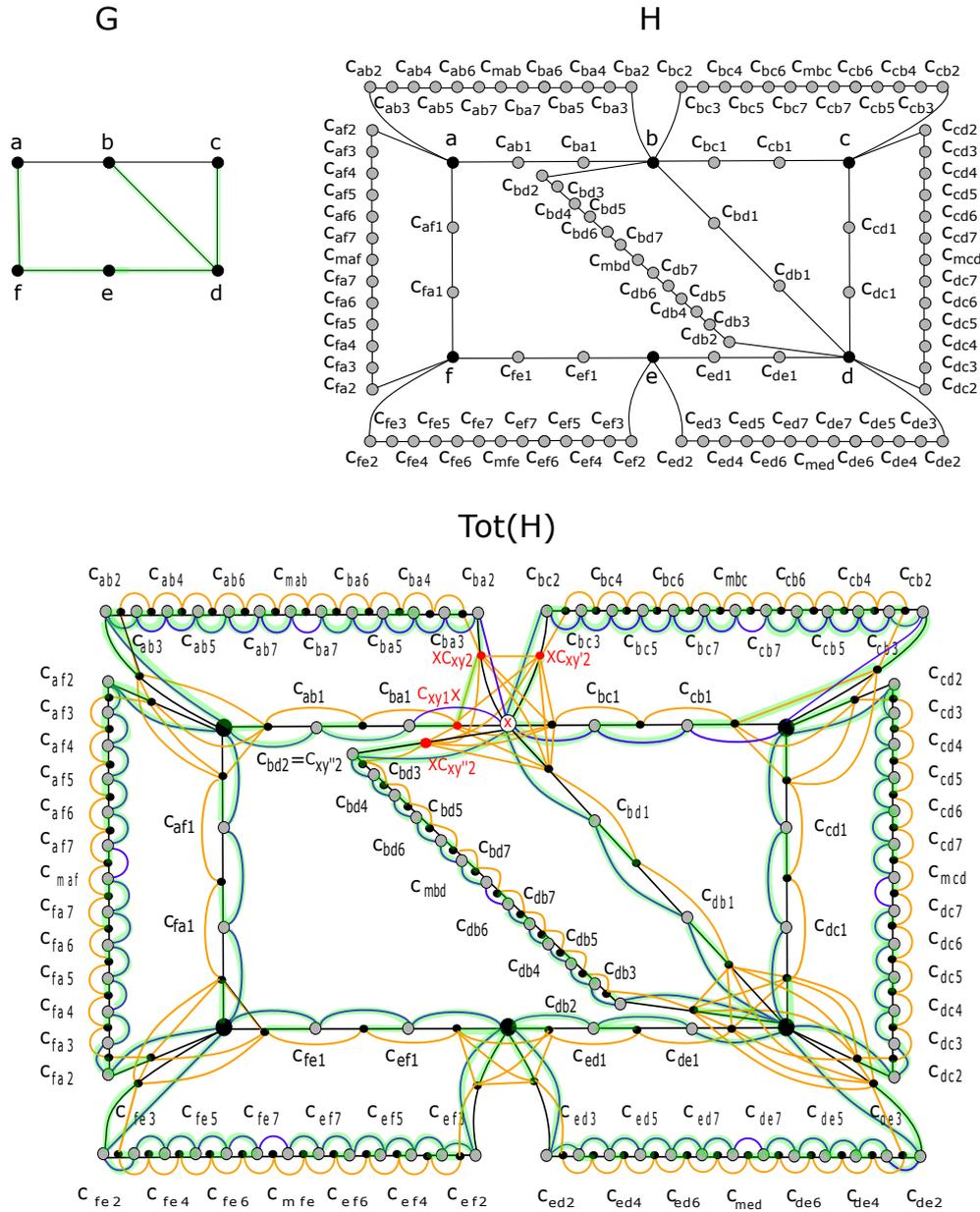


Figura 3.5: (a) Grafo  $G$  e sua árvore 4-geradora em verde. (b) Grafo  $H$  obtido da [Construção 14](#) sobre  $G$ . (c) Grafo total  $Tot(H)$  e sua árvore de solução 18-geradora (arestas marcadas em verde) com base na árvore 4-geradora de  $G$  provada no [Teorema 15](#).

caminho em  $X$  dado por  $y, yC_{yx1}, C_{yx1}, C_{yx1}C_{xy1}, C_{xy1}, C_{xy1}x, x$ , considere o caminho em  $T''$  de tamanho 3 por  $y, C_{yx1}, C_{xy1}x$ . Portanto,  $d_{T''}(C_{xy1}x, x) \leq 15$ , pelo caminho de tamanho no máximo 12 entre  $x$  e  $y$  mais o caminho anterior de tamanho 3 de  $y$  a  $C_{xy1}x$ .

O vértice  $x$  em  $G$  produz um subgrafo completo de  $Tot(H)$  de tamanho  $2d_G(x) + 1$ , uma vez que para cada aresta incidente em  $x$  associamos duas arestas incidentes a ele ao criar um  $C_{17}$  em  $H$ , mais o próprio vértice  $x$ . Tal subgrafo completo  $K_{2d_G(x)+1}$  contém um par de vértices nos quais a árvore geradora  $T''$  de  $Tot(H)$  que acabamos de apresentar acima está a uma distância máxima 18 considerando os seguintes casos:  $x$  é adjacente a um

vértice desejado, como  $xc_{xy'2}$ , e  $c_{xy1}x$  é adjacente ao outro vértice desejado em  $T''$ , como  $xc_{xy2}$ , com a distância total igual a 17, conforme provado anteriormente;  $x$  é adjacente a ambos os vértices desejados em  $T''$ , com distância total igual a 2;  $x$  não é adjacente a um vértice desejado, como  $xc_{xy''2}$ , e  $c_{xy1}x$  é adjacente ao outro vértice desejado em  $T''$ , como  $xc_{xy2}$ , com distância total igual a 18 por um caminho de tamanho 15 de  $x$  a  $c_{xy1}x$  mais as arestas  $c_{xy1}x$   $xc_{xy2}$ ,  $x$   $c_{xy''2}$  e  $c_{xy''2}$   $xc_{xy''2}$ ;  $x$  não é adjacente a um vértice desejado, como  $c_{xy1}x$ , e  $c_{xy1}x$  não é adjacente a outro vértice desejado, como  $xc_{xy''2}$ , logo a distância de  $c_{xy1}x$  a  $xc_{xy''2}$  é igual a 17 por um caminho de tamanho 15 de  $c_{xy1}x$  a  $x$  mais as arestas  $x$   $c_{xy''2}$  e  $c_{xy''2}$   $xc_{xy''2}$ .

Da mesma forma, o vértice  $c_{xy2}$  é adjacente a  $x$  em  $\text{Tot}(H)$ , e então com caminho de tamanho no máximo 15 de  $c_{xy1}x$  a  $x$ , usamos mais duas arestas adicionando a  $T''$  as arestas  $c_{xy1}x$   $xc_{xy2}$  e  $xc_{xy2}$   $c_{xy2}$ , com distância no máximo 17. Por fim, observe que de  $c_{xy1}$  a  $c_{xy2}$  em relação a uma não-aresta de  $T_G$  é necessário usar mais uma aresta comparando ao caso em que  $xy$  é uma aresta de  $T_G$ . Porque quando  $xy$  não é aresta, usamos em  $T''$  as arestas  $c_{xy1}$   $c_{xy1}x$ ,  $c_{xy1}x$   $xc_{xy2}$  e  $xc_{xy2}$   $c_{xy2}$ , enquanto quando  $xy$  é uma aresta usamos em  $T''$  as arestas  $c_{xy1}$   $x$  e  $x$   $c_{xy2}$ . Isso implica que existe um par de vértices em qualquer  $C_{17}$  associado a uma não-aresta em  $T''$  de  $\text{Tot}(H)$  a uma distância 18.

Por outro lado, considere  $H$  o grafo obtido de  $G$  na [Construção 14](#) tal que  $H$  tem uma árvore total 18-geradora  $T''$ . Para cada  $C_{17}$  em  $H$  deve haver pelo menos uma não-aresta em  $T''$  para evitar a criação de ciclos. Dado um par de vértices adjacentes em um  $C_{17}$  que forma uma não-aresta em  $T''$ , se usarmos todas as outras arestas de  $C_{17}$  para construir o caminho entre elas em  $T''$ , então a distância seria 17. Observe que não podemos ter em  $T''$  uma única não-aresta para cada  $C_{17}$ , caso contrário criaríamos um ciclo longo em  $T''$ . Portanto, existe um par de vértices adjacentes  $u$  e  $v$  em um  $C_{17}$  de  $H$  tais que o caminho entre eles em  $T''$  deve passar por vértices de outros  $C_{17}$  de  $H$ .

Assim, construímos uma árvore geradora de solução de  $G$ ,  $T_G$ , como segue: Se em  $T''$  associado a um  $X = C_{17}$  de uma aresta  $uv \in G$  usamos apenas arestas de  $X$  para um caminho entre dois vértices de  $X$  em  $T''$  então usamos a aresta  $uv$  em  $T_G$ ; Se em  $T''$  associado a um  $X = C_{17}$  de uma aresta  $uv \in G$  usamos arestas de  $X$  e algumas outras arestas de  $C_{17}$  distintas de  $X$  para um caminho entre dois vértices de  $X$  em  $T''$  então  $uv$  é uma não-aresta de  $T_G$ .

Como  $\text{Tot}(H)$  é 18-admissível, se entre  $x$  e  $y$  pertencentes a um mesmo  $X = C_{17}$  teremos que usar outros ciclos para o caminho entre  $x$  e  $y$ , em todos esses ciclos distintos de  $X$ , precisamos usar necessariamente o vértice de  $Cx'y'1$ , sendo  $x'$  e  $y'$  vértices do

caminho entre  $x$  e  $y$  em  $T_G$ . Caso contrário, teríamos que  $d_{T''}(x, y) > 18$  ao usar vértices de  $C_{x'y'i}$ ,  $i \in \{2, 3, 4, 5, 6, 7\}$ . Portanto, para ter  $d_{T''}(x, y) = 18$  só podemos ter um caminho usando arestas de ciclos distintos de  $X$  e para cada um desses ciclos usamos apenas 3 arestas. Portanto, considerando  $d_{T''}(x, y) = 18$ , analisamos cada caminho possível entre  $x$  e  $y$  em  $T''$  da seguinte forma:

- (i)  $18 = 3 \times 6$ , o que significa que passamos por 3 arestas de cada um dos 6 ciclos em  $\text{Tot}(H)$ . Portanto, 6 é o número de arestas de  $G$  relacionadas ao caminho de  $x$  a  $y$  em  $T_G$ , este caso não é possível porque tanto  $x$  quanto  $y$  não pertencem a mais de um  $C_{17}$ , e portanto deve haver outras arestas, além das 18 arestas já utilizadas, necessárias para estar em um caminho entre  $x$  e  $y$ , conforme representado na [Figura 3.6\(a\)](#).
- (ii)  $18 = 3 \times 5 + 3$ , o que significa que passamos por 3 arestas de cada um dos 5 ciclos mais 3 arestas em  $\text{Tot}(H)$ . Este caso poderia acontecer se  $x$  e  $y$  pertencessem a um mesmo  $C_{17}$ , e então  $x$  (ou  $y$ ) estivessem a uma distância 1 (ou 2) de um vértice que pertence a dois  $C_{17}$ 's. Porém, observe que existem vértices adjacentes  $x$  e  $y$  em  $\text{Tot}(H)$  que não pertencem a um mesmo  $C_{17}$ . Isso só acontece quando  $xy$  é uma aresta de  $L(H)$  Portanto, além das 18 arestas onde  $18 = 3 \times 5 + 3$ , precisamos usar mais 3 arestas para completar o caminho entre  $x$  e  $y$ , já que ambos os vértices pertencem a ciclos diferentes. Outra maneira possível, conforme descrito na [Figura 3.6\(b\)](#), é  $x$  (ou  $y$ ) estar a uma distância 3 de um vértice que pertence a dois  $C_{17}$ 's. Neste caso,  $x$  e  $y$  possuem outros vértices adjacentes onde é necessário usar mais arestas para construir o caminho de  $x$  ou  $y$  até esses vértices.
- (iii)  $18 = 3 \times 4 + 6$ , o que significa que passamos por 3 arestas de cada um dos 4 ciclos mais 6 arestas em  $\text{Tot}(H)$ . O único caso possível é que entre  $x$  e  $y$  de um mesmo  $C_{17}$  exista uma distância tal que ao considerar  $x'$  e  $y'$  de ciclos diferentes, sua distância seja no máximo 18 onde  $18 = 3 \times 4 + 6$ , ao usar 3 arestas em cada um dos quatro  $C_{17}$  do caminho e outras 6 arestas para os ciclos  $x'$  e  $y'$ .

Concluimos então que  $G$  é 4-admissível por causa dos quatro  $C_{17}$ 's que tornam  $\text{Tot}(H)$  18-admissível. □

Como decidir se um grafo arbitrário tem uma árvore 4-geradora é NP-completo ([CAI; CORNEIL, 1995](#)), como consequência imediata do [Teorema 15](#), temos o seguinte [Corolário 16](#).

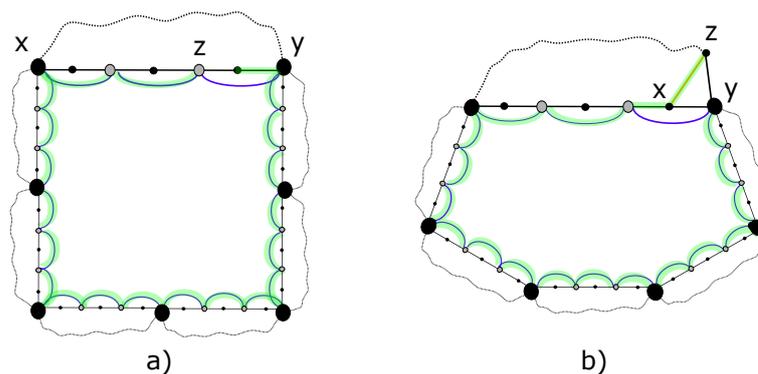


Figura 3.6: (a) Caso i)  $18 = 3 \times 6$  não é possível. (b) Caso ii)  $18 = 3 \times 5 + 3$ , um exemplo onde existe outro vértice com distância maior que 18 de  $y$ .

**Corolário 16.** *ÁRVORE TOTAL 18-ADMISSIBILIDADE é NP-completo.*

Observe que o Teorema 15 pode ser estendido para qualquer valor de  $\sigma(G) = k \geq 4$  e  $\sigma''(G) = 15 + (k - 3)3$  usando ciclos de tamanho  $14 + (k - 3)3$ . Ou seja,  $\sigma(G) = 4$  e  $\sigma''(G) = 18$ ,  $\sigma(G) = 5$  e  $\sigma''(G) = 21$ , e assim por diante.

Ademais, no Teorema 48 adaptamos a solução do grafo-meio para também ser uma solução para grafos total, e dessa forma, conseguimos reduzir a NP-completude da determinação da 18-admissibilidade para 17-admissibilidade para grafos totais.

É NP-completo decidir  $\sigma(G) = 5$  para grafos bipartidos (BRANDSTADT et al., 2007). Observe que o grafo  $H$  obtido pela Construção 14 substitui cada aresta por ciclos de tamanho 20 e multiplica por três o tamanho dos ciclos originais dos grafos bipartidos, resultando em um grafo bipartido. Portanto, segue o próximo corolário.

**Corolário 17.** *ÁRVORE TOTAL ADMISSIBILIDADE é NP-completo para grafos bipartidos.*

É NP-completo decidir o valor de  $\sigma(G)$  para grafos planares (FEKETE; KREMER, 2001). E, a transformação de Construção 14, que substitui as arestas do grafo por ciclos, aplicada a um grafo planar  $G$ , obtém um grafo planar  $H$ . Além disso, o mesmo raciocínio da prova de Teorema 15 também se aplica a ARESTA ÁRVORE-ADMISSIBILIDADE. Portanto, segue o próximo corolário.

**Corolário 18.** *ÁRVORE TOTAL ADMISSIBILIDADE é NP-completo para grafos planares.*

### 3.2.3 Grafos total 4-admissíveis

Como acabamos de provar que decidir TOTAL  $t$ -ADMISSIBILIDADE é uma tarefa difícil, a seguir, analisamos valores particulares de  $t$  e determinamos casos polinomiais tratáveis.

Apresentamos condições suficientes para que os grafos sejam total 4-admissíveis, o que nos permite apresentar algoritmos de tempo polinomial para determinar  $\sigma''(G)$  para classes de grafos como junção de grafos e grafos *split*.

**Lema 19.** *Se um grafo  $G$  tem um vértice  $v$  tal que cada aresta de  $G$  tem um vértice em  $N(v)$  como um de seus extremos, então  $\sigma''(G) \leq 4$ .*

*Demonstração.* Considere uma árvore total geradora  $T''$  de  $G$  como segue. A raiz de  $T''$  é  $v$ . O vértice raiz  $v$  de  $T''$  é adjacente aos vértices de  $N(v)$  e aos vértices-aresta de  $N_E(v)$ . Como cada aresta de  $G$  tem um extremo em  $N(v)$ , os vértices restantes e vértices-aresta de  $T''$  podem ser adjacentes aos vértices de  $N(v)$ . Portanto, o diâmetro desta árvore total geradora é no máximo 4, o que implica que  $\sigma''(G) \leq 4$ .  $\square$

**Lema 20.** *Se um grafo  $G$  tem uma aresta  $uv$  tal que cada vértice de  $G$  é adjacente a  $u$  ou a  $v$ , então  $\sigma''(G) \leq 4$ .*

*Demonstração.* Considere uma árvore total geradora  $T''$  de  $G$  como segue. A raiz de  $T''$  é o vértice-aresta  $uv$ . Todos os vértices-aresta em  $N_E(uv)$  têm  $uv$  como pai em  $T''$ . Como todos os vértices são adjacentes a  $u$  ou  $v$ , os vértices restantes e vértices-aresta podem ser adjacentes aos vértices-aresta em  $N_E(uv)$ . Portanto, esta árvore total geradora tem diâmetro 4, o que implica  $\sigma''(G) \leq 4$ .  $\square$

Figura 3.7(a) e Figura 3.7(b) representam as árvores total geradoras de grafos 4-admissíveis que satisfazem as condições suficientes dos Lema 19 e do Lema 20, respectivamente.

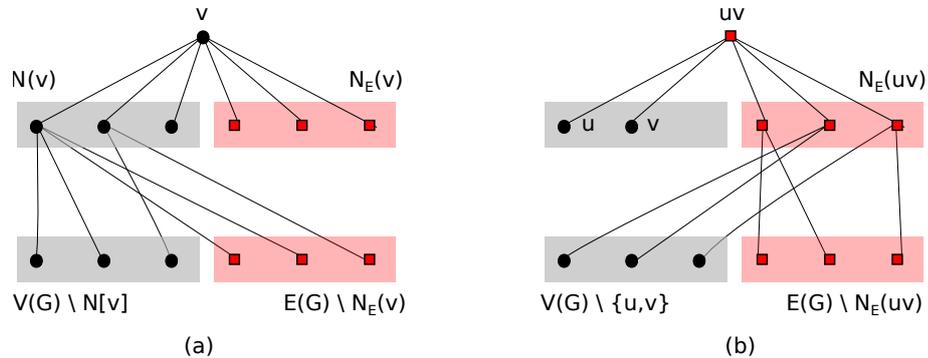


Figura 3.7: (a) Árvore Total geradora do Lema 19. (b) Árvore Total geradora do Lema 20.

Observe que um grafo completo  $K_3$  com vértices  $a, b, c$  tem  $\sigma''(K_3) = 3$  com árvore total 3-geradora  $T''$  onde o vértice raiz é o vértice-aresta  $uv$ , que é adjacente a todos os

vértices-aresta e vértices  $a$  e  $b$  em  $T''$  e o vértice restante  $c$  é adjacente ao vértice-aresta  $ac$ . A seguir, tratamos do caso  $n \geq 4$ .

**Lema 21.** *O grafo completo  $K_4$  tem  $\sigma'' = 4$ .*

*Demonstração.* Considere o grafo completo  $K_4$  com vértices  $a, b, c$  e  $d$ . Uma árvore total 4-geradora de  $K_4$  pode ser obtida conforme descrito no [Lema 20](#).

Afirmamos que não existe uma árvore total 3-geradora  $T''$  de  $K_4$ . Por uma questão de contradição, suponha que exista tal  $T''$ , ou: (i) o vértice raiz de  $T''$  é um vértice  $a$  para o qual seus filhos poderiam incluir  $N_E(a) \cup N(a)$  ou (ii) o vértice raiz de  $T''$  é um vértice-aresta  $ab$  para o qual seus filhos podem incluir  $a, b, N_E(ab)$ .

No caso (i) existem três arestas  $bc, bd$  e  $cd$  que são adjacentes entre elas no grafo total e precisam ser adjacentes a alguns vértices ou vértices-aresta de  $N(a) \cup N_E(a)$  em  $T''$ . No entanto, cada elemento em  $N(a) \cup N_E(a)$  é adjacente a apenas duas dessas três arestas. Portanto, a terceira aresta deve ser adjacente a outro elemento, uma contradição com  $T''$  ser uma árvore total 3-geradora. No caso (ii), os elementos restantes não conectados ao vértice-raiz contêm a aresta  $dc$  e os vértices  $d$  e  $c$ . Esses três elementos são adjacentes entre eles no grafo total e precisam ser adjacentes a alguns vértices ou arestas de  $N_E(ab) \cup \{a, b\}$ . No entanto, cada elemento de  $N_E(ab) \cup \{a, b\}$  em  $T''$  é adjacente apenas a dois desses três elementos. Portanto, o terceiro elemento deve ser adjacente a um filho diferente do vértice raiz, uma contradição com  $T''$  ser uma árvore total 3-geradora.  $\square$

A propriedade de grafos serem  $t$ -admissíveis não é hereditária, ou seja, um grafo  $G$  pode ser  $t$ -admissível, enquanto um subgrafo induzido de  $G$  pode possuir índice de extensão maior do que  $t$ . Apesar disso, provamos no [Lema 22](#) que se um grafo possui um subgrafo com índice de extensão total igual a 4 então o índice do grafo está limitado inferiormente a 4.

**Lema 22.** *Se  $G$  possui um subgrafo  $H$  com  $\sigma''(H) = 4$  então  $\sigma''(G) \geq 4$ .*

*Demonstração.* Seja  $H$  um subgrafo de  $G$  que satisfaz  $\sigma''(H) = 4$ . Uma vez que  $G$  possuirá formas de encurtar caminhos entre qualquer par de vértices de  $H$  havendo  $z \in V(G) \setminus V(H)$  vizinho a todos os vértices de  $V(H)$ , considere o vértice  $z \in G \setminus H$  vizinho a todos os vértices de  $H$ . Temos que analisar como pode ser formada a árvore de solução do grafo total de  $H' = G[V(H) \cup \{z\}]$ . Note que  $z$  não é vizinho a nenhum vértice de  $L(H)$  e é vizinho a todos os vértices-arestas que possuam rotulação  $zx$ , donde  $x$  é qualquer vértice de  $H$ . Dessa forma, a melhor forma de encurtar caminhos é construir uma árvore que

$z$  seja vizinho a todos os vértices de  $H$  e a todos os vértices  $zx$ , para  $x \in V(H)$ . Além disso, note que os vértices-arestas de  $L(H)$  podem ser vizinhos aos vértices-arestas  $zx$ , para  $x$  sendo um vértice de  $H$ , ou aos vértices de  $H$ . No primeiro caso, note que não existe vértice-aresta  $zx$  que possa ser vizinho a todos os vértices-arestas de  $L(H)$ , pois esse vértice pode ser vizinho somente a aqueles que possuam rótulo  $x$  em  $L(H)$ . Portanto, há dois vizinhos  $xy$  e  $vy$  em  $L(H)$  cujo caminho na árvore solução passe por  $zx$ ,  $z$  e por  $zy$ , tornando assim a distância igual a 4. No segundo caso, análogo ao anterior, não existe um vértice de  $H$  que seja vizinho a todos os vértices de  $L(H)$ . Dessa forma, existem dois vértices-arestas de  $L(H)$  que são vizinhos e sua distância em qualquer árvore solução tenha que usar aresta para  $H$ , depois para  $z$ , depois para  $H$  e depois para  $L(H)$ , tornando assim, a distância igual a 4. Dessa forma, a adição de um novo vértice não faz diminuir o parâmetro, concluindo o que queríamos demonstrar.  $\square$

**Corolário 23.** *Se  $G$  possui um subgrafo  $K_4$  então  $\sigma''(G) \geq 4$ .*

*Demonstração.* Como o subgrafo  $K_4$  possui  $\sigma''(G) = 4$  pela [Lema 21](#), logo  $G$  possui um subgrafo  $H$  com  $\sigma''(H) = 4$  e de acordo com o [Lema 22](#)  $\sigma''(G) \geq 4$ .  $\square$

**Corolário 24.** *Qualquer grafo completo  $K_n$  tem  $\sigma''(K_n) = 4$ , para  $n \geq 4$ .*

*Demonstração.* Qualquer grafo  $K_n$ , para  $n \geq 4$ , possui subgrafo  $K_4$ , logo, pelo [Corolário 23](#),  $\sigma''(K_n) \geq 4$ . Qualquer grafo  $K_n$  possui uma aresta  $uv$  tal que cada vértice de  $K_n$  é adjacente a  $u$  ou a  $v$ , então  $\sigma''(K_n) \leq 4$  de acordo com o [Lema 20](#).  $\square$

**Teorema 25.** *Existe um algoritmo de tempo polinomial para determinar  $\sigma''(G)$  para grafos split  $G = (S, K, E)$ , onde  $S$  é um conjunto independente e  $K$  é uma clique de tamanho máximo.*

*Demonstração.* Seja  $v$  um vértice da clique de  $G$ . Observe que cada aresta de  $G$  tem uma de suas extremidades em  $N(v)$ . Portanto, pelo [Lema 19](#),  $\sigma''(G) \leq 4$ . Como os grafos com  $\sigma'' = 3$  são  $K_4$ -livres, os limites  $|K|$  dos grafos *split* com  $\sigma''(G) = 3$  devem ser no máximo 3. Portanto, se  $|K| \geq 4$ , temos que  $\sigma''(G) = 4$ .

Se  $|K| = 1$ , então  $G$  é um grafo estrela. Portanto,  $G$  é uma árvore e, pela [Proposição 12](#),  $\sigma''(G) = 2$ . Se  $|K| = 2$ , há dois casos: não há vértice de  $S$  adjacente a  $u$  e  $v$ . Portanto  $G$  é uma árvore e, pela [Proposição 12](#),  $\sigma''(G) = 2$ ; existe um vértice de  $S$  adjacente a  $u$  e  $v$ . Neste caso consideramos este vértice como parte de  $K$  com  $|K| = 3$ .

Considere agora  $|K| = 3$  com  $K = \{a, b, c\}$  e  $N_s(x)$  os vizinhos do vértice  $x \in K$  que estão em  $S$ . Se  $N_s(a) \cap N_s(b) = N_s(a) \cap N_s(c) = N_s(b) \cap N_s(c) = \emptyset$ , então  $\sigma''(G) = 3$  com

uma árvore total geradora  $T''$  onde o vértice-aresta  $ab$  é sua raiz, vértices  $a, b$  e os vértices-aresta em  $N_E(a)$  e  $N_E(b)$  são adjacentes a  $ab$ , vértices em  $N_s(a)$  (resp.  $N_s(b)$ ) adjacentes a  $a$  (resp.  $b$ ), vértice  $c$  adjacente ao vértice-aresta  $ac$ , os vértices-aresta restantes em  $N_E(c)$  adjacentes a  $ac$ , e os vértices em  $N_s(c)$  adjacentes a  $c$ . Se  $N_s(a) \cap N_s(b) \neq \emptyset$ ,  $N_s(a) \cap N_s(c) = \emptyset$ , e  $N_s(b) \cap N_s(c) = \emptyset$ , então  $\sigma''(G) = 3$  com árvore total geradora  $T''$  onde o vértice-aresta  $ab$  é sua raiz, vértices  $a, b$  e vértices-aresta em  $N_E(a)$  e  $N_E(b)$  são adjacentes a  $ab$ , o vértice  $c$  é adjacente ao vértice-aresta  $ac$ , os vértices restantes em  $N_s(a) \cup N_s(b)$  são adjacentes a qualquer um  $a$  ou  $b$ , os vértices-aresta restantes em  $N_E(c)$  são adjacentes a  $ac$ , os vértices restantes em  $N_s(c)$  são adjacentes a  $c$ . Se  $N_s(a) \cap N_s(c) \neq \emptyset$  e  $N_s(b) \cap N_s(c) \neq \emptyset$ , já que estamos considerando grafos  $K_4$ -livres, então há um vértice  $e \in N_s(a) \cap N_s(b)$  e um vértice  $d \in N_s(b) \cap N_s(c)$  tal que  $\{a, b, c, d, e\}$  induz um grafo de gema. Tal grafo de gema induzido implica  $\sigma''(G) = 4$  (Lema 22).  $\square$

Vários grafos podem ser construídos por uma série de operações de junção, como cografos. Dados dois grafos  $G_1$  e  $G_2$ , denotamos  $G_1 \wedge G_2$  o grafo da junção entre  $G_1$  e  $G_2$ .

**Teorema 26.** *Existe um algoritmo de tempo polinomial para determinar  $\sigma''(G)$  para grafos junção  $G = G_1 \wedge G_2$ .*

*Demonstração.* Seja  $uv$  uma aresta entre um vértice  $u$  de  $G_1$  e um vértice  $v$  de  $G_2$ . Observe que todos os vértices de  $G$  são adjacentes a  $u$  ou  $v$ . Portanto, pelo Lema 20,  $\sigma''(G) \leq 4$ .

A determinação de  $\sigma'' \leq 2$  é polinomial para qualquer grafo arbitrário. Como os grafos com  $\sigma'' = 3$  são  $C_4$ -livres e  $K_4$ -livres, ou:  $G_1$  não tem arestas e  $G_2$  é uma clique com tamanho no máximo dois com vértices  $u$  e  $v$  ou  $G_1$  é uma clique com tamanho no máximo dois com vértices  $u$  e  $v$  e  $G_2$  não tem arestas. Em ambos os casos temos uma total árvore geradora  $T''$  de tais grafos usando um vértice  $v$  da clique  $uv$  como raiz de  $T''$ . O vértice  $v$  de  $T''$  é adjacente a  $N(v)$  e  $N_E(v)$  e as arestas restantes de  $G$  são adjacentes a  $u$ . Caso contrário,  $G$  tem  $C_4$  ou um  $K_4$  como subgrafo induzido e  $\sigma''(G) = 4$ , pelo Lema 22.  $\square$

**Corolário 27.** *Existe um algoritmo de tempo polinomial para determinar  $\sigma''(G)$  para cografos.*

Tabela 3.1 resume uma relação entre os parâmetros  $\sigma, \sigma'$  e  $\sigma''$ .

	$\sigma(G)$	$\sigma'(G)$	$\sigma''(G)$
$K_n, n \geq 5$	2	4	4
<i>Grafo split</i>	$\leq 3$	$\leq 4$	$\leq 4$
<i>Grafo junção</i>	$\leq 3$	$\leq 4$	$\leq 4$
$C_n$	$n - 1$	$n - 1$	$n$
$P_n, n > 1$	1	1	2
Árvore, distinta de $P_n$	1	2	2
$C_n$ -multi-ciclo	$2n - 3$	$n + 1$	$n + 2$
<i>Grafo planar</i>	NP – Completo	NP – Completo	NP – Completo
<i>Grafo bipartido</i>	NP – Completo	NP – Completo	NP – Completo

Tabela 3.1: Coluna  $\sigma''(G)$  e grafos  $C_n$ -multi-ciclos são alguns dos resultados apresentados nesta dissertação.  $\sigma''(G) > \sigma(G)$  para grafos completos, ciclos, caminhos e árvores. Enquanto  $\sigma'(G) = \sigma(G)$  para grafos ciclos e caminhos e  $\sigma'(G) > \sigma(G)$  para grafos completos e árvores. Os três parâmetros são diferentes para grafos  $C_n$ -multi-ciclos e, contraintuitivamente,  $\sigma(G) > \sigma''(G) > \sigma'(G)$ . Enquanto  $\sigma(G)$  está limitado superiormente a 3 para grafos *split* e junção,  $\sigma'(G)$  e  $\sigma''(G)$  estão limitados a 4. Decidir os 3 parâmetros para grafos planares e bipartidos é NP-Completo. Uma questão que permanece em aberto é conhecer alguma classe de grafos cuja complexidade de determinar o índice de extensão seja distinta da complexidade de determinar o seu índice de aresta ou total.

# 4 Grafos Clique-Aumentantes

Este capítulo analisa em detalhes a relação entre os índices de extensão antes e depois da operação de clique-aumentante e determina limites para determinadas classes de grafos. Inicia a análise da operação para cliques de tamanho 2, cliques de tamanho 3 em grafos planares até cliques de tamanhos arbitrários em grafos linha. Mostra que grafos-meio e grafos-quase-totais são subgrafos de grafos totais e também pertencem a classe de grafos clique-aumentantes.

## 4.1 Grafos Clique-Aumentantes

Grafos-meio e grafos-quase-totais são subgrafos de grafos totais. A [Figura 1.3](#) mostra os grafos  $M(G)$  e  $QT(G)$  subgrafos do grafo  $Tot(G)$  da [Figura 1.2](#). Nossa abordagem a seguir é considerar a admissibilidade para esses grafos, uma vez que a propriedade de ser  $t$ -admissível não é hereditária (ou seja, subgrafos de um grafo  $t$ -admissível podem não ser  $t$ -admissíveis). Ao focar em grafos-meio, há casos em que a admissibilidade  $\sigma(M(G))$  é igual a  $\sigma(L(G))$ , mas também há casos em que  $\sigma(M(G))$  é igual a  $\sigma(Tot(G))$ .

Considerando  $G$  isomorfo aos grafos ciclos  $C_n$ , fica claro que  $\sigma(G) = n - 1$ ,  $\sigma'(G) = n - 1$ ,  $\sigma''(G) = n$  e  $\sigma(M(G)) = \sigma''(G) = n$ . Enquanto ao considerar  $G$  isomorfo a classe de grafo  $C_n$ -multi-ciclo, temos que  $\sigma(G) = 2n - 3$ ,  $\sigma'(G) = n + 1$ ,  $\sigma''(G) = n + 2$  e  $\sigma(M(G)) = \sigma'(G) = n + 1$ .

Portanto, como a admissibilidade para grafos-meio (e quase-total) nem sempre é igual a  $\sigma'$  nem a  $\sigma''$ , a seguir trataremos desses casos, em que ambos podem ser vistos como subclasses de grafos clique-aumentantes, definidos abaixo.

**Construção 28.** *Dado um grafo  $G$ , construa um grafo clique-aumentante  $H = CA(G)$  da seguinte forma: escolha um conjunto arbitrário de cliques de  $G$ ; Para cada clique selecionada  $X$ , adicione um novo vértice  $u_X$  ao grafo e todas as arestas entre  $u_X$  e os vértices de  $X$ , ou seja,  $N(u_X) = X$ .*

Note que, na [Construção 28](#), não é obrigatório escolher cliques maximais.

**Lema 29.** *Grafos-meio e grafos-quase-totais são grafos clique-aumentantes.*

*Demonstração.* Para grafos-meio  $M(G)$ , podemos construir um grafo clique-aumentante  $CA(L(G))$  a partir de  $L(G)$  escolhendo o conjunto de cliques que possuem um mesmo vértice de  $G$ . Dado  $V(G) = \{v_1, \dots, v_n\}$ , portanto na [Construção 28](#), a partir do grafo linha  $L(G)$  escolhemos as  $n$  cliques, cada uma com os vértices tendo rótulo  $v_i$ , para  $i = 1, \dots, n$ . Para grafos-quase-totais  $QT(G)$ , podemos construir um grafo clique-aumentante  $CA(G)$  a partir de  $G$  escolhendo o conjunto de cliques que sejam pares de vértices adjacentes de  $G$ .  $\square$

Como consequência direta da [Construção 28](#) temos o [Teorema 30](#).

**Teorema 30.** *Um grafo  $G$  é um grafo clique-aumentante se e somente se  $G$  tem um vértice simplicial.*

Como podemos verificar a existência de um vértice simplicial em tempo polinomial ([GOLUMBIC, 2004](#)), então o [Teorema 30](#) resulta em uma caracterização em tempo polinomial de grafos clique-aumentantes. Como outra consequência do [Teorema 30](#) temos que qualquer grafo cordal é um grafo clique-aumentante, e claramente o inverso não é verdadeiro. Além disso, qualquer grafo cordal pode ser obtido com uma série de operações clique-aumentantes, uma vez que qualquer grafo cordal possui uma ordenação de eliminação perfeita baseada nos vértices simpliciais em sua construção ([GOLUMBIC, 2004](#)). Portanto, todo o conhecimento que já foi obtido sobre a  $t$ -admissibilidade para grafos cordais ([BRANDSTÄDT; DRAGAN; LE et al., 2004; PANDA; DAS, 2010](#)) pode ser usado para determinar a  $t$ -admissibilidade para grafos cordais clique-aumentantes.

A seguir, investigamos os grafos obtidos na [Construção 28](#) e analisamos se o índice de extensão é alterado ou não comparando com o índice de extensão do grafo de entrada.

## 4.2 Alternando tamanhos de cliques

Na [Subseção 4.2.1](#) tratamos do caso onde as cliques escolhidas na [Construção 28](#) possuem tamanho dois; Na [Subseção 4.2.2](#) para lidar com casos onde cliques de tamanho três podem ser escolhidas, analisamos a operação clique-aumentante para grafos planares; Enquanto na [Subseção 4.2.3](#) consideramos um caso mais geral onde as cliques escolhidas na [Construção 28](#) possuem tamanho arbitrário, baseado em algumas restrições.

### 4.2.1 Grafos Clique-aumentantes escolhendo cliques de tamanho 2

Primeiro, tratamos de escolhas particulares de conjuntos de cliques, como segue.

**Lema 31.** *Seja  $H$  qualquer grafo obtido na [Construção 28](#) a partir de um grafo  $G$ , temos que  $\sigma(H) \leq \sigma(G) + 1$ .*

*Demonstração.* Suponha que para uma aresta  $uv$  de  $G$ , na [Construção 28](#) adicionamos um vértice  $w$  e as arestas  $uw$  e  $vw$ . Considere uma árvore  $t$ -geradora  $T$  de  $G$  e adicione uma das duas arestas adicionadas a  $T$  na [Construção 28](#), sem perda de generalidade considere  $uw \in E(T)$  e  $vw \notin E(T)$ . Para a aresta não utilizada em  $T$ ,  $vw$ , a distância entre  $v$  e  $w$  é no máximo  $t + 1$ , pois no pior caso  $uv$  está a uma distância  $t$  em  $T$ .  $\square$

**Lema 32.** *Seja  $H$  obtido de  $G$  na [Construção 28](#) adicionando um vértice a todas as arestas de  $G$ .  $H$  satisfaz  $\sigma(H) \geq \sigma(G) + 1$ .*

*Demonstração.* Dada qualquer árvore geradora de solução  $T$  de  $H$  analisamos abaixo as possíveis interseções entre  $T$  e  $G$ .

Caso 1) A intersecção entre a árvore de solução de  $H$  e o grafo  $G$  é uma árvore geradora de  $G$ . Como pelos vértices de  $G$  temos uma árvore geradora de  $G$ , com relação aos novos vértices de  $H$  devemos ter usado uma aresta única para a árvore de solução de  $H$ . Portanto, o caminho  $\sigma(G)$  será aumentado em uma unidade, semelhante ao [Lema 31](#).

Caso 2) A intersecção entre a árvore de solução de  $H$  e o grafo  $G$  é uma floresta geradora distinta de uma árvore de  $G$ . Como em  $H$  temos uma árvore geradora, existem duas árvores da floresta de  $G$  que devemos usar duas novas arestas  $vw$  e  $wu$  associadas a um vértice adicionado  $w$  na [Construção 28](#). Porém, esse caminho com duas arestas poderia ser encurtado usando a aresta  $uv$  de  $G$ , sem perder a estrutura das árvores da floresta de  $G$ . No final, quando alcançamos uma árvore geradora de  $H$  unindo duas árvores da floresta, temos o Caso 1) anterior.  $\square$

Como consequência de [Lema 31](#) e [Lema 32](#), temos [Corolário 33](#).

**Corolário 33.** *Seja  $H$  obtido de  $G$  da [Construção 28](#) adicionando um vértice a todas as arestas de  $G$ .  $H$  satisfaz  $\sigma(H) = \sigma(G) + 1$ .*

**Corolário 34.** *Decidir o índice de extensão para grafos clique-aumentantes é NP-completo.*

Observe que a partir dos argumentos de [Lema 29](#), grafos-quase-totais podem ser obtidos adicionando um novo vértice  $w$  para cada aresta  $uv$  e tornando-o adjacente a ambos

os extremos  $u$  e  $v$ . Do [Corolário 33](#), isso implica que  $\sigma(QT(G)) = \sigma(G) + 1$ . Como decidir 4-admissibilidade é um problema NP-completo, temos o [Corolário 35](#) abaixo.

**Corolário 35.** *Decidir a 5-admissibilidade para grafos quase-totais é NP-completo.*

Neste ponto, observe que mesmo que os grafos-meio e os grafos-quase-totais sejam grafos clique-aumentantes, não temos nenhum resultado de complexidade para os grafos-meio. Uma abordagem possível para determinar a complexidade do índice de extensão para grafos-meio é lidar com quando  $M(G) \approx QT(G')$ , ou seja, os casos em que os grafos são isomorfos aos grafos-quase-totais. A seguir, mostramos quando isso acontece.

**Teorema 36.** *Dados os grafos  $G$  e  $G'$ , temos que  $M(G) \approx QT(G')$  então  $L(G)$  é  $K_3$ -livre.*

*Demonstração.* Suponha que  $L(G)$  tem  $K_3$ . Então a construção de  $M(G)$  tem um novo vértice adjacente de todos os vértices deste  $K_3$ . Portanto, a construção de  $M(G)$  é distinta da construção de  $QT(G')$ , pois para cada  $QT(G')$  é necessário adicionar um novo vértice de grau 2 em relação a cada aresta do grafo. Na construção de  $M(G)$ , a adição de um novo vértice  $u$  adjacente aos vértices de  $K_3$  faz com que  $u$  tenha grau pelo menos 3. Então  $M(G)$  não é isomorfo a  $QT(G')$  quando  $L(G)$  possui  $K_3$ .  $\square$

Do [Teorema 36](#), temos  $CA(L(G)) \approx M(G) \approx QT(L(G))$  então  $L(G)$  é  $K_3$ -livre. Para  $L(G)$  ser  $K_3$ -livre,  $G$  é um caminho ou um ciclo  $C_n, n \geq 4$ . Portanto,  $\sigma(L(G))$  pode ser determinado em tempo polinomial quando  $L(G)$  é  $K_3$ -livre, porque  $\sigma'(C_n) = n - 1$  e  $\sigma'(P_n) = 1$ . Isso implica que  $\sigma(QT(L(G))) = n$  ou  $2$  para  $L(G)$   $K_3$ -livre. Se  $G$  é um caminho, então  $M(G) \not\approx QT(G')$ , pois  $L(G)$  também seria um caminho e a construção de  $M(G)$  adiciona vértices de grau 2 adjacentes aos extremos de cada aresta de  $L(G)$  mais 2 vértices de grau 1 no início e fim do caminho de  $L(G)$ . Portanto, para  $M(G) \approx QT(G')$ ,  $G$  precisa ser um ciclo  $C_n, n \geq 4$ .

Determinar  $\sigma(QT(G))$  é NP-completo para  $G$  arbitrário ([Corolário 35](#)). Couto, Cunha e Posner ([COUTO; CUNHA; POSNER, 2021](#)) provaram que a determinação de  $\sigma(L(G))$  é NP-completa mesmo que  $G$  seja bipartido. Porém, concluímos que  $\sigma(L(G))$  pode ser determinado em tempo polinomial quando  $L(G)$  é bipartido, uma vez que  $L(G)$  bipartido implica que é  $K_{1,3}$ -livre (porque é um grafo linha e os grafos linhas são grafos sem garras) e  $K_3$ -livre (porque é um grafo bipartido, logo não possui ciclos ímpares), portanto, esses grafos têm  $\Delta \leq 2$ . Logo, temos o seguinte [Teorema 37](#).

**Teorema 37.** *Dado um grafo  $G$ ,  $\sigma(L(G))$  pode ser determinado em tempo polinomial para  $L(G)$  um grafo bipartido.*

A tratabilidade de decidir  $\sigma(M(G))$  será abordada na [Subseção 4.2.3](#) ([Corolário 49](#)).

No [Lema 29](#), notamos que a determinação do índice de extensão para grafos clique-aumentantes é NP-completa mesmo se adicionarmos um novo vértice adjacente a ambos os extremos de cada aresta de  $G$ . Agora, estamos interessados em determinar o índice de extensão para grafos clique-aumentantes mesmo quando existem arestas sem novos vértices adicionados. Portanto, [Proposição 38](#), [Proposição 39](#), [Teorema 41](#) e [Corolário 43](#) lidam com alguns desses possíveis casos.

**Proposição 38.** *Seja  $uv$  uma aresta de  $G$ . Se  $d_T(u,v) = \sigma(G)$ , para uma árvore de solução  $T$  de  $G$ , e  $CA(G)$  é obtido de  $G$  adicionando um vértice  $w$  e as arestas  $uw$  e  $vw$  em  $E(CA(G))$ , então  $\sigma(CA(G)) \leq \sigma(G) + 1$ .*

*Demonstração.* Considere  $T$  uma árvore geradora de  $G$  tal que  $d_T(u,v) = \sigma(G)$ . Para obter uma solução para  $CA(G)$ , podemos considerar uma árvore  $T'$  de  $CA(G)$  como a mesma árvore  $T$  e adicionar uma aresta  $uw$  a ela. Como  $d_T(u,v) = \sigma(G)$ , então  $d_{T'}(w,v) \leq \sigma(G) + 1$ .  $\square$

Com base no [Proposição 38](#) não podemos dizer que  $\sigma(CA(G)) = \sigma(G) + 1$ . Por exemplo, se  $G$  é um  $C_4$  e adicionamos um novo vértice  $w$  apenas a uma aresta  $uv$  onde  $d_T(u,v) = \sigma(G) = 3$ , ainda mantemos  $\sigma(CA(G)) = 3$ .

**Proposição 39.** *Seja  $uv$  uma aresta de  $G$ . Se  $d_T(u,v) < \sigma(G)$ , para uma solução  $T$  de  $G$ , e  $CA(G)$  é obtido de  $G$  adicionando um vértice  $w$  tal que  $uw, vw \in E(G)$ , então  $\sigma(CA(G)) = \sigma(G)$ .*

*Demonstração.* Se adicionarmos um vértice  $w$  à aresta  $uv$  onde  $d_T(u,v) < \sigma(G)$ , a distância entre  $u$  e  $v$  ainda é a mesma em  $CA(G)$ , construindo  $T'$  com as mesmas arestas de  $T$  e adicionando a aresta  $uw$ . Portanto  $d_{T'}(u,w) = 1$  e  $d_{T'}(v,w) \leq \sigma(G)$ . Então  $\sigma(CA(G)) = \sigma(G)$ .  $\square$

Portanto, como consequência do [Proposição 39](#) temos [Corolário 40](#).

**Corolário 40.** *Decidir o índice de extensão de  $CA(G)$  é NP-completo mesmo sem adicionar um novo vértice adjacente a ambos os extremos de cada aresta de  $G$  para obter  $CA(G)$ .*

**Teorema 41.** *Seja  $H$  obtido de  $G$  na [Construção 28](#) adicionando um novo vértice adjacente a todos os pares de vértices adjacentes de  $G$  que pertencem a todos os caminhos de tamanho  $\sigma(G)$  de qualquer árvore de solução de  $G$  então  $\sigma(H) \geq \sigma(G) + 1$ .*

*Demonstração.* Sejam  $u$  e  $v$  dois vértices adjacentes de  $G$  com  $d_T(u,v) = \sigma(G)$  em uma árvore de solução  $T$  de  $G$ . Se construirmos um grafo  $H$  adicionando um vértice em  $G$  adjacente a ambos os extremos de cada aresta do caminho entre  $u$  e  $v$  em  $T$ , e seja  $w$  um vértice adicionado a  $G$  adjacente a ambos os extremos da aresta  $uv$ , então é possível obter uma árvore de solução  $T'$  de  $H$  baseada na árvore de solução  $T$  de  $G$  adicionando apenas uma nova aresta adjacente a cada novo vértice adicionado, incluindo a aresta  $uw$ .

Observe que adicionamos novos vértices associados a todas as arestas de todos os caminhos  $\sigma(G)$  de qualquer árvore de solução de  $G$ , isso implica que em  $H$  não podemos construir uma solução baseada em uma árvore de solução de  $G$  com  $\sigma(H) = \sigma(G)$ , pois para qualquer caminho  $\sigma(G)$  possível, temos uma aresta que aumenta o caminho. Portanto  $\sigma(H) \geq \sigma(G) + 1$ .  $\square$

**Corolário 42.** *Se  $G$  é um ciclo é necessário adicionar um vértice adjacente a todos os pares de vértices adjacentes de  $G$  para construir um grafo  $H$  que satisfaça  $\sigma(H) \geq \sigma(G) + 1$ .*

*Demonstração.* Um grafo ciclo de tamanho  $n$  tem  $n$  árvores de solução diferentes removendo uma aresta única de  $G$  por árvore, então todos os pares de vértices adjacentes de  $G$  pertencem a um caminho de tamanho  $\sigma(G)$  de qualquer árvore de solução. Portanto é necessário adicionar um novo vértice adjacente a cada par de vértices adjacentes de  $G$  e isso satisfaz o [Teorema 41](#).  $\square$

Uma questão natural é: é possível adicionar vértices que não correspondam a todas as arestas do grafo e mesmo assim o índice de extensão aumenta? A resposta é verdadeira. Observe que se tivermos um grafo  $G$  formado por um ciclo  $C_n$  e um vértice pendente (ou seja, um vértice de grau 1) adjacente a um vértice do ciclo, podemos adicionar  $n$  novos vértices adjacentes a cada uma das  $n$  arestas de  $C_n$  para construir um grafo  $H$  com  $\sigma(H) = n$ , que é maior que  $\sigma(G) = n - 1$ .

A construção acima satisfaz o [Teorema 41](#) porque a aresta única de  $G$  onde não adicionamos um novo vértice não pertence a nenhum caminho de solução de tamanho  $\sigma(G)$  de  $G$ . Com base neste último exemplo, podemos generalizar a ideia de aumentar o

índice de extensão mesmo sem adicionar vértices em relação a todas as arestas do grafo, como provado no [Corolário 43](#).

**Corolário 43.** *Seja  $H$  obtido de  $G$  na [Construção 28](#) adicionando um vértice adjacente a todos os pares de vértices adjacentes, exceto as pontes de  $G$ , então  $\sigma(H) \geq \sigma(G) + 1$ .*

*Demonstração.* Todas as pontes do grafo pertencem a todas as árvores de solução, caso contrário a solução seria uma floresta, ou seja, um grafo desconexo. Suponha que adicionemos um vértice  $w$  adjacente a uma ponte  $uv$  para construir um grafo  $H'$ . Todas as árvores de solução de  $G$  têm a aresta  $uv$ . A solução para  $H'$  tem a aresta  $uv$  e  $uw$  (ou  $wv$ ) ou mesmo as arestas  $uw$  e  $wv$ .

Em todas as árvores de solução aumentamos o índice de extensão, porque nenhum vértice do componente conexo que contém  $u$  em  $G \setminus \{uv\}$  é adjacente a qualquer vértice do componente conexo que contém  $v$  em  $G \setminus \{uv\}$ , exceto  $v$ . Portanto, como não é necessário adicionar novos vértices adjacentes aos extremos das pontes, se adicionarmos vértices em relação a todas as outras arestas, a partir do [Lema 32](#), temos que  $\sigma(H) \geq \sigma(G) + 1$ .  $\square$

No [Corolário 43](#) provamos que não é necessário adicionar vértices em relação a todas as arestas de  $G$  para aumentar o índice de extensão, mas essas arestas são pontes. Daí surge outra questão: é possível adicionar vértices que não correspondem a todas as arestas do grafo e com relação às arestas que não adicionamos novos vértices, existem arestas pertencentes a ciclos (arestas que não são pontes) e mesmo assim o índice de extensão aumenta? A resposta ainda é verdadeira. Por exemplo, considere um *grafo prisma de base triangular*  $G$  (grafo obtido pela união de dois triângulos, onde cada triângulo é uma base, e adicionamos arestas entre quaisquer dois vértices correspondentes das bases), que tem  $\sigma(G) = 3$ , basta adicionar um vértice adjacente a cada  $u$  e  $u'$ , onde  $u$  e  $u'$  são vértices correspondentes das duas bases triangulares, obtendo o grafo  $H$ . Não há pontes neste grafo e o índice de extensão aumenta para  $\sigma(H) = 4$ . Além disso, podemos aumentar o índice de extensão adicionando apenas 2 vértices adjacentes a cada  $u$  e  $u'$ , mesmo sem adicionar 3 às 3 arestas  $uu'$  existentes no prisma.

Pode-se perguntar se para todos os grafos prisma o índice de extensão aumenta ao adicionar um vértice para cada aresta  $uu'$ , onde  $u$  e  $u'$  são vértices correspondentes das duas bases obtendo o grafo resultante  $H$ , semelhante ao caso onde as bases são triângulos. Porém, observe que mesmo considerando cada base como um  $C_n$ , para  $n \geq 4$  em um grafo prisma  $G$ , esta propriedade de aumentar o índice de extensão não é válida, pois  $\sigma(G) \geq 4$

antes da adição de vértices como acabamos de mencionar, e essa adição forma distâncias de no máximo 4 em uma árvore de solução de  $H$ .

### 4.2.2 Grafos Planares Clique-aumentantes escolhendo cliques de tamanho 3

Fekete e Kremer (FEKETE; KREMER, 2001) provaram que a determinação do índice de extensão para grafos planares é NP-completa. Um resultado seminal em (SEDLÁČEK, 1964) diz que um grafo  $G$  tem um grafo linha planar  $L(G)$  se e somente se  $G$  é planar de grau máximo  $\Delta(G) = 4$  e todo vértice de grau 4 é um vértice de corte. Com base nisso, nosso interesse é investigar grafos clique-aumentantes de grafos planares que também resultem em grafos planares. Como para determinar índices de extensão podemos restringir o problema em grafos biconexos (ou seja, sem vértice de corte), uma vez que  $\sigma(G) = \max_{B_i \in \mathcal{B}(G)} \{\sigma(G[B_i])\}$ , pois  $\mathcal{B}(G)$  é o conjunto de todas as componentes biconexas maximais de  $G$ , podemos assumir que  $G$  é planar e  $\Delta(G) = 3$  para considerar o caso onde  $L(G)$  é planar.

**Proposição 44.** *Qualquer grafo linha planar  $L(G)$  de um grafo planar biconexo  $G$  satisfaz  $\Delta(L(G)) = 4$ .*

*Demonstração.* Como  $\Delta(G) = 3$ , observe que dois vértices adjacentes  $u$  e  $v$  de  $G$  com grau 3 cada se associam à aresta  $uv$  que é um vértice de  $L(G)$  adjacente aos quatro vértices restantes correspondentes às outras duas arestas incidentes a  $u$  e a  $v$  em  $G$ .  $\square$

Cada  $v \in V(G)$  com  $d_G(v) = 3$  produz um  $K_3$  em  $L(G)$ , então temos [Proposição 45](#).

**Proposição 45.** *Todo grafo linha planar  $L(G)$  de um grafo planar biconexo  $G$  tem clique de tamanho máximo  $\omega(L(G)) \leq 3$ .*

Com base na [Proposição 44](#) e na [Proposição 45](#), obtemos um grafo clique-aumentante  $CA(L(G))$  de um grafo linha planar  $L(G)$  da seguinte maneira: Escolha um número arbitrário de  $K_3$  em  $L(G)$ ; Adicione um novo vértice adjacente a cada  $K_3$  que escolhemos em  $L(G)$ . Como consequência direta desta última construção, temos que  $H = CA(L(G))$  é um grafo planar. Além disso, temos a [Proposição 46](#).

**Proposição 46.** *O grafo  $H$  obtido a partir de um grafo linha planar  $L(G)$  adicionando novos vértices adjacentes a cada  $K_3$  tem  $\Delta(H) = 8$ .*

*Demonstração.* A partir da [Proposição 44](#),  $\Delta(L(G)) = 4$  e então escolhemos os quatro  $K_3$ 's que contêm um vértice  $v$  com  $d_{L(G)}(v) = 4$  produz  $d_H(v) = 8$ .  $\square$

Como consequência da [Proposição 46](#) perguntamos sobre o índice de extensão de grafos planares com  $\Delta = 6$ . Para resolver esta questão investigamos duas perspectivas: O que se sabe sobre o índice de extensão de grafos planares? O que se sabe sobre o índice de extensão de grafos de graus limitados?

Em relação à primeira questão, Fekete e Kremer ([FEKETE; KREMER, 2001](#)) provaram que se um grafo planar  $G$  tem grau de face limitado, então  $\sigma(G)$  pode ser determinado em tempo polinomial. Como  $G$  tem grau de face limitado, então seu grafo dual tem grau limitado, e todo grafo planar é grafo dual de algum outro grafo (especificamente, todo grafo planar é dual de seu grafo dual), isso implica que  $H = CA(L(G))$  satisfaz a condição de Fekete e Kremer. Em relação à segunda questão, Papoutsakis ([PAPOUTSAKIS, 2018](#)) desenvolveu um algoritmo eficiente para determinar o índice de extensão de grafos com  $\Delta$  limitado. Assim, temos o [Teorema 47](#).

**Teorema 47.** *Dado um grafo  $H = CA(L(G))$  obtido a partir de um grafo linha planar  $L(G)$  adicionando novos vértices adjacentes a cada  $K_3$ , podemos determinar  $\sigma(H)$  em tempo polinomial.*

### 4.2.3 Grafos Linhas Clique-aumentantes escolhendo cliques de tamanhos arbitrários

A partir dos grafos obtidos na [Construção 14](#), num argumento semelhante ao dado no [Teorema 15](#), pode-se provar que  $G$  tem uma árvore 4-geradora se e somente se o grafo-meio de  $H$ ,  $M(H)$ , onde  $H$  é obtido de  $G$  na [Construção 14](#), tem uma árvore 17-geradora.

**Teorema 48.** *Um grafo  $G$  tem uma árvore 4-geradora se e somente se o grafo-meio  $M(H)$  do grafo  $H$  obtido de  $G$  na [Construção 14](#) tem uma árvore 17-geradora.*

*Demonstração.* O grafo-meio  $M(H)$ , diferente do grafo  $Tot(H)$ , não possui arestas correspondentes a  $E(H)$ , pois a construção do grafo-meio subdivide essas arestas e liga cada vértice-aresta criado dessa subdivisão aos seus vértices-aresta adjacentes no grafo linha  $L(H)$  de  $H$ . Podemos utilizar o mesmo argumento dado no [Teorema 15](#) para construir a árvore de  $M(H)$  a partir da árvore de  $G$ , porém, utilizamos agora arestas de  $L(H)$  para construir o caminho na árvore  $T_{M(H)}$ , Caso a aresta  $uv \in T_G$ , é criado um caminho de tamanho 3 em  $T_{M(H)}$  utilizando os vértices  $uc_{uv1}, c_{uv1}c_{vu1}, c_{uv1}v, vc_{vz1}$ , onde  $vz \in E(G)$  e  $vz$  é

adjacente a  $uv$  em  $G$ . Dessa forma, caso haja um caminho de tamanho 4 do vértice  $u$  para o vértice  $w$  em  $T_G$  haverá um caminho de tamanho 12 em  $T_{M(H)}$  de  $uc_{uv1}$  a  $wc_{wv'1}$ . Caso a aresta  $uv \notin T_G$ , é criado um caminho em  $T_{M(H)}$  pelos vértices  $uc_{uv1}, c_{uv1}c_{vu1}, c_{uv1}v, vc_{uv2}$  de forma que o vértice  $vc_{vz1}$  não esteja neste caminho, impedindo que o caminho da aresta  $uv$  em  $T_{M(H)}$  feche um ciclo com outros caminhos de  $T_{M(H)}$  correspondentes à arestas de  $E(T_G)$ . Para os outros vértices e arestas de  $M(H)$  também utilizamos arestas de  $L(H)$  para construir o caminho na árvore  $T_{M(H)}$ , exceto as arestas que ligam os vértices  $c_{uv7}$  aos vértices  $c_{mu7}$ . Para completar a construção de  $T_{M(H)}$ , selecionamos arestas resultantes da subdivisão das arestas de  $H$  em  $M(H)$  até que todos os vértice sejam ligados à árvore e sem formar ciclos. A [Figura 4.1\(a\)](#) apresenta uma árvore 17-geradora do grafo-meio de  $H$  da [Figura 3.5\(b\)](#). Para o caminho dos ciclos  $C_{17}$  que correspondem a arestas pertencentes à  $T_G$ , os vértices  $c_{mu7}$  e  $c_{mu7}c_{vu7}$  (ou  $c_{mu7}c_{uv7}$ ) tem distância 17, a maior neste caminho. Para o caminho dos  $C_{17}$  que correspondem a arestas pertencentes a  $E(G) \setminus T_G$ , além desses vértices citados, existem os vértices  $vc_{vu2}$  e  $vc_{vu'2}$  com distância máxima 17.

Com os mesmos argumentos da prova do [Teorema 15](#), precisamos construir  $T(G)$  a partir de  $T_{M(H)}$  e existe um par de vértices adjacentes pertencentes a um mesmo  $C_{17}$  de  $H$  tais que o caminho entre eles em  $T_{M(H)}$  deve passar por vértices de outros  $C_{17}$  de  $H$ . Tomamos como referência a distância 3 entre os vértices do tipo  $xc_{xy1}$ , marcados em vermelho na [Figura 4.1\(a\)](#), que estejam em caminhos adjacentes. Assim, analogamente ao [Teorema 15](#), é trivial mostrar que o caminho máximo 17 em  $T_{M(H)}$  é obtido por  $17 = 3 \times 4 + 5$ , onde 4 é o caminho máximo em  $T_G$ .  $\square$

Como consequência direta do [Teorema 48](#), temos o [Corolário 49](#) a seguir.

**Corolário 49.** *Decidir o índice de extensão para grafos-meio é NP-completo.*

Além disso, a construção obtida no [Teorema 48](#) pode ser adaptada para obter uma redução para o problema da total-17-admissibilidade, determinada no [Corolário 50](#).

**Corolário 50.** *Árvore total-17-Admissibilidade é NP-completo.*

*Demonstração.* Considere o mesmo grafo  $H$  da [Construção 14](#) obtido de um grafo  $G$ . Se  $G$  é 4-admissível então obtenha uma árvore de  $Tot(H)$  pelas arestas do grafo linha  $L(G)$  de  $G$  (como na [Figura 4.1\(b\)](#)). Note que cada aresta  $uv$  de  $G$  que pertence a uma árvore de  $G$  associa a um caminho de tamanho 3 pelos vértices  $uc_{uv1}$  e  $c_{vu1}v$ . Assim, analogamente ao [Teorema 15](#), a árvore de  $Tot(H)$  será 17-admissível se e somente se  $G$  for 4-admissível.  $\square$

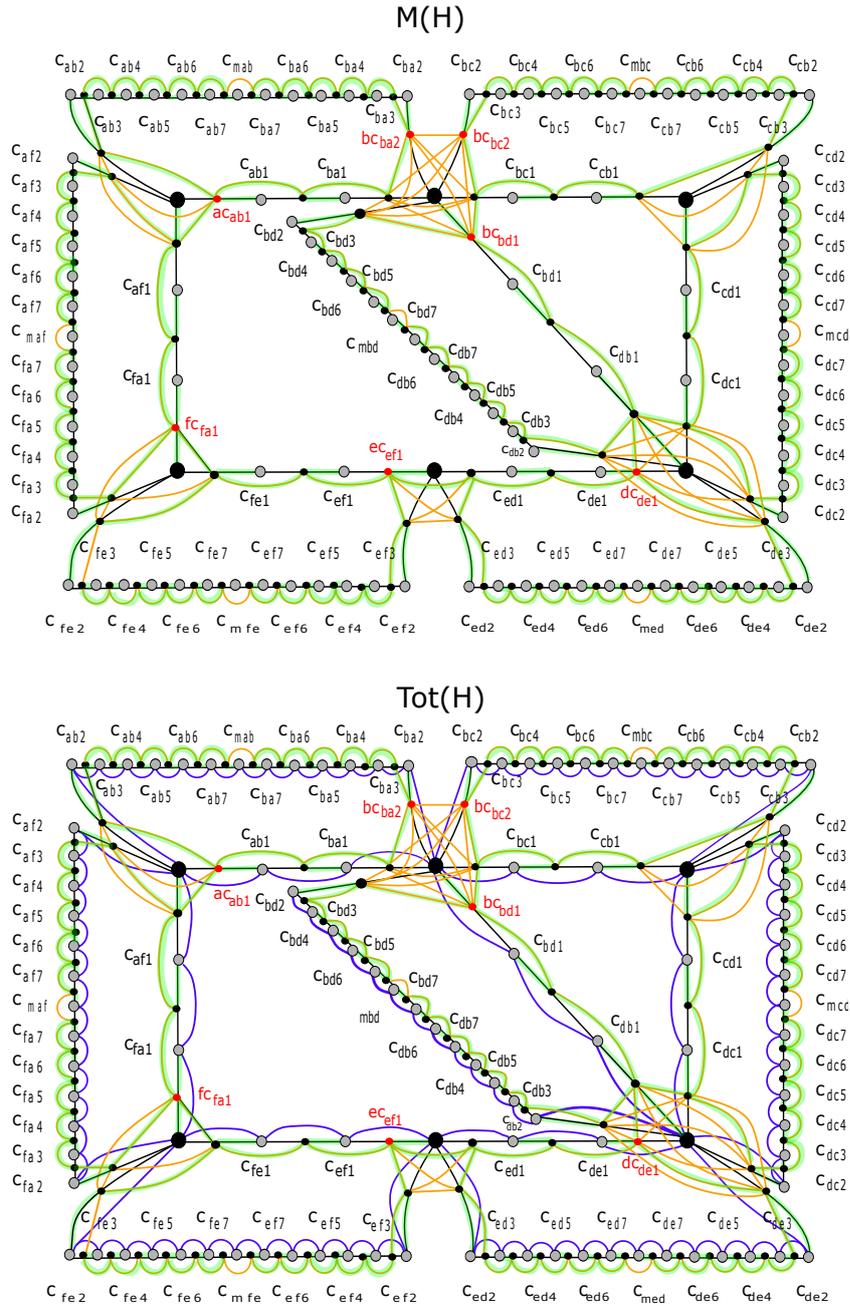


Figura 4.1: (a) Árvore 17-geradora de  $M(H)$  (b) Árvore 17-geradora de  $Tot(H)$ .

Sabe-se que um grafo  $G$  é um grafo linha se as arestas de  $G$  podem ser particionadas em subgrafos completos máximos de modo que nenhum vértice esteja em mais de dois dos subgrafos (LEHOT, 1974).

**Lema 51.** *Seja  $G'$  um grafo linha  $L(G)$  com partição de arestas em subgrafos completos maximais. Se  $H = CA(G')$  é obtido de  $G'$  adicionando um novo vértice em relação a qualquer clique maximal de  $G'$ , então  $H$  é um grafo linha.*

*Demonstração.* As novas adições de vértices consideradas para obter  $H$  mantêm  $H$  com as

mesmas partições de arestas que  $G'$  em subgrafos completos máximos e então preservam a propriedade de que nenhum vértice se encontra em mais de dois dos subgrafos completos.  $\square$

Análogo ao [Lema 31](#) e ao [Lema 32](#) temos o [Lema 52](#), quando um grafo linha  $G'$  é biconexo e distinto de um grafo completo.

**Lema 52.** *Seja  $G' = L(G)$  um grafo linha biconexo distinto de um grafo completo e seja  $\mathcal{C}$  uma partição de aresta de  $G'$  em cliques maximais. O grafo  $H$  obtido de  $G'$  adicionando cada vértice adjacente a cada clique maximal de  $\mathcal{C}$  tem  $\sigma(H) = \sigma(G') + 1$ .*

*Demonstração.* Considere uma árvore  $t$ -geradora  $T$  de  $L(G)$  e adicione uma aresta a  $T$  das duas arestas adicionadas anteriormente, sem perda de generalidade considere  $uw \in E(T)$  e  $vw \notin E(T)$ . Para a aresta não utilizada em  $T$ ,  $vw$ , a distância entre  $v$  e  $w$  é no máximo  $t + 1$ , pois no pior caso  $uv$  está a uma distância  $t$  em  $T$ .

Por outro lado, dada qualquer árvore geradora de solução  $T$  de  $H$ , analisamos abaixo as possíveis interseções entre  $T$  e  $L(G)$  (denotadas como  $L(G) = G'$ ).

Caso 1) A intersecção entre a árvore de solução de  $H$  e o grafo  $G'$  é uma árvore geradora de  $G'$ . Como pelos vértices de  $G'$  temos uma árvore geradora de  $G'$ , com relação aos novos vértices de  $H$  devemos ter usado uma aresta única para a árvore solução de  $H$ . Consequentemente, o caminho  $\sigma(G')$  será aumentado em uma unidade, semelhante ao limite superior garantido acima.

Caso 2) A intersecção entre a árvore de solução de  $H$  e o grafo  $G'$  é uma floresta geradora distinta de uma árvore de  $G'$ . Como em  $H$  temos uma árvore geradora, existem duas árvores da floresta de  $G'$  que devemos usar duas novas arestas  $vw$  e  $wu$  associadas a um vértice adicionado  $w$ . Porém, esse caminho com duas arestas poderia ser encurtado usando a aresta  $uw$  de  $G'$ , sem perder a estrutura das árvores da floresta de  $G'$ . No final, quando alcançamos uma árvore geradora de  $H$  unindo duas árvores da floresta, temos o Caso 1 anterior).  $\square$

Como Couto, Cunha e Posner provaram que determinar  $\sigma'(G) = \sigma(L(G))$  é NP-completo para  $t = 8$  ([COUTO; CUNHA; POSNER, 2021](#)), então do [Lema 51](#) e do [Lema 52](#) temos o [Teorema 53](#).

**Teorema 53.** *Decidir a 9-ADMISSIBILIDADE para grafos  $H$  do [Lema 52](#) é NP-completo.*

Couto, Cunha e Posner também mostraram que decidir a ARESTA 3-ADMISSIBILIDADE é um problema solucionável em tempo polinomial. Portanto, temos o [Teorema 54](#).

**Teorema 54.** *Decidir  $\sigma(H) = 4$  é um problema solucionável em tempo polinomial para grafos  $H$  do [Lema 52](#).*

## 5 Conclusões e Trabalhos Futuros

Este trabalho apresentou o índice de extensão total dos grafos, que é equivalente ao índice de extensão dos grafos totais. Desenvolveu estratégias para determinar índices de extensão total para diversas classes de grafos e provou que **ÁRVORE TOTAL 17-ADMISSIBILIDADE** é **NP-completo**. Esta **NP-completude** pode ser estendida a qualquer **ÁRVORE TOTAL  $(14 + (k - 3)3)$ -ADMISSIBILIDADE**, para  $k \geq 4$ , e como consequência estabeleceu que **ÁRVORE TOTAL ADMISSIBILIDADE** é **NP-completo** mesmo para grafos bipartidos ou planares.

Além disso, alguns subgrafos de grafos totais são grafos clique-aumentantes, operação proposta. Grafos-meio e grafos-quase-totais são grafos clique-aumentantes, como algumas outras classes de grafos, como grafos cordais e subclasses de grafos linha e grafos planares. Para vários desses casos, provou-se resultados e propriedades de **NP-completude** que produzem valores exatos para o índice de extensão.

Este trabalho estabeleceu também algumas relações, algumas gerais e outras particulares para classes de grafos. Por exemplo, provou-se que o índice de extensão total de um grafo  $G$  está limitado superiormente ao seu índice de extensão, ou seja  $\sigma''(G) \leq \sigma(G) + 2$ . Contudo, busca-se entender melhor a relação entre os parâmetros  $\sigma$ ,  $\sigma'$  e  $\sigma''$  como, por exemplo, se existe uma caracterização estrutural para grafos que possuem  $\sigma''(G) = \sigma(G) + 2$ .

Como pesquisa futura, existem intervalos de valores do índice de extensão que ainda podem ser analisados tanto da aresta-admissibilidade como da total-admissibilidade. Ou seja, Couto, Cunha e Posner ([COUTO; CUNHA; POSNER, 2021](#)) provaram que o problema da aresta 3-admissibilidade é solucionável em tempo polinomial e que o da aresta 8-admissibilidade é **NP-completo**. Como ficariam as complexidades desses problemas para valores entre 4 e 7? Da mesma forma, como ficariam as complexidades do problema de total-admissibilidade para valores entre 3 e 16?

Há uma intuição de que as complexidades de 3-total-admissibilidade e 3-admissibilidade para grafos clique-aumentantes sejam solucionáveis em tempo polinomial. O interessante

disso é que grafos cordais são clique-aumentantes e a classificação de 3-admissibilidade para grafos cordais é um problema em aberto. Dessa forma, como trabalho futuro, focar em grafos clique-aumentantes pode ser um bom caminho para caracterizar os grafos cordais 3-admissíveis.

Este trabalho apresentou também alguns resultados de condições suficientes que fazem o índice de extensão de grafos clique-aumentantes variar como, por exemplo, quando adicionamos um vértice adjacente a todas as arestas do grafo e outros em que o parâmetro não varia. Com isso, pretende-se buscar condições que sejam necessárias e suficientes para que essa variação do parâmetro ocorra.

A [Tabela 5.1](#) sumariza os resultados de complexidade computacional conhecidos na literatura e obtidos nesta dissertação.

$t$ -admissibilidade	$G$	$L(G)$	$Tot(G)$	$M(G)$	$QT(G)$	$CA(G)$
Polinomial	$\leq 2$	$\leq 3$	$\leq 2$	$\leq 2$	$\leq 2$	$\leq 2$
NP-Completo	$\geq 4$	$\geq 8$	$\geq 17$	$\geq 17$	$\geq 5$	$\geq 9$

Tabela 5.1: Resultados da coluna  $G$  obtidos por Cai e Corneil ([CAI; CORNEIL, 1995](#)). Resultados da coluna  $L(G)$  obtidos por Couto, Cunha e Posner ([COUTO; CUNHA; POSNER, 2021](#)). As colunas seguintes são de alguns dos resultados obtidos nesta dissertação. Na linha NP-Completo, as colunas  $Tot(G)$ ,  $M(G)$ ,  $QT(G)$  e  $CA(G)$  são dos [Corolário 50](#), [Corolário 49](#), [Corolário 35](#) e [Teorema 53](#), respectivamente.

# REFERÊNCIAS

- BHATT, Sandeep et al. Optimal simulations of tree machines. In: IEEE. FOUNDATIONS of Computer Science. [S. l.: s. n.], 1986. P. 274–282.
- BONDY, John Adrian; MURTY, Uppaluri Siva Ramachandra et al. **Graph theory with applications**. [S. l.]: Citeseer, 1976. v. 290.
- BRANDSTADT, Andreas et al. Tree spanners for bipartite graphs and probe interval graphs. **Algorithmica**, Springer, v. 47, n. 1, p. 27–51, 2007.
- BRANDSTÄDT, Andreas; DRAGAN, Feodor F; LE, Hoàng-Oanh et al. Tree spanners on chordal graphs: complexity and algorithms. **Theor. Comput. Sci.**, Elsevier, v. 310, n. 1-3, p. 329–354, 2004.
- CAI, Leizhen; CORNEIL, Derek G. Tree spanners. **SIAM J. Discrete Math.**, SIAM, v. 8, n. 3, p. 359–387, 1995.
- CHURCH, A. **An Unsolvable Problem of Elementary Number Theory**. [S. l.]: Amer.. J. Math., 1936. P. 345–363.
- COOK, A. The complexity of theorem-proving procedures. In: ACM. PROCEEDINGS of the third annual ACM symposium on Theory of computing. [S. l.: s. n.], 1971. P. 151–158.
- COUTO, Fernanda; CUNHA, Luís; POSNER, Daniel. Edge Tree Spanners. **Graphs and Combinatorial Optimization: from Theory to Applications: CTW2020**, Springer International Publishing, p. 195–207, 2021.
- COUTO, Fernanda; CUNHA, Luís Felipe I. Hardness and Efficiency on Minimizing Maximum Distances for Graphs With Few  $P_4$ 's and  $(k, \ell)$ -graphs. **Electron. Notes Theor. Comput. Sci.**, Elsevier, v. 346, p. 355–367, 2019.
- \_\_\_\_\_. Hardness and efficiency on minimizing maximum distances in spanning trees. **Theor. Comput. Sci.**, Elsevier, v. 838, p. 168–179, 2020.
- \_\_\_\_\_. Hardness and efficiency on t-admissibility for graph operations. **Discrete Appl. Math.**, North-Holland, v. 304, p. 342–348, 2021.

- COUTO, Fernanda; CUNHA, Luís Felipe I. Tree t-Spanners of a Graph: Minimizing Maximum Distances Efficiently. **COCOA, LNCS**, v. 11346, p. 46–61, jan. 2018.
- CUNHA, Luís Felipe I et al. Advancing the transposition distance and diameter through lonely permutations. **SIAM J. Discrete Math.**, SIAM, v. 27, n. 4, p. 1682–1709, 2013.
- FEKETE, Sándor P; KREMER, Jana. Tree spanners in planar graphs. **Discrete Appl. Math.**, Elsevier, v. 108, n. 1-2, p. 85–103, 2001.
- GAREY, Michael R.; JOHNSON, David S. **Computers and Intractability: A Guide to the Theory of NP-Completeness**. [S. l.]: W. H. Freeman Co, 1979.
- GOLUMBIC, Martin Charles. **Algorithmic Graph Theory and Perfect Graphs**. [S. l.]: Elsevier, 2004.
- GÓMEZ, Renzo; MIYAZAWA, Flávio K; WAKABAYASHI, Yoshiko. Tree 3-Spanners on Generalized Prisms of Graphs. In: SPRINGER. **LATIN, LNCS**. [S. l.: s. n.], 2022. P. 557–573.
- HAMADA, Takashi; YOSHIMURA, Izumi. Traversability and connectivity of the middle graph of a graph. **Discrete Mathematics**, Elsevier, v. 14, n. 3, p. 247–255, 1976.
- HASSIN, Refael; TAMIR, Arie. On the Minimum Diameter Spanning Tree Problem. **Inf. Process. Lett.**, v. 53, n. 2, p. 109–111, 1995.
- KARP, R. M. **Complexity of Computer Computations**. [S. l.]: Springer, 1972. P. 85–103.
- LEHOT, Philippe GH. An optimal algorithm to detect a line graph and output its root graph. **Journal of the ACM (JACM)**, ACM New York, NY, USA, v. 21, n. 4, p. 569–575, 1974.
- LEVIN, L. A. Universal Sequential Search Problems. **Probl. Peredachi Inf**, Springer, v. 9, p. 115–116, 1973.
- PANDA, BS; DAS, Anita. Characterization and Recognition of Tree 3-Spanner Admissible Directed Path Graphs of Diameter Three. In: SPRINGER. **INTERNATIONAL Workshop on Graph-Theoretic Concepts in Computer Science**. [S. l.: s. n.], 2018. P. 369–381.
- \_\_\_\_\_. Tree 3-spanners in 2-sep chordal graphs: Characterization and algorithms. **Discrete Appl. Math.**, Elsevier, v. 158, n. 17, p. 1913–1935, 2010.
- PAPOUTSAKIS, Ioannis. Tree spanners of bounded degree graphs. **Discrete Appl. Math.**, Elsevier, v. 236, p. 395–407, 2018.

PELEG, David; ULLMAN, Jeffrey D. An optimal synchronizer for the hypercube.

**SIAM J. Comput.**, SIAM, v. 18, n. 4, p. 740–747, 1989.

SEDLÁČEK, J. **Some properties of interchange graphs (in Theory of graphs and its applications, M. Fiedler, editor), 145-150.** [S. l.]: Praha, 1964.

TURING, A. On Computable Numbers, with an application to the Entscheidungsproblem. n. 1, p. 1–36, 1936.