

Problemas em grafos com poucos P_4 's e em grafos indiferença

Vagner Pedrotti

Este exemplar corresponde à redação final da Tese devidamente corrigida e defendida por Vagner Pedrotti e aprovada pela Banca Examinadora.

Campinas, 19 de agosto de 2011.



Célia Picinin de Mello (Orientadora)

Tese apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.

FICHA CATALOGRÁFICA ELABORADA POR
MARIA FABIANA BEZERRA MÜLLER – CRB8/6162
BIBLIOTECA DO INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E
COMPUTAÇÃO CIENTÍFICA – UNICAMP

P343p Pedrotti, Vagner, 1980-
Problemas em grafos com poucos P4's e em grafos
indiferença / Vagner Pedrotti. – Campinas, SP : [s.n.],
2011.

Orientador: Célia Picinin de Mello.
Tese (doutorado) - Universidade Estadual de
Campinas, Instituto de Computação.

1. Teoria dos grafos. 2. Otimização combinatória.
3. Algoritmos em grafos. 4. Grafos perfeitos. I. Mello,
Célia Picinin de, 1950-. II. Universidade Estadual de
Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em inglês: Problems on graphs with few P4's and indifference graphs

Palavras-chave em inglês:

Graph theory

Combinatorial optimization

Graph algorithms

Perfect graphs

Área de concentração: Ciência da Computação

Titulação: Doutor em Ciência da Computação

Banca examinadora:

Célia Picinin de Mello [Orientador]

Cláudia Linhares Sales

Sulamita Klein

Flávio Keidi Miyazawa

Orlando Lee

Data da defesa: 19-08-2011

Programa de Pós-Graduação: Ciência da Computação

TERMO DE APROVAÇÃO

Tese Defendida e Aprovada em 19 de agosto de 2011; pela Banca examinadora composta pelos Professores Doutores:



Profª. Drª. Cláudia Linhares Sales - membro
Departamento de Computação / UFC



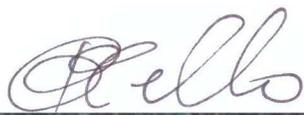
Profª. Drª. Sulamita Klein - membro
IM e COPPE / UFRJ



Prof. Dr. Flavio Keidi Miyazawa - membro
IC / UNICAMP



Prof. Dr. Orlando Lee - membro
IC / UNICAMP



Profª. Drª. Célia Picinin de Mello - orientadora
IC / UNICAMP

Problemas em grafos com poucos P_4 's e em grafos indiferença

Vagner Pedrotti

Agosto de 2011

Banca Examinadora:

- Célia Picinin de Mello (Orientadora)
- Cláudia Linhares Sales - DC - Universidade Federal do Ceará
- Sulamita Klein - IM e COPPE - Universidade Federal do Rio de Janeiro
- Flávio Keidi Miyazawa - IC - Universidade Estadual de Campinas
- Orlando Lee - IC - Universidade Estadual de Campinas
- Simone Dantas (Suplente) - IM - Universidade Federal Fluminense
- João Meidanis (Suplente) - IC - Universidade Estadual de Campinas
- Christiane Neme Campos (Suplente) - IC - Universidade Estadual de Campinas

Resumo

Nesta tese de doutoramento são considerados três problemas em grafos, para os quais são obtidos resultados quando a entrada é restrita a algumas classes. Todos os problemas são problemas de otimização combinatória sobre grafos simples e apresentam diferentes classificações de complexidade. Em dois casos, o estudo focou classes de grafos com “poucos P_4 's” e o uso da decomposição modular. No último caso, considerou-se uma subclasse dos grafos de intervalos e a aplicação de uma técnica conhecida como *pullback*.

O primeiro problema estudado é o Problema dos Separadores Minimais, para o qual são conhecidos algoritmos polinomiais em toda classe de grafos que possuir um número polinomial de separadores minimais. Serão dados, como contribuição deste trabalho, um algoritmo linear para listar os separadores minimais de grafos P_4 -carregados estendidos e limitantes justos no número e tamanho dos separadores minimais destes grafos, bem como de algumas de suas subclasses, P_4 -carregada, P_4 -arrumada e P_4 -leve. Estes resultados estendem um algoritmo anterior para grafos P_4 -esparsos, ao mesmo tempo que incluem estas classes de grafos entre as que possuem um número de separadores minimais limitado por um função linear no número de vértices do grafo.

Em seguida, será tratado o Problema de Empacotamento de Cliques, uma extensão do problema de emparelhamento máximo. Para a maioria das classes de grafos mais importantes, o problema é NP-Difícil. A contribuição apresentada resolve este problema em tempo polinomial (para qualquer tamanho fixo de clique) em grafos P_4 -arrumados, através de uma técnica similar a utilizada para os cografos. Infelizmente, para as superclasses mais estudadas da classe P_4 -arrumada, este problema é NP-Difícil, o que é um indício de que a técnica utilizada foi totalmente aproveitada em relação às classes com poucos P_4 's.

Por fim, será estudado o Problema da Coloração Total Forte, uma variação do problema clássico da coloração total, que foi introduzido há pouco tempo e ainda tem sua complexidade computacional desconhecida. Como esperado, existem algoritmos polinomiais apenas para classes bastante simples de grafos. Além da complexidade, outro importante ponto em aberto para o problema é a conjectura de que o número de cores necessárias na solução do problema para um grafo G seria limitado por $\Delta(G) + 3$. A técnica do *pullback*, já utilizada para os Problemas de Coloração de Arestas e Coloração

Total em grafos dualmente cordais será estendida, resultando em um algoritmo linear para grafos indiferença (também conhecido como grafos de intervalos próprios). Este algoritmo produz uma solução que valida a conjectura nesta classe de grafos.

Estas contribuições confirmam a importância da decomposição modular em algoritmos para classes de grafos com “poucos P_4 's” e ampliam o uso da técnica do *pullback* para variações dos problemas clássicos de coloração.

Abstract

In this doctoral thesis, three problems on graphs are considered and results are given for them when the input is restricted to some graph classes. All the problems are combinatorial optimization problems on simple graphs and have distinct classifications of complexity. In two of them, the research focused on graph classes known as graphs with “few P_4 ’s” and on the use of modular decomposition on such graphs. In the last problem, a subclass of interval graphs was studied with respect to the application of the technique known as *pullback*.

The first problem studied is the Minimal Separator Problem. For this problem, there exists polynomial time algorithms for every class of graphs which has a polynomial number of minimal separators. A linear-time algorithm, that lists all minimal separators of extended P_4 -laden graphs, is presented. Moreover, tight bounds on the number and on the total size of minimal separators are given for extended P_4 -laden graphs and for some of their subclasses: the P_4 -laden, P_4 -tidy, and P_4 -lite graphs. This result extends a previous algorithm for P_4 -sparse graphs and gives, for the above classes, better bounds on the number of minimal separators that were already known to be polynomial.

Then, the Clique Packing Problem is analyzed. The problem is an extension of the classical Maximum Matching Problem and is NP-Hard for almost all graph classes. The contribution presented solves the problem in polynomial time (for any fixed clique size) in P_4 -tidy graphs through a technique similar to that used for cographs. However, the most well-known superclasses of P_4 -tidy graphs contains split graphs, for which this problem is NP-Hard. This is an evidence that the technique was fully explored with respect of graph classes with few P_4 ’s.

At last, the Strong Total Coloring Problem is considered. It is a recently introduced variation of the classical Total Coloring Problem and its complexity is still unknown. As expected, there are quite few graph classes for which the problem has a polynomial time algorithm. Besides its complexity, another important open question for this problem is a conjecture which states that $\Delta(G) + 3$ colors are sufficient for coloring any graph G . A known technique, called *pullback*, used for edge and total coloring of dually chordal graphs is extended to derive a linear time algorithm for indifference graphs (also known

as proper interval graphs). This algorithm produces solutions that validate the conjecture for this graph class.

These contributions assert the importance of modular decomposition in algorithms for graph classes with “few P_4 ’s” and broaden the pullback technique to variations of classical coloring problems.

Agradecimentos

Algumas pessoas tiveram um papel especial na minha vida antes e durante o doutoramento, contribuindo de forma significativa para este trabalho. Espero que continuem comigo nos desafios de ainda virão.

Agradeço profundamente minha mãe, por todo suporte e apoio ao longo de tantos anos. Agradeço especialmente a Edna, por sua alegria contagiante e enorme persistência, sempre positiva, ao meu lado. Agradeço também a minha orientadora, que em muito contribuiu para o resultado deste trabalho, sempre prestativa e crítica muito construtiva.

Não poderia deixar de agradecer os muitos amigos que estiveram comigo nesse período. Neste papel, eu incluo meus primos e tios, que sempre me incentivaram e meus colegas da Unicamp, dividindo os momentos alegres e difíceis deste período.

Pelo suporte financeiro a este projeto, agradeço à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

Sumário

Resumo	v
Abstract	vii
Agradecimentos	ix
1 Introdução	1
2 Fundamentação Teórica	4
2.1 Noções elementares e Grafos	4
2.2 Módulos	7
2.3 Decomposição Modular	9
2.4 Classes de grafos	11
2.4.1 Cografos	14
2.4.2 P_4 -esparsa	15
2.4.3 P_4 -leve	17
2.4.4 P_4 -carregada	18
2.4.5 Extensões	20
3 Separadores Minimais de Grafos P_4-carregados estendidos	21
3.1 Introdução	21
3.2 Classe P_4 -esparsa	23
3.3 Aplicação da Decomposição Modular	25
3.4 Classe P_4 -carregada estendida	27
3.4.1 Grafos <i>split</i>	29
3.4.2 Outros casos	30
3.4.3 Algoritmo	30
3.5 Limitantes	31
3.5.1 Fórmula geral	31
3.5.2 Estudo de grafos	32

3.5.3	Estudo de classes	40
3.6	Limitantes justos	41
3.6.1	Construção padrão	42
3.6.2	Estudo das subclasses	44
3.7	Conclusão	45
4	Empacotamento de Cliques de Grafos P_4-arrumados	46
4.1	Introdução	46
4.2	Resultados para cografos	48
4.2.1	Grafos união	49
4.2.2	Grafos join	50
4.2.3	Solução para cografos	52
4.3	Considerações para a classe P_4 -arrumada	54
4.3.1	Aranhas	55
4.3.2	Pseudo-aranhas	60
4.3.3	Outros casos	67
4.3.4	O Algoritmo	68
4.4	Conclusão	68
5	Coloração Total Forte de Grafos Indiferença	69
5.1	Introdução	69
5.2	Preliminares	72
5.3	Grafos indiferença	72
5.4	Conclusões	75
6	Conclusões	77
	Bibliografia	79

Lista de Tabelas

3.1	Separadores minimais de P_5 , $\overline{P_5}$ e C_5	30
3.2	Parâmetros para separadores minimais de aranhas.	34
3.3	Parâmetros para separadores minimais de pseudo-aranhas.	36
3.4	Resumo da Propriedade 3.5 para algumas famílias de grafos.	40
3.5	Propriedade 3.5 para alguns grafos.	40
3.6	Limitantes dos separadores minimais das classes de interesse.	41

Lista de Figuras

2.1	Exemplos de módulos.	7
2.2	Módulos fortes e não fortes num grafo.	8
2.3	Um grafo e duas de suas partições de congruência.	9
2.4	Uma árvore de decomposição modular.	10
2.5	Árvore de decomposição modular rotulada.	11
2.6	Diagrama de Hasse das classes, adaptado de [32].	12
2.7	Grafos de cinco vértices com um P_4 induzido e um vértice adicional.	13
2.8	Um cografo e suas decomposições.	15
2.9	Aranhas gorda e magra.	16
2.10	Um grafo da classe P_4 -leve e sua decomposição modular.	17
2.11	Um grafo da classe P_4 -carregada.	20
3.1	Grafos usados para exemplificar o conceito de separadores minimais.	22
3.2	Grafos união e <code>join</code> para exemplo de separadores minimais.	24
3.3	Aranhas para exemplo de separadores minimais.	24
3.4	Relação entre separadores minimais e módulos.	26
3.5	Separadores minimais em grafos <code>split</code>	30
3.6	Pseudo-aranhas para exemplos de separadores minimais.	36
3.7	Construção padrão para grafos com limitantes justos.	42
4.1	Exemplos de soluções do PEC.	46
4.2	Empacotamentos de cliques com $n_3 = 1$ e $n_4 = 1$	49
4.3	Tentativas de empacotamento de cliques em um grafo união.	49
4.4	Tentativas de empacotamento de cliques em um grafo <code>join</code>	51
4.5	Exemplo de co-árvore binária.	53
4.6	Um grafo da classe P_4 -arrumada e uma ADMQB.	55
4.7	Empacotamento de cliques em uma aranha.	56
4.8	Pseudo-aranhas com $ S = K + 1$	61
4.9	Pseudo-aranhas com $ K = S + 1$	64
5.1	Coloração total e total forte de um grafo.	70

5.2	Coloração total forte como exemplo de pullback.	74
-----	---	----

Capítulo 1

Introdução

Esta tese reúne os resultados acumulados durante a pesquisa realizada no curso de doutorado. O principal objetivo desta pesquisa foi obter resultados teóricos com imediata aplicação em algoritmos para solução de problemas de otimização combinatória em grafos. Para fundamentar o trabalho, foram escolhidos problemas que são conhecidamente complexos (NP-Difíceis) ou cuja dificuldade é desconhecida, quando não existe restrição sobre a entrada. Todavia, em todos os casos considerados, existem algoritmos com complexidade temporal polinomial para resolver estes problemas quando o grafo de entrada é condicionado de alguma forma.

Dado o conhecimento e familiaridade prévios com a decomposição modular [60], os primeiros problemas considerados foram escolhidos dentre problemas que já haviam sido resolvidos em algumas classes de grafos por algoritmos que envolvem o uso intrínseco desta decomposição. A decomposição modular tem sido usada para resolver vários problemas, como a orientação transitiva de grafos de comparabilidade [56] e reconhecimento de classes de grafos, tais como P_4 -carregada [31] e várias de suas subclasses [32].

O interesse na decomposição modular justifica-se pela evolução nos algoritmos que a produzem, geralmente obtidas pela exploração de suas propriedades. O primeiro algoritmo para computar a árvore de decomposição modular de um grafo é de Cowan, James e Stanton [23] e sua complexidade temporal é $O(n^4)$. Subsequentemente, surgiram algoritmos mais eficientes, como os de Ehrenfeucht et al. [29] e de Muller e Spinrad [57], ambos $O(n^2)$. Finalmente, em 1994, Spinrad e McConnell [56] propuseram o primeiro algoritmo linear para a decomposição modular. O interesse pela decomposição modular foi aumentando à medida que sua obtenção tornava-se mais eficiente. Além disso, mesmo após o primeiro algoritmo linear, a pesquisa por algoritmos de implementações mais simples continuou e produziu novos algoritmos, como o de Tedder et al. [68]. Uma recente compilação de vários algoritmos foi publicada por Habib e Paul [39].

Nesta tese, são considerados dois problemas em grafos, restritos a algumas classes

específicas, que fazem parte de uma família de classes conhecidas como classes de grafos com “poucos P_4 ’s”. Nestes grafos, a decomposição modular é extensivamente utilizada para algoritmos de reconhecimento e para a solução de problemas clássicos de otimização, como coloração de vértices e clique máxima. No topo da hierarquia destas classes está a P_4 -carregada estendida, que reúne subclasses definidas ao longo do tempo, tais como: cografos, P_4 -leve, P_4 -esparsa, P_4 -redutível, P_4 -extensível [21, 44, 42, 48, 45], além de outras extensões e variações [33, 32, 11]. Muitas destas classes possuem aplicações específicas, das quais podem-se citar o número guloso [1], para a classe P_4 -carregada estendida; o emparelhamento máximo [30] e problemas de coloração [14], para a classe P_4 -arrumada; o comportamento do operador clique [28], para uma pequena restrição da classe P_4 -arrumada; e clique máxima, cobertura mínima por cliques, grafo sanduíche, partições em cliques e conjuntos independentes [47, 12, 25], para a classe P_4 -esparsa.

O primeiro problema considerado neste trabalho é o Problema dos Separadores Minimais, que pode ser resolvido por um algoritmo de tempo polinomial (em relação ao tamanho do grafo) para cada separador minimal [5]. Todavia, quando o grafo pertence à classe P_4 -esparsa, é conhecido um algoritmo linear para listar todos os separadores minimais do grafo [58]. Este algoritmo já faz uso da decomposição modular, de forma que o foco do trabalho neste problema também foi o de estender a técnica utilizada para superclasses. Como fruto do trabalho, o problema foi resolvido em tempo linear para a classe P_4 -carregada estendida, através do método proposto, que constrói os separadores minimais de um grafo a partir de sua decomposição modular. Além disso, foram obtidas expressões que fornecem limitantes justos para o número e tamanho da descrição dos separadores minimais dos grafos P_4 -carregados estendidos, P_4 -carregados, P_4 -arrumados e P_4 -leves. Com isto, o problema de contagem de separadores minimais para estas classes, proposto em [58], foi resolvido. Ainda é possível estender este trabalho, considerando outras classes ou superclasses dos grafos P_4 -carregado estendido, desde que se demonstre que os grafos satisfazem um conjunto de restrições.

O segundo problema tratado é o Problema do Empacotamento de Cliques. Este problema foi resolvido em tempo polinomial (fixando o tamanho das cliques) para cografos por Guruswami et al. [36]. Este algoritmo opera recursivamente na co-árvore do grafo, reconstruindo-o a partir de seus vértices e construindo um conjunto de soluções intermediárias em cada nó. Nosso estudo sobre o problema culminou na extensão da técnica usada para resolver o problema em tempo polinomial para grafos na classe P_4 -arrumada, substituindo a co-árvore pela árvore de decomposição modular. Este resultado, de certa forma, está no limite de possíveis extensões quando se considera a hierarquia das classes com “poucos P_4 ’s”, uma vez que a próxima candidata seria a classe P_4 -carregada ou P_4 -carregada estendida e ambas contêm os grafos **split**, para os quais o problema é NP-Difícil.

O último problema estudado envolve uma variação de um problema clássico de coloração. Esta variação está entre os problemas em que procuram-se colorações mínimas de vértices, arestas e ambos (total) de forma que elementos adjacentes ou incidentes tenham cores distintas. Um limitante bem conhecido da coloração de arestas de um grafo G , $\Delta(G) + 1$, foi dado por Vizing, que também estabeleceu uma conjectura para o limitante da coloração total, $\Delta(G) + 2$. A variação que será estudada pode ser aplicada à coloração de arestas ou à coloração total e resume-se em proibir que, em vértices adjacentes, sejam utilizados os mesmos conjuntos de cores. Estes problemas serão chamados de Problema da Coloração de Arestas Forte e Problema da Coloração Total Forte, respectivamente. Para eles, existem conjecturas que $\Delta(G) + 2$ [75] e $\Delta(G) + 3$ [74] seriam limitantes no número mínimo de cores necessárias, respectivamente.

Como resultado do terceiro problema estudado, neste trabalho concluiu-se que um algoritmo de complexidade de tempo linear já conhecido para o Problema da Coloração Total restrito a classe de grafos dualmente cordais, também soluciona o Problema da Coloração Total Forte para grafos que satisfazem restrições adicionais. Estes grafos formam uma subclasse dos grafos indiferença, também conhecidos como grafos de intervalos unitários. Além disso, a técnica utilizada produz uma coloração total forte com no máximo $\Delta(G) + 3$ cores para qualquer grafo indiferença, mostrando que a conjectura proposta por Zhang et al. é válida nesta classe de grafos. Observa-se que o Problema da Coloração Total Forte foi recentemente introduzido e ainda tem sua complexidade temporal desconhecida para grafos em geral. Até o momento, este problema é solucionado eficientemente apenas em poucas classes de grafos.

Neste último problema, a decomposição modular não é utilizada e a classe envolvida não está na categoria de classes com “poucos P_4 's”. Todavia, o estudo desta classe possibilitou o uso de ferramentas e técnicas distintas para abordar um problema relativamente novo e com grandes oportunidades de pesquisa.

A organização desta tese fornece as notações, definições e terminologias comuns no Capítulo 2, o qual também apresenta a decomposição modular e várias das classes de grafos com “poucos P_4 's”. Os Capítulos 3, 4 e 5 tratam, respectivamente, do Problema dos Separadores Minimais, do Problema do Empacotamento de Cliques e do Problema da Coloração Total Forte. Por fim, as conclusões são dadas no Capítulo 6.

Capítulo 2

Fundamentação Teórica

Neste capítulo são dadas a nomenclatura e notações para as noções fundamentais de grafos, bem como são definidas e dadas as principais características da decomposição modular de grafos e de algumas classes de grafos, objeto de estudo. Estes fundamentos são necessários para a compreensão do trabalho descrito nos capítulos subsequentes.

2.1 Noções elementares e Grafos

Neste trabalho, os conjuntos seguem a noção matemática de conjuntos. Apenas para esclarecimento, são descritas algumas operações e características importantes.

O conjunto vazio será representado por \emptyset . Um conjunto é dito *unitário* se contém apenas um elemento. As operações de união, intersecção e diferença entre conjuntos serão representadas por \cup , \cap e \setminus . A união e intersecção de vários conjuntos pode ser representada por $\bigcup_{C \in \mathcal{C}} C$ e $\bigcap_{C \in \mathcal{C}} C$, sendo \mathcal{C} um conjunto de conjuntos ou por $\bigcup_u^{i=p} C_i$ e $\bigcap_u^{i=p} C_i$, sendo p e u inteiros e C_i um conjunto, para i inteiro entre p e u . A cardinalidade do conjunto A (número de elementos do mesmo) é expressa como $|A|$.

Dado um conjunto S , define-se uma *partição* P de S como sendo um conjunto $P = \{P_1, P_2, \dots, P_n\}$, tal que $P_i \subset S$ para $1 \leq i \leq n$, $P_i \cap P_j = \emptyset$ para $1 \leq i \leq n$, $1 \leq j \leq n$ e $i \neq j$, e $\bigcup_n^{i=1} P_i = S$. Os elementos do conjunto que forma a partição P são chamados *partes*.

Um grafo simples $G = (V(G), E(G))$ (não orientado) é dado por um conjunto de vértices $V(G)$ e um conjunto de arestas $E(G)$. Cada elemento de $E(G)$ é um par não ordenado contendo dois elementos de $V(G)$. Nesta tese, o termo grafo será usado para designar um grafo simples.

Um grafo é *trivial* se contém apenas um vértice e nenhuma aresta. Os vértices que formam uma aresta são chamados *extremos* da aresta. Dois vértices que são extremos de uma aresta são ditos *adjacentes* ou *vizinhos*.

Um *subgrafo* $H = (V(H), E(H))$ de um grafo G é um grafo tal que $V(H) \subseteq V(G)$ e $E(H) \subseteq E(G)$. Além disso, se toda aresta de G cujos extremos pertencem ambos a $V(H)$ for também uma aresta de H , então H é um *subgrafo induzido* de G . O subgrafo induzido de G , cujo conjunto de vértices é X , é representado por $G[X]$.

A *vizinhança* de um vértice v é o conjunto de vértices u para os quais existe uma aresta $\{v, u\}$ no grafo e é representada por $N_G(v)$. Define-se, também $N_G[v] = N_G(v) \cup \{v\}$. O *grau* de um vértice v , por vezes denotado por $d_G(v)$ é o número de arestas que o tem como extremo. O grau máximo de um grafo é o maior grau entre todos os seus vértices e é denotado por $\Delta(G)$.

Um *caminho* em um grafo simples G de um vértice a a um vértice b é uma sequência de vértices (v_1, v_2, \dots, v_n) , sem repetição, tal que $a = v_1$, $b = v_n$ e $\{v_i, v_{i+1}\} \in E(G)$ para $1 \leq i < n$. Os vértices a e b são chamados extremos do caminho e os demais, vértices internos.

Um grafo é *conexo* se para todo par de vértices distintos, x e y , existe um caminho de x a y no grafo. Um *componente conexo* de um grafo qualquer é um subconjunto maximal de seus vértices cujo subgrafo induzido é conexo.

Um *separador* S de um grafo G é um subconjunto de seus vértices tal que $G[V(G) \setminus S]$ é desconexo. Note que, se G é desconexo, \emptyset é um separador de G .

Um *grafo caminho* de n vértices, representado por P_n , é o grafo cujos vértices são v_1, v_2, \dots, v_n e cujas arestas são $\{\{v_i, v_{i+1}\} : 1 \leq i < n\}$.

Um *ciclo* de n vértices (C_n), $n \geq 3$, é obtido do P_n adicionando a aresta $\{v_n, v_1\}$.

Um *grafo completo* de n vértices (K_n) é um grafo no qual todo par de vértices distintos são adjacentes. Um grafo com n vértices e nenhuma aresta será representado por S_n .

Uma *clique* é um subconjunto dos vértices de um grafo que induz um grafo completo. Já um subconjunto dos vértices de um grafo que induz um grafo sem arestas é chamado *conjunto independente*. Os maiores valores entre os tamanhos das cliques e dos conjuntos independentes de um grafo G são representados por $\omega(G)$ e $\alpha(G)$, respectivamente.

O *complemento* de um grafo G será representado por \overline{G} e é definido de modo que $V(\overline{G}) = V(G)$ e $\{x, y\} \in E(\overline{G}) \Leftrightarrow \{x, y\} \notin E(G)$.

Dados dois grafos G_1 e G_2 , a *união* destes grafos é representada por $G_1 \cup G_2$ e resulta no grafo $(V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$. A *união disjunta* destes dois grafos é igual a união deles, caso $V(G_1) \cap V(G_2) = \emptyset$. Caso contrário, renomeiam-se os vértices de G_2 de modo que nenhum deles esteja contido em $V(G_1)$ e, então, toma-se a união dos grafos. Por fim, define-se o *join* dos grafos G_1 e G_2 , $G_1 + G_2 = (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2) \cup \{\{x, y\} : x \in V(G_1), y \in V(G_2)\})$.

O isomorfismo entre dois grafos, G e H , é representado por $G \simeq H$.

Uma *árvore* é um grafo simples e conexo que não possui nenhum subgrafo induzido isomorfo a um ciclo. Os vértices de uma árvore também são chamados de *nós*. Opcional-

mente, um de seus vértices pode ser escolhido como sendo a *raiz* da árvore e, neste caso, dado um nó n qualquer, mas distinto da raiz, todos os nós que ficam no único caminho que liga n à raiz, excluindo o próprio n e incluindo a raiz, são os ancestrais de n (a raiz não tem ancestrais). O único ancestral de um nó (exceto a raiz) que é seu vizinho é seu *pai*. Todos os vizinhos de um nó, exceto seu pai, se houver, são seus *filhos* e todos os nós de grau um, exceto a raiz, são chamados *folhas*.

Um grafo é *bipartido* se seu conjunto de vértices pode ser particionado em duas partes, A e B , de forma que não exista nenhuma aresta do grafo com ambos os extremos em A ou em B . Um grafo é *bipartido completo* se é bipartido e para quaisquer $a \in A$ e $b \in B$, o grafo contém a aresta $\{a, b\}$.

Um grafo é *cordal* se não possui nenhum subgrafo induzido isomorfo ao C_n para $n \geq 4$. Dado um grafo G qualquer, o menor número de arestas que devem ser adicionadas ao mesmo para que ele se torne um grafo cordal é conhecido como *preenchimento mínimo* e é denotado por $\Phi(G)$.

Um grafo é *fracamente cordal* se não contiver nenhum C_n ou $\overline{C_n}$ para $n \geq 4$. Um grafo G é *dualmente cordal* se existir uma ordem linear nos seus vértices, (v_1, v_2, \dots, v_n) tal que, todo v_i possua um vizinho máximo em $G_i = G[v_1, v_2, \dots, v_i]$. Um vértice u é vizinho máximo de v_i , se $N_{G_i}[w] \subseteq N_{G_i}[u]$ para todo $w \in N_{G_i}[v_i]$. Ainda, um grafo é *fortemente cordal* se todos os seus subgrafos induzidos (incluindo o próprio grafo) são dualmente cordais.

Um grafo é *split* se o seu conjunto de vértices admite uma partição em dois conjuntos K e S que induzem um grafo completo e um grafo sem arestas, respectivamente. Uma partição deste tipo é chamada *partição split* e denotada por (K, S) .

Um grafo G é *perfeito* se, para todo subgrafo induzido H de G , $\chi(H) = \omega(H)$.

Um grafo G é *perfeitamente ordenável* se admite uma ordem o para seus vértices tal que todo subgrafo induzido H de G tem seus vértices coloridos de forma ótima por um algoritmo guloso (que atribui a menor cor possível para cada vértice) avaliando seus vértices segundo a ordem o restrita aos vértices de H .

Como uma subclasse dos grafos perfeitamente ordenáveis estão os grafos *brittle*, para os quais todo subgrafo induzido possui um vértice que não é extremo ou não é vértice interno de nenhum P_4 induzido.

Uma propriedade relativa a grafos é dita *hereditária* se, quando satisfeita por um grafo G , também é satisfeita por todo subgrafo induzido de G . Uma classe é hereditária se a pertinência à classe é uma propriedade hereditária, ou seja, se G pertence à classe, todo subgrafo induzido de G também pertence à classe. Como exemplos, tem-se as classes de grafos bipartidos, cordais e perfeitos.

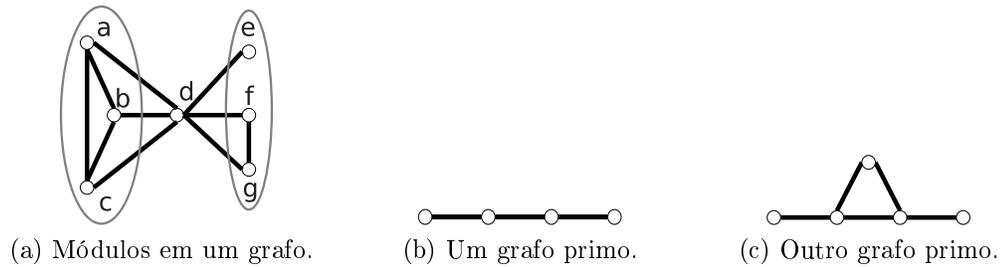


Figura 2.1: Exemplos de módulos.

2.2 Módulos

Um subconjunto não vazio, M , dos vértices de um grafo G é um *módulo* de G se, para todo vértice v de $V(G) \setminus M$, ou $(N_G(v) \cap M = \emptyset)$ ou $(M \subseteq N_G(v))$. De outra forma, um módulo é um conjunto de vértices de um grafo indistinguíveis pelos demais vértices. A Figura 2.1a contém dois exemplos de módulos de um grafo. Note que os vértices e, f e g não são vizinhos de nenhum dos vértices do módulo $\{a, b, c\}$, enquanto d contém em sua vizinhança todos os vértices deste módulo. Para maior clareza, perceba que o conjunto de vértices $A = \{d, e\}$ não forma um módulo, embora d seja vizinho de todos os vértices do grafo não contidos em A e o vértice e não seja vizinho de nenhum dos vértices do grafo não contidos em A . Esta seria uma noção de certa forma inversa ao conceito de módulo, o qual não é respeitado pelo conjunto A , dado que o vértice f é adjacente a d e não a e .

São chamados de *módulos triviais* de um grafo G , os módulos $V(G)$ e $\{v\}$, para todo $v \in V(G)$. Os demais módulos também são conhecidos por *conjuntos homogêneos*. Um grafo cujos únicos módulos são os módulos triviais é chamado de *grafo primo*, como os grafos (b) e (c) da Figura 2.1. Grafos primos são comuns, como os grafos C_n para $n \geq 5$ e P_n para $n \geq 4$, por exemplo. Note que, se M é módulo de G , M também é módulo de \overline{G} e, desta forma, se G é primo, \overline{G} também é primo. Além disso, se G é primo, G e \overline{G} são conexos.

Dados dois módulos distintos X e Y de um grafo G , o relacionamento destes módulos traz informações sobre o grafo ou sobre outros módulos. São duas as situações de interesse. Primeiro, se os módulos forem disjuntos ($X \cap Y = \emptyset$), então ou as arestas $\{\{x, y\} : x \in X, y \in Y\}$ pertencem a G (e os módulos são chamados *adjacentes*) ou G não contém nenhuma de tais arestas (e os módulos são chamados *não-adjacentes*). Se ambos os módulos interceptam-se, mas nenhum contém o outro, diz-se que os módulos são *sobrepostos* e, neste caso, os seguintes conjuntos também são módulos de G : $X \setminus Y$, $Y \setminus X$, $X \cap Y$, $X \cup Y$ e $(X \cup Y) \setminus (X \cap Y)$.

Alguns módulos relacionam-se com outros módulos de maneira diferenciada, sem que se sobreponham a nenhum outro. Uma vez que tais módulos têm papel destacado na

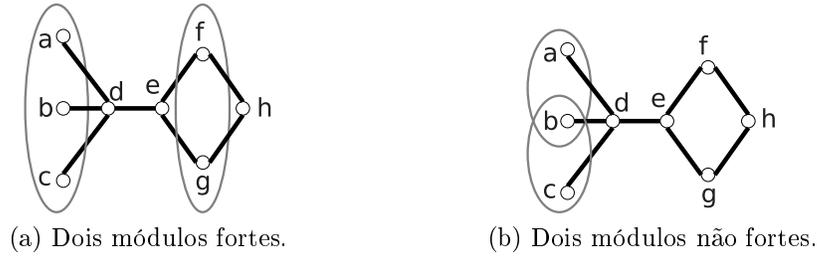


Figura 2.2: Módulos fortes e não fortes num grafo.

decomposição modular, diz-se que um módulo M de um grafo G é *forte* se, para todo módulo P de G , ou $P \cap M = \emptyset$, ou $P \subseteq M$ ou $M \subseteq P$. Observa-se na Figura 2.2 dois módulos fortes no item (a) e dois módulos sobrepostos (não fortes) para o mesmo grafo em (b). Para verificar tal afirmação, note que o grafo não possui outros módulos além dos destacados na figura e dos módulos triviais.

Define-se uma *partição de congruência* \mathcal{P} como uma partição dos vértices de um grafo G de forma que cada parte é um módulo de G . Desta forma, como todas as partes de \mathcal{P} são módulos dois a dois disjuntos, pelo exposto acima, são dois a dois adjacentes ou não-adjacentes. Representa-se a relação de adjacência entre as partes de \mathcal{P} por um grafo denominado grafo quociente, definido a seguir.

Dado um grafo G e uma partição de congruência $\mathcal{P} = \{M_1, M_2, \dots, M_k\}$ para G , o *grafo quociente* de G em relação a \mathcal{P} é o grafo $G|\mathcal{P}$ dado por $V(G|\mathcal{P}) = \{M_1, M_2, \dots, M_k\}$ e $E(G|\mathcal{P}) = \{\{M_i, M_j\} : M_i, M_j \in \mathcal{P}, M_i \text{ e } M_j \text{ adjacentes em } G\}$. Não será feita distinção na notação entre um vértice de $G|\mathcal{P}$ e a correspondente parte de \mathcal{P} , de forma que um vértice de $G|\mathcal{P}$ será tomado como módulo de G e vice-versa, quando cabível. Para facilidade de notação, o grafo quociente pode ser denotado sem a definição de \mathcal{P} , como $G|\{M_1, M_2, \dots, M_k\}$ e, neste caso, podem ser omitidos os módulos unitários. Na Figura 2.3 estão representadas duas partições de congruência do mesmo grafo, com seus respectivos grafos quociente.

Nota-se também que o grafo $G[X]$ é isomorfo ao grafo quociente $G|\mathcal{P}$ se $X = \{v_1, v_2, \dots, v_k\}$ e $v_i \in M_i$, $1 \leq i \leq k$. Além disso, denominam-se *fatores*, os subgrafos induzidos por cada módulo contido em \mathcal{P} em G .

Quando, para um grafo G , todos os módulos de uma partição de congruência \mathcal{P} são fortes, então pode-se caracterizar todos os módulos de G a partir do grafo quociente e dos grafos fatores. Um conjunto $M \subseteq V(G)$ é um módulo de G se, e somente se, M for módulo de um dos grafos fatores ou M for a união das partes de \mathcal{P} que formam um módulo do grafo quociente.

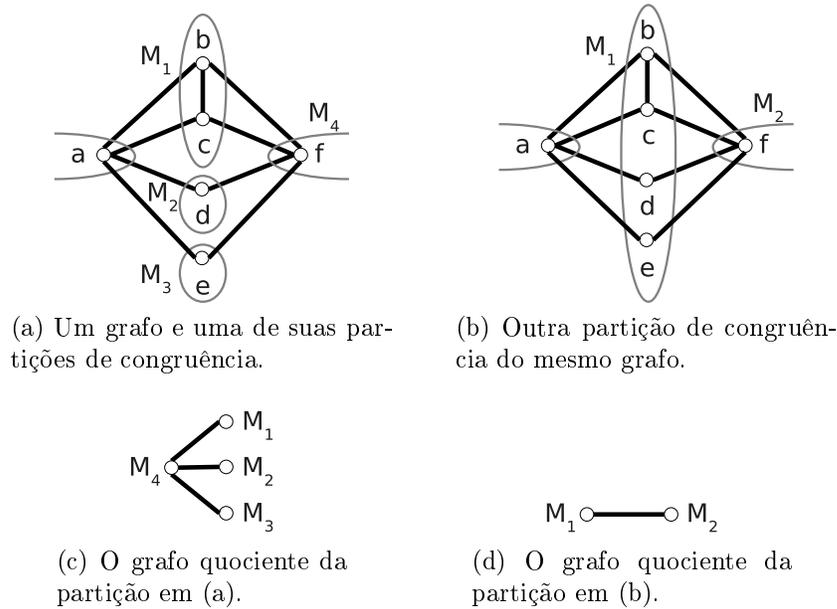


Figura 2.3: Um grafo e duas de suas partições de congruência.

2.3 Decomposição Modular

Pode-se decompor os módulos de um grafo G usando partições de congruência e então decompor novamente cada grafo fator, em sucessão, até que todos os grafos fatores sejam grafos triviais. Esta decomposição pode ser representada como uma árvore, cujos nós correspondem a módulos de G . A raiz da árvore corresponde ao módulo trivial $V(G)$ e os filhos de cada nó correspondem às partes de uma partição de congruência do subgrafo induzido pelo módulo associado àquele nó.

Como visto, se forem usadas para tal decomposição partições de congruência cujas partes sejam módulos fortes, com auxílio dos grafos quociente de cada partição, pode-se obter todos os módulos do grafo. Contudo, é interessante restringir ainda mais o tipo de partição utilizada, pois partições em módulos fortes não são únicas (na Figura 2.3 as duas partições exibidas para o grafo contêm apenas módulos fortes) e podem ser degeneradas, como partições nas quais cada parte é formada por um vértice do grafo. Além disso, seria necessário identificar módulos nos grafos quocientes para determinar os módulos do grafo.

A *decomposição modular* de um grafo é definida como a decomposição recursiva do conjunto de vértices do grafo em módulos fortes maximais próprios. Um módulo é um *módulo forte maximal próprio* se é um módulo maximal entre os módulos fortes do grafo, excluindo o próprio conjunto de vértices do grafo. A árvore que representa esta decomposição é chamada de *árvore de decomposição modular* (ADM) e está exemplificada na Figura 2.4.

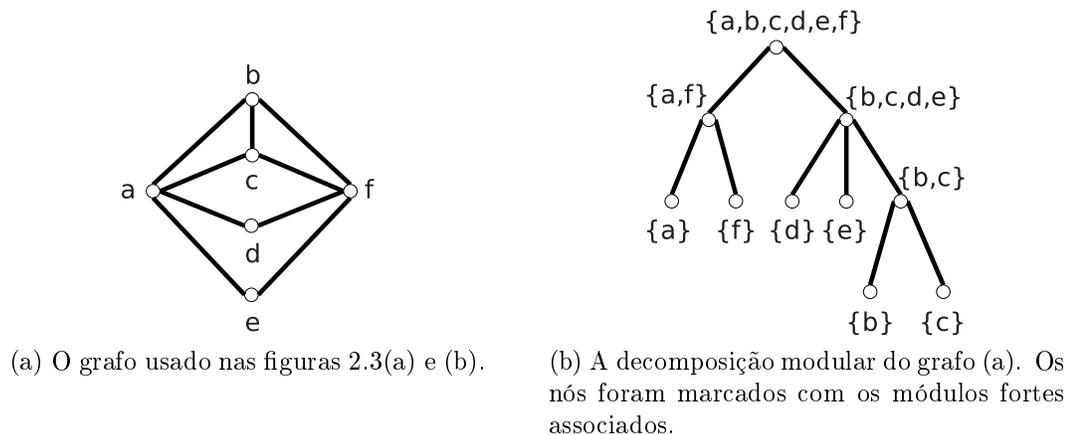


Figura 2.4: Uma árvore de decomposição modular.

Como a partição dos vértices de um grafo em módulos fortes maximais próprios é única, também é única a árvore de decomposição modular de um grafo. Além disso, a árvore de decomposição modular de um grafo contém um nó para cada módulo forte. Ou seja, outra maneira teórica de construir a árvore é encontrar todos os módulos fortes de um grafo, criar um nó para cada um deles e então definir como pai de um nó o menor módulo forte que o contém.

Além disso, associa-se a cada nó da árvore de decomposição modular, um grafo quociente determinado pela partição de congruência usada no nó e pelo subgrafo induzido pelo módulo representado. De acordo com a condição satisfeita pelo grafo quociente, rotula-se cada nó interno da árvore de decomposição modular. Se o grafo quociente for completo, o nó é chamado de *serial* e é rotulado com S ; se não contiver arestas, o nó é chamado de *paralelo* e é rotulado com P ; e se for primo, é chamado de *vizinhança* e é rotulado com N . Os nós rotulados como paralelos ou seriais também são conhecidos como *degenerados* enquanto os rotulados como vizinhança são conhecidos também por *primos*. A Figura 2.5 exibe a árvore de decomposição modular previamente apresentada na Figura 2.4, adicionando os rótulos dos nós internos.

Note ainda a relação que existe entre o subgrafo H induzido pelo módulo representado num nó da árvore de decomposição modular e o tipo do nó. Se o tipo do nó for serial, então o grafo quociente correspondente é completo com dois ou mais vértices e, portanto, seu complemento é desconexo, por conseguinte \overline{H} é desconexo; se for paralelo, então o grafo quociente correspondente não possui arestas mas possui dois ou mais vértices, sendo, portanto, desconexo e, conseqüentemente, H também é desconexo; e se for vizinhança, H e \overline{H} são conexos uma vez que o grafo quociente correspondente ao nó é conexo e seu complemento também o é.

Ainda, nota-se que todos os módulos de um grafo G são caracterizados pela sua árvore

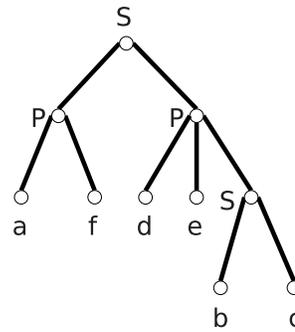


Figura 2.5: Árvore de decomposição modular rotulada.

de decomposição modular T . Ou seja, M é um módulo de G se, e somente se, M é um nó de T ou M é a união de filhos de um nó degenerado de T .

2.4 Classes de grafos

Nesta seção serão estudadas algumas classes que podem ser definidas em termos da decomposição modular de seus grafos, embora, a princípio, muitas delas tenham sido estudadas sem auxílio desta poderosa ferramenta. O uso da decomposição modular em tais classes destaca-se por proporcionar soluções lineares ou polinomiais para problemas que, para grafos em geral, são NP-completos ou ainda indeterminados, como o isomorfismo.

Todas as classes de grafos avaliadas nesta seção são conhecidas como classes de grafos com “poucos” P_4 's. Isto porque estas classes restringem a presença de P_4 induzidos em seus grafos, o que limita, de certa forma, a densidade local destes subgrafos. Além disso, todas estas classes possuem caracterizações interessantes expressas em termos de propriedades da árvore de decomposição modular de seus grafos.

A Figura 2.6 exhibe classes de grafos nesta condição e a relação de continência entre elas. Em seguida, algumas notações são introduzidas e, então, algumas destas classes são definidas e caracterizadas.

Entre as classes de grafos com poucos P_4 's, pode-se dar mais ênfase as classes P_4 -carregada e P_4 -carregada estendida. Giakoumakis [31] introduziu estas classes e também mostrou que problemas como ciclo máximo, coloração de vértices e clique máxima são resolvidos em tempo linear para os grafos destas classes. A classe P_4 -carregada estendida contém muitas classes bem estudadas, tais como P_4 -carregada e P_4 -arrumada, que por sua vez, contém as classes P_4 -leve, P_4 -extensível, P_4 -esparsa e cografos [32]. Cada uma destas subclasses possuem aplicações específicas. Por exemplo, o número de dispersão (*scattering number*) e emparelhamento máximo são resolvidos em tempo linear em grafos da classe P_4 -arrumada [32, 30].

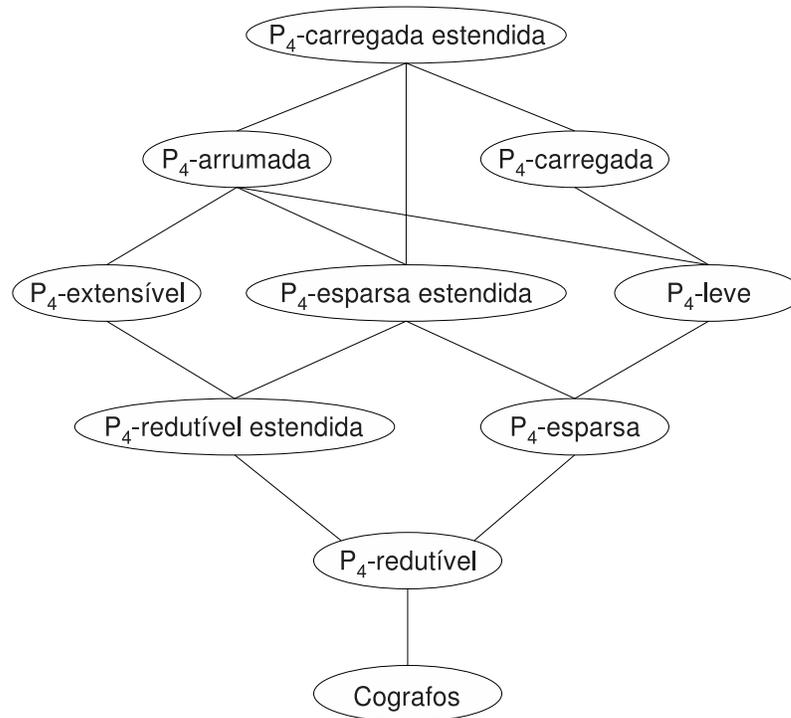


Figura 2.6: Diagrama de Hasse das classes, adaptado de [32].

As restrições que versam sobre estas classes de grafos limitam a ocorrência de subgrafos induzidos isomorfos ao P_4 e o relacionamento que estes subgrafos mantêm entre si ou com os demais vértices do grafo. Dada a importância do P_4 para tais classes, são introduzidas abaixo algumas notações que facilitarão o uso destes conceitos.

Dado um grafo G e um inteiro $n > 1$, define-se $P_n(G) = \{P \subseteq V(G) : G[P] \simeq P_n\}$. Em especial, $P_4(G)$ contém todos os conjuntos de quatro vértices que induzem em G um grafo isomorfo ao P_4 .

Considere um grafo G para o qual existem $P \in P_4(G)$ e $v \in V(G) \setminus P$. Tome seu subgrafo $H = G[P \cup \{v\}]$. Se $|P_4(H)| \geq 2$, então o vértice v é chamado de *parceiro* de P em G e denota-se por $R(G, P)$ o conjunto de todos os vértices parceiros de P em G . Note que, neste caso, H é isomorfo a um dos grafos Z_1 a Z_7 da Figura 2.7. Caso contrário, H é isomorfo a algum dos grafos dos casos I , U e B da mesma figura, nos quais $|P_4(H)| = 1$.

A seguir serão estudadas algumas classes com “poucos P_4 ”. Os cografos, grafos P_4 -esparços, grafos P_4 -leves e grafos P_4 -carregados são detalhados nas seções 2.4.1, 2.4.2, 2.4.3 e 2.4.4. As classes P_4 -esparça estendida, P_4 -arrumada e P_4 -carregada estendida são tratadas em conjunto pela Seção 2.4.5. Todas estas classes são classes hereditárias.

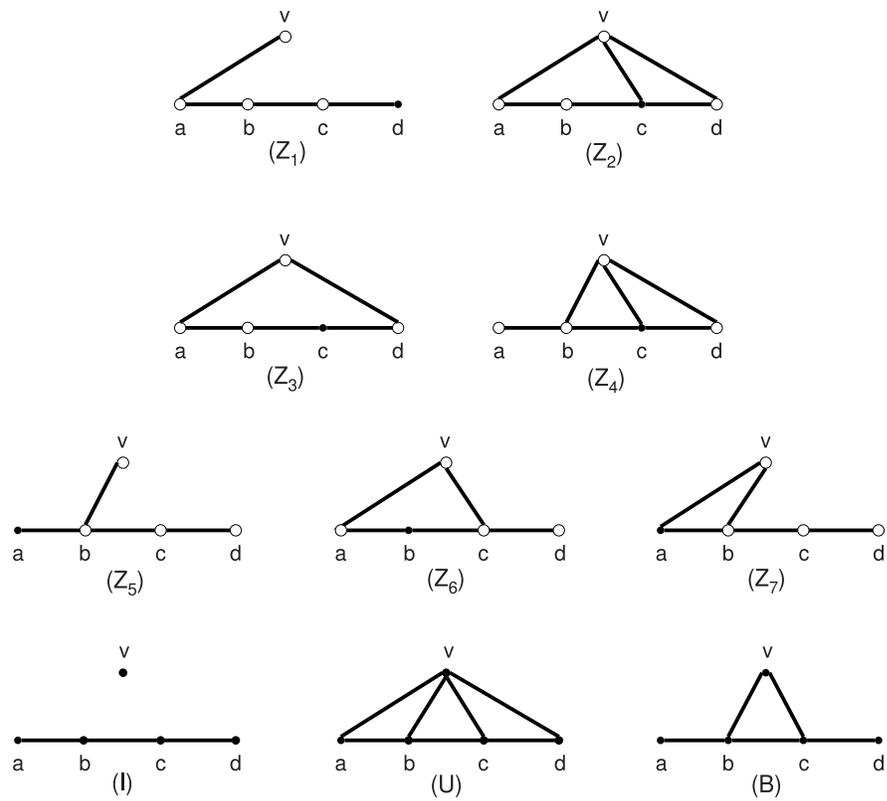


Figura 2.7: Grafos de cinco vértices com um P_4 induzido (por $\{a, b, c, d\}$) e um vértice adicional (v). Os grafos Z_1 a Z_7 possuem ao menos outro P_4 induzido no mesmo grafo, destacado por vértices vazados, de forma que, nestes grafos, v é parceiro do P_4 induzido por $\{a, b, c, d\}$.

2.4.1 Cografos

A classe dos cografos (do inglês *cograph*) foi definida por vários autores em trabalhos independentes e, assim, apareceu na literatura com vários sinônimos, entre eles, D^* -grafos, grafos P_4 restritos e HD-grafos. Contudo, a primeira definição da classe [53] é recursiva e denomina um grafo simples por *cografo* se o mesmo satisfizer uma das seguintes condições:

- Possuir um único vértice; ou
- seu complemento for um cografo; ou
- for a união disjunta de dois cografos (veja 2.1).

Da própria definição segue que pode-se decompor qualquer cografo até seus vértices isolados, alternadamente tomando seus componentes conexos e complementando-os. Vê-se facilmente que esta decomposição é única e pode ser representada como uma árvore na qual os nós correspondem às duas operações usadas e as folhas correspondem aos vértices do grafo. Contudo, esta árvore de decomposição contém nós com um único filho, da operação de complemento, que em seguida serão divididos em seus componentes.

Ao aglutinar estas duas operações, define-se a *co-árvore* como a árvore de decomposição de cografos em subgrafos induzidos pelos vértices dos componentes conexos do grafo (rótulo 0 para operação de união) ou pelos vértices dos componentes conexos do complemento do grafo (rótulo 1 para operação *join*). Um exemplo de um cografo e sua co-árvore estão na Figura 2.8.

A co-árvore e a árvore de decomposição modular de um cografo são isomorfas e seus rótulos são correspondentes, ou seja, os rótulos 0 e 1 da co-árvore correspondem aos rótulos paralelo e serial da árvore de decomposição modular, respectivamente (esta relação pode ser vista na Figura 2.8). Com isso, os cografos são os únicos grafos cuja decomposição modular não contém nós do tipo vizinhança.

Pode-se reconstruir um cografo a partir de seus vértices usando a co-árvore e as operações de união e *join*. O grafo correspondente a um nó de rótulo 0 (1) é obtido pela união (*join*) dos grafos correspondentes aos filhos deste nó na co-árvore.

As adjacências entre vértices de um cografo são expressas na sua co-árvore, uma vez que dois vértices de um cografo são vizinhos se, e somente se, o ancestral comum a ambos, mais distante da raiz, é um nó rotulado 1 . Já que exibir a co-árvore é suficiente para descrever o cografo, a hereditariedade da classe (veja 2.1) é provada mostrando que dado um cografo G e sua co-árvore T , ao remover um vértice de G é possível adaptar T para representar o novo grafo.

Lema 2.1 ([21]). *Todo subgrafo induzido de um cografo é um cografo.*

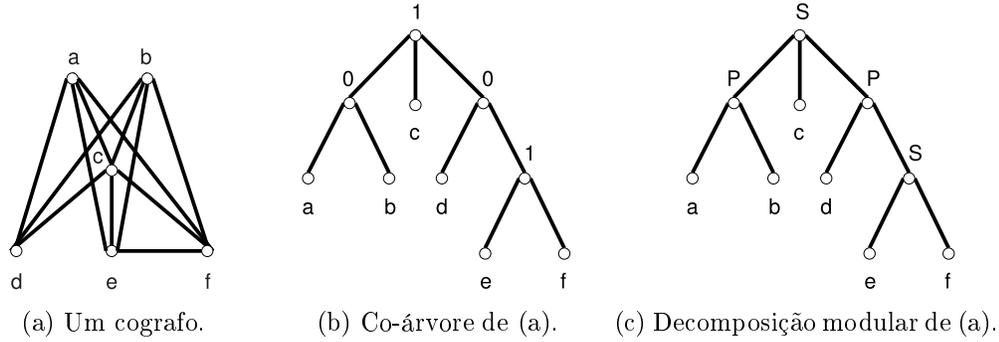


Figura 2.8: Um cografo e suas decomposições.

Apesar dos cografos terem sido introduzidos por Lerchs [53], também foram estudados independentemente por vários pesquisadores, resultando em uma riqueza de caracterizações, das quais destacamos a seguinte [21]: um grafo G é um cografo se, e somente se, não possui subgrafos induzidos isomorfos ao P_4 .

Também é conhecido um algoritmo linear para construção da co-árvore e reconhecimento da classe [22], anterior aos algoritmos lineares de decomposição modular (para grafos gerais), que também podem ser usados para este fim, sendo, porém, muito mais complexos.

2.4.2 P_4 -esparsa

Esta classe de grafos foi introduzida por Hoàng em sua dissertação de mestrado [42], na qual forneceu algumas caracterizações equivalentes e mostrou que estes grafos são perfeitos (veja 2.1) e perfeitamente ordenáveis (veja 2.1).

Um grafo G pertence à classe P_4 -esparsa (do inglês P_4 -sparse) se, todo subgrafo H induzido em G , com cinco vértices, satisfizer $|P_4(H)| \leq 1$, ou, de forma equivalente, se para todo $P \in P_4(G)$, tem-se que $R(G, P) = \emptyset$. A seguir, examina-se uma família de grafos especiais para esta classe e então introduz-se algumas caracterizações equivalentes.

Um grafo G é uma *aranha* (do inglês *spider*) se $V(G)$ pode ser particionado em conjuntos K , S e R de forma que:

- K é uma clique, S é um conjunto independente e $|K| = |S| \geq 2$;
- Existe uma bijeção $b : K \rightarrow S$ tal que:
 - $\forall k \in K, N_G(k) \cap S = \{b(k)\}$, caso em que G é chamado de *aranha magra*; ou
 - $\forall k \in K, N_G(k) \cap S = S \setminus \{b(k)\}$, caso em que G é chamado de *aranha gorda*.

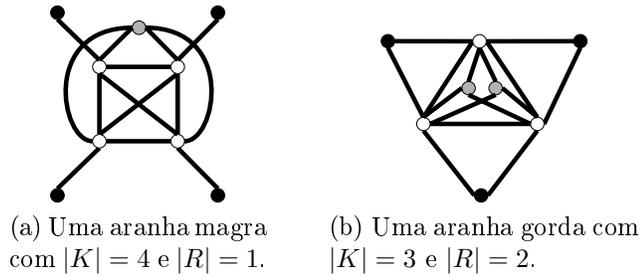


Figura 2.9: Aranhas gorda e magra.

- $\forall r \in R, (K \subseteq N_G(r))$ e $(N_G(r) \cap S) = \emptyset$.

A partição dos vértices de uma aranha segundo a definição acima será representada como (K, S, R) e referida como *partição canônica* ou simplesmente partição. Esta partição está bem definida, uma vez que é única. A Figura 2.9 contém exemplos de aranhas gordas e magras. Vértices vazados pertencem à K , vértices negros pertencem à S e os vértices em tom cinza à R .

Note que, qualquer subgrafo isomorfo ao P_4 em uma aranha está contido no subgrafo $G[R]$ ou no subgrafo $G[K \cup S]$ e, neste último caso, seus extremos (vértices de grau 1) estão em S e os vértices internos em K .

A decomposição modular de uma aranha tem como raiz um nó vizinhança. Além disso, todos os vértices em K e S são módulos fortes triviais, filhos da raiz, e R , quando existir, será um módulo forte representado por outro filho da raiz.

Giakoumakis e Vanherpe propuseram para esta classe uma caracterização relacionada a decomposição modular das aranhas.

Teorema 2.2 ([33]). *Um grafo G pertence à classe P_4 -esparsa se, e somente se, para todo nó vizinhança de sua árvore de decomposição modular, seu grafo quociente for isomorfo a uma aranha e seus fatores forem todos grafos triviais, exceto possivelmente aquele correspondente ao conjunto R , da partição canônica (K, S, R) do grafo quociente.*

Em outra caracterização, por restrições locais, as aranhas apresentam papel importante.

Teorema 2.3 ([46]). *Um grafo G é um grafo P_4 -esparso se, e somente se, todo subgrafo H induzido em G , com ao menos dois vértices, satisfaz uma das seguintes afirmações:*

- H é desconexo;
- \overline{H} é desconexo;
- H é isomorfo a uma aranha.

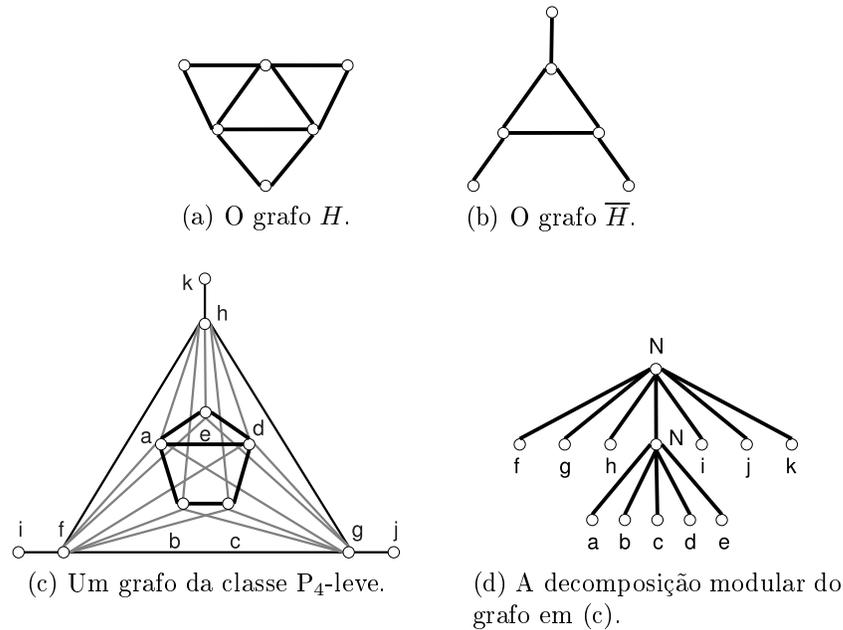


Figura 2.10: Um grafo da classe P_4 -leve e sua decomposição modular.

A classe também possui uma caracterização por subgrafos proibidos, assim como os cografos, dada abaixo.

Lema 2.4 ([46]). *Um grafo G é um grafo P_4 -esparso se, e somente se, não contém nenhum subgrafo induzido isomorfo a qualquer dos grafos $\{Z_1, Z_2, \dots, Z_7\}$ da Figura 2.7.*

2.4.3 P_4 -leve

Esta classe foi introduzida por Jamison e Olariu ([44]) (neste denominada P_4 -lite) e contém todo grafo G cujos subgrafos induzidos de até seis vértices possuem no máximo dois P_4 induzidos ou são isomorfos ao grafo H ou ao grafo \overline{H} da Figura 2.10 (que possuem seis vértices e três subgrafos isomorfos ao P_4). A mesma figura contém um exemplo de grafo da classe, embora o reconhecimento destes grafos fique mais claro após a caracterização dada a seguir.

Uma *pseudo-aranha* (do inglês *pseudo-spider*) é um grafo obtido de uma aranha de partição canônica (K, S, R) , substituindo um vértice v contido em $K \cup S$ por um par de novos vértices, v' e v'' , e fazendo ambos adjacentes aos mesmos vértices aos quais v era adjacente. Os novos vértices podem ser ou não adjacentes. A partição canônica da pseudo-aranha é obtida da partição canônica da aranha, removendo o vértice v de seu conjunto e adicionando, ao mesmo, o par de novos vértices, v' e v'' . Note que esta partição é única para uma pseudo-aranha. A decomposição modular de uma pseudo-aranha também tem

como raiz um nó vizinhança, cujo grafo quociente é uma aranha.

Com base no trabalho de Giakoumakis et al. [32], tem-se uma caracterização construtiva destes grafos.

Teorema 2.5 ([32]). *Um grafo G é um grafo P_4 -leve se, e somente se, todo subgrafo H induzido em G , com ao menos dois vértices, satisfaz uma das seguintes afirmações:*

- H é desconexo;
- \overline{H} é desconexo;
- H é isomorfo a uma aranha;
- H é isomorfo a uma pseudo-aranha; ou
- H é isomorfo a P_5 ou a $\overline{P_5}$.

No mesmo trabalho é apresentado um algoritmo de reconhecimento linear baseado na árvore de decomposição modular, do qual obtém-se outra caracterização para a classe:

Teorema 2.6 ([32]). *Um grafo G pertence à classe P_4 -leve se, e somente se, para todo nó vizinhança de sua árvore de decomposição modular, seu grafo quociente é isomorfo ao P_5 , ao $\overline{P_5}$ ou a uma aranha (vide Seção 2.4.2) e seus fatores são todos grafos triviais, exceto quando o grafo quociente for isomorfo a uma aranha, caso em que o fator correspondente ao conjunto R da partição canônica do grafo quociente não tem restrições adicionais e mais um fator (correspondente a um vértice de K ou S) pode possuir dois vértices.*

É fácil notar que esta classe contém todos os grafos P_4 -esparsos. Ainda, mostra-se que os grafos da classe são `brittle` [44] (veja 2.1) e, portanto, perfeitamente ordenáveis.

2.4.4 P_4 -carregada

Giakoumakis introduziu a classe P_4 -carregada (do inglês P_4 -laden [31]) como contendo todo grafo cujos subgrafos induzidos com até seis vértices contém no máximo dois P_4 induzidos distintos ou são grafos `split`.

Uma vez que os grafos H e \overline{H} da Figura 2.10 são casos especiais de grafos `split`, verifica-se trivialmente que esta classe contém propriamente a classe P_4 -leve. Além disso, foi dada uma caracterização da classe segundo sua decomposição modular, para a qual introduz-se a seguinte notação: seja G um grafo `split` e $\{K, S\}$ uma partição de seus vértices em uma clique e um conjunto independente. Denotam-se, por $S(G)$, o subconjunto de S formado por vértices que não são vizinhos de pelo menos um vértice em K , ou seja, $S(G) = \{s \in S : N_G(s) \subsetneq K\}$; por $K(G)$, a vizinhança de $S(G)$ (que está contida em K), isto é, $K(G) = N_G(S(G))$ e por $R(G)$, os demais vértices de G .

Teorema 2.7 ([31]). *Um grafo pertence à classe P_4 -carregada se, e somente se, o grafo quociente de cada nó vizinhança de sua árvore de decomposição modular for isomorfo:*

- ao P_5 ou ao $\overline{P_5}$ e seus fatores forem grafos triviais; ou
- a uma aranha e seus fatores forem todos grafos triviais, exceto o fator correspondente ao conjunto R (da partição canônica da aranha), se não vazio, e mais outro vértice qualquer, cujo fator pode conter dois vértices; ou
- a um grafo *split* G e os fatores correspondentes aos vértices em $S(G)$ forem conjuntos independentes e os correspondentes aos vértices em $K(G)$ forem cliques.

Desta caracterização, também deriva-se a caracterização construtiva. Para tal, se um grafo H é formado a partir de um *split* G com uma partição *split* (K, S) pela substituição de vértices em $R(G)$ por subgrafos quaisquer, então H é chamado um grafo *pseudo-split*.

Teorema 2.8. *Um grafo G é um grafo P_4 -carregado se, e somente se, todo subgrafo H induzido em G , com ao menos dois vértices, satisfaz uma das seguintes afirmações:*

- H é desconexo;
- \overline{H} é desconexo;
- H é isomorfo a uma pseudo-aranha;
- H é isomorfo a P_5 ou a $\overline{P_5}$; ou
- H é um grafo *pseudo-split* (inclui as aranhas).

No trabalho em que foi definida, também foi apresentado um algoritmo linear de reconhecimento da classe que valida os critérios definidos no Teorema 2.7. Outra característica interessante da classe é que seus grafos são *brittle*, a exemplo de sua subclasse P_4 -leve.

Na Figura 2.11 está exposto um grafo da classe e sua decomposição modular. Note em (c) que o grafo quociente do nó de rótulo vizinhança é um grafo *split*, pois pode-se particionar seus vértices na clique $\{b, d, e\}$ e no conjunto independente $\{a, c, f\}$. O único fator não trivial é o conjunto independente $\{a', a''\}$, que corresponde ao vértice a do grafo quociente, pertencente a $S(G)$.

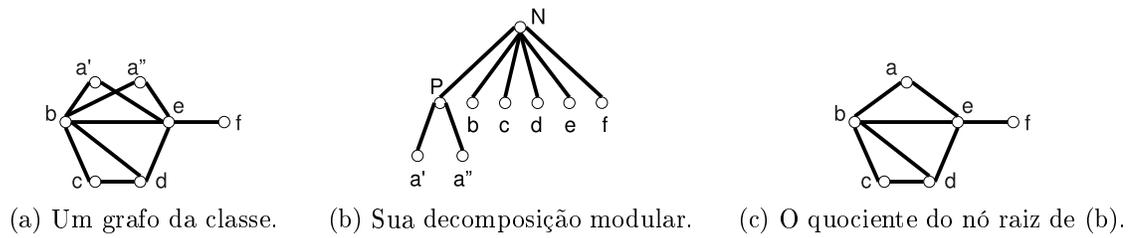


Figura 2.11: Um grafo da classe P_4 -carregada.

2.4.5 Extensões

As classes de grafos P_4 -esparsa, P_4 -leve e P_4 -carregada, descritas anteriormente, foram estendidas através de um método comum, produzindo as classes P_4 -esparsa estendida [33], P_4 -arrumada [32] (do inglês *P_4 -tidy*) e P_4 -carregada estendida [31]. O método usado consiste em permitir que grafos das classes estendidas possuam subgrafos induzidos isomorfos ao C_5 . Desta forma, estas classes podem ser caracterizadas construtivamente, com base na caracterização da classe que foi estendida, bastando permitir o C_5 como subgrafo induzido. A caracterização pela decomposição modular também pode ser obtida, permitindo que qualquer nó vizinhança de um grafo de uma destas classes tenha como grafo quociente o C_5 e seus fatores sejam grafos triviais.

Capítulo 3

Separadores Minimais de Grafos P_4 -carregados estendidos

Neste capítulo, descreve-se um algoritmo de complexidade temporal linear para listar todos os separadores minimais de um grafo da classe P_4 -carregada estendida. Para tanto, demonstra-se que para qualquer grafo da classe P_4 -carregada estendida, o número de separadores minimais do grafo e a soma da cardinalidade de todos estes separadores são limitados por funções lineares no tamanho do grafo. Além disso, são dados limitantes diferenciados e justos para a classe P_4 -carregada estendida e para algumas de suas subclasses: P_4 -carregada, P_4 -arrumada e P_4 -leve.

3.1 Introdução

Um *separador* é um conjunto de vértices de um grafo tal que, ao serem removidos, obtém-se um grafo desconexo (note que o conjunto \emptyset é um separador de qualquer grafo desconexo). Um separador também é conhecido como *corte de vértices*. Dados dois vértices distintos do grafo, a e b , um separador é dito *ab-separador* se, ao remover seus vértices do grafo, não existe mais caminho entre a e b , ou seja, a e b ficaram em componentes conexos distintos do grafo resultante. Um *ab-separador minimal* é um *ab-separador* que não contém propriamente nenhum outro *ab-separador* do grafo. Por fim, um *separador minimal* de um grafo é um *ab-separador minimal*, para algum par de vértices a e b do grafo. O Problema dos Separadores Minimais (PSM) consiste em determinar todos os separadores minimais de um grafo.

Como ilustração destes conceitos, considere os grafos na Figura 3.1. No caso (a), os separadores minimais do grafo são os conjuntos $\{a, c, e\}$, $\{b, c, d\}$, $\{e\}$, $\{f, g\}$ e $\{e, h\}$. Note que o conjunto destes separadores não é minimal. Contudo, cada separador é minimal para um par de vértices específico. Por exemplo, $\{e, h\}$ é um *fg-separador minimal*. Já

o caso (b) ilustra uma família infinita de grafos para os quais o número de separadores minimais é exponencial no tamanho do grafo. Isto é fácil de ver, considerando que os ab -separadores minimais são conjuntos $\{s_1, s_2, \dots, s_n\}$ nos quais $s_i = x_i$ ou $s_i = y_i$, para todo $1 \leq i \leq n$. Logo, há, pelo menos, 2^n separadores minimais num grafo com $2n + 2$ vértices e $3n$ arestas.

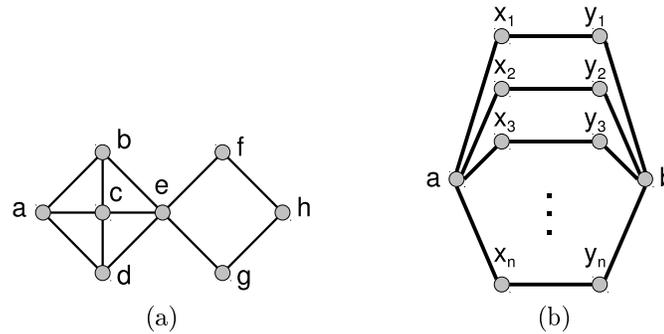


Figura 3.1: Grafos usados para exemplificar o conceito de separadores minimais.

Os separadores minimais de um grafo relacionam-se a outros conceitos em teoria dos grafos, como por exemplo a triangularização mínima [59, 41], bem como são utilizados em algoritmos para variados problemas, como variações do Problema da Filogenia Perfeita [37, 38]. Além disso, Bouchitté e Todinca [9, 10] mostraram que o *treewidth* e o preenchimento mínimo podem ser computados em tempo polinomial em qualquer classe de grafos cujos separadores minimais forem limitados polinomialmente no tamanho do grafo. Note que ambos os problemas são casos especiais de triangularizações e são NP-Difíceis para grafos arbitrários [2, 73]. Bodlaender e Rotics usaram a técnica da decomposição modular para mostrar que se os grafos primos obtidos na decomposição possuem um número polinomial de separadores minimais, então os problemas acima citados são resolvidos em tempo polinomial. Os autores também demonstraram que estes problemas podem ser resolvidos em tempo polinomial em algumas classes especiais de grafos com um número exponencial de separadores minimais [8].

A literatura científica contém uma série de algoritmos para listar todos os separadores minimais de um grafo. O primeiro deles, de Kloks e Kratsch, tem complexidade temporal $O(n^5 R)$ [51], no qual R é o número de separadores minimais do grafo. Em 2000, Berry et al. [5] publicaram um algoritmo de complexidade temporal $O(n^3 R)$. Convém notar que os grafos fracamente cordais, cordais, trapezoidais e de permutação possuem um número polinomial de separadores minimais (veja revisões em [58, 51]). Embora os grafos fracamente cordais contenham tanto os grafos cordais como os P_4 -carregados, o número de separadores minimais de grafos cordais é $O(n)$ [66] enquanto que para os grafos fracamente cordais é $O(n + m)$ [6]. Além disso, existem algoritmos para encontrar os separadores

minimais de grafos cordais [52, 7] e P_4 -esparsos [58] em tempo linear, de grafos fracamente cordais [6] em tempo $O(m^2)$ e de grafos planares 3-conexos [55] em tempo linear por separador.

Neste capítulo, será considerado o Problema dos Separadores Minimais restrito à classe P_4 -carregada estendida, que contém tanto a classe P_4 -arrumada como os grafos `split`. Como resultado deste trabalho, a classe P_4 -carregada estendida figura entre as classes de grafos que possuem um algoritmo de tempo linear para este problema e cujo número de separadores minimais é $O(n)$. Ambos resultados são melhorias em relação aos grafos fracamente cordais, uma superclasse da P_4 -carregada.

Na Seção 3.2 serão descritos os resultados conhecidos para o PSM restrito à classe P_4 -esparsa. Já na Seção 3.3 a técnica utilizada para a classe P_4 -esparsa é generalizada para grafos arbitrários através da decomposição modular. Na Seção 3.4 são caracterizados os separadores minimais de grafos da classe P_4 -carregada estendida. Limitantes no número e tamanho destes separadores minimais são dados na Seção 3.5. Na Seção 3.6 os limitantes dados são demonstrados justos. Finalmente, as conclusões deste trabalho são resumidas na Seção 3.7.

No restante deste capítulo as seguintes notações serão utilizadas: $\lambda(G)$ representará o conjunto de todos os separadores minimais de um grafo G , $l_G = |\lambda(G)|$ e $s_G = \sum_{X \in \lambda(G)} |X|$. Os dois últimos parâmetros representam o número e o tamanho da descrição de todos os separadores minimais de um grafo.

3.2 Classe P_4 -esparsa

Nesta seção descrevem-se os resultados já publicados por Nikolopoulos e Palios [58] sobre os separadores minimais de grafos P_4 -esparsa. Tais resultados compreendem a descrição dos separadores minimais de alguns tipos de grafos e suas composições.

Um grafo da classe P_4 -esparsa é um grafo união, `join` ou uma aranha e todos seus subgrafos induzidos também satisfazem esta condição. Assim, ao tratar estes três casos, pode-se computar todos os separadores minimais da classe. Este estudo foi feito e culminou no seguinte resultado.

Lema 3.1 (Nikolopoulos e Palios [58]). *Considere um grafo G .*

- *Se G é desconexo e B_1, B_2, \dots, B_p , $p \geq 2$, são seus componentes conexos, então $\lambda(G) = \{\emptyset\} \cup \bigcup_{i=1}^p \lambda(B_i)$.*
- *Se \overline{G} é desconexo e B_1, B_2, \dots, B_p , $p \geq 2$, são os complementos dos componentes conexos de \overline{G} , então $T \in \lambda(G)$ se, e somente se, $T = T_i \cup (V(G) \setminus V(B_i))$ e $T_i \in \lambda(B_i)$.*

- Se G é uma aranha com partição (K, S, R) então $T \in \lambda(G)$ se, e somente, se
 - $T = K \cup T_H$ e $T_H \in \lambda(H)$, sendo $H = G[R]$; ou
 - $T = \{v\}$, $v \in K$ e G é uma aranha magra; ou
 - $T = K \setminus \{v\}$, $v \in K$ e G é uma aranha gorda.

Para ilustrar os separadores minimais de grafos união e join, dados pelo Lema 3.1, considere a Figura 3.2. O grafo união do item (a) tem como separadores minimais os conjuntos \emptyset (b_1c_1 -separador minimal) e $\{a_2\}$ (a_1a_3 -separador minimal), que são os separadores minimais de cada operando, além do conjunto vazio. Já no item 3.2b, apresenta-se um grafo join tal que $B_1 = \{a_1, a_2, a_3, a_4\}$ e $B_2 = \{b_1, b_2\}$. Este grafo tem como separadores minimais o conjunto $\{a_1, a_2, a_3, a_4\}$, que é a união do conjunto \emptyset (b_1b_2 -separador minimal) com B_1 , e os conjuntos $\{a_2, b_1, b_2\}$ e $\{a_3, b_1, b_2\}$, obtidos unindo B_2 com os separadores minimais de $G[B_1]$, $\{a_2\}$ e $\{a_3\}$, respectivamente.

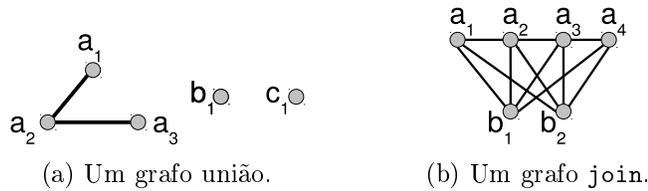


Figura 3.2: Grafos união e join para exemplo de separadores minimais.

Além disso, na Figura 3.3 há exemplos de aranhas, cujos separadores minimais são dados, também, pelo Lema 3.1. No item (a), os separadores minimais são $\{k_1\}$, $\{k_2\}$ e $\{k_3\}$ (dados pelas vizinhanças dos vértices em S) e o separador $\{h_2, k_1, k_2, k_3\}$ obtido unindo K ao separador minimal $\{h_2\}$, de $G[R]$, onde $R = \{h_1, h_2, h_3\}$. A aranha do item (b) tem como separadores minimais os conjuntos $\{k_1, k_2\}$, $\{k_1, k_3\}$ e $\{k_2, k_3\}$, que são os vizinhos de cada vértice em S , já que $R = \emptyset$.

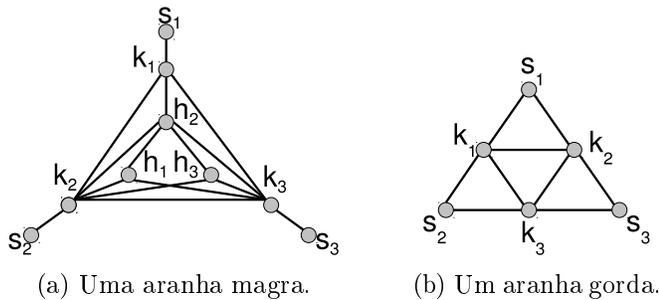


Figura 3.3: Aranhas para exemplo de separadores minimais.

A fim de analisar o tempo gasto pelo algoritmo, é preciso considerar o número e o tamanho total dos separadores minimais dos grafos. Nikolopoulos e Palios conduziram tal estudo e chegaram ao seguinte resultado.

Teorema 3.2 (Nikolopoulos e Palios [58]). *Seja G um grafo P_4 -esparso com n vértices e m arestas. Se G é desconexo, ele possui no máximo $(2n - 1)/3$ separadores minimais, e se G é conexo, possui no máximo $(2n - 2)/3$ separadores minimais. Em qualquer caso, o tamanho total da descrição dos separadores minimais de G não excede $4m/3$.*

3.3 Aplicação da Decomposição Modular

Observa-se que a descrição dos separadores minimais obtida por Nikolopoulos e Palios, dada no Lema 3.1, é dividida em três casos: grafos união, join e aranhas. Em cada um destes casos, a árvore de decomposição modular do grafo tem um nó raiz com rótulo distinto: paralelo, serial ou vizinhança, respectivamente. Ainda mais, ao considerar o grafo quociente do nó raiz, ele será um grafo sem arestas, completo ou uma aranha prima (tal que $|R| \leq 1$). Os dois primeiros são os únicos casos de grafos quocientes não primos em uma árvore de decomposição modular, enquanto o terceiro tipo é um caso especial de nó vizinhança.

A relação entre os grafos considerados no Lema 3.1 e propriedades das suas árvores de decomposição modular motivou o estudo da caracterização dos separadores minimais em situações mais gerais. A seguir, a questão será abordada para um grafo H qualquer, obtendo os separadores minimais deste grafo a partir dos separadores minimais do grafo induzido por módulos de H e dos separadores minimais do respectivo grafo quociente, obtido contraindo tais módulos a um vértice. Note que, se o grafo H não possuir um módulo não-trivial (com pelo menos dois e no máximo $|V(H)| - 1$ vértices), então o lema abaixo não adiciona informações sobre os separadores minimais de H .

Teorema 3.3. *Seja H um grafo, M_1, M_2, \dots, M_p um conjunto de módulos de H , dois a dois disjuntos e tais que sua união forma $V(H)$ (eles são uma partição de $V(H)$ em módulos), e $G = H|\{M_1, M_2, \dots, M_p\}$. Seja v_i , $1 \leq i \leq p$, o vértice de G representante do módulo M_i de H . Então, T é um separador minimal de H se, e somente se:*

- $T = T' \cup \left(\bigcup_{v_k \in N_G(v_i)} M_k \right)$ se T' é um separador minimal de $H[M_i]$; ou
- $T = \bigcup_{v_k \in T'} M_k$ e T' é um separador minimal de G .

Demonstração. Com o objetivo de determinar os separadores minimais de H , serão examinados os pares de vértices não adjacentes $\{a, b\}$. Se a e b pertencem ao mesmo módulo M_i , é fácil perceber que $T = T' \cup \left(\bigcup_{v_k \in N_G(v_i)} M_k \right)$ é um separador minimal de H se, e

somente se, T' é um separador minimal de $H[M_i]$. De fato, uma vez que M_i é um módulo de H , os vértices de qualquer M_k , tal que $v_k \in N_g(v_i)$, são todos adjacentes a todos os vértices de M_i e, portanto, T é um $\{a, b\}$ -separador de H se, e somente se, T' é um ab -separador de $H[M_i]$. Além disso, a minimalidade de T segue da minimalidade de T' .

Se a e b pertencerem a módulos distintos, M_i e M_j , considere o grafo contraído G . Para cada caminho P de a a b em G , existe em H um conjunto de caminhos obtidos de P pela substituição de cada vértice V_x de P por qualquer outro vértice v_y de M_x . Logo, a remoção de $T = \bigcup_{v_k \in T'} M_k$ desconecta a de b em H se, e somente se, a remoção de T' desconecta v_i e v_j em G . Além disso, a minimalidade de T segue da minimalidade de T' . Por fim, observa-se que se T é um ab -separador minimal de H , então T também é um separador minimal com respeito a qualquer outro par de vértices x e y , $x \in M_i$ e $y \in M_j$. \square

A Figura 3.4 exemplifica a construção dos separadores minimais de um grafo H de acordo com o Teorema 3.3. O item (a) representa um separador minimal S de H que também é separador minimal do grafo G , mostrado em (d), pois S não contém os vértices d_1 e d_2 . Já o separador minimal apresentado no item (b) é obtido a partir do separador minimal de G dado em (e), substituindo o vértice a pelo módulo M . O último caso, representado pelo item (c), ocorre quando um separador minimal de H é formado tomando um separador minimal de $H[M]$ ($\{b_2\}$) e acrescentando os vértices adjacentes a b em G , conforme (f).

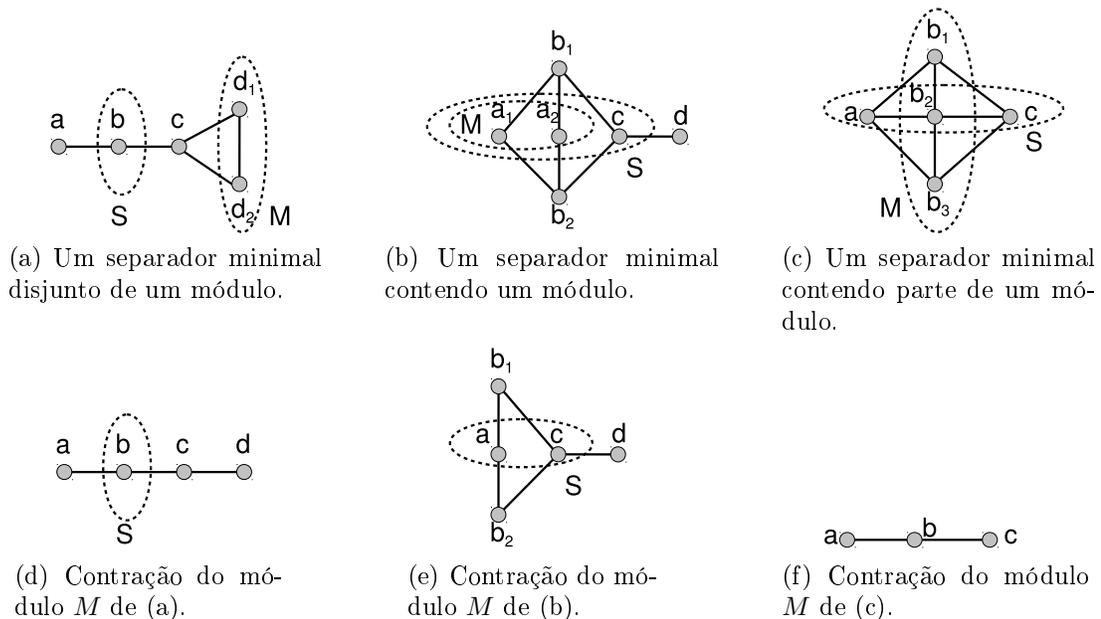


Figura 3.4: Relação entre separadores minimais e módulos.

A aplicação repetida dos resultados do Teorema 3.3 levam imediatamente a uma caracterização dos separadores minimais de qualquer grafo, baseado em sua decomposição modular.

Nota-se também que parte dos resultados dados por Nikolopoulos e Palios no Lema 3.1 podem ser provados usando o Teorema 3.3. Isto porque num grafo desconexo seus componentes conexos são módulos e num grafo cujo complemento é desconexo, os componentes conexos do complemento do grafo também são seus módulos. O Teorema 3.3 também caracteriza os separadores minimais de aranhas não primas a partir dos separadores minimais de aranhas primas.

O Teorema 3.3 fornece uma estratégia para computar os separadores minimais de um grafo a partir da sua decomposição modular. Dado um grafo G qualquer e sua árvore de decomposição modular, T , particionam-se os vértices de G em módulos fortes M_1, M_2, \dots, M_t , representados pelas subárvores T_1, T_2, \dots, T_t , de T , obtidas removendo-se a raiz de T . Pela contração dos módulos M_1, M_2, \dots, M_t , obtém-se o grafo quociente Q de G , cujos separadores minimais não podem ser computados pelo mesmo algoritmo (já que Q pode ser primo ou isomorfo a G). Já os separadores minimais dos grafos induzidos por cada módulo M_i podem ser computados recursivamente, uma vez que a ADM de $G[M_i]$ é T_i . Então, a simples aplicação do Teorema 3.3 produz $\lambda(G)$.

O Algoritmo 1 computa recursivamente os separadores minimais de um grafo qualquer usando a estratégia descrita. Relembre que a árvore de decomposição modular T de um grafo, bem como a construção dos grafos quocientes associados a cada nó interno de T pode ser feita em tempo linear no tamanho de G . O Algoritmo 1 apenas executa uma busca em profundidade na árvore T , computando, para cada nó, os separadores do subgrafo induzido pelo correspondente módulo. Logo, o tempo total do algoritmo depende de s_G e da complexidade para computar os separadores minimais dos grafos quocientes.

3.4 Classe P_4 -carregada estendida

Nesta seção, o PSM será solucionado para uma superclasse dos grafos P_4 -esparcos, a classe P_4 -carregada estendida, definida na Seção 2.4.5. Para tanto, é necessário computar os separadores minimais de todos os grafos que podem surgir como quocientes da decomposição modular de grafos da classe P_4 -carregada estendida, ou seja, grafos `split`, C_5 , P_5 e $\overline{P_5}$, além dos quocientes dos grafos união e `join`, já considerados anteriormente. Note que aranhas primas são grafos `split`.

Algoritmo 1 Separadores minimais usando a decomposição modular (sepmin).

Entrada: Um grafo G e sua ADM, T .

Saída: $\lambda(G)$ contém todos os separadores minimais de G .

$\lambda(G) \leftarrow \emptyset$

Se $|V(G)| \geq 2$ **então**

Obtenha o grafo quociente Q da raiz de T

Computar $\lambda(Q)$ usando outro algoritmo

Para cada separador minimal $S \in \lambda(Q)$ **faça**

$S' \leftarrow \emptyset$

Para cada vértice $v_i \in S$ **faça**

$S' \leftarrow S' \cup M_i$

Fim Para

$\lambda(G) \leftarrow \lambda(G) \cup \{S'\}$

Fim Para

Para cada M_i **faça**

$\lambda(G_i) \leftarrow \text{sepmin}(G_i, T_i)$

Para cada separador minimal $S \in \lambda(M_i)$ **faça**

$S' \leftarrow S$

Para cada vértice $v_j \in N_Q(v_i)$ **faça**

$S' \leftarrow S' \cup M_j$

Fim Para

$\lambda(G) \leftarrow \lambda(G) \cup \{S'\}$

Fim Para

Fim Para

Fim Se

3.4.1 Grafos *split*

Como visto, os grafos *split* assumem um papel importante como grafos quociente de nós da decomposição modular de um grafo da classe P_4 -carregada estendida. Logo, o primeiro passo para considerar esta classe é, naturalmente, caracterizar os separadores minimais de um grafo *split*.

Lema 3.4. *Sejam G um grafo *split* e (K, S) uma partição *split* de G com K maximal. Então, $P \in \lambda(G)$ se, e somente se, $P = N_G(v)$, $v \in S$.*

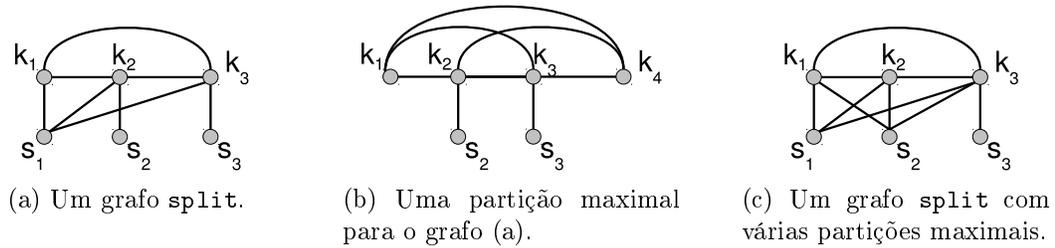
Demonstração. Considere um grafo *split* G com partição *split* (K, S) tal que K é uma clique maximal. Note que, se G for um grafo completo, $S = \emptyset$ bem como $\lambda(G) = \emptyset$.

Se $v \in S$ e $P = N_G(v)$ então existe um vértice $u \in K \setminus P$ uma vez que K é maximal. Logo, P é um uv -separador pois v não tem vizinhos em $G[V(G) \setminus P]$. Além disso, P é um uv -separador minimal uma vez que para qualquer $w \in P$, w é um vértice adjacente a ambos u e v .

Agora, considere P um separador minimal de G . Então, existe um $\{u, v\} \subset V(G)$ tal que P é um uv -separador minimal de G e uma das seguintes condições deve ser satisfeita:

- $u \in K$ e $v \in S$: Note que P deve conter todos os vértices em $N_G(v) \cap N_G(u)$, que é igual a $N_G(v)$ uma vez que $N_G(v) \subset K \setminus \{u\} \subseteq N_G(u)$. Portanto, P é um uv -separador minimal somente se $P = N_G(v)$.
- $\{u, v\} \subseteq S$: Uma vez que $N_G(u) \subset K$ e $N_G(v) \subset K$, segue que $N_G(u) \subseteq P$ ou $N_G(v) \subseteq P$. Caso contrário, existem $x \in N_G(u) \setminus P$ e $y \in N_G(v) \setminus P$ que formam um caminho (u, x, y, v) em $G[V(G) \setminus P]$. Agora, se P é um uv -separador minimal, P está contido em $N_G(v)$ ou $N_G(u)$ uma vez que estes conjuntos são uv -separadores. \square

Apresenta-se na Figura 3.5, um grafo *split* (a) com uma partição (K, S) . Note que os vizinhos de s_1 não formam um separador minimal, isto porque nesta partição, K não é maximal. O mesmo grafo é apresentado em (b) (fazendo $k_4 = s_1$), agora com uma partição (K, S) na qual K é maximal. Observa-se que, neste grafo, os separadores minimais são os vizinhos de cada vértice em S , ou seja, $\{k_2\}$ e $\{k_3\}$. Por fim, exibe-se em (c) um grafo que possui múltiplas partições (K, S) com K maximal, uma delas obtida movendo s_1 para K e outra, movendo s_2 para K . Note que, independentemente da escolha, os separadores minimais dados pelo Lema 3.4 são sempre os mesmos, $\{k_3\}$ e $\{k_1, k_2, k_3\}$.

Figura 3.5: Separadores minimais em grafos *split*.

3.4.2 Outros casos

Por fim, basta considerar um pequeno conjunto de grafos, que são permitidos como grafos quocientes de nós da árvore de decomposição modular de um grafo P_4 -carregado estendido. Para tal, considere um P_5 com vértices $V(P_5) = \{a, b, c, d, e\}$ e arestas $E(P_5) = \{\{a, b\}, \{b, c\}, \{c, d\}, \{d, e\}\}$ e um C_5 obtido adicionando a aresta $\{e, a\}$ ao P_5 . Os separadores minimais dos grafos em questão são fornecidos na Tabela 3.1.

Grafo	Separadores minimais
P_5	$\{b\}, \{c\}$ e $\{d\}$
$\overline{P_5}$	$\{a, b\}, \{a, e\}$ e $\{d, e\}$
C_5	$\{a, c\}, \{a, d\}, \{b, d\}, \{b, e\}$ e $\{c, e\}$

Tabela 3.1: Separadores minimais de P_5 , $\overline{P_5}$ e C_5 .

3.4.3 Algoritmo

Em vista dos resultados das seções anteriores, pode-se listar os separadores minimais de qualquer grafo da classe P_4 -carregada estendida. Para tal, estende-se o Algoritmo 1, a fim de computar os separadores minimais dos grafos quocientes que podem surgir nos nós da árvore de decomposição modular de um grafo desta classe. Como já é sabido, estes grafos são os grafos completos, sem arestas, *split* ou isomorfos a C_5 , P_5 ou $\overline{P_5}$. O algoritmo então identifica em qual desses casos se enquadra cada nó da árvore de decomposição modular e computa seus separadores minimais de acordo com os lemas 3.1 e 3.4.

Cada nó da ADM de um grafo da classe P_4 -carregada estendida pode ser classificado (quando for um nó do tipo vizinhança, o grafo quociente deve ser identificado como um grafo *split* ou um dos três casos simples acima descritos) em tempo total linear [31], através do reconhecimento de grafos *split* [40].

Todos os passos descritos estão representados no Algoritmo 2, o qual tem como entrada um grafo primo da classe P_4 -carregada estendida G . Note que aranhas não são tratadas

explicitamente, pois são grafos **split** (quando primas). Além disso, pseudo-aranhas não são consideradas, pois não são grafos primos e seus grafos quociente são aranhas.

Algoritmo 2 Separadores minimais para grafos quociente de grafos P_4 -carregados estendidos (sepminquo).

Entrada: Um grafo G da classe P_4 -carregada estendida que seja primo.

Saída: $\lambda(G)$ contém todos os separadores minimais de G .

Se G for isomorfo a C_5 , P_5 ou $\overline{P_5}$ **então**

 Produzir $\lambda(G)$ conforme valores pré-computados.

Senão Se G for um grafo sem arestas (quociente de grafos união) **então**

$\lambda(G) \leftarrow \{\emptyset\}$

Senão

$\lambda(G) \leftarrow \emptyset$

Se G for um grafo **split** **então**

Para cada vértice $s_i \in S$ **faça**

$\lambda(G) \leftarrow \lambda(G) \cup \{N_G(s_i)\}$

Fim Para

Senão Se G não for um grafo completo (quociente de grafos **join**) **então**

 Falha pois G não satisfaz os pré-requisitos.

Fim Se

Fim Se

3.5 Limitantes

O algoritmo apresentado na seção anterior calcula os separadores minimais de qualquer grafo da classe P_4 -carregada estendida em tempo linear, desde que o tamanho total da descrição de todos eles, assim como o número total de separadores minimais, sejam lineares no tamanho do grafo. Nesta seção, será mostrado que este fato é válido para os grafos P_4 -carregados estendidos e também serão dadas constantes que limitam, em relação ao tamanho do grafo, o tamanho e número de separadores minimais, para a classe e várias subclasses.

3.5.1 Fórmula geral

O objetivo geral desta seção é mostrar que existem constantes α e β tais que $l_G \leq \alpha n_G$ e $s_G \leq \beta m_G$, para qualquer grafo G da classe P_4 -carregada estendida. Deste modo, esta classe se assemelha ao grafos cordais, ambas com $O(n)$ separadores minimais e se diferencia dos grafos fracamente cordais, com $n + m$ separadores minimais. Note que, pelo Teorema 3.2, se G for P_4 -esparso, então estas relações valem para $\alpha = 2/3$ e $\beta = 4/3$.

Em seu trabalho, Nikolopoulos e Palios mostraram que seus limitantes para grafos P_4 -esparcos eram justos exibindo uma família de cografos que satisfazem os limites dados. Uma análise desta família mostra que, à medida que tomamos grafos maiores na sequência, estabelece-se a relação $\beta = 2\alpha$, que também vale para grafos P_4 -esparcos. A construção desta família, será estendida a outras classes de grafos na Seção 3.6.1, de forma que esta relação seja mantida.

A seguir, serão computados os limitantes para l_G e s_G para qualquer grafo G da classe P_4 -carregada estendida e também para várias de suas subclasses (P_4 -carregada, P_4 -arrumada e P_4 -leve). Serão obtidos valores distintos de α para cada uma dessas classes. Tais limitantes serão estabelecidos com auxílio da Propriedade que segue.

Propriedade 3.5. *Para um grafo G e um número real $\alpha \geq 0$, as seguintes condições são satisfeitas:*

$$s_G \leq 2\alpha m_G \quad l_G \leq \begin{cases} \alpha n_G - 1/3 & , \text{ se } G \text{ é desconexo;} \\ \alpha n_G - 2/3 & , \text{ se } G \text{ é conexo.} \end{cases}$$

Note que todo grafo satisfaz as condições da Propriedade 3.5 para algum valor positivo de α . Além disso, se um grafo satisfaz tais condições para $\alpha = x$, ele também as satisfaz para qualquer $\alpha \geq x$. A Propriedade 3.5 será usada para classificar famílias de grafos de acordo com o menor valor de α para o qual as condições da propriedade valem. Estes valores mínimos de α fornecem limitantes para l_G e s_G nos grafos de cada família.

3.5.2 Estudo de grafos

Nesta seção, serão consideradas famílias de grafos às quais um grafo G da classe P_4 -carregada estendida pode pertencer. Estas famílias estão relacionadas à caracterização da classe via decomposição modular, dada no Teorema 2.7 da Seção 2.4.4. Para cada caso serão dadas condições suficientes para que o grafo G satisfaça a Propriedade 3.5.

Embora os grafos `união` e `join` já tenham sido considerados por Nikolopoulos e Palios, tais grafos, nas classes que serão tratadas neste capítulo, podem originar-se tendo como operandos grafos que não foram considerados naquele trabalho, como pseudo-aranhas. Desta forma, os grafos `união` e `join` serão analisados de forma mais genérica.

Lema 3.6. *Se $G = \bigcup_{i=1}^p G_i$, $p \geq 2$, é uma união de grafos conexos G_i , então G satisfaz a Propriedade 3.5 para α se cada G_i satisfaz a Propriedade 3.5 para α .*

Demonstração. Seja $G = \bigcup_{i=1}^p G_i$, $p \geq 2$ uma união de grafos conexos G_i . Pelo Lema 3.1,

$$l_G = 1 + \sum_{i=1}^p l_{G_i} \leq 1 + \sum_{i=1}^p \left(\alpha n_{G_i} - \frac{2}{3} \right) = \alpha n_G + \frac{3-2p}{3} \leq \alpha n_G - \frac{1}{3}$$

$$s_G = \sum_{i=1}^p s_{G_i} \leq \sum_{i=1}^p 2\alpha m_{G_i} = 2\alpha m_G.$$

□

Lema 3.7. *Se $G = G_1 + G_2 + \dots + G_p$, $p \geq 2$, é um grafo join, então G satisfaz a Propriedade 3.5 para α se cada G_i satisfaz a Propriedade 3.5 para α .*

Demonstração. Seja $G = G_1 + G_2 + \dots + G_p$, $p \geq 2$ um grafo join. Pelo Lema 3.1,

$$l_G = \sum_{i=1}^p l_{G_i} \leq \sum_{i=1}^p \left(\alpha n_{G_i} - \frac{1}{3} \right) = \alpha n_G - \frac{p}{3} \leq \alpha n_G - \frac{2}{3}$$

$$s_G = \sum_{i=1}^p [(n_G - n_{G_i})l_{G_i} + s_{G_i}] \leq \sum_{i=1}^p [(n_G - n_{G_i})\left(\alpha n_{G_i} - \frac{1}{3}\right) + 2\alpha m_{G_i}]$$

Uma vez que $m_G = \sum_{i=1}^p \left[\frac{n_{G_i}}{2}(n_G - n_{G_i}) + m_{G_i} \right]$, para que $s_G \leq 2\alpha m_G$ é necessário

$$\sum_{i=1}^p \left[(n_G - n_{G_i})\left(\alpha n_{G_i} - \frac{1}{3}\right) + 2\alpha m_{G_i} \right] \leq 2\alpha \sum_{i=1}^p \left[\frac{n_{G_i}}{2}(n_G - n_{G_i}) + m_{G_i} \right]$$

que é válido uma vez que $\alpha n_{G_i} - \frac{1}{3} \leq \alpha n_{G_i}$, $\forall i, 1 \leq i \leq p$.

□

Pelo Teorema 3.2 os cografos satisfazem a Propriedade 3.5 para qualquer $\alpha \geq 2/3$. Embora os lemas anteriores não imponham tal restrição e todo cografo possa ser completamente decomposto pelas operações de união e join, ela surge como consequência da condição $l_G \leq \alpha n_G - 2/3$ aplicada ao cografo elementar, que é o grafo trivial (um grafo conexo de 1 vértice e sem nenhum separador).

Em seguida, considerar-se-ão as aranhas, de forma a reafirmar os resultados já conhecidos e ampliar seu escopo, permitindo que o subgrafo induzido pela parte R seja qualquer. Para tanto, a Tabela 3.2 consolida os parâmetros relevantes para uma aranha G , com partição (K, S, R) , de ambas as espécies: magra e gorda, representando por n_H , m_H , l_H e s_H os respectivos parâmetros do grafo $H = G[R]$ e considerando $k = |K| = |S|$. Alguns parâmetros recebem uma nomenclatura na tabela (como m_t e s_T , nos quais t e T referem-se a aranhas magras e gordas, respectivamente) para que suas expressões aritméticas sejam referenciadas na avaliação das pseudo-aranhas.

Espécie	Número de vértices	Número de separadores (l_G)
Ambas	$n_G = n_H + 2k$	$l_G = l_H + k$

(a) Cardinalidade.

Espécie	Número de arestas	Tamanho dos separadores (s_G)
Magra	$m_t = m_H + (k^2 + k)/2 + kn_H$	$s_t = s_H + l_H k + k$
Gorda	$m_T = m_H + 3(k^2 - k)/2 + kn_H$	$s_T = s_H + l_H k + k(k - 1)$

(b) Tamanho.

Tabela 3.2: Parâmetros para separadores minimais de aranhas.

Lema 3.8. *Seja G uma aranha com partição (K, S, R) e $H = G[R]$. Se $R \neq \emptyset$ e H satisfaz a Propriedade 3.5 para α , então G satisfaz a Propriedade 3.5 para $\max\{\alpha, 2/3\}$. Se $R = \emptyset$, então G satisfaz a Propriedade 3.5 para $\alpha = 2/3$.*

Demonstração. Seja G uma aranha com partição (K, S, R) . Note que $k \geq 2$. Pelo Lema 3.1 e uma vez que R pode ser vazio (mas $l_H \geq 0$) tem-se que $l_G = k + l_H \leq k + \alpha n_H$. Dado que G é conexo, deseja-se que $l_G \leq \alpha n_G - 2/3$.

$$k + \alpha n_H \leq \alpha(n_H + 2k) - \frac{2}{3} \iff k \leq 2\alpha k - \frac{2}{3}$$

$$\alpha \geq \frac{k + \frac{2}{3}}{2k} \text{ que é verdadeiro se } \alpha \geq \frac{2}{3}$$

Para provar que $s_G \leq 2\alpha m_G$, dois casos são analisados. O primeiro trata o caso em que G é uma aranha magra e o outro quando G é uma aranha gorda. Em ambos os casos, o limitante é obtido da hipótese sobre H e das equações dadas na Tabela 3.2.

Se G é uma aranha magra tem-se que:

$$s_G = s_H + l_H k + k \leq 2\alpha m_H + \alpha n_H k + k$$

$$s_G \leq 2\alpha m_G \text{ se } 2\alpha m_H + \alpha n_H k + k \leq 2\alpha(m_H + n_H k + \frac{k^2}{2} + \frac{k}{2})$$

$$\alpha n_H k + k \leq \alpha(2n_H k + k^2 + k)$$

$$\alpha n_H + 1 \leq \alpha(2n_H + k + 1)$$

$$\alpha \geq \frac{1}{n_H + k + 1} \text{ que é válido se } \alpha \geq 1/3.$$

Finalmente, se G é uma aranha gorda:

$$\begin{aligned}
s_G &= s_H + l_H k + k^2 - k \leq 2\alpha m_H + \alpha n_H k + k^2 - k \\
s_G \leq 2\alpha m_G \text{ se } 2\alpha m_H + \alpha n_H k + k^2 - k &\leq 2\alpha(m_H + n_H k + \frac{3k^2}{2} - \frac{3k}{2}) \\
\alpha n_H k + k^2 - k &\leq \alpha(2n_H k + 3k^2 - 3k) \\
\alpha n_H + k - 1 &\leq \alpha(2n_H + 3k - 3) \\
\text{que é verdadeiro se } k - 1 &\leq \alpha(3k - 3) \\
\text{e, conseqüentemente, se } \alpha &\geq 1/3.
\end{aligned}$$

□

Com isto, os resultados de Nikolopoulos e Palios sobre grafos P_4 -esparcos (Teorema 3.2) também podem ser concluídos dos lemas 3.6, 3.7 e 3.8, dado que todos os possíveis grafos quocientes em nós da árvore de decomposição modular de um grafo P_4 -esparso satisfazem a Propriedade 3.5 para $\alpha \geq 2/3$.

A fim de considerar a classe P_4 -carregada estendida e subclasses, que motivaram este estudo, serão consideradas a seguir as pseudo-aranhas. Pode-se caracterizar os separadores minimais de tais grafos com a aplicação direta do Teorema 3.3 sobre os separadores minimais de uma aranha, uma vez que os vértices v' e v'' da pseudo-aranha, que resultam da divisão do vértice v da aranha, formam um módulo. Além disso, a contração de tal módulo na pseudo-aranha, produz a aranha original.

Corolário 3.9. *Sejam G' uma aranha com partição (K, S, R) e G uma pseudo-aranha obtida de G' dividindo um vértice v em dois vértices v' e v'' . Então, $T \in \lambda(G)$ se, e somente se,*

- $T = T'$, onde $T' \in \lambda(G')$ e $v \notin T'$; ou
- $T = (T' \setminus \{v\}) \cup \{v', v''\}$, onde $T' \in \lambda(G')$ e $v \in T'$; ou
- $T = N_G(v') = N_G(v'')$, se $v \in K$ e v' não é adjacente a v'' .

Na Figura 3.6 são apresentadas uma aranha (a), cujos separadores minimais são $\{k_1\}$, $\{k_2\}$ e $\{k_3\}$, além de duas pseudo-aranhas derivadas desta aranha. No caso (b), os separadores minimais são idênticos aos da aranha, já que nenhum deles contém v e o separador minimal de v' e v'' é $\{k_1\}$, já considerado. Já para a pseudo-aranha em (c), o separador minimal $\{k_1\}$ da aranha, torna-se $\{v', v''\}$, pois $v = k_1$, e o grafo tem um novo $v'v''$ -separador minimal, $\{s_1, k_2, k_3\}$.

Agora, os valores de l_G e s_G para uma pseudo-aranha G com partição (K, S, R) serão calculados usando o Corolário 3.9. Seja $k = \min\{|K|, |S|\}$ e $H = G[R]$. Se G é uma

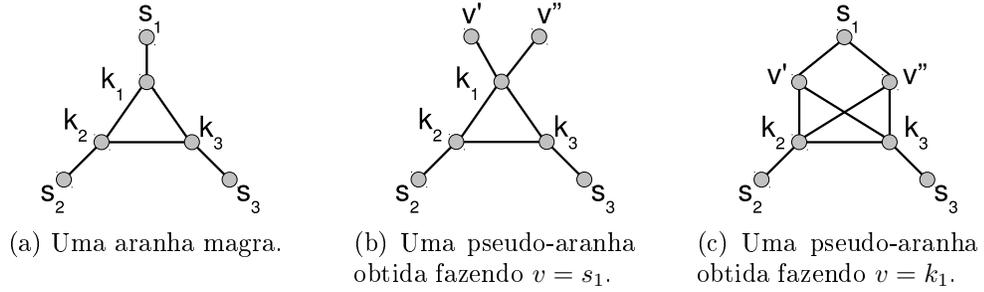


Figura 3.6: Pseudo-aranhas para exemplos de separadores minimais.

pseudo-aranha obtida de uma aranha com partição (K', S', R) pela divisão de um vértice $v \in K'$ em dois vértices não adjacentes, v' e v'' , então $l_G = l_H + k + 1$. Para os demais tipos de pseudo-aranhas, tem-se que $l_G = l_H + k$. Note também que $n_G = n_H + 2k + 1$ para qualquer pseudo-aranha. Finalmente, na Tabela 3.3 são fornecidos os valores de m_G e s_G , baseados nos valores de s_t , m_t , s_T , e m_T para aranhas, dados na Tabela 3.2.

Espécie	$\{v', v''\} \notin E(G)$		$\{v', v''\} \in E(G)$	
	$v \in S$	$v \in K$	$v \in S$	$v \in K$
Magra	$m_t + 1$	$m_t + k + n_H$	$m_t + 2$	$m_t + k + n_H + 1$
Gorda	$m_T + k - 1$	$m_T + 2k + n_H - 2$	$m_T + k$	$m_T + 2k + n_H - 1$

(a) Número de arestas em pseudo-aranhas.

Espécie	$v \in S$	$v \in K$	
	—	$\{v', v''\} \in E(G)$	$\{v', v''\} \notin E(G)$
Magra	s_t	$s_t + l_H + 1$	$s_t + l_H + k + n_H + 1$
Gorda	s_T	$s_T + l_H + k - 1$	$s_T + l_H + 3k + n_H - 3$

(b) Tamanho dos separadores minimais de pseudo-aranhas.

Tabela 3.3: Parâmetros para separadores minimais de pseudo-aranhas.

Agora, estabelecem-se os limitantes para separadores minimais de pseudo-aranhas.

Lema 3.10. *Seja G uma pseudo-aranha com partição (K, S, R) e $H = G[R]$. Se $R \neq \emptyset$ e H satisfaz a Propriedade 3.5 para α , então o grafo G satisfaz a Propriedade 3.5 para $\max\{\alpha, 11/15\}$. Se $R = \emptyset$, então G satisfaz a Propriedade 3.5 para $\alpha = 11/15$.*

Demonstração. Seja G uma pseudo-aranha com partição (K, S, R) . Lembre-se que $k \geq 2$. Portanto, pelo Corolário 3.9 e hipótese sobre H , tem-se que $l_G \leq k + l_H + 1 \leq k + \alpha n_H + 1$.

Logo, para que $l_G \leq \alpha n_G - 2/3$, é necessário que

$$\begin{aligned} k + \alpha n_H + 1 &\leq \alpha(n_H + 2k + 1) - \frac{2}{3} \\ k + 1 &\leq 2\alpha k + \alpha - \frac{2}{3} \\ \alpha &\geq \frac{k + \frac{5}{3}}{2k + 1}, \text{ que é satisfeito para } \alpha \geq \frac{11}{15}. \end{aligned}$$

Por definição de pseudo-aranhas, existem dois vértices v' e v'' em K ou em S que podem ser contraídos para obter uma aranha G' . Considere os seguintes casos:

- $\{v', v''\} \subset S$: Pelo Corolário 3.9, $s_G = s_{G'}$ e, pelo Lema 3.1, $s_G \leq 2\alpha m_{G'} < 2\alpha m_G$, para $\alpha \geq 1/3$.
- $\{v', v''\} \subset K$ e v' não é adjacente a v'' : Nesse caso, são usadas as equações dadas nas Tabelas 3.2 e 3.3.

Seja G uma pseudo-aranha magra. Portanto, $s_G = s_H + l_H(k + 1) + 2k + n_H + 1 \leq 2\alpha m_H + \alpha n_H(k + 1) + 2k + n_H + 1$ e $m_G = m_H + n_H k + n_H + (k^2 + 3k)/2$. Note que se $R = \emptyset$, H não existe e a inequação $l_H \leq \alpha n_H - 1/3$ é falsa para qualquer valor de α . Para contornar esta situação, usa-se a inequação relaxada $l_H \leq \alpha n_H$, que é válida para $R = \emptyset$ assim como para $R \neq \emptyset$. Logo, $s_G \leq 2\alpha m_G$ se

$$\begin{aligned} \alpha n_H(k + 1) + 2k + n_H + 1 &\leq \alpha(2n_H k + 2n_H + k^2 + 3k), \\ \text{que é satisfeita se (a) } \alpha n_H(k + 1) + n_H &\leq 2\alpha n_H(k + 1) \end{aligned}$$

$$\alpha \geq \frac{1}{k + 1}, \text{ verdadeira se } \alpha \geq \frac{1}{3}$$

$$\text{e se (b) } 2k + 1 \leq \alpha(k^2 + 3k)$$

$$\alpha \geq \frac{2}{k + 3} + \frac{1}{k^2 + 3k}, \text{ verdadeira se } \alpha \geq \frac{1}{2}.$$

Caso contrário, G é uma pseudo-aranha gorda e, portanto, $s_G = s_H + l_H(k + 1) + k^2 + 2k + n_H - 3 \leq 2\alpha m_H + \alpha n_H(k + 1) + k^2 + 2k + n_H - 3$ e $m_G = m_H + n_H k + n_H + (3k^2 + k)/2 - 2$. Usando-se a mesma inequação relaxada que do caso anterior, tem-se que $s_G \leq 2\alpha m_G$ se

$$\alpha n_H(k + 1) + k^2 + 2k + n_H - 3 \leq \alpha(2n_H k + 2n_H + 3k^2 + k - 4),$$

$$\text{que é satisfeita se (a) } \alpha n_H(k + 1) + n_H \leq 2\alpha n_H(k + 1) \rightarrow \alpha \geq \frac{1}{3}$$

$$\text{e (b) } k^2 + 2k - 3 \leq \alpha(3k^2 + k - 4) \rightarrow \alpha \geq \frac{1}{2}$$

- $\{v', v''\} \subset K$ e v' é adjacente a v'' : Seja G^* a pseudo-aranha obtida de G removendo a aresta $\{v', v''\}$. Portanto, pelo Corolário 3.9 e resultados anteriores, segue que $s(G) < s(G^*) \leq 2\alpha m_{G^*} < 2\alpha m_G$ para $\alpha \geq \max\{\alpha, 1/2\}$. \square

Este resultado sugere que um grafo G que contenha, como subgrafo induzido, uma pseudo-aranha possui limitantes para l_G e s_G maiores que os limitantes válidos para grafos P_4 -esparcos. O mesmo ocorre para os grafos **split**, conforme o resultado a seguir.

Lema 3.11. *Um grafo **split** G conexo e com pelo menos dois vértices, satisfaz $l_G \leq n_G - 2$ e $s_G \leq m_G$.*

Demonstração. Seja G um grafo **split** com partição **split** (K, S) , tal que K é maximal. Diretamente do Lema 3.4, tem-se que $l_G \leq |S|$. Além disso, uma vez que K é maximal e o grafo tem ao menos dois vértices, $|K| \geq 2$ e $|S| \leq n_G - 2$. Cada separador minimal de G é um conjunto $P = N_G(v)$, $v \in S$. Portanto, cada vértice $u \in P$ pode ser associado a uma aresta distinta $\{u, v\}$ em G . Logo, $s_G \leq m_G$. \square

Como consequência direta do Lema 3.11, pode-se estabelecer outro limitante para os grafos **split**.

Corolário 3.12. *Um grafo **split** satisfaz a Propriedade 3.5 para qualquer $\alpha \geq 1$.*

Todavia, em grafos da classe P_4 -carregada estendida, permite-se que nós da árvore de decomposição modular correspondam a grafos **pseudo-split**, obtidos a partir de grafos **split**, nos quais alguns vértices podem ser substituídos por outro grafo da classe P_4 -carregada estendida (veja a definição na Seção 2.4.4). Neste caso, se um nó da árvore representa um grafo **pseudo-split** G , então o grafo quociente deste nó será um grafo **split** primo, G' . Logo, $|R(G')| \leq 1$, pois $R(G')$ é um módulo de G' . Dado o grafo quociente G' , a pseudo-aranha G pode ser produzida a partir de G' expandindo vértices em $K(G')$ para grafos completos, vértices em $S(G')$ para grafos sem arestas e, por fim, o vértice em $R(G')$ para um grafo da classe P_4 -carregada estendida (pois a classe é hereditária). Note que, ao expandir apenas os vértices em $S(G')$ e $K(G')$, o grafo resultante ainda é um grafo **split**, G'' , tal que $|R(G'')| \leq 1$. Logo, basta considerar a expansão de um único vértice em um grafo **split** para tratar este caso, feita no lema a seguir.

Lema 3.13. *Sejam G um grafo, M um módulo de G , $2 \leq |M| < n_G$, $G' = G \setminus \{M\}$ e v um vértice representante de M . Se G' é um grafo **split** conexo com partição **split** (K', S') tal que $v \in K'$, $N_G(v) \cap S' = \emptyset$, então G satisfaz a Propriedade 3.5 para α se $G \setminus M$ também satisfaz esse fato para α , com $\alpha \geq 1$.*

Demonstração. Seja $H = G[M]$. Note que, por construção de G' ,

$$n_G = n_{G'} + n_H - 1 \quad \text{e} \quad m_G = m_{G'} + m_H + (|K'| - 1)(n_H - 1).$$

Por hipótese, (K', S') é uma partição **split** de G' . Como v não tem vizinhos em S' , então K' é maximal e, pelo Lema 3.4, v não participa de qualquer separador minimal de G' . Logo, pelo Teorema 3.3, tem-se que

$$l_G = l_{G'} + l_H \quad \text{e} \quad s_G = s_{G'} + s_H + l_H(|K'| - 1).$$

Como o grafo G' é conexo e $n_{G'} \geq 2$, segue, pelo Lema 3.11, que $l_{G'} \leq n_{G'} - 2$ e $s_{G'} \leq m_{G'}$.

Primeiramente, será provada a restrição para l_G , ou seja, $l_G \leq \alpha n_G - 2/3$.

$$\begin{aligned} l_{G'} + l_H &\leq \alpha(n_{G'} + n_H - 1) - 2/3 \\ \text{pela hipótese sobre } H, l_H &\leq \alpha n_H - 1/3 \\ \text{portanto, } l_{G'} &\leq \alpha(n_{G'} - 1) - 1/3. \end{aligned}$$

Uma vez que $l_{G'} \leq n_{G'} - 2$, basta que $0 \leq \alpha(n_{G'} - 1) - n_{G'} + 5/3$.

$$\text{Ou seja, } \alpha \geq \frac{n_{G'} - 5/3}{n_{G'} - 1} \text{ que é verdadeiro se } \alpha \geq 1.$$

Agora, resta provar que $s_G \leq 2\alpha m_G$.

$$\begin{aligned} s_{G'} + s_H + l_H(|K'| - 1) &\leq 2\alpha(m_{G'} + m_H + (n_H - 1)(|K'| - 1)) \\ \text{pela hipótese sobre } H, s_H &\leq 2\alpha m_H \\ \text{portanto, } s_{G'} + l_H(|K'| - 1) &\leq 2\alpha(m_{G'} + (n_H - 1)(|K'| - 1)) \\ \text{novamente pela hipótese sobre } H, l_H(|K'| - 1) &\leq (\alpha n_H - 1/3)(|K'| - 1) \\ \text{logo, } s_{G'} &\leq \alpha(2m_{G'} + (n_H - 2)(|K'| - 1)) + |K'|/3 - 1/3 \\ \text{como } G' \text{ é um grafo } \text{split}, s_{G'} &\leq m_{G'} \\ \text{e basta que } 0 &\leq \alpha(2m_{G'} + (n_H - 2)(|K'| - 1)) + |K'|/3 - 1/3 - m_{G'}. \end{aligned}$$

$$\text{Então, } \alpha \geq \frac{m_{G'} - (|K'| - 1)/3}{2m_{G'} + (n_H - 2)(|K'| - 1)}$$

e assim é suficiente que $\alpha \geq 1/2$.

(Note que $m_{G'} \geq 1$ uma vez que $n_{G'} \geq 2$ e G' é um grafo conexo por hipótese.) □

Nesta seção, foram dados limitantes para uma série de famílias de grafos, baseando-se nos requisitos definidos pela Propriedade 3.5. A Tabela 3.4 resume os resultados para consulta e comparação dos valores mínimos para α permitidos em cada caso. Note que

os resultados são válidos para grafos união e join, desde que cada operando satisfaça a proposição para o valor de α em questão. Ainda, para aranhas e pseudo-aranhas, é necessário que o subgrafo induzido pela cabeça (parte R) também satisfaça a proposição para o mesmo α .

Grafo	Menor α	Referência (Lema)
União	-	3.6
Join	-	3.7
Aranha	2/3	3.8
Pseudo-aranha	11/15	3.10
Split modificado	1	3.13

Tabela 3.4: Resumo da Propriedade 3.5 para algumas famílias de grafos.

Por fim, a decomposição modular de um grafo P_4 -carregado estendido pode conter nós do tipo vizinhança que correspondem a um conjunto finito de subgrafos induzidos. Tais grafos foram listados na Seção 3.4.2 juntamente com seus separadores minimais. Desta forma, calculam-se os valores mínimos de α que satisfazem a Propriedade 3.5 para estes grafos na Tabela 3.5.

Grafo (G)	n_G	m_G	l_G	s_G	Menor α
K_1	1	0	0	0	2/3
P_5	5	4	3	3	11/15
$\overline{P_5}$	5	6	3	6	11/15
C_5	5	5	5	10	17/15

Tabela 3.5: Propriedade 3.5 para alguns grafos.

3.5.3 Estudo de classes

Após analisar as famílias de grafos na Seção 3.5.2, tem-se recursos suficientes para estabelecer limitantes para os separadores minimais de quaisquer grafos na classe P_4 -carregada estendida. Além disso, serão dados limitantes melhores para algumas de suas subclasses.

Em todas as classes novas consideradas, a decomposição modular de um grafo da classe pode conter nós correspondentes a grafos união, join e aranhas, assim como nos grafos P_4 -esparsos. Além disso, os nós podem corresponder a pseudo-aranhas e, conforme o caso, a grafos `split` ou a um C_5 , P_5 ou seu complemento ($\overline{P_5}$). Cada um destes grafos satisfaz a Propriedade 3.5 para valores mínimos de α resumidos na Tabela 3.4. Como consequência, em cada classe, valerá a maior restrição para α entre todas as possíveis. O

resultado está dado na Tabela 3.6, que identifica o tipo de grafo permitido em cada classe e o menor limitante α válido.

Classe	Pseudo-aranha	P_5 e $\overline{P_5}$	C_5	Split	α
P_4 -leve	•	•			11/15
P_4 -carregada	•	•		•	1
P_4 -arrumada	•	•	•		17/15
P_4 -carregada estendida	•	•	•	•	17/15

Tabela 3.6: Limitantes dos separadores minimais das classes de interesse.

Assim, prova-se o principal resultado deste capítulo.

Teorema 3.14. *Se G é um grafo P_4 -leve, então $l_G \leq 11n_G/15$ e $s_G \leq 22m_G/15$. Se G é um grafo P_4 -carregado, então $l_G \leq n_G$ e $s_G \leq 2m_G$. Se G é um grafo P_4 -arrumado ou um P_4 -carregado estendido, então $l_G \leq 17n_G/15$ e $s_G \leq 34m_G/15$.*

Demonstração. É suficiente mostrar que a Propriedade 3.5 é válida para grafos P_4 -leves com $\alpha = 11/15$, para grafos P_4 -carregados com $\alpha = 1$ e para P_4 -arrumados e P_4 -carregados estendidos com $\alpha = 17/15$.

A prova segue por indução no número de vértices do grafo, na qual a base é um entre quatro grafos conexos especiais, cujos separadores minimais obedecem aos parâmetros expostos na Tabela 3.5, que também fornece os menores valores de α para que tais grafos satisfaçam a Propriedade 3.5.

Dado um grafo de uma das classes consideradas, ou ele é um dos grafos da base da indução, ou é um grafo união, join, aranha, pseudo-aranha ou split (dependendo da classe considerada). Em cada caso, os valores mínimos de α para cada um destes grafos foi demonstrado nos Lemas 3.6, 3.7, 3.8 e 3.10 e do Corolário 3.12.

Para cada classe, toma-se o maior valor mínimo exigido para α entre os possíveis grafos permitidos, resumidos na Tabela 3.6. Assim, nos casos de grafos união e join, os operandos, por hipótese de indução satisfazem a Propriedade 3.5 para o valor de α desejado. Nos casos de pseudo-aranhas e aranhas, o grafo induzido pela cabeça, se houver, também atenderá à propriedade desejada. \square

3.6 Limitantes justos

Os limites apresentados na seção anterior para a classe P_4 -carregada estendida e algumas de suas subclasses são, de fato, justos no sentido que não há melhor razão para limitar l_G e s_G em relação a n_G e m_G , respectivamente, que seja válida para todos os grafos em cada classe. Isto será provado nesta seção, fazendo uso de uma construção já usada por Nikolopoulos e Palios.

3.6.1 Construção padrão

A família da construção padrão é formada por uma sequência infinita de grafos, na qual o primeiro grafo G_0 é qualquer grafo conexo dado, $G_1 = G_0 \cup G_0$, $G_2 = G_1 + G_1$ e assim sucessivamente, cada grafo sendo, alternadamente, a união ou join entre duas cópias do grafo anterior na sequência. Nikolopoulos e Palios utilizaram esta construção fazendo G_0 ser o grafo trivial (K_1).

A Figura 3.7 ilustra a construção padrão. No item (a), está representado o grafo G_2 quando G_0 é o grafo trivial, de forma que este é um grafo da construção original de Nikolopoulos e Palios. Já no item (b), está representado o mesmo grafo G_2 quando G_0 é formado por dois vértices não-adjacentes. Por fim, o item (c) ilustra a árvore de decomposição modular de um grafo da sequência. Nas folhas desta árvore deve-se incluir a árvore de decomposição modular do grafo G_0 escolhido.

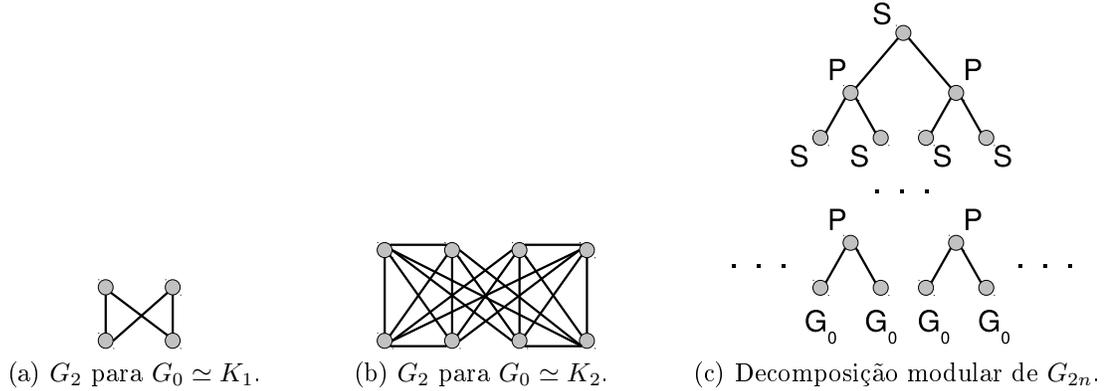


Figura 3.7: Construção padrão para grafos com limitantes justos.

No caso mais genérico em que considera-se um grafo G_0 conexo qualquer, conforme se avança na sequência, as razões l_{G_i}/n_{G_i} e s_{G_i}/m_{G_i} tendem a limites que dependem apenas dos parâmetros do grafo G_0 , base da sequência. Fato este dado pelo Lema 3.15, no qual simplifica-se a notação da seguinte forma: $n_i = n_{G_i}$, $m_i = m_{G_i}$, $l_i = l_{G_i}$ e $s_i = s_{G_i}$.

Lema 3.15. *Dado um grafo base G_0 para produzir a sequência infinita de grafos definidos acima, as razões l_i/n_i e s_i/m_i tornam-se tão próximas de $(l_0 + 2/3)/n_0$ e $2(l_0 + 2/3)/n_0$ quanto se queira, respectivamente, usando valores crescentes de i .*

Demonstração. Como mostrado no Lema 3.1, pode-se inferir que:

$$l_i = 2l_{i-1}, \text{ para } i = 0 \pmod{2} \text{ e } l_i = 2l_{i-1} + 1, \text{ para } i = 1 \pmod{2},$$

o que, para $i = 0 \pmod{2}$ e grande o suficiente, leva a:

$$\begin{aligned} l_i &= 2l_{i-1} = 4l_{i-2} + 2 = 4(4l_{i-4} + 2) + 2 = \dots \\ &= 2^i l_0 + 2 \sum_{j=0}^{i/2-1} 4^j = 2^i \left(l_0 + \frac{2}{3} \right) - \frac{2}{3}. \end{aligned}$$

Agora, uma vez que $n_i = n_0 \cdot 2^i$, tem-se que, para $i = 0 \pmod{2}$, $i \geq 2$:

$$\lim_{i \rightarrow \infty} \frac{l_i}{n_i} = \lim_{i \rightarrow \infty} \left(\frac{l_0 + \frac{2}{3}}{n_0} - \frac{\frac{2}{3}}{2^i n_0} \right) = \frac{l_0 + \frac{2}{3}}{n_0}.$$

Considere, agora, a relação entre s_i e m_i . Pelo Lema 3.1:

$$s_i = 2s_{i-1}, \text{ para } i = 1 \pmod{2} \text{ e } s_i = 2s_{i-1} + 2l_{i-1}n_{i-1}, \text{ para } i = 0 \pmod{2}.$$

Para $i = 0 \pmod{2}$, $i \geq 2$, s_i pode se escrito como

$$s_i = 2s_{i-1} + 2l_{i-1}n_{i-1} = 4s_{i-2} + 2 \frac{l_i n_i}{2} = 4s_{i-2} + \frac{l_i n_i}{2}.$$

Portanto, para $i = 0 \pmod{2}$ e grande o suficiente, segue que

$$s_i = 4 \left(4s_{i-4} + \frac{l_{i-2} n_{i-2}}{2} \right) + \frac{l_i n_i}{2} = 4^2 s_{i-4} + 4 \frac{l_{i-2} n_{i-2}}{2} + \frac{l_i n_i}{2}.$$

$$\text{Generalizando, } s_i = 2^i s_0 + \sum_{j=1}^{i/2} \frac{4^{i/2-j} l_{2j} n_{2j}}{2} = 2^i s_0 + \frac{2^i}{2} \sum_{j=1}^{i/2} \frac{l_{2j} n_{2j}}{4^j}$$

$$\text{no entanto, } \frac{l_{2j} n_{2j}}{4^j} = \frac{(4^j (l_0 + \frac{2}{3}) - \frac{2}{3}) 4^j n_0}{4^j} = 4^j n_0 \left(l_0 + \frac{2}{3} \right) - \frac{2n_0}{3}$$

$$\text{e } \sum_{j=1}^{i/2} \frac{l_{2j} n_{2j}}{4^j} = \sum_{j=1}^{i/2} 4^j n_0 \left(l_0 + \frac{2}{3} \right) - \sum_{j=1}^{i/2} \frac{2n_0}{3} = 4n_0 \left(l_0 + \frac{2}{3} \right) \frac{2^i - 1}{3} - \frac{in_0}{3},$$

$$\text{os quais implicam que } s_i = 2^i \left(s_0 + 2n_0 \left(l_0 + \frac{2}{3} \right) \frac{2^i - 1}{3} - \frac{in_0}{6} \right).$$

Do Lema 3.1 tem-se que

$$m_i = 2m_{i-1}, \text{ para } i = 1 \pmod{2} \text{ e } m_i = 2m_{i-1} + n_{i-1}^2, \text{ para } i = 0 \pmod{2}.$$

Portanto, para $i = 0 \pmod{2}$, $i \geq 2$, segue que

$$\begin{aligned} m_i &= 2m_{i-1} + n_{i-1}^2 = 4m_{i-2} + n_{i-1}^2 = 4(m_{i-2} + n_{i-2}^2) \\ &= 4(4(m_{i-4} + n_{i-4}^2) + n_{i-2}^2) = 4^2 m_{i-4} + 4^2 n_{i-4}^2 + 4n_{i-2}^2 \\ &= 2^i m_0 + \sum_{j=0}^{i/2-1} 4^{i/2-j} n_{2j}^2 = 2^i m_0 + \sum_{j=0}^{i/2-1} 4^{i/2-j} (2^{2j} n_0)^2 \\ &= 2^i m_0 + n_0^2 \sum_{j=0}^{i/2-1} 4^{j+i/2} = 2^i m_0 + 2^i n_0^2 \frac{2^i - 1}{3}. \end{aligned}$$

Finalmente, a razão s_i/m_i pode ser estabelecida:

$$\begin{aligned} \frac{s_i}{m_i} &= \frac{2^i(s_0 + 2n_0(l_0 + \frac{2}{3})\frac{2^i-1}{3} - \frac{in_0}{6})}{2^i m_0 + 2^i n_0^2 \frac{2^i-1}{3}} = \frac{s_0 + 2n_0(l_0 + \frac{2}{3})\frac{2^i-1}{3} - \frac{in_0}{6}}{m_0 + n_0^2 \frac{2^i-1}{3}} \\ &= \frac{6s_0 + 4n_0(l_0 + \frac{2}{3})(2^i - 1) - in_0}{6m_0 + 2n_0^2(2^i - 1)}. \end{aligned}$$

Logo, obtém-se

$$\lim_{i \rightarrow \infty} \frac{s_i}{m_i} = \frac{2(l_0 + \frac{2}{3})}{n_0}.$$

□

3.6.2 Estudo das subclasses

A partir da construção padrão estudada na Seção 3.6.1, serão exibidos grafos cujos separadores minimais se aproximam dos limitantes dados na Seção 3.5.3 tanto quanto se desejar. Desta forma, nenhuma relação menor que as exibidas é válida para todos os grafos de cada classe considerada.

Teorema 3.16. *Os limitantes dados no Teorema 3.14 são justos para cada classe de grafos considerada.*

Demonstração. Tomando $G_0 = P_5$, cada G_i seria um grafo P_4 -leve. Já considerando $G_0 = C_5$, cada G_i seria um grafo P_4 -arrumado.

Pelo Lema 3.15, se $G_0 = P_5$, existem grafos P_4 -leves cujas razões l_i/n_i e s_i/m_i são as mais próximas a $11/15$ e $22/15$, respectivamente, quanto se queira. Da mesma forma, usando-se $G_0 = C_5$ pode-se produzir grafos P_4 -arrumados, cujas razões se aproximam de $17/15$ e $34/15$.

Resta mostrar que existe uma família de grafos P_4 -carregados para os quais a razão l_G/n_G se aproxima de 1 enquanto s_G/m_G , de 2. Para tanto, considera-se a seguinte sequência de grafos **split**:

Seja H_j um grafo **split** com uma clique $K = \{k_1, k_2, \dots, k_j\}$. Considere $\mathcal{C} = \{Y \subset K : Y \neq \emptyset\}$ e para cada $X \in \mathcal{C}$, cria-se um vértice v_X em S tal que $N_{H_j}(v_X) = X$. Por construção e pelo Lema 3.11, $\lambda(H_j) = \mathcal{C}$.

Observe que $l_{H_j} = 2^j - 2$ e $s_{H_j} = j2^{j-1} - j$, enquanto que $n_{H_j} = 2^j + j - 2$ e $m_{H_j} = j2^{j-1} - j + j(j-1)/2$. Isto permite mostrar que

$$\lim_{j \rightarrow \infty} \frac{l_{H_j}}{n_{H_j}} = \lim_{j \rightarrow \infty} \frac{2^j - 2}{2^j + j - 2} = 1 \text{ e } \lim_{j \rightarrow \infty} \frac{s_{H_j}}{m_{H_j}} = \lim_{j \rightarrow \infty} \frac{j2^{j-1} - j}{j2^{j-1} + \frac{j(j-1)}{2}} = 1.$$

Agora, se for usado como G_0 um grafo *split* H_j nesta sequência, serão produzidos grafos G_i tais que l_i/n_i tende a $(l_{H_j} + 2/3)/n_{H_j}$ e s_i/m_i a $2(l_{H_j} + 2/3)/n_{H_j}$. Usando valores crescentes de j para H_j ,

$$\lim_{j \rightarrow \infty} \frac{l_{H_j} + \frac{2}{3}}{n_{H_j}} = 1 + \frac{\frac{2}{3}}{2^j + j - 2} = 1 \text{ e } \lim_{j \rightarrow \infty} 2 \frac{l_{H_j} + \frac{2}{3}}{n_{H_j}} = 2 + \frac{\frac{4}{3}}{2^j + j - 2} = 2.$$

□

3.7 Conclusão

O problema da listagem dos separadores minimais foi abordado, dissertando-se sobre seus aspectos e informações sobre o caso geral. Também foi descrito um algoritmo eficiente para resolver o problema na classe P_4 -esparsa, apresentado por Nikolopoulos e Palios.

Em seguida, foi exposta uma generalização daquela técnica, fruto de nossa contribuição, fazendo uso direto da decomposição modular e dos grafos quociente para obter os separadores minimais de um grafo qualquer. Então, aplicou-se a técnica em um conjunto de superclasses dos grafos P_4 -esparsos, dado pelas classes P_4 -leve, P_4 -arrumada, P_4 -carregada e P_4 -carregada estendida.

Para cada uma das classes consideradas foram dados limitantes para o número e tamanho total dos separadores minimais dos grafos que elas contêm. Em todos os casos, o número de separadores minimais do grafo é limitado linearmente pelo número de vértices do grafo e o tamanho total destes separadores, no tamanho do grafo. Desta forma, o algoritmo baseado na técnica, lista todos os separadores minimais de um grafo de tais classes em tempo linear.

Embora sejam todos lineares, os limitantes para algumas classes diferem na constante. Além disso, demonstrou-se que os limitantes são justos, exibindo famílias de grafos de cada uma das classes, tais que seus separadores minimais se aproximam indefinidamente dos limites estabelecidos.

Uma versão preliminar dos resultados obtidos foi apresentada no *V Latin-American Algorithms, Graphs and Optimization Symposium*, realizado em Gramado em novembro de 2009 e foram publicados, na forma de resumo estendido [63]. O trabalho completo foi redigido na forma de um Relatório Técnico do Instituto de Computação da UNICAMP [64]. Além disso, foi submetido a um periódico internacional e está em fase de revisão.

Capítulo 4

Empacotamento de Cliques de Grafos P_4 -arrumados

Neste capítulo, será descrita uma solução do Problema de Empacotamento de Cliques em grafos pertencentes a classe P_4 -arrumada. O algoritmo proposto adiciona a classe P_4 -arrumada às poucas classes de grafos para as quais sabe-se resolver o problema em tempo polinomial. Este fato é interessante pois sabe-se que o problema é NP-Difícil mesmo quando restrito a algumas classes de grafos.

4.1 Introdução

O *Problema do Empacotamento de Cliques* (PEC) é um problema de otimização combinatoria no qual deseja-se conhecer qual é o maior número de cliques duas a duas disjuntas e cada uma de cardinalidade r , que existem em um grafo dado.

Na Figura 4.1 pode-se ver soluções do PEC em um mesmo grafo para três casos, quando $r = 2$ (a), quando $r = 3$ (b) e quando $r = 4$ (c).

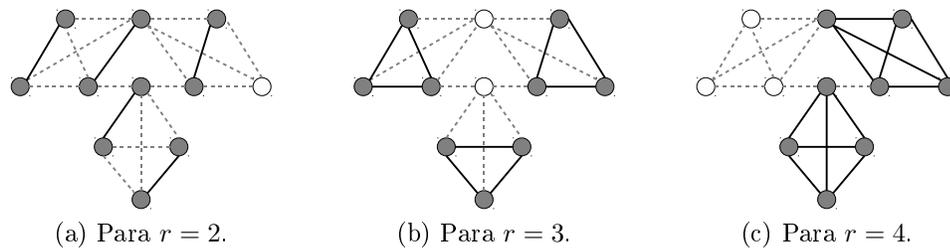


Figura 4.1: Exemplos de soluções do PEC.

Este problema pode ser considerado uma especialização do Problema de Empacotamento de Subgrafos em Grafos. Neste problema deseja-se encontrar subgrafos de um dado

grafo que sejam disjuntos nos vértices e que cada um deles seja isomorfo a um grafo de um conjunto pré-definido \mathcal{F} . O objetivo a atingir pode variar, sendo comuns a maximização do número de vértices ou arestas dos subgrafos escolhidos. Para reduzir o PEC a este problema mais genérico, basta considerar $\mathcal{F} = \{K_r\}$.

O PEC também é conhecido como Problema do Emparelhamento Multidimensional, uma vez que pode-se tomá-lo como uma extensão do clássico Problema do Emparelhamento Máximo (que corresponde ao PEC quando $r = 2$), para o qual são conhecidos vários algoritmos polinomiais. Entretanto, como consequência de um resultado em Kirkpatrick e Hell [49] sobre a complexidade do Problema de Empacotamento de Subgrafos em Grafos, o PEC é NP-Difícil para grafos arbitrários e $r \geq 3$.

Alguns resultados recentes limitam o escopo em que o problema é tratável, mostrando que ele continua NP-Difícil mesmo quando o grafo de entrada é restringido a algumas classes de grafos. Entre as classes com resultados deste tipo estão, para qualquer $r \geq 3$, os grafos cordais, totais e linha; para $r = 3$ ou $r = 4$, os grafos planares e, para $r \geq 4$, os grafos `split` (resultados de Guruswami et al. [36]) e os grafos planares cúbicos (provado em 2008 por Cerioli et al. [15]).

Na vertente oposta, são raros os casos em que foram publicados algoritmos polinomiais para o PEC restrito a classes de grafos. Parece haver apenas um para valores arbitrários de r – tratando os cografos (no mesmo trabalho de Guruswami et al. [36]). Já para o caso especial em que $r = 3$, há um algoritmo para grafos indiferença [54].

Um caso especial do PEC merece nota: quando deseja-se saber se é possível particionar os vértices do grafo em cliques de tamanho r . Esta variante corresponde a um emparelhamento multidimensional perfeito e existem algoritmos polinomiais que a resolvem para classes de grafos nas quais o PEC é NP-Difícil. Entre tais classes estão os grafos cordais e fortemente cordais [24] e os grafos `split` [36]. Claramente, caso o grafo não possua uma partição de vértices deste tipo, estes algoritmos não fornecem uma solução para o PEC. Além disso, esta variante do PEC é um problema NP-Completo para grafos em geral quando $r \geq 3$ [50, 49].

Dada a dificuldade do problema, algoritmos aproximados também foram apresentados para casos especiais do PEC e problemas relacionados. Quando $r = 3$, existe um algoritmo 1,2-aproximado para o PEC de grafos com grau máximo menor que 5 [54]. Também há um algoritmo 2-aproximado para o Problema do Empacotamento de Subgrafos em Grafos quando $\mathcal{F} = \{K_2, K_3, \dots, K_t\}$ e outros algoritmos com melhores aproximações para $t \in \{3, 4, 5\}$ [16].

Com o objetivo de apresentar um algoritmo polinomial para o PEC em grafos da classe P_4 -arrumada, na Seção 4.2 será descrito o algoritmo polinomial para cografos de Guruswami et al., na Seção 4.3 serão detalhadas as extensões necessárias para contemplar a classe P_4 -arrumada e será apresentado o algoritmo para a mesma e, por fim, as

conclusões deste trabalho estarão na Seção 4.4.

Durante as análises de complexidade, se G é o grafo de entrada para o problema, n e m representarão, respectivamente, $|V(G)|$ e $|E(G)|$.

4.2 Resultados para cografos

Nesta seção, serão descritos os resultados apresentados para o PEC em cografos por Guruswami et al. [36]. Posteriormente, estes resultados serão estendidos para superclasses dos cografos.

Como visto na Seção 2.4.1, todo cografo pode ser decomposto recursivamente em cografos menores (com menos vértices) por apenas duas operações, num processo eficiente. Este fato é um convite para algoritmos recursivos nesta classe, que precisam apenas construir soluções a partir de soluções menores, considerando apenas dois casos: grafos união e `join` (veja a Seção 2.1). Conhecidas as soluções do PEC para grafos de uma família, é fácil combinar tais soluções para resolver o PEC em grafos união dos elementos da família. Todavia, o mesmo não é verdade para grafos `join`. Desta forma, para resolver o PEC em cografos, define-se um problema mais genérico, mas que pode ser mais facilmente resolvido recursivamente tanto em grafos união como em grafos `join`.

Um conjunto P de cliques de um grafo G , duas a duas disjuntas, será denominado um *empacotamento do tipo* (n_2, \dots, n_r) se, para cada inteiro i , $2 \leq i \leq r$, P contiver exatamente n_i cliques de cardinalidade i e, além disso, não contiver cliques unitárias ou de tamanho maior que r . Na Figura 4.2, pode-se ver dois empacotamentos de cliques de tipos distintos para o mesmo grafo.

Dados um grafo G e inteiros não negativos n_3, n_4, \dots, n_r , o problema a ser resolvido é o seguinte: determinar o maior inteiro não negativo n_2 tal que existe um empacotamento do tipo (n_2, \dots, n_r) para G . A solução deste problema será representada por uma função $f(G, n_3, \dots, n_r)$. Caso G não admita um empacotamento do tipo $(0, n_3, \dots, n_r)$, o problema não tem solução e a função f não está definida para tais argumentos. Na Figura 4.2b, $f(G, 1, 1) = 2$ já que não restaram vértices no grafo para empacotar novas cliques de cardinalidade 2.

Nota-se que, se $f(G, n_3, \dots, n_r) = j$, então existe um empacotamento do tipo (j, n_3, \dots, n_r) para G , mas não existe um do tipo $(j + 1, n_3, \dots, n_r)$. Observe, também, que computar $f(G, 0, 0, \dots, 0)$ corresponde ao emparelhamento máximo de G (o maior número de K_2 disjuntos no grafo).

Observe que para computar f não basta encontrar um empacotamento P do tipo $(0, n_3, \dots, n_r)$ em G , remover de G os vértices correspondentes às cliques em P (juntamente com as arestas incidentes a tais vértices) e computar o emparelhamento máximo no grafo resultante. A falha de tal técnica é que podem existir vários empacotamentos

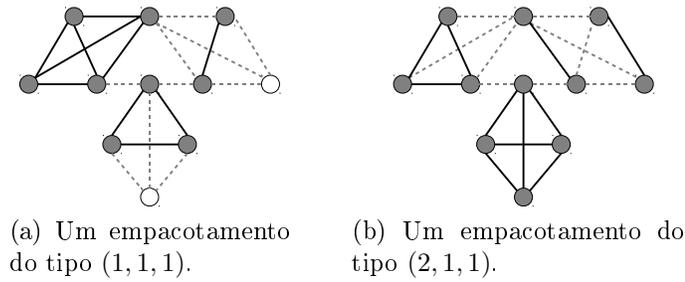


Figura 4.2: Empacotamentos de cliques com $n_3 = 1$ e $n_4 = 1$.

deste tipo e, para cada um, o emparelhamento máximo obtido pode ter tamanho distinto.

Claramente, o PEC pode ser resolvido através da função f computando o maior inteiro k , $1 \leq k \leq n/r$, tal que $f(G, 0, 0, \dots, k)$ está definida. Deste modo, resolver o PEC em cografos equivale a computar recursivamente $f(G, n_3, \dots, n_r)$ quando G é um grafo união ou join.

4.2.1 Grafos união

Suponha que G seja a união disjunta de p grafos, G_1, G_2, \dots, G_p . Então cada clique de G estará completamente contida em um dos grafos G_i , $1 \leq i \leq p$. A função $f(G, n_3, \dots, n_r)$ pode ser computada para cada possível distribuição das n_l , $3 \leq l \leq r$, cliques de G nos grafos G_i .

Na Figura 4.3 pode-se observar as duas possíveis maneiras de empacotar um triângulo (K_3) em um grafo união com dois componentes conexos, a fim de computar $f(G, 1)$. Em cada uma delas, o exemplo reflete o maior número de K_2 que podem ser empacotados juntamente com o K_3 .

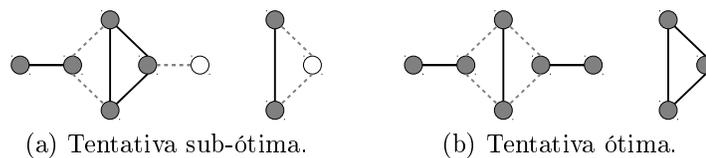


Figura 4.3: Tentativas de empacotamento de cliques em um grafo união.

Para facilitar a análise, observa-se que a união disjunta de grafos é uma operação associativa, de forma que $G = G_1 \cup (G_2 \cup (\dots G_p))$, ou seja, G pode ser obtido aplicando-se $p - 1$ operações de união disjunta com dois operandos cada. Desta forma, apenas o caso

em que $G = G_1 \cup G_2$ será considerado, cuja solução é dada por

$$f(G, n_3, \dots, n_r) = \max\{f(G_1, n'_3, \dots, n'_r) + f(G_2, n''_3, \dots, n''_r) : \\ n'_i, n''_i \in \mathcal{Z}^+ \text{ e } n'_i + n''_i = n_i \text{ para todo } 3 \leq i \leq r\}$$

e avaliada apenas quando as duas chamadas recursivas estiverem definidas.

Note que o número total de combinações avaliadas por este método é $\prod_{i=3}^r (1 + n_i)$. Como $n_i \leq n/i$, são consideradas $O(n^{r-2})$ possibilidades.

4.2.2 Grafos join

Considere um grafo G produto da operação `join` entre p grafos, G_1, G_2, \dots, G_p . Neste caso, uma clique de G pode conter vértices de mais de um dos grafos G_i . De fato, a união de até p cliques, cada uma contida em um grafo G_i distinto, forma uma clique de G .

Assim como para grafos união, a operação que forma os grafos `join` é associativa e a análise será resumida ao caso em que G é o grafo `join` de G_1 e G_2 . Note que, os vértices de cada clique do empacotamento de G podem estar particionados entre G_1 e G_2 de diversas maneiras, enquanto que, para um grafo união, $G_1 \cup G_2$, existem apenas duas maneiras – ou a clique está contida em G_1 ou em G_2 . Isto torna o cômputo de $f(G, n_3, \dots, n_r)$ mais complexo para grafos `join`.

Cada clique de um empacotamento P do tipo (n_2, \dots, n_r) para G terá parte de seus vértices em G_1 e parte em G_2 (uma das partes pode ser vazia). Para cada $i, 1 \leq i \leq r$, das n_i cliques em P , $n_{i,j}$ representará o número destas cliques com j vértices em G_1 e $i - j$ vértices em G_2 . Note que $n_{1,0}$ e $n_{1,1}$ são sempre nulos pois P não contém cliques unitárias, mas estão definidos apenas para facilitar a notação. Portanto, o empacotamento P é formado combinando dois empacotamentos P' e P'' , de G_1 e G_2 , respectivamente, de forma que $n_i = \sum_{j=0}^i n_{i,j}$. Representa-se por n'_i e n''_i , para $1 \leq i \leq r$, o número de cliques de cardinalidade i nos empacotamentos P' e P'' , respectivamente. Assim, $n'_i = \sum_{j=i}^r n_{j,i}$ e $n''_i = \sum_{j=i}^r n_{j,j-i}$. Note que, cada uma das n'_i cliques de P' é parte de uma clique de P de cardinalidade maior ou igual a i . Além disso, as respectivas cliques em P , que contribuem para n'_i , não são necessariamente da mesma cardinalidade.

Na Figura 4.4 pode-se observar dois empacotamentos dos tipos $(0, 1)$ e $(1, 1)$ para um grafo `join`. Considere como operandos do grafo `join` o subgrafo G_1 , induzido pelos três vértices à esquerda e o G_2 , induzido pelos três vértices à direita. No caso (a), $n''_3 = 1$ pois $n_{3,0} = 1$. Já no caso (b), que corresponde ao valor ótimo de $f(G, 1)$, tem-se que $n'_1 = 2$, $n''_2 = 1$ e $n''_1 = 1$, pois $n_{3,1} = 1$ e $n_{2,1} = 1$.

Dado um empacotamento P do tipo (n_2, \dots, n_r) para um grafo `join` $G = G_1 + G_2$,

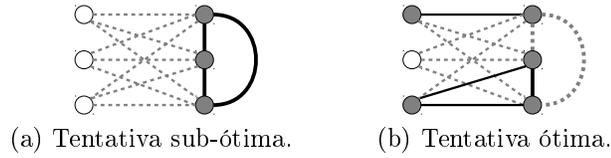


Figura 4.4: Tentativas de empacotamento de cliques em um grafo join.

verifica-se que as seguintes condições são atendidas:

$$n_i = \sum_{j=0}^i n_{i,j}, \text{ para } 2 \leq i \leq r \quad (4.1)$$

$$n'_2 \leq f(G_1, n'_3, \dots, n'_r) \text{ e } n''_2 \leq f(G_2, n''_3, \dots, n''_r) \quad (4.2)$$

$$|V(G_1)| \geq n'_1 + \sum_{i=2}^r in'_i \text{ e } |V(G_2)| \geq n''_1 + \sum_{i=2}^r in''_i \quad (4.3)$$

A primeira condição é válida pela maneira como os vértices das cliques em P são particionados em G_1 e G_2 , formando os empacotamentos de cliques P' e P'' . A segunda condição é válida pois existem os empacotamentos de cliques P' e P'' para G_1 e G_2 , que, ignorando cliques unitárias, são dos tipos (n'_2, \dots, n'_r) e (n''_2, \dots, n''_r) , respectivamente. Note que $f(G_1, n'_3, \dots, n'_r)$ pode ser maior que n'_2 , pois a função f procura por um empacotamento de cliques com o número máximo de K_2 (o mesmo vale para G_2 e n''_2) e que P' e P'' podem conter cliques de tamanho 1, partes de uma clique maior em G . A última condição decorre do fato que os grafos G_1 e G_2 contêm as cliques em P' e P'' , respectivamente, e que estas cliques são duas a duas disjuntas.

Acima, foi concluído que as Condições 4.1 a 4.3 são satisfeitas por um empacotamento de cliques P de G . Estas mesmas condições são também suficientes para a existência de um empacotamento do tipo (n_2, \dots, n_r) para G , que é composto a partir de empacotamentos de cliques para G_1 e G_2 .

Para computar $f(G, n_3, \dots, n_r)$, Guruswami et al. propuseram um algoritmo que, dado um valor para n_2 , avalia todas as possíveis atribuições de valores aos inteiros não-negativos $n_{i,j}$, até que estes satisfaçam as Condições 4.1 a 4.3. Inicialmente, $n_2 = 0$ e este será incrementado a cada iteração bem-sucedida. Quando uma iteração falha, o valor de n_2 anterior é retornado. No caso da primeira iteração falhar, a função estará indefinida.

Este procedimento já computa f num grafo join, contudo uma pequena melhoria pode ser feita, considerando a seguinte propriedade.

Lema 4.1 ([36]). *Se P é um empacotamento do tipo (n_2, \dots, n_r) para um grafo $G = G_1 + G_2$, então existe um empacotamento Q do mesmo tipo para G , tal que Q não contém*

duas cliques distintas A e B tais que $A \subseteq V(G_1)$ e $B \subseteq V(G_2)$. Além disso, nenhuma clique de Q possui um vértice de G que não pertença a nenhuma clique de P .

A demonstração do Lema 4.1 se dá por construção e consiste em substituir cliques $A = \{u_1, u_2, \dots, u_s\} \subseteq V(G_1)$ e $B = \{v_1, v_2, \dots, v_t\} \subseteq V(G_2)$, com $s, t \geq 2$, que ocorrem em P , pelas cliques $A' = \{v_1, u_2, \dots, u_s\}$ e $B' = \{u_1, v_2, \dots, v_t\}$, respectivamente, para obter um empacotamento Q com a propriedade desejada.

A propriedade descrita no lema anterior essencialmente significa que não é preciso analisar nenhuma atribuição de valores aos inteiros $n_{i,j}$ na qual, para algum k e l , $n_{k,0} \geq 1$ e $n_{l,l} \geq 1$, visto que existe um empacotamento do mesmo tipo para o qual esta condição não acontece. Logo, pode-se acrescentar às restrições anteriores a seguinte:

$$\sum_{i=1}^r n_{i,0} = 0 \text{ ou } \sum_{i=1}^r n_{i,i} = 0 \quad (4.4)$$

O número total de atribuições de valores aos inteiros $n_{i,j}$ que é considerado pelo procedimento acima descrito a fim de computar $f(G, n_3, \dots, n_r)$ será analisado a seguir.

Ignorando a última restrição, cada clique de tamanho i no grafo G pode possuir de 0 a i vértices em G_1 , resultando em $i + 1$ possibilidades. Ao tomar-se n_i cliques de tamanho i , podemos distribuí-las em $i + 1$ cestos, num total de $\binom{n_i+i}{n_i}$ possibilidades, que é $O(n_i^i)$. Esta análise decorre da associação dessa contagem com o clássico problema de contagem de maneiras de distribuir n_i bolas indistinguíveis em $i + 1$ cestos distinguíveis. Como as escolhas para n_i e n_j , $i \neq j$ são independentes, o total de possibilidades a analisar, considerando todos os parâmetros n_2, n_3, \dots, n_r de uma avaliação de f é $O(n^{2+3+\dots+r})$, visto que $n_i \leq n$, para qualquer $2 \leq i \leq r$.

Considerando a restrição 4.4, pode-se executar a análise da complexidade da computação de $f(G, n_3, \dots, n_r)$ em dois passos. Primeiro, consideram-se todas as possibilidades de atribuições aos inteiros $n_{i,j}$ que desconsideram cliques contidas em G_1 , depois faz-se o mesmo para G_2 . Logo, o número de formas distintas para dividir os vértices de uma clique de tamanho i foi reduzido de $i + 1$ para i . Seguindo o raciocínio anterior, resulta que o tempo total para avaliar uma atribuição de parâmetros de f é $O(n^{1+2+\dots+(r-1)})$.

4.2.3 Solução para cografos

O algoritmo proposto que resolve o PEC para os cografos computa a árvore de decomposição modular (que é isomorfa a co-árvore) para o grafo de entrada em tempo linear [22]. Cada nó desta árvore corresponde a um subgrafo induzido do grafo original, que também é um cografo. Além disso, cada nó que não é folha é rotulado com P ou S se o grafo

representado pelo nó for a união ou o join, respectivamente, dos grafos representados pelos filhos (veja mais sobre a co-árvore na Seção 2.4.1).

Posteriormente, a co-árvore é transformada em uma árvore binária, que será chamada *co-árvore binária*. Isto é feito da seguinte forma: cada nó com quatro ou mais filhos ($f_1, f_2, f_3, f_4, \dots, f_k$) é substituído por um nó com dois filhos, o primeiro deles terá como filhos $f_1, f_2, \dots, f_{\lfloor k/2 \rfloor}$ e o segundo, $f_{\lfloor k/2 \rfloor + 1}, \dots, f_k$. Um nó com três filhos f_1, f_2, f_3 é substituído por dois nós A e B . Os filhos de A serão f_1 e B e os de B serão f_2 e f_3 . Note que a co-árvore de G não possui nós com um único filho.

Na construção da co-árvore binária, toda vez que um nó é substituído por novos nós, estes têm o mesmo rótulo do nó removido. Esta árvore decompõe o cografo repetidamente pelas operações binárias de união e join até seus vértices isolados. Logo, a co-árvore binária pode ser usada diretamente para aplicação das técnicas descritas nas seções 4.2.1 e 4.2.2.

Na Figura 4.5 pode-se examinar um cografo no item (a) e sua co-árvore no item (b). A co-árvore binária obtida a partir do procedimento descrito acima é dada em (c). Note que, a forma escolhida para construir a co-árvore binária evita que se formem longas cadeias de nós inexistentes em relação a co-árvore original, desbalanceando demasiadamente a árvore. Todavia, este cuidado não é necessário - um nó com vários filhos poderia ser dividido em dois e seus filhos distribuídos livremente. Note que a co-árvore de um cografo é sempre a mesma, salvo por isomorfismos, mas a co-árvore binária não, pois depende da escolha da divisão dos filhos de cada nó.

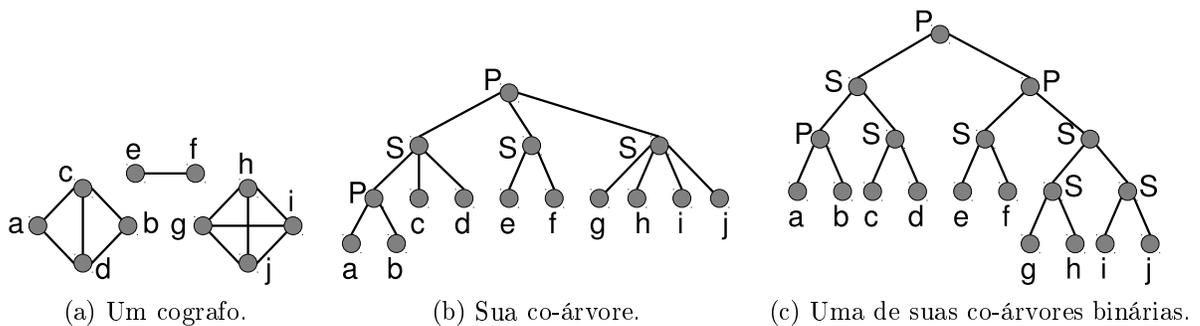


Figura 4.5: Exemplo de co-árvore binária.

O algoritmo usa a conhecida técnica da programação dinâmica, armazenando em cada nó da árvore uma tabela que contém o valor da função f no grafo correspondente ao nó, para cada sequência de parâmetros n_3, n_4, \dots, n_r que for necessária. As tabelas são preenchidas através de uma busca em profundidade, das folhas em direção a raiz. Desta forma, na avaliação de um nó, os valores de f para seus filhos já estão calculados e armazenados nas respectivas tabelas, podendo ser obtidos em tempo constante. Para as folhas da árvore, a tabela é preenchida com zeros.

Considere uma avaliação de $f(G', n_3, n_4, \dots, n_r)$ em um grafo G' correspondente a um nó qualquer da co-árvore binária. Verifica-se facilmente que a soma $\sum_{i=3}^r n_i$ será maior ou igual às somas $\sum_{i=3}^r n'_i$ e $\sum_{i=3}^r n''_i$ das chamadas recursivas de f nos filhos do nó, independentemente do nó representar uma operação de união ou `join`.

Recorde que o objetivo final é computar $f(G, 0, 0, \dots, k)$ no grafo G de entrada, para todos valores inteiros de k no intervalo $1 \leq k \leq \gamma$, sendo $\gamma = \lfloor n/r \rfloor$. Logo, a soma computada no parágrafo anterior está limitada a γ e esta restrição propaga-se para todos os nós da co-árvore binária. Assim, em cada nó, basta computar os valores de f para os parâmetros que obedecem a este limite. Como exercício de combinatória, verifica-se que o número de possibilidades que atende a estas condições é $\binom{\gamma+r-2}{\gamma}$ e esta função é $O(\gamma^{r-2})$ que, por sua vez, é $O(n^{r-2})$.

Por fim, o número de nós da co-árvore binária é $O(n)$ e número total de chamadas de f computadas em todos os nós é $O(n^{r-1})$. Assim, o tempo total do algoritmo é $O(n^{2+\dots+(r-1)}n^{r-1})$. Note que, embora polinomial, o algoritmo pode ser impraticável para valores elevados de n ou r .

4.3 Considerações para a classe P_4 -arrumada

A técnica descrita na Seção 4.2 resolve o PEC para cografos em tempo polinomial. As características essenciais da técnica são: a existência de uma árvore de decomposição do grafo obtida em tempo polinomial e que permita reconstruí-lo a partir de grafos-base (no caso, vértices isolados) através de um conjunto de operações; a possibilidade de computar cada chamada de f em tempo polinomial em qualquer nó da árvore de decomposição, considerando que f já foi computada em seus filhos para todos parâmetros necessários; e que f seja computável em tempo polinomial para os grafos-base (trivial no caso de vértices isolados).

Desta forma, pode-se estender a técnica a qualquer classe cujos grafos satisfaçam tais condições básicas. Naturalmente, surgem como candidatos superclasses dos cografos. Todavia, a abrangência dos candidatos está limitada já que o problema é NP-Difícil mesmo quando restrito a algumas classes de grafo, por exemplo, a classe dos grafos `split`. Entre as várias opções, aqui será estudada a classe P_4 -arrumada, por conter várias outras (como a P_4 -esparsa e a P_4 -reduzível, por exemplo), mas cujas superclasses imediatas, dentre as que temos conhecimento, está a P_4 -carregada estendida, que contém os grafos `split`.

A árvore de decomposição utilizada para grafos da classe P_4 -arrumada é a árvore de decomposição modular (veja Seção 2.3), que pode ser obtida em tempo linear [56]. Os nós desta árvore estão rotulados como P , S , ou N . Os dois primeiros correspondem aos rótulos da co-árvore e às operações de união e `join` dos grafos representados pelos filhos do nó. O terceiro rótulo corresponde a um tipo distinto de nó, que, para grafos da classe

P_4 -arrumada, pode representar os seguintes grafos [32]:

- C_5 , P_5 ou $\overline{P_5}$; ou
- uma aranha; ou
- uma pseudo-aranha.

Um processamento similar ao aplicado na co-árvore também é executado, tornando os nós de rótulos P e S todos binários, a fim de usar as técnicas conhecidas para grafos união e *join*. A árvore obtida é denominada *árvore de decomposição modular quase binária* (ADMQB). Um exemplo desse processamento é dado na Figura 4.6, na qual apresenta-se um grafo (a) e sua árvore de decomposição modular (b). Para chegar a ADMQB (c), apenas substituiu-se o nó de rótulo S por um par deles, deixando o nó de rótulo N intocado, embora possua mais de dois filhos.

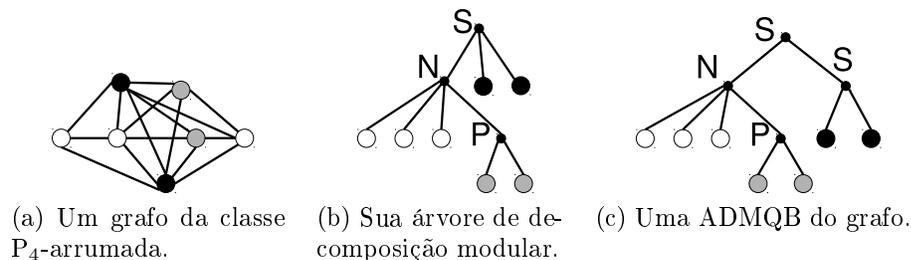


Figura 4.6: Um grafo da classe P_4 -arrumada e uma ADMQB.

O algoritmo é executado da mesma forma que para os cografos. Em cada nó da ADMQB é armazenada uma tabela com o resultado de f para as atribuições de parâmetros necessárias. Tais tabelas são preenchidas das folhas em direção à raiz por uma busca em profundidade. O processamento dos nós rotulados com P e S é o mesmo usado para os cografos. Contudo, ao atingir um nó do tipo N , utilizam-se novas técnicas para computar f . Caso o nó corresponda a um C_5 , P_5 ou $\overline{P_5}$, a tabela é preenchida com valores pré-computados. Caso o nó seja uma aranha ou pseudo-aranha, algoritmos apropriados, descritos nas próximas subseções, serão executados.

4.3.1 Aranhas

As aranhas foram definidas na Seção 2.4.2. Relembre que seu conjunto de vértices pode ser particionado em três conjuntos conhecidos como K , S e R . Note que se uma aranha G é tal que $R \neq \emptyset$, então $G[K \cup R] = G[K] + G[R]$ e $G[S \cup R] = G[S] \cup G[R]$. Logo, qualquer clique de uma aranha estará contida em $K \cup R$ ou em $K \cup S$ e, neste último

caso, a clique conterá no máximo um vértice de S pois S é um conjunto independente de G .

Dado um empacotamento de cliques P para uma aranha G representa-se por n_i^S , para $2 \leq i \leq r$, o número de cliques de tamanho i em P que contenham um vértice de S . As demais cliques de P são cliques do grafo $G[K] + G[R]$, se $R \neq \emptyset$, ou de $G[K]$, caso contrário. No primeiro caso, serão usadas as mesmas definições dos inteiros $n_{i,j}$, n'_i e n''_i já usadas para os grafos *join* na Seção 4.2.2 considerando $G_1 = G[K]$ e $G_2 = G[R]$. No outro caso, por conveniência, serão usadas as mesmas definições, considerando $G_1 = G[K]$ e G_2 um grafo vazio. Logo, $n''_i = 0$ e $n_{i,j} = 0$ para $0 \leq j < i$ e $1 \leq i \leq r$.

Como ilustração destes conceitos, considere as aranhas dadas na Figura 4.7. Em ambos exemplos, as arestas que ligam cada vértice h_i com cada vértice k_i foram omitidas, para diminuir a poluição visual. No caso (a), a aranha admite um empacotamento do tipo $(1, 0, 1)$, como o formado pelas cliques $\{k_1, k_2, h_1, h_2\}$ e $\{s_3, k_3\}$. Para este empacotamento, $n_{4,2} = 1$ e $n_2^S = 1$, fazendo $n'_2 = 1$ e $n''_2 = 1$. Já no caso (b), toma-se como exemplo um empacotamento do tipo $(2, 1)$ constituído pelas cliques $\{k_1, k_2, s_3\}$, $\{h_1, k_3\}$ e $\{h_2, h_3\}$, ao qual correspondem os inteiros $n_{2,1} = 1$, $n_{2,0} = 1$ e $n_3^S = 1$, implicando $n'_1 = 1$, $n''_1 = 1$ e $n''_2 = 1$.

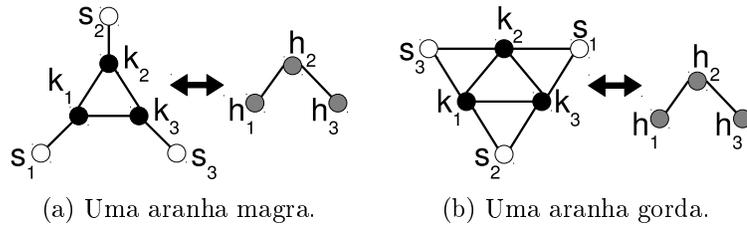


Figura 4.7: Empacotamento de cliques em uma aranha.

Sendo que existem dois tipos de aranhas, as magras e as gordas, do ponto de vista de empacotamento de cliques, a diferença entre elas é que na aranha magra uma clique que contenha um vértice de S poderá conter no máximo um outro vértice (pertencente a K), enquanto que numa aranha gorda, tal clique pode conter até $|K| - 1$ vértices em K .

Da mesma forma que para os grafos *join*, serão estudadas as condições necessárias para que exista um empacotamento de cliques de uma aranha. A seguir, determinam-se as condições suficientes sobre os inteiros $n_{i,j}$ e n_i^S para construir um empacotamento de cliques em conformidade com tais inteiros.

Primeiramente, dado um empacotamento P do tipo (n_2, \dots, n_r) para uma aranha G com partição (K, S, R) , valem as seguintes condições (que podem ser verificadas para as aranhas da Figura 4.7).

$$n_i = \sum_{j=0}^i n_{i,j} + n_i^S, \text{ para } 2 \leq i \leq r \quad (4.5)$$

$$n_2'' \leq f(G[R], n_3'', \dots, n_r'') \text{ se } R \neq \emptyset \quad (4.6)$$

$$|S| \geq \sum_{i=2}^r n_i^S, |K| \geq n_1' + \sum_{i=2}^r i n_i' + \sum_{i=2}^r (i-1) n_i^S \text{ e } |R| \geq n_1'' + \sum_{i=2}^r i n_i'' \quad (4.7)$$

$$n_i^S = 0, \text{ para todo } i \geq 3 \text{ (} i \geq |K| + 1 \text{) se a aranha é magra (gorda)} \quad (4.8)$$

A primeira condição é verdadeira desde que cada clique em P contém exatamente um vértice de S ou está contida em $K \cup R$. Para $R \neq \emptyset$, a segunda condição decorre do fato que as interseções das cliques em P com R origina um empacotamento de cliques em $G[R]$ contendo n_i'' cliques de cardinalidade i em $G[R]$, para $1 \leq i \leq r$. Note que considerar f para $G[K]$ equivale a verificar se K contém vértices suficientes, fato incluído na próxima condição. A terceira condição é satisfeita pela estrutura das aranhas e pelas definições de n_i' , n_i'' e n_i^S . Note que a restrição sobre a cardinalidade de S é redundante pois $|S| = |K| \geq \sum_{i=2}^r (i-1) n_i^S \geq \sum_{i=2}^r n_i^S$ e foi mantida pois será utilizada na próxima subseção. Além das condições anteriores, adaptadas das condições 4.1 a 4.3, P satisfaz também a condição 4.8 pois cliques do empacotamento que contém vértices de S não podem ter cardinalidade maior que 2 para aranhas magras e maior que $|K|$ para aranhas gordas.

Finalmente, a fim de apresentar um procedimento para computar $f(G, n_3, \dots, n_r)$ para uma aranha G , demonstra-se que as condições acima são não apenas necessárias, mas também suficientes para determinar a existência de empacotamentos de cliques em aranhas. Relembre-se que, dos inteiros $n_{i,j}$ obtém-se n_i' e n_i'' .

Lema 4.2. *Seja G uma aranha. A aranha G possui um empacotamento de cliques P do tipo (n_2, \dots, n_r) se, e somente se, existirem inteiros não negativos $n_{i,j}$ e n_i^S , para $2 \leq i \leq r$ e $0 \leq j \leq i$, satisfazendo as condições 4.5 a 4.8.*

Demonstração. Seja G uma aranha com partição (K, S, R) e P um empacotamento do tipo (n_2, \dots, n_r) para G . Os argumentos descritos anteriormente mostram que existem inteiros $n_{i,j}$ e n_i^S associados ao empacotamento P satisfazendo as condições 4.5 a 4.8. Note que estas condições refletem a estrutura das aranhas.

Agora, dada uma aranha G e os inteiros $n_{i,j}$ e n_i^S , para $2 \leq i \leq r$ e $0 \leq j \leq i$, que satisfazem as condições 4.5 a 4.8, será construído um empacotamento de cliques P do tipo (n_2, \dots, n_r) para G .

Primeiramente, rotule cada vértice de K por k_i , $1 \leq i \leq |K|$ e os ordene como $O_K = (k_1, k_2, \dots, k_{|K|})$. Se G for uma aranha magra, rotule como s_i o vértice de S

adjacente a k_i , para $1 \leq i \leq |S|$, e considere a sequência $O_S = (s_1, s_2, \dots, s_{|S|})$. Caso contrário (G é uma aranha gorda), rotule como s_i o único vértice em S que não é adjacente a k_i , para $1 \leq i \leq |S|$, e considere a sequência $O_S = (s_{|S|}, s_1, s_2, \dots, s_{|S|-1})$.

Em seguida, cada clique de P será construída, começando pelas que contém vértices em S . Para cada inteiro $2 \leq i \leq r$, constrói-se n_i^S cliques de tamanho i com um vértice em S . Cada clique é formada removendo-se o primeiro vértice de O_S e os $i - 1$ primeiros vértices de O_K e tomando-os como um conjunto.

Agora, mostra-se que cada conjunto adicionado à P é uma clique. Se G for uma aranha gorda, cada um dos conjuntos formados usará um vértice de S e ao menos um vértice de K , mas não todos, pela condição 4.8. Logo, pela ordenações O_S e O_K , os vértices s_i e k_i nunca estarão ambos no mesmo conjunto, que, portanto, será uma clique. Se G for uma aranha magra, cada conjunto conterá os vértices adjacentes s_i e k_i , uma vez que, pela condição 4.8, as únicas cliques possíveis tem tamanho 2.

Pode-se, agora, formar as cliques que não contém vértices em S .

Considere os inteiros n'_j para $1 \leq j \leq r$. Forma-se um empacotamento de cliques P' com n'_j cliques de cardinalidade j , obtidas removendo-se, para cada j , jn'_j vértices remanescentes em O_K . Também considere um empacotamento P'' do tipo (n''_2, \dots, n''_r) para $G[R]$, no qual incluem-se n''_1 vértices isolados de $G[R]$ que ainda não pertenciam a cliques em P'' . Agora, para cada $n_{i,j} > 0$, $1 \leq j < i \leq r$, repete-se $n_{i,j}$ vezes o seguinte procedimento: remove-se uma clique de tamanho j de P' e uma clique de tamanho $i - j$ de P'' , unem-se as duas cliques e adiciona-se a clique resultante a P . Findo o procedimento, restam em P' e P'' as cliques que correspondem aos inteiros $n_{i,i}$ e $n_{i,0}$, respectivamente. Adicionando à P todas as cliques restantes em P' e em P'' , tem-se o empacotamento desejado.

Como $G[K]$ é um grafo completo, P'' é um empacotamento de cliques para $G[R]$ e $G[K \cup R] = G[K] + G[R]$, os conjuntos acima produzidos evidentemente são cliques.

A condição 4.7 garante a existência de vértices suficientes em K , S e R para produzir P' e P'' . Além disso, a condição 4.6 garante a existência do empacotamento P'' .

Note que, se $R = \emptyset$, então $n_i = n'_i + n_i^S$, para $2 \leq i \leq r$, pelas condições 4.5 e 4.7. Logo, o processo é resumido a criar P' e adicionar suas cliques à P para obter o empacotamento desejado.

Finalmente, a condição 4.5, garante que o número correto de cliques de tamanhos 2 a r foram adicionadas a P . \square

Assim como para grafos *join*, o algoritmo proposto para computar $f(G, n_3, \dots, n_r)$, sendo G uma aranha, começa com $n_2 = 0$ e a cada iteração bem-sucedida incrementa n_2 . Uma iteração consiste em avaliar todas as possibilidades de atribuições aos inteiros $n_{i,j}$ e n_i^S até que se encontre uma satisfazendo as condições do Lema 4.2. Quando nenhuma das

atribuições satisfaz tais condições, a iteração falha e o valor anterior de n_2 é retornado. Caso a primeira iteração falhe (com $n_2 = 0$), a função estará indefinida.

Antes de analisar o tempo gasto pelo algoritmo, considere uma pequena simplificação. Dado que, para calcular $f(G, n_3, \dots, n_r)$ é necessário determinar o maior valor de n_2 tal que exista um empacotamento de cliques do tipo (n_2, \dots, n_r) , a seguinte propriedade mostra-se útil, quando G é uma aranha.

Lema 4.3. *Se P é um empacotamento do tipo (n_2, \dots, n_r) para uma aranha G , então existe um empacotamento de cliques Q do mesmo tipo para G tal que toda clique de tamanho 2 está contida em R ou contém um vértice de S . Além disso Q e P contêm as mesmas cliques de cardinalidade maior que 2.*

Demonstração. Considere os inteiros $n_{i,j}$ e n_i^S associados a P . É necessário mostrar que existe quantidade suficiente de vértices livres em S para construir Q a partir de P . De fato, toda clique em P contida em $K \cup S$ possui no máximo 1 vértice em S . Logo, o número de vértices em S não contidos em cliques de P é, pelo menos, $\sum_{i=1}^r in'_i \geq 2n_{2,2} + n_{2,1}$. Assim, as seguintes transformações podem ser executadas: $n_2^S \leftarrow n_2^S + n_{2,1} + n_{2,2}$, $n_{2,2} \leftarrow 0$ e $n_{2,1} \leftarrow 0$. Por inspeção, observa-se que os inteiros, após as transformações, satisfazem as condições 4.5 a 4.8 e, pelo Lema 4.2, existe um empacotamento de cliques Q para a aranha G . \square

Note que as transformações executadas na demonstração do lema anterior alteram apenas os inteiros $n_{2,2}$, $n_{2,1}$ e n_2^S , não afetando a forma como cliques de cardinalidade maior que 2 são formadas. Na verdade, as transformações podem ser feitas *in loco*, removendo as cliques de P que correspondem aos inteiros $n_{2,1}$ e $n_{2,2}$ e usando seus vértices juntamente com vértices livres de S para compor as novas cliques e adicioná-las novamente a P .

Desta propriedade, resulta imediatamente que poderíamos adicionar a expressão $n_{2,1} + n_{2,2} = 0$ às condições necessárias para existência de um empacotamento de cliques em uma aranha, de modo que $n_2 = n_{2,0} + n_2^S$. Ainda, dados os inteiros $n_{i,j}$ e n_i^S que satisfazem as condições já conhecidas para um empacotamento P do tipo $(0, n_3, n_4, \dots, n_r)$, pode-se imediatamente calcular N_2 , o maior valor de n_2 tal que exista um empacotamento do tipo (n_2, n_3, \dots, n_r) para a mesma aranha. Desta forma, o algoritmo precisaria apenas analisar as possíveis maneiras de distribuir as cliques de tamanho entre 3 e r . Então, para cada possibilidade, calculam-se as quantidades de vértices dos conjuntos S e K que não são utilizados pelas cliques contidas em P : $F_S = |S| - \sum_{i=3}^r n_i^S$ e $F_K = |K| - \sum_{i=1}^r in'_i - \sum_{i=3}^r (i-1)n_i^S$. Uma vez que os inteiros $n_{2,0}$ e n_2^S podem ser maximizados de forma independente, $N_2 = \min\{F_S, F_K\} + f(G[R], n_3'', \dots, n_r'') - n_2''$, se $R \neq \emptyset$, ou $N_2 = \min\{F_S, F_K\}$, caso contrário. Os cálculos podem ser ainda simplificados, observando-se que $F_K \leq F_S$ e, portanto, $\min\{F_S, F_K\} = F_K$.

Com isto, inicia-se a análise da complexidade temporal do algoritmo. Considere, a princípio, que $R \neq \emptyset$. Uma vez que, para aranhas magras, nenhuma clique de tamanho maior que 2 pode usar vértices em S , então a contagem de possibilidades analisadas é a mesma que para o grafo $G[K]+G[R]$. Logo o tempo gasto é $O(n^{2+3+\dots+(r-1)})$. Para aranhas gordas, num cenário de pior caso, qualquer clique desejada pode ser empacotada utilizando um vértice de S . Assim, o número de possibilidades analisadas para n_i cliques de tamanho i é $\binom{n_i+i+1}{n_i}$. Todavia, as cliques contidas em $K \cup R$ constituem um empacotamento de $G[K] + G[R]$ e, assim, aplicando o Lema 4.1, o número de possibilidades é reduzido para $\binom{n_i+i}{n_i}$, que é $O(n_i^i)$. Portanto, o tempo total gasto é $O(n^{3+4+\dots+r})$.

Se $R = \emptyset$ e G for uma aranha magra, todas as cliques desejadas de tamanho maior que 2 precisam estar contidas em K e, como exposto nos comentários do Lema 4.3, considera-se $n_2 = 0$. Logo $n'_i = n_i$, para todo $3 \leq i \leq r$, e uma única avaliação da condição 4.7 é suficiente para determinar se o empacotamento é viável. O cálculo de N_2 também é imediato, resultando em um tempo total de $O(n)$ para contar os vértices em K e S . Se G for uma aranha gorda, numa análise simples, cada clique contém ou não um vértice de S . Logo, para as n_i cliques de tamanho i , haverá $\binom{n_i+1}{n_i} = n_i + 1$ possibilidades. Assim, o total de possibilidades é $O(n^{r-2})$.

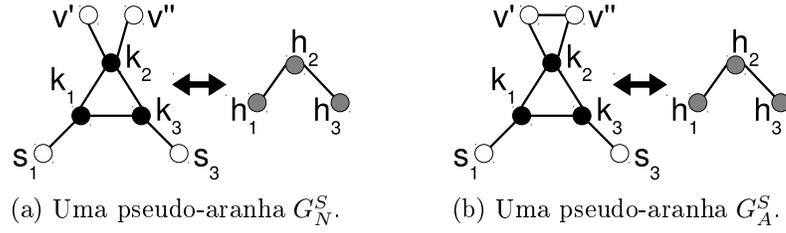
4.3.2 Pseudo-aranhas

Relembre que uma pseudo-aranha é um grafo que é obtido de uma aranha por uma única alteração: tome um vértice v de K ou S e o divida em dois vértices v' e v'' , cada um deles tendo a mesma vizinhança do vértice v . Além disso, v' e v'' podem ou não ser adjacentes. Veja mais sobre pseudo-aranhas na Seção 2.4.3.

Para facilitar a notação, se G é uma aranha, G_N^K e G_A^K representarão as pseudo-aranhas obtidas de G substituindo um vértice $v \in K$ por dois, não-adjacentes e adjacentes, respectivamente. De modo análogo, G_N^S e G_A^S representarão as pseudo-aranhas obtidas quando $v \in S$. Os novos vértices sempre serão chamados de v' e v'' e os conjuntos K , S e R da aranha G são estendidos para a pseudo-aranha, removendo v do conjunto ao qual pertencia e adicionando ao mesmo conjunto os vértices v' e v'' . Além disso, as mesmas definições de $n_{i,j}$, n_i^S e seus derivados serão usadas para um empacotamento de cliques de uma pseudo-aranha.

Na Figura 4.8 são apresentadas duas pseudo-aranhas obtidas com base na aranha da Figura 4.7a, tomando $v = s_2$.

A análise das pseudo-aranhas é dividida em quatro casos, sem distinguir se a pseudo-aranha originou-se de uma aranha magra ou gorda. Serão dadas condições para o empacotamento de cliques em todos os casos de pseudo-aranhas nos lemas 4.4, 4.5 e 4.7 e o procedimento para calcular f será dado para cada um deles. Além disso, duas propriedades

Figura 4.8: Pseudo-aranhas com $|S| = |K| + 1$.

apresentadas nos lemas 4.6 e 4.8 serão usadas para restringir a busca de empacotamentos de cliques durante a computação de f , melhorando a complexidade do algoritmo nos casos cobertos pelos lemas 4.5 e 4.7.

Primeiramente, considere o caso mais simples, que pode ser reduzido à análise de uma aranha.

Lema 4.4. *Seja G uma aranha, então $f(G_N^S, n_3, \dots, n_r) = f(G, n_3, \dots, n_r)$.*

Demonstração. Se P é um empacotamento do tipo (n_2, \dots, n_r) para G , pode-se transformá-lo em um empacotamento P' do mesmo tipo para G_N^S apenas trocando v por v' (ou v'') em alguma clique de P que contenha v , se houver.

Agora, seja P um empacotamento de cliques para G_N^S do tipo (n_2, \dots, n_r) e seja (K, S, R) a partição canônica de G_N^S . Se G é uma aranha magra, então pelo menos um dos vértices v' ou v'' não faz parte de nenhuma clique de P , pois ambos os vértices tem como único vizinho o mesmo vértice de K . Logo, P também é um empacotamento do tipo (n_2, \dots, n_r) para G . Já se G é uma aranha gorda, então P pode conter cliques distintas C' e C'' tais que $v' \in C'$ e $v'' \in C''$. Se este for o caso, como $|S| > |K|$, existe um vértice $u \in S$ não usado por nenhuma clique de P . Pela definição de aranha gorda, os vizinhos de u contêm $C' \setminus \{v'\}$ ou $C'' \setminus \{v''\}$. No primeiro caso, faça $P' = (P \setminus \{C'\}) \cup \{u\} \cup (C' \setminus \{v'\})$ e, no outro caso, proceda de forma análoga, obtendo, em ambos os casos, um empacotamento P' do tipo (n_2, \dots, n_r) para G . \square

É fácil observar, na Figura 4.8a, que, em qualquer empacotamento de cliques para G_N^S , v' ou v'' não fará parte de nenhuma clique. Logo, o algoritmo que resolve o PEC para aranhas, definido na Seção 4.3.1, também resolve o PEC para G_N^S , bastando desconsiderar v'' , por exemplo.

Em seguida, considerar-se-á uma pseudo-aranha na qual v' e v'' são adjacentes. Neste caso, ambos podem participar de uma mesma clique. Define-se o inteiro $n_i^e \in \{0, 1\}$, $2 \leq i \leq r$, para representar se um empacotamento de cliques contém uma clique de tamanho i contendo v' e v'' . Note que, esta possível clique não é levada em conta pelos inteiros n_i^S ,

que só consideram cliques com um único vértice em S . Desta forma, um empacotamento do tipo (n_2, \dots, n_r) de uma pseudo-aranha G_A^S satisfaz as seguintes condições.

$$n_i = \sum_{j=0}^i n_{i,j} + n_i^S + n_i^e, \text{ para } 2 \leq i \leq r \quad (4.9)$$

$$n_2'' \leq f(G[R], n_3'', \dots, n_r'') \text{ se } R \neq \emptyset \quad (4.10)$$

$$|K| \geq n_1' + \sum_{i=2}^r [in_i' + (i-1)n_i^S + (i-2)n_i^e] \text{ e } |R| \geq n_1'' + \sum_{i=2}^r in_i'' \quad (4.11)$$

$$|S| \geq \sum_{i=2}^r (n_i^S + 2n_i^e) \quad (4.12)$$

$$n_i^S = 0, \text{ para todo } i \geq 3 \text{ (} i \geq |K| + 1 \text{) se a aranha é magra (gorda)} \quad (4.13)$$

$$n_i^e = 0, \text{ para todo } i \geq 4 \text{ (} i \geq |K| + 2 \text{) se a aranha é magra (gorda)} \quad (4.14)$$

$$\sum_{i=2}^r n_i^e \leq 1, \text{ para } 2 \leq i \leq r \quad (4.15)$$

Note que as condições 4.9 a 4.13 são análogas às condições 4.5 a 4.8 para aranhas, apenas considerando os inteiros n_i^e , quando necessário. As condições 4.14 e 4.15 refletem as possibilidades válidas para a única clique possível contendo os vértices v' e v'' . Além disso, note que a condição 4.12 considera explicitamente o tamanho de S , invalidando atribuições aos inteiros n_i^S e n_i^e que usem mais vértices do que disponíveis neste conjunto, ao passo que para aranhas essa restrição estava indiretamente garantida.

Como exemplo, a pseudo-aranha da Figura 4.8b possui um empacotamento do tipo $(1, 2)$ dado pelas cliques $\{v', v'', k_2\}$, $\{h_1, h_2, k_3\}$ e $\{k_1, h_3\}$. Para este empacotamento, $n_3^e = 1$, $n_{3,1} = 1$ e $n_{2,1} = 1$, implicando $n_1' = 2$, $n_1'' = 1$ e $n_2'' = 1$. Pode-se verificar que tal empacotamento satisfaz as condições apresentadas acima.

Lema 4.5. *Se G é uma aranha, então G_A^S possui um empacotamento de cliques P do tipo (n_2, \dots, n_r) se, e somente se, existem inteiros não-negativos, $n_{i,j}$, n_i^S e n_i^e satisfazendo as condições 4.9 a 4.15 para G_A^S .*

Demonstração. Seja G uma aranha. Se o grafo G_A^S possui um empacotamento de cliques P do tipo (n_2, \dots, n_r) verifica-se, por inspeção, que os inteiros $n_{i,j}$, n_i^S e n_i^e , associados a P , satisfazem as condições propostas neste lema.

Agora, se existem inteiros $n_{i,j}$, n_i^S e n_i^e , para $2 \leq i \leq r$ e $0 \leq j \leq i$, que satisfazem as condições 4.9 a 4.15, é dado a seguir um procedimento para construir um empacotamento P do tipo (n_2, \dots, n_r) para G_A^S .

Primeiro, se $n_i^e = 0$, para todo $2 \leq i \leq r$, então facilmente verifica-se que os inteiros $n_{i,j}$ e n_i^S satisfazem as condições 4.5 a 4.8 para G e, pelo Lema 4.2, existe um empacotamento de cliques P' do tipo (n_2, \dots, n_r) para G . Se houver uma clique em P' contendo v , este vértice é substituído por v' ou v'' . Desta forma, obtém-se o empacotamento do tipo (n_2, \dots, n_r) para G_A^S .

Suponha, agora, que $n_i^e \neq 0$ para algum $2 \leq i \leq r$. Pelas condições 4.14 e 4.15, existe um único inteiro j tal que $n_j^e = 1$, i.e., para todo inteiro k , $2 \leq k \leq r$ e $k \neq j$, $n_k^e = 0$. Desta forma, aplica-se a seguinte transformação: $n_{j-1}^S \leftarrow n_{j-1}^S + 1$, se $j \geq 3$. Logo, os inteiros $n_{i,j}$ e n_i^S satisfazem as condições do Lema 4.2 e G possui um empacotamento P' do tipo $(n_2, \dots, n_{j-1} + 1, n_j - 1, \dots, n_r)$, se $j \geq 3$ ou do tipo $(n_2 - 1, n_3, \dots, n_r)$ se $j = 2$.

Para construir P , no primeiro caso, tome qualquer clique C de P' com $j - 1$ vértices e contendo um vértice de S (tal clique existe por construção). No outro caso, tome para C uma clique unitária contendo um vértice de S não usado em P' (que existe pela condição 4.11). Seja $\{u\} = C \cap S$. Altere os rótulos dos vértices da aranha de modo que $u = v$ (basta trocar u com v e $b^{-1}(u)$ com $b^{-1}(v)$, conforme a definição de aranha na Seção 2.4.2). Por fim, $P = (P' \setminus C) \cup (\{v', v''\} \cup (C \setminus \{v\}))$ é o empacotamento desejado. \square

Visto que, para computar f deseja-se sempre o maior número possível de cliques de tamanho 2 no empacotamento, a seguinte propriedade usa esse fato para limitar um pouco as possibilidades a serem consideradas.

Lema 4.6. *Se G é uma aranha, $G' = G_A^S$, P é um empacotamento de cliques do tipo (n_2, \dots, n_r) para G' e P é máximo para n_2 (ou seja, não existe um empacotamento do tipo $(n_2 + 1, n_3, \dots, n_r)$ para G'), então existe um empacotamento de cliques P' para G' , do mesmo tipo, tal que v' e v'' pertencem a mesma clique, ou seja, $n_i^e = 1$ para algum $2 \leq i \leq r$.*

Demonstração. Suponha que v' e v'' não fazem parte da mesma clique de P . Se ambos v' e v'' não estiverem contidos em nenhuma clique de P , então P não é máximo para n_2 pois $P \cup \{\{v', v''\}\}$ seria um empacotamento de cliques para G_A^S com mais cliques de tamanho 2. Então, sem perda de generalidade, seja $v' \in C' \in P$ e $u \in (C' \setminus \{v'\})$. Como $u \in K$, basta permutar v'' por u em suas respectivas cliques. Ou seja, se houver um $C'' \in P$ tal que $v'' \in C''$, considere $P' = (P \setminus \{C', C''\}) \cup \{(C' \cup \{v''\}) \setminus \{u\}, (C'' \cup \{v\}) \setminus \{v''\}\}$. Se não existir C'' , considere $P' = (P \setminus \{C'\}) \cup \{(C' \cup \{v''\}) \setminus \{u\}\}$. Verifica-se por inspeção que P' é o empacotamento desejado. \square

Por fim, serão avaliadas as pseudo-aranhas obtidas quando v' e v'' estão contidos em K . Se eles forem adjacentes, K será uma clique e um empacotamento de cliques para a pseudo-aranha satisfará as mesmas condições 4.5 a 4.7, que os empacotamentos de cliques para aranhas satisfazem. Contudo a condição 4.8 fica substituída pelas duas condições

abaixo, pois se a pseudo-aranha em questão for magra, pode-se formar uma clique com v' , v'' e o vértice de S adjacente a ambos.

$$n_i^S = 0, \text{ para todo } i \geq 4 \text{ (} i \geq |K| + 1 \text{) se a aranha é magra (gorda)} \quad (4.16)$$

$$n_3^S \leq 1, \text{ se a pseudo-aranha é magra.} \quad (4.17)$$

Já se os vértices v' e v'' não forem adjacentes, nenhuma clique da pseudo-aranha pode usar todos vértices de K , pois esta parte não é uma clique. Assim, um empacotamento de cliques para a pseudo-aranha satisfará as mesmas condições 4.5 a 4.7, além das seguintes.

$$n'_i = 0 \text{ e } n_i^S = 0, \text{ para todo } |K| \leq i \leq r \quad (4.18)$$

$$n_i^S = 0, \text{ para todo } i \geq 3 \text{ (} i \geq |K| \text{) se a aranha é magra (gorda)} \quad (4.19)$$

Na Figura 4.9 encontram-se dois exemplos de pseudo-aranhas, obtidos fazendo $v = k_2$ na aranha da Figura 4.7a, aos quais tais condições se aplicam. Para o caso (a), tem-se um empacotamento do tipo $(1, 0, 0, 1)$ dado por $\{s_2, v''\}$ e $\{v', k_1, k_3, h_1, h_2\}$ tal que $n_2^S = 1$ e $n_{5,3} = 1$. Já para o caso (b), existe um empacotamento do tipo $(0, 0, 0, 0, 1)$ formado pela clique $\{k_1, v', v'', k_3, h_1, h_2\}$.

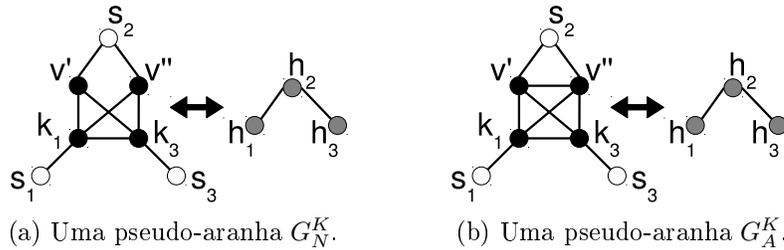


Figura 4.9: Pseudo-aranhas com $|K| = |S| + 1$.

As condições descritas além de necessárias para a existência de um empacotamento de cliques para tais pseudo-aranhas, também são suficientes, como afirma o lema a seguir.

Lema 4.7. *Se G' é uma aranha e $G = G_N'^K$ ou $G = G_A'^K$, então G possui um empacotamento de cliques P do tipo (n_2, \dots, n_r) se, e somente se, existem inteiros não-negativos, $n_{i,j}$, n_i^S satisfazendo as condições 4.5 a 4.7 para G e também as condições 4.16 e 4.17, se $G = G_A'^K$, ou as condições 4.18 e 4.19, se $G = G_N'^K$.*

Demonstração. Dada uma aranha G' e os inteiros $n_{i,j}$ e n_i^S satisfazendo as condições deste lema, será formado um empacotamento de cliques P , para $G = G_N'^K$ ou $G = G_A'^K$, como

desejado. Para tal considere que K' , S e R são a partição canônica da aranha G' e K , S e R , da pseudo-aranha G .

Note que, como $|K'| = |K| - 1$ pode ser impossível obter um empacotamento com os parâmetros solicitados para G' . Contudo, a seguir será mostrado que pode-se obter um empacotamento de cliques P' para G' com todas as cliques desejadas para P , exceto possivelmente uma. Esta clique ausente é substituída em P' por outra, que possuirá um vértice em K' a menos que o desejado em P para K . Então, através de uma troca de vértices e usando v' e v'' , acrescenta-se um vértice a esta clique para obter P .

Quando for necessário obter um empacotamento de cliques P' , conforme descrito acima, serão utilizados os inteiros $\bar{n}_{i,j}$ e \bar{n}_i^S que inicialmente são idênticos aos valores de $n_{i,j}$ e n_i^S para representar a estrutura de P' .

Considera-se um caso especial quando $G = G_A^K$, $n_3^S = 1$ e G for uma pseudo-aranha magra. Nesta situação aplicam-se as seguintes transformações: $\bar{n}_3^S \leftarrow 0$ e $\bar{n}_2^S \leftarrow n_2^S + 1$. Verifica-se por inspeção que os inteiros $\bar{n}_{i,j}$ e \bar{n}_i^S satisfazem as condições 4.5 a 4.8 para G' , o que implica na existência de um empacotamento P' do tipo $(n_2 + 1, n_3 - 1, n_4, \dots, n_r)$ para G' . Por construção, P' conterà uma clique C com dois vértices, um deles em S e o outro em K . Seja $\{u\} = C \cap K$. Se $u \neq v$, troque, em cada clique de P' , u por v e $b^{-1}(u)$ por $b^{-1}(v)$. Logo $C \cap K = \{v\}$ e, ao substituir v por v' e v'' , forma-se o empacotamento de cliques P , desejado para G .

Se os inteiros dados não configuram o caso especial e a primeira parte da condição 4.7 (a que envolve K) não for satisfeita com igualdade, então há sobra de vértices em K e verifica-se por inspeção que os inteiros $n_{i,j}$ e n_i^S satisfazem as condições 4.5 a 4.8 para G' . Logo, pelo Lema 4.2, G' possui um empacotamento P' do tipo (n_2, \dots, n_r) . Se houver uma clique $C' \in P'$ tal que $v \in C'$, substitua v por v' . Assim, obteve-se o empacotamento $P = P'$ para G .

Agora, consideram-se os casos distintos do caso especial e sem sobra de vértices em K . Se $n'_i \geq 1$ para algum inteiro i , será feita uma transformação nos inteiros para que uma dentre as cliques desejadas contenha um vértice a menos de K . Logo, selecione inteiros k e l , $1 \leq l \leq k$ e $2 \leq k \leq r$ tais que $n_{k,l} \geq 1$ (caso *a*) e faça $\bar{n}_{k,l} \leftarrow n_{k,l} - 1$ e $\bar{n}_{k-1,l-1} \leftarrow n_{k-1,l-1} + 1$, se $k \geq 3$. Caso contrário, $n_k^S \geq 1$ para algum k , $3 \leq k \leq r$ (caso *b*) e aplica-se a seguinte transformação: $\bar{n}_k^S \leftarrow n_k^S - 1$ e $\bar{n}_{k-1}^S \leftarrow n_{k-1}^S + 1$. Estas transformações garantem que os inteiros $\bar{n}_{i,j}$ e \bar{n}_i^S satisfazem as condições 4.5 a 4.8 para G' , que, pelo Lema 4.2, possui um empacotamento P' .

Em seguida, a partir de P' será produzido o empacotamento de cliques P desejado. Primeiro, considere que $G = G_N^K$, no qual v' e v'' não podem pertencer a mesma clique.

Suponha que foi aplicado o caso *a*. Se $k = 2$ e $l = 2$, então existe um vértice $x \in K'$ não usado por nenhuma clique de P' . Se $k = 2$ e $l = 1$, então existe um vértice $x \in R$ não usado por nenhuma clique de P' . Nestes dois casos, forma-se $C' = \{x\}$. Por fim, se $k \geq 3$,

então existe uma clique $C' \in P'$ tal que $C' \in K' \cup R$ e $|C' \cap K'| = l - 1$. Em todos os casos, pela condição 4.18, existe um vértice $u \in K' \setminus C'$. Alteram-se os rótulos dos vértices de G' de modo que $u = v$ (conforme feito no Lema 4.2, corrigindo as cliques em P'). Agora, adiciona-se v' à C' , faz-se $P = (P' \setminus \{C'\}) \cup \{C' \cup \{v'\}\}$ e, se houver uma clique $C'' \in P'$ tal que $v \in C''$, substitui-se v por v'' , fazendo $P \leftarrow (P \setminus \{C''\}) \cup \{(C'' \setminus \{v\}) \cup \{v''\}\}$. Chega-se, assim, ao empacotamento P desejado.

Considere agora que foi aplicado o caso *b*. Note que a pseudo-aranha G é gorda, caso contrário, ou haveria vértices livres em K ou a condição 4.19 seria violada.

Portanto, $k \geq 3$ e existe $C' \in P'$ tal que $|C'| = k - 1$ e $C' \cap S = \{u\}$. Uma vez que $k \leq |K| - 1$ (pela condição 4.18), $|C'| \leq |K| - 2 = |K'| - 1$ e $|C' \cap K'| \leq |K'| - 2$. Ou seja, existem ao menos dois vértices em $K' \setminus C'$ e, pela definição de aranha gorda, ao menos um deles, denominado t , é adjacente a u . Altere os rótulos dos vértices de G' de modo que $t = v$, adicione a C' o vértice v' e troque v por v'' em qualquer clique de P' que contivesse v para chegar ao empacotamento desejado. Ou seja, $P = (P' \setminus \{C'\}) \cup \{C' \cup \{v'\}\}$ e, caso exista $C'' \in P'$ tal que $v \in C''$, faça $P \leftarrow (P \setminus \{C''\}) \cup \{(C'' \setminus \{v\}) \cup \{v''\}\}$.

Como última tarefa, P será produzido a partir de P' no caso em que $G = G_A^K$. Neste caso, v' e v'' podem pertencer a mesma clique.

Considere que foi aplicado o caso *a*. Se $k = 2$, então existe um vértice $x \in K'$ ou $x \in R$ não utilizado por nenhuma clique de P' . Se $x \in K'$, altere os rótulos dos vértices de G' para que $x = v$ e faça $P = P' \cup \{v', v''\}$. Já se $x \in R$, tome a clique $C' \in P'$ tal que $v \in C'$ e faça $C = (C' \setminus \{v\}) \cup \{v'\}$ e $P = (P' \setminus \{C'\}) \cup \{C, \{x, v''\}\}$. Se $k \geq 3$, então existe uma clique $C' \in P'$ tal que $|C'| = k - 1$ e $C' \in K' \cup R$. Se existir $x \in C' \cap K'$, altere os rótulos dos vértices de G' para que $x = v$ e faça $C = (C' \setminus \{v\}) \cup \{v', v''\}$ e $P = (P' \setminus \{C'\}) \cup \{C\}$. Caso contrário, tome qualquer $x \in C'$ ($x \in R$) e $C'' \in P'$ tal que $v \in C''$ e faça $C''_1 = (C'' \setminus \{v\}) \cup \{v'\}$, $C'_1 = (C' \cup \{v''\})$ e $P = (P' \setminus \{C', C''\}) \cup \{C''_1, C'_1\}$.

Suponha, por fim, que foi aplicado o caso *b*. Logo $k \geq 3$ e existe uma clique C' em P' , tal que $|C'| = k - 1$ e $C' \cap K' \neq \emptyset$. Seja $u \in C' \cap K'$ e altere os rótulos dos vértices de G' tal que $v = u$. Produz-se P trocando v pelo par v' e v'' em C' , i.e., $P = (P' \setminus \{C'\}) \cup \{(C' \setminus \{v\}) \cup \{v', v''\}\}$. \square

Com isto, foram determinadas as condições necessárias e suficientes para existir um empacotamento de cliques em todos os tipos de pseudo-aranhas.

Por fim, para computar f , é necessário determinar N_2 para um dado empacotamento do tipo (n_2, \dots, n_r) , para todos os tipos de pseudo-aranhas, exceto a G_N^S , que já foi considerada no Lema 4.4. Para tal, considere a propriedade abaixo.

Lema 4.8. *Se G é uma aranha, G' é uma pseudo-aranha tal que $G' = G_A^S$, $G' = G_N^K$ ou $G' = G_A^K$ e P é um empacotamento de cliques de G' do tipo (n_2, \dots, n_r) , então existe um empacotamento de cliques P' para G' , do mesmo tipo, tal que, $n_{2,2} = 0$ e $n_{2,1} \leq 1$.*

Demonstração. Como para aranhas, seja F_S o número de vértices em S não utilizados por nenhuma clique em P , ou seja, $F_S = |S| - \sum_{i=2}^r (n_i^S + 2n_i^e)$ com $n_i^e = 0$ para todo $2 \leq i \leq r$ se a pseudo-aranha G' não for do tipo G_A^S . Em todos os casos, o empacotamento P será tal que $F_S \geq \sum_{i=1}^r in'_i - 1 \geq 2n_{2,2} + n_{2,1} - 1$. Logo, pode-se aplicar as seguintes transformações nos inteiros, a fim de tornar $n_{2,2} = 0$, de forma semelhante ao Lema 4.3. Dado que $x = \min\{F_S - n_{2,2}, n_{2,1}\}$, faça: $n_2^S \leftarrow n_2^S + n_{2,2} + x$, $n_{2,2} \leftarrow 0$ e $n_{2,1} \leftarrow n_{2,1} - x$.

Verifica-se facilmente que, após as transformações, os inteiros $n_{i,j}$ e n_i^S continuam satisfazendo as condições dos lemas 4.5 ou 4.7, dado que os inteiros originais, dados pelos parâmetros de P , obedeciam às respectivas condições. \square

O Lema 4.8 fornece, para pseudo-aranhas, uma simplificação semelhante a que o Lema 4.3 fornece para aranhas. Ou seja, pode-se adicionar as condições $n_{2,2} = 0$ e $n_{2,1} \leq 1$ às restrições sobre os inteiros na busca por empacotamentos.

Assim, pode-se considerar o tempo gasto pelo algoritmo para computar o valor de f , dados os inteiros n_2, n_3, \dots, n_r , em cada tipo de pseudo-aranha G obtida de uma aranha G' . De acordo com o Lema 4.4, se $G = G_N^S$, então o algoritmo tomará $O(n^{3+4+\dots+r})$, no pior caso. No caso em que $G = G_A^K$, a análise é similar do que para uma aranha, na qual, para cada $3 \leq i \leq r$, as n_i cliques de tamanho i podem ser distribuídas em $r+1$ maneiras e o valor máximo de n_2 pode ser calculado em tempo constante, resultando no mesmo tempo $O(n^{3+4+\dots+r})$.

Para $G = G_A^S$, pode-se escolher um valor de $2 \leq j \leq r$ tal que $n_j \geq 1$ para que $n_j^e = 1$ (note que, mesmo que $n_i = 0$, para todo $3 \leq i \leq r$, sempre é possível fazer $n_2 = 1$) e então, para cada uma das r possibilidades, considera-se que as cliques restantes são distribuídas da mesma forma que para aranhas entre os inteiros $n_{i,j}$ e n_i^S e computa-se o máximo de n_2 para cada possibilidade. Como r é uma constante, o tempo total resultante também é $O(n^{3+4+\dots+r})$.

Por fim, se $G = G_N^K$, a distribuição de cliques oferece apenas algumas possibilidades a menos que para o caso em que $G = G_A^K$ (aquelas em que todos os vértices de K são usados em conjunto em uma clique). Deste modo, neste caso o tempo gasto também é da mesma ordem.

4.3.3 Outros casos

Como observado na Seção 4.3, a função $f(G, n_3, \dots, n_r)$ também deve ser computada para os grafos C_5 , P_5 e $\overline{P_5}$ pois estes estão presentes em grafos P_4 -arrumados. Nestes casos, os valores de f são dados a seguir.

$$\begin{aligned} f(C_5, 0, 0, \dots, 0) &= 2 & f(P_5, 0, 0, \dots, 0) &= 2 \\ f(\overline{P}_5, 0, 0, \dots, 0) &= 2 & f(\overline{P}_5, 1, 0, \dots, 0) &= 1 \end{aligned}$$

Em todos demais casos, f está indefinida pois o empacotamento desejado não existe.

4.3.4 O Algoritmo

Como já foi exposto, o algoritmo apresentado para computar o PEC em grafos da classe P_4 -arrumada usa a mesma ideia do algoritmo desenvolvido por Guruswami et al.

As três subseções anteriores descreveram os passos para computar f nas aranhas, pseudo-aranhas e demais grafos P_4 -arrumada de acordo com os Lemas 4.2, 4.5, 4.4 e 4.7. Além disso, analisaram a complexidade temporal em cada caso, considerando as melhorias decorrentes dos Lemas 4.3, 4.6 e 4.8. Como resultado, no pior caso, o tempo gasto pelo algoritmo é $O(n^{3+4+\dots+r})$ para computar uma invocação da função f no grafo correspondente a qualquer nó da ADMQB de um grafo da classe P_4 -arrumada.

De maneira análoga aos cografos, o número de nós da ADMQB é $O(n)$ e são computadas, no máximo, $O(n^{r-2})$ invocações de f por nó. Como resultado, o algoritmo leva $O(n^{3+4+\dots+r}n^{r-1})$ para computar o PEC. Note que esta complexidade é maior que a do algoritmo para cografos por um fator de n^{r-2} .

4.4 Conclusão

Neste capítulo, o problema do PEC foi descrito, assim como a técnica de solução conhecida para cografos. Em seguida, o problema foi resolvido para a classe de grafos P_4 -arrumada, utilizando uma técnica semelhante e tratando adequadamente aranhas e pseudo-aranhas – subgrafos característicos de grafos dessa classe – e descreveu-se um algoritmo polinomial para o problema, dado um r fixo.

Este é um dos resultados que compõe esta tese e a técnica de solução utiliza a decomposição modular como importante ferramenta que guia o algoritmo pelo paradigma da divisão e conquista.

Uma versão menor do trabalho foi apresentada no *Tercer Taller Latinoamericano de Clanes en Gráficas*, ocorrido em Guanajuato, México em outubro de 2008. Além disso, uma versão completa foi publicada como Relatório Técnico do Instituto de Computação da UNICAMP [62] e uma versão está sendo preparada para submissão a periódico.

Capítulo 5

Coloração Total Forte de Grafos Indiferença

Neste capítulo, será estudada uma variante do problema clássico da coloração total de grafos, chamada coloração total distinguidora de vértices e também conhecida por coloração total forte.

O estudo deste problema será restringido à classe dos grafos indiferença, uma subclasse bastante conhecida e estudada dos grafos de intervalos. Como principal resultado, provou-se limitantes superiores para a coloração total forte de grafos indiferença e, para alguns destes grafos com características específicas, o problema foi resolvido. Os limitantes dados são suficientes para estabelecer que grafos indiferença satisfazem a conjectura de Zhang et al. sobre coloração total forte.

Além disso, as técnicas utilizadas para as provas teóricas são construtivas, de modo que um algoritmo de coloração aproximado é facilmente obtido a partir das construções dos limitantes. Este algoritmo tem complexidade linear e resolve o problema para um subconjunto dos grafos indiferença.

5.1 Introdução

Uma *coloração total* de um grafo é uma coloração de seus vértices e arestas de forma que vértices adjacentes, arestas adjacentes e arestas e vértices incidentes tenham cores distintas. *Coloração de arestas* e *coloração de vértices* são casos parciais da coloração total, nos quais apenas arestas ou vértices são coloridos, respectivamente. Representa-se por $C(v)$ o conjunto de cores associado ao vértice v numa coloração total do grafo, ou seja, a cor do vértice v e as cores de todas as arestas incidentes a v . Uma coloração total de um grafo é *distinguidora de vértices adjacentes*, se para todo par de vértices adjacentes, u e v , do grafo, os conjuntos $C(u)$ e $C(v)$ são distintos. Neste texto, uma coloração total distingui-

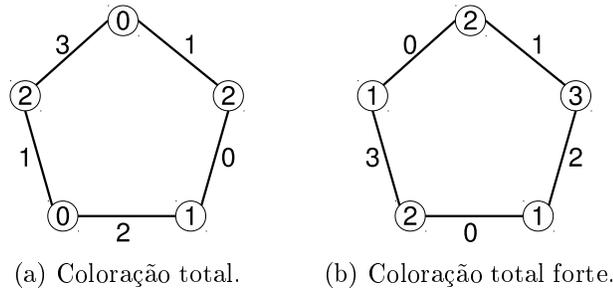


Figura 5.1: Coloração total e total forte de um grafo.

dora de vértices adjacentes será chamada de *coloração total forte*. O número mínimo de cores para colorir um grafo em cada um destes problemas é chamado, respectivamente, de *número cromático total*, *índice cromático*, *número cromático* e *número cromático total forte*. Estes parâmetros são denotados por $\chi''(G)$, $\chi'(G)$, $\chi(G)$ e $\chi_a''(G)$, respectivamente.

Na Figura 5.1 apresentam-se duas colorações totais para um C_5 , ambas com o número mínimo de cores. No item (a), a coloração não é uma coloração total forte, pois observam-se vértices adjacentes que usam o mesmo conjunto de cores, $\{0, 1, 2\}$. Em (b) a coloração é total forte para o mesmo grafo. Note que nem sempre é possível obter uma coloração total forte com $\chi''(G)$ cores.

O tópico deste capítulo é abordar o problema do número cromático total forte para grafos simples. Este problema foi introduzido por Zhang et al. [74], que também determinaram o valor do número cromático total forte para algumas famílias de grafos tais como grafos completos, ciclos, grafos bipartidos completos, leques (grafos *join* entre um caminho e um grafo trivial), rodas (grafos *join* entre um ciclo e um grafo trivial) e árvores. Além disso, os autores propuseram a seguinte conjectura:

Conjectura 5.1. *Se G é um grafo conexo, simples e com pelo menos dois vértices, então $\chi_a''(G) \leq \Delta(G) + 3$.*

São bastante conhecidos na literatura os limitantes superiores para o índice cromático de um grafo simples G , $\chi'(G) \leq \Delta(G) + 1$, dado por Vizing [69], e os dados pela *Conjectura da Coloração Total*, $\chi''(G) \leq \Delta(G) + 2$, apresentada, independentemente, por Behzad [4] e Vizing [69, 70].

O limitante superior dado na Conjectura 5.1 é justo, uma vez que o número cromático total forte de um grafo completo é $\chi_a''(K_p) = p + 2 = \Delta(K_p) + 3$, quando $p \geq 3$ é ímpar [74]. Tomando $2t + 1 = p$, e associando cada vértice do K_p a um inteiro entre 1 e p , esta coloração atribui a cor j ao vértice j e às arestas $\{j+2, j+2t-1\}$, $\{j+3, j+2t-2\}$, \dots , $\{j+t-1, j+t+2\}$ e $\{j+t, j+t+1\}$, para cada j de 1 a p . Além disso, as arestas $\{2t, 1\}$ e $\{p, 2\}$ recebem as cores p e 1, respectivamente, e, para j de 1 a $2t-1$, a aresta $\{j, j+2\}$

recebe a cor $p + 1$, se $j \bmod 4 = 1$ ou 2 , ou a cor $p + 2$, se $j \bmod 4 = 0$ ou 3 . Além disso, a Conjectura 5.1 foi confirmada para grafos com $\Delta(G) = 3$ [18, 43, 71], hipercubos [17], grafos periplanares [72] e grafos Halin [19].

Note que $\chi''(K_p) = p$ para um grafo completo K_p com número ímpar de vértices. Assim, $\chi''_a(K_p) = \chi''(K_p) + 2$. Já quando p é par, $\chi''(K_p) = \chi''_a(K_p) = p + 1$ e uma coloração total forte do K_p com o número mínimo de cores pode ser obtida a partir de uma coloração total do grafo K_{p+1} (com $p + 1$ cores), apenas removendo um de seus vértices.

A coloração total distinguidora de vértices adjacentes de um grafo simples está relacionada com a coloração de arestas distinguidora de vértices, que foi considerada pela primeira vez por Burris e Schelp [13] e, posteriormente, foi discutida por muitos outros autores (Bazgan et al. [3] provaram que $n + 1$ é um limitante superior do problema, por exemplo). Nesta forma da coloração, quaisquer dois vértices devem possuir conjuntos de cores distintos. Posteriormente, foi estudado o problema em que essa exigência é aplicada apenas a vértices adjacentes [17, 75].

Sabe-se há tempo que é NP-Difícil determinar o índice cromático e o número cromático total de um grafo [67, 69]. O estado da complexidade do problema de determinar o número cromático total forte de um grafo ainda não é conhecido. Parte da dificuldade do problema da coloração total forte provavelmente se deve a existência de uma dificuldade adicional para construir esse tipo de coloração: dado um grafo e uma coloração total forte para ele, a mesma coloração pode não ser válida para alguns de seus subgrafos. Isto nunca acontece com as colorações de vértices, arestas e total, nas quais todo subgrafo aceita a mesma coloração do grafo original.

Neste capítulo, o estudo será direcionado para grafos indiferença. Estes são uma subclasse de grafos de intervalos (veja, por exemplo, [35]) no qual todos os intervalos têm o mesmo tamanho (unitário). Eles também podem ser caracterizados como grafos de intervalos livres de $K_{1,3}$ induzido [34] e são reconhecidos em tempo linear [20, 26]. Grafos indiferença também podem ser caracterizados por uma ordem linear: seus vértices podem ser linearmente ordenados de forma que os vértices contidos em uma mesma clique maximal são consecutivos [65]. Esta ordem é chamada *ordem indiferença*.

Problemas de coloração foram considerados para grafos indiferença, mas ambos os problemas de determinar o índice cromático e o número cromático total destes grafos continuam abertos. Todavia, alguns resultados parciais são conhecidos para um grafo indiferença G . Se $\Delta(G)$ é ímpar, então $\chi'(G) = \Delta(G)$, caso contrário, $\chi'(G) \leq \Delta(G) + 1$ (a última inequação é dada pelo teorema de Vizing). Ainda, se $\Delta(G)$ é par, $\chi''(G) = \Delta(G) + 1$, caso contrário, $\chi''(G) \leq \Delta(G) + 2$, i.e., a Conjectura da Coloração Total está confirmada [27]. Estes resultados foram obtidos através do uso de uma técnica de rolagem (*pullback*) para computar o número cromático total de um grafo indiferença.

Na Seção 5.2 são apresentadas notações usadas no texto e resultados conhecidos relativos ao tema. Os resultados novos são dados na Seção 5.3 e, por fim, as conclusões do trabalho são dadas na Seção 5.4.

5.2 Preliminares

Neste trabalho, será usada a técnica de rolagem a fim de dar uma coloração total forte para grafos indiferença. Esta técnica foi usada anteriormente para atacar os problemas da coloração de arestas e da coloração total de grafos dualmente cordais, uma superclasse dos grafos indiferença.

Para maior agilidade, alguns resultados conhecidos [27] serão reapresentados a seguir.

Um *pullback* de G para G' é uma função $f : V(G) \rightarrow V(G')$, tal que se $\{u, v\} \in E(G)$, então $\{f(u), f(v)\} \in E(G')$ e f é injetiva quando restrita a $N_G[v]$, para todo $v \in V(G)$. Este tipo de função nos permite usar cores de vértices e arestas numa coloração de G' para os vértices e arestas de G , pois as restrições impostas às colorações de vértices, de arestas e total são preservadas pela função.

Será denotado por G^2 o grafo que satisfaz $V(G^2) = V(G)$ e $\{u, v\} \in E(G^2)$ se, e somente se, u e v são distintos e sua distância em G é no máximo 2.

Lema 5.2. *Existe um **pullback** de um grafo G para um grafo completo K_p se, e somente se, $\chi(G^2) \leq p$.*

Lema 5.3. *Se G é um grafo dualmente cordal, então $\chi(G^2) \leq \Delta(G) + 1$.*

Uma vez que um grafo indiferença G é um grafo dualmente cordal, dos lemas 5.2 e 5.3, obtém-se diretamente o seguinte resultado.

Corolário 5.4. *Se G é um grafo indiferença, existe um **pullback** de G para $K_{\Delta(G)+1}$.*

Neste texto, quando uma cor α está contida em $C(v)$, diz-se que α aparece em v , caso contrário, diz-se que α falta em v .

5.3 Grafos indiferença

Nesta seção, um *pullback* a partir de um grafo completo será considerado para dar uma coloração total forte para grafos indiferença. Esta coloração será ótima se o grafo satisfizer algumas propriedades adicionais, discutidas adiante. Caso contrário, a coloração será uma aproximação, suficiente para provar a Conjectura 5.1 para grafos indiferença.

Para este objetivo, uma coloração total de um grafo completo K_p , sendo p um inteiro ímpar, será considerada. É sabido que $\chi''(K_p) = p$. Rotule os vértices de K_p com

$\{v_0, \dots, v_{p-1}\}$ e considere uma coloração total de K_p dada pela seguinte função $c : V(K_p) \cup E(K_p) \rightarrow \{0, \dots, p-1\}$.

$$\begin{aligned} c(v_i) &= 2i \bmod p, \text{ e} \\ c(v_i, v_j) &= (i+j) \bmod p. \end{aligned}$$

É fácil observar que todas as cores disponíveis aparecem em cada vértice de K_p , uma vez que c é uma coloração total de K_p com p cores.

Lema 5.5. *Sejam G um grafo indiferença e p um inteiro ímpar tal que $p \geq \Delta(G) + 2$. Então, G possui uma coloração total forte com p cores.*

Demonstração. Seja v_0, \dots, v_{n-1} uma ordem indiferença de um grafo indiferença G e seja p um inteiro ímpar tal que $p \geq \Delta(G) + 2$. Considere uma função $\pi : V(G) \cup E(G) \rightarrow \{0, \dots, p-1\}$ definida abaixo.

$$\begin{aligned} \pi(v_i) &= c(v_{i \bmod p}) = 2i \bmod p, \text{ e} \\ \pi(v_i, v_j) &= c(v_{i \bmod p}, v_{j \bmod p}) = (i+j) \bmod p \end{aligned}$$

Pelo Corolário 5.4, existe uma função pullback de G para um grafo completo K_p . A função π é um pullback de G para um grafo completo K_p e, logo, π é uma coloração total para G . De fato, a sequência $(0, 2, \dots, p-1, 1, 3, \dots, p-2)$ de cores é atribuída repetidamente, pela função π , aos vértices de G seguindo a ordem indiferença considerada. Uma vez que, para qualquer vértice $v_i \in V(G)$, $|N_G[v_i]| \leq \Delta(G) + 1 < p$, a função π é injetiva quando restrita a $N[v_i]$. Além disso, se $\{v_i, v_j\} \in E(G)$, a cor $\pi(v_i, v_j)$ é a mesma cor da aresta $\{v_r, v_s\} \in E(K_p)$ com $r = i \bmod p$ e $s = j \bmod p$.

Na Figura 5.2 estão ilustrados no item (a) uma coloração total de um K_5 conforme a função c e, no item (b), a coloração total forte produzida pela função π sobre um grafo indiferença de grau máximo 3. Os vértices do grafo indiferença estão em uma ordem indiferença.

Agora, será demonstrado que π é uma coloração total forte de G . Para tanto, mostra-se que para cada vértice v_i de G existe uma cor α que falta em v_i e ao mesmo tempo aparece em cada vértice v_j adjacente a v_i tal que $j > i$. Desta forma, para todos pares de vértices adjacentes de G , existe uma cor que aparece em um deles e esta mesma cor falta no outro, provando o lema.

Seja v_i um vértice de G e v_j o vértice mais à direita de v_i na ordem indiferença, que seja adjacente a v_i . Por construção, v_i não é adjacente a v_{j+1} , se existir. Ainda mais, v_i não é adjacente a v_{j+1-p} (se existir), caso contrário v_i teria grau $p-1 \geq \Delta(G) + 1$, uma contradição.

Para continuar a prova, considere a ilustração na Figura (c).

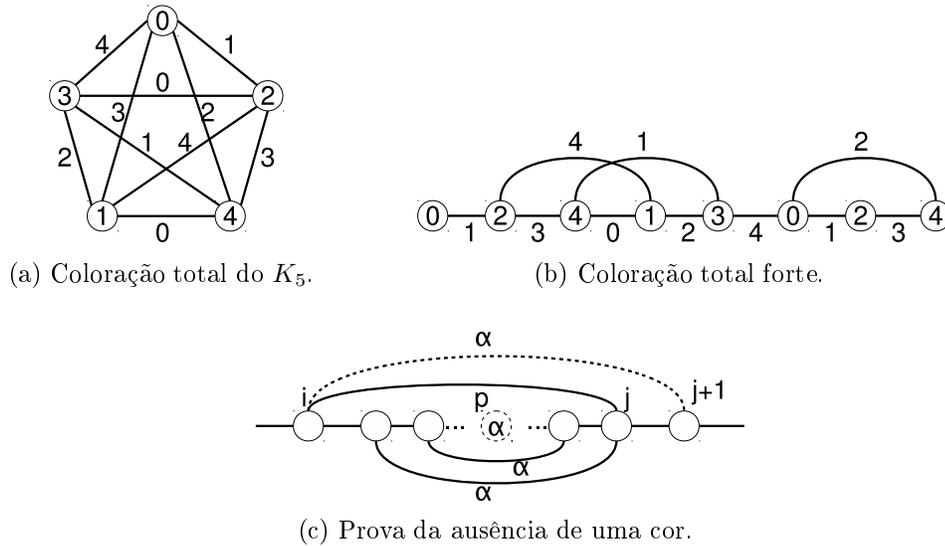


Figura 5.2: Coloração total forte como exemplo de pullback.

Dado que π é um pullback de um K_p , a cor α que π associaria a (v_i, v_{j+1}) poderia estar associada apenas a arestas (v_i, v_k) tais que $k = j + 1 + lp$, $l \in \mathbb{Z}$. Isto implica que $v_k \notin N_G[v_i]$ e, portanto, v_i não possui arestas incidentes coloridas por π com a cor α . Além disso, $\pi(v_i) \neq \alpha$. De fato, se $\pi(v_i) = \alpha = \pi(v_i, v_{j+1})$, então $2i \bmod p = i + j + 1 \bmod p$. Logo, $i = j + 1 + lp$, $l \in \mathbb{Z}$. Por outro lado, pela escolha de $j + 1$, isto é impossível, já que $i < j + 1$ e $j + 1 - p < i$. Portanto, $\alpha \notin C(v_i)$.

Para terminar a prova, seja $\alpha = \pi(v_i, v_{j+1})$. Agora, será mostrado que a cor α aparece em todos os vértices entre v_{i+1} e v_j na ordem indiferença.

Pela definição de ordem indiferença, $G[\{v_{i+1}, \dots, v_j\}]$ é um grafo completo pois $\{v_i, v_j\} \in E(G)$. Logo, pela definição de π , $\alpha = \pi(v_i, v_{j+1}) = \pi(v_{i+k}, v_{j+1-k})$, para $1 \leq k < (j - i + 1)/2$. Adicionalmente, se $j + 1 - i$ for par, o vértice $v_{(j+1-i)/2}$ também possui a cor α . Logo, a cor α aparece em todos os vértices entre v_{i+1} e v_j e o lema está provado. \square

Note que, para qualquer grafo G com pelo menos dois vértices de grau máximo adjacentes, é válido que $\chi''_a(G) \geq \Delta(G) + 2$. As afirmações a seguir são consequências imediatas do Lema 5.5.

Corolário 5.6. *Se G é um grafo indiferença com grau máximo ímpar, então G tem uma coloração total forte com $\Delta(G) + 2$ cores. Além disso, se G possui ao menos dois vértices de grau máximo adjacentes, então $\chi''_a(G) = \Delta(G) + 2$.*

Corolário 5.7. *Se G é um grafo indiferença com grau máximo par, então G tem uma coloração total forte com $\Delta(G) + 3$ cores.*

Estes resultados fornecem limitantes superiores para a coloração total forte de grafos indiferença suficientes para afirmar que tais grafos satisfazem a conjectura da coloração total forte (Conjectura 5.1).

Teorema 5.8. *Se G é um grafo indiferença, então $\chi_a''(G) \leq \Delta(G) + 3$.*

A fim de complementar os resultados anteriores, agora, será considerado um caso especial de grafos indiferença com grau par, para o qual será apresentada uma coloração total com o número mínimo de cores, também obtida pelo uso de um pullback de um grafo completo, que é uma coloração total forte com número mínimo de cores.

Lema 5.9. *Se G é um grafo indiferença com grau máximo par e não possui dois vértices de grau máximo adjacentes, então $\chi_a''(G) = \Delta(G) + 1$.*

Demonstração. Neste caso, $\chi_a''(G) = \chi''(G)$. Por argumentos similares aos do Lema 5.5, se $p = \Delta(G) + 1$ (que é um número inteiro ímpar), a função π ainda é um pullback de G para K_p e, portanto, π é uma coloração total de G .

Uma vez que não existem dois vértices de grau máximo adjacentes em G , então $\chi_a''(G) \geq \Delta(G) + 1$. Se v for um vértice de grau máximo de G , então todas as cores usadas aparecem em v . Um vizinho u de v tem pelo menos uma cor faltando (pois ele não tem grau máximo), logo $C(u) \neq C(v)$.

Como no Lema 5.5, tome uma ordem indiferença de G e sejam v_i um vértice que não tem grau máximo e v_j o vizinho de v_i mais à direita na ordem. Uma vez que, $i < j + 1$ e $j + 1 - p = j - \Delta(G) < i$ (pois $j < i + \Delta(G)$), a cor $\alpha = \pi(v_i, v_{j+1})$ não aparece em v_i . Por argumentos semelhantes aos usados no Lema 5.5, a cor α aparece em todos os vértices entre v_{i+1} e v_j , o que completa a prova. \square

5.4 Conclusões

Neste capítulo, foi demonstrado que a Conjectura 5.1 vale para todos os grafos indiferença. A técnica usada proporcionou a elaboração de um algoritmo aproximado de tempo linear que colore um grafo indiferença qualquer. Além disso, se o grafo tem grau máximo par e não possui vértices de grau máximo adjacentes ou se o grafo tem grau máximo ímpar e pelo menos dois vértices de grau máximo adjacentes, o algoritmo garantidamente encontra uma coloração com número mínimo de cores.

Estes resultados foram apresentados no *Latin American Workshop on Cliques in Graphs*, ocorrido em Itaipava em novembro de 2010 e foram submetidos à revista *Matemática Contemporânea* [61].

Uma continuidade natural para este trabalho é considerar o problema da coloração de arestas distinguidora de vértices adjacentes para grafos indiferença. Infelizmente, a técnica

utilizada parece não ser facilmente adaptável. Por exemplo, no grafo da Figura 5.2b, o primeiro vértice com a cor 3 e o vértice seguinte são adjacentes e suas arestas incidentes compartilham o mesmo conjunto de cores $\{1, 2, 4\}$. Além disso, o Lema 5.5 usa a cor do vértice $v_{(j+1-i)/2}$ para garantir a distinção entre $C(v_i)$ e $C(v_{(j+1-i)/2})$, quando $j - i$ é par. Algo que não é possível na coloração de arestas. Outra vertente é analisar a Conjectura 5.1 no contexto dos grafos de intervalos.

Capítulo 6

Conclusões

O foco deste trabalho foi a solução de problemas combinatórios em grafos, principalmente pelo uso da decomposição modular, a fim de contribuir para expandir o uso desta poderosa ferramenta, que tem sido aplicada com sucesso em uma ampla gama de problemas. Nesta direção, naturalmente estudam-se as classes de grafos com “poucos P_4 's”, para as quais a decomposição modular tem um vínculo maior.

O leitor foi familiarizado com a decomposição modular e com as classes de grafos com “poucos P_4 's” no Capítulo 2, no qual também estão compiladas as notações e fundamentos necessários para o desenvolvimento dos assuntos posteriores.

O Problema dos Separadores Minimais foi estudado no Capítulo 3. Embora sejam conhecidos algoritmos polinomiais para este problema em qualquer classe de grafos cujo número de separadores minimais seja limitado polinomialmente, o mais eficiente entre estes algoritmos leva tempo $O(n^3)$ por separador. Nossas contribuições para este problema são um algoritmo linear que lista todos os separadores minimais dos grafos P_4 -carregados estendidos e o estabelecimento de limites justos para a quantidade e tamanho total destes separadores minimais em grafos da classe P_4 -carregada estendida e em algumas de suas subclasses: P_4 -carregada, P_4 -arrumada e P_4 -leve. Os limitantes obtidos para o número de separadores minimais são todos $O(n)$, melhorando os anteriormente conhecidos, $O(n+m)$. O algoritmo proposto é resultado de uma técnica que foi desenvolvida para obter os separadores minimais de um grafo qualquer a partir dos separadores minimais dos grafos quocientes de sua árvore de decomposição modular, generalizando o trabalho anterior para a classe P_4 -esparsa.

O Capítulo 4 aborda o segundo problema considerado, o Problema do Empacotamento de Cliques, que é inerentemente difícil mesmo para classes de grafos que podem ser vistas como “simples”, no sentido que são classes cujo reconhecimento, bem como problemas clássicos de otimização são resolvidos em tempo polinomial ou mesmo linear. A decomposição modular já havia sido utilizada para resolver o Problema do Empacotamento de Cliques

quando a entrada é um cografo. Nossa contribuição é uma extensão deste algoritmo para tratar, em tempo polinomial, grafos de uma superclasse, a P_4 -arrumada. Nota-se que uma nova extensão do algoritmo no mesmo sentido é pouco provável, visto que as superclasses mais estudadas da P_4 -arrumada contém os grafos `split`, para os quais este problema é intratável.

O último problema considerado é o Problema da Coloração Total Forte, no Capítulo 5. Este estudo foge do escopo da decomposição modular e das classes de grafos com “poucos P_4 's”, oferecendo novos desafios e oportunidades. O problema é uma variação recentemente apresentada do clássico problema de coloração total e ainda não teve sua complexidade computacional determinada. Nosso estudo demonstrou que uma técnica conhecida como *pullback* e utilizada para coloração total de grafos dualmente cordais, pode ser adaptada para coloração total forte de grafos indiferença. Estes grafos formam uma subclasse bem conhecida dos grafos de intervalos que, por sua vez, são uma subclasse dos grafos dualmente cordais. A coloração produzida pelo algoritmo proposto para um grafo indiferença G usa no máximo $\Delta(G) + 3$ cores, validando a Conjectura de Zhang et al. nesta classe de grafos. Além disso, o algoritmo produz, em alguns casos bem determinados, a coloração ótima.

As contribuições apresentadas por esta tese de doutorado permitem que alguns problemas em grafos sejam resolvidos em classes de grafos para as quais não havia algoritmo anterior ou de uma forma mais eficiente. Para tal, foram propostas extensões e generalizações de técnicas conhecidas a fim de ampliar seu escopo.

Referências Bibliográficas

- [1] J. C. S. Araújo e C. L. Sales. Grundy number on P_4 -classes. *Electronic Notes in Discrete Mathematics*, 35:21–27, 2009. LAGOS'09 - V Latin-American Algorithms, Graphs and Optimization Symposium.
- [2] S. Arnborg, D. G. Corneil e A. Proskurowski. Complexity of finding embeddings in a k -tree. *SIAM J. Algebra. Discr. Methods*, 8(2):277–284, 1987.
- [3] C. Bazgan, A. Harkat-Benhamdine, H. Li e M. Wozniak. On the vertex-distinguishing proper edge-colorings of graphs. *J. Comb. Theory B*, 75(2):288–301, 1999.
- [4] M. Behzad. *Graphs and their chromatic numbers*. Tese de doutorado, Michigan State University, 1965.
- [5] A. Berry, J.-P. Bordat e O. Cogis. Generating all the minimal separators of a graph. *Internat. J. Foundations Comput. Sci.*, 11(3):397–403, 2000.
- [6] A. Berry, J.-P. Bordat e P. Heggernes. Recognizing weakly triangulated graphs by edge separability. *Nordic J. of Computing*, 7:164–177, September de 2000.
- [7] A. Berry e R. Pogorelcnik. A simple algorithm to generate the minimal separators and the maximal cliques of a chordal graph. *Inform. Process. Lett.*, 111(11):508–511, 2011.
- [8] H. L. Bodlaender e U. Rotics. Computing the treewidth and the minimum fill-in with the modular decomposition. *Algorithmica*, 36:375–408, 2008.
- [9] V. Bouchitté e I. Todinca. Treewidth and minimum fill-in: Grouping the minimal separators. *SIAM J. Comput.*, 31:212–232, January de 2002.
- [10] V. Bouchitté e I. Todinca. Listing all potential maximal cliques of a graph. *Theor. Comput. Sci.*, 276(1-2):17–32, 2002.
- [11] A. Brandstädt e R. Mosca. On variations of P_4 -sparse graphs. *Discrete Appl. Math.*, 129:521–532, August de 2003.

- [12] R. S. F. Bravo, S. Klein, L. T. Nogueira e F. Protti. Characterization and recognition of P_4 -sparse graphs partitionable into k independent sets and l cliques. *Discrete Appl. Math.*, 159(4):165–173, 2011.
- [13] A. C. Burris e R. H. Schelp. Vertex-distinguishing proper edge-colorings. *J. Graph Theory*, 26:73–82, 1997.
- [14] V. Campos, C. L. Sales, K. Maia, N. Martins e R. Sampaio. Restricted coloring problems on graphs with few P_4 's. LAGOS'11 - VI Latin-American Algorithms, Graphs and Optimization Symposium, 2011.
- [15] M. R. Cerioli, L. Faria, T. O. Ferreira, C. A. J. Martinhon, F. Protti e B. Reed. Partition into cliques for cubic graphs: Planar case, complexity and approximation. *Discrete Appl. Math.*, 156:2270–2278, 2008.
- [16] F. Chataigner, G. Manić, Y. Wakabayashi e R. Yuster. Approximation algorithms and hardness results for the clique packing problem. *Discrete Appl. Math.*, 157:1396–1406, 2009.
- [17] M. Chen e X. Guo. Adjacent vertex-distinguishing edge and total chromatic number of hypercubes. *Inform. Process. Lett.*, 109:599–602, 2009.
- [18] X. Chen. On the adjacent vertex distinguishing total coloring numbers of graphs with $\Delta = 3$. *Discrete Math.*, 308(17):4003–4007, 2008.
- [19] X. Chen e Z. Zhang. AVDTC numbers of generalized halin graphs with maximum degree at least 6. *Acta Mathematicae Applicatae Sinica, English Series*, 24:55–58, 2008.
- [20] D. G. Corneil, H. Kim, S. Natarajan, S. Olariu e A. P. Sprague. Simple linear time recognition of unit interval graphs. *Inform. Process. Lett.*, 55:99–104, 1995.
- [21] D. G. Corneil, H. Lerchs e L. S. Burlingham. Complement reducible graphs. *Discrete Appl. Math.*, 3(3):163–174, 1981.
- [22] D. G. Corneil, H. Lerchs e L. S. Burlingham. A linear recognition algorithm for cographs. *SIAM J. Computing*, 14(4):926–934, 1985.
- [23] D. D. Cowan, L. O. James e R. G. Stanton. Graph decomposition for undirected graphs. Em *3rd S-E Conf. Combinatorics, Graph Theory and Computing*, páginas 281–290. Utilitas Math., 1972.

- [24] E. Dahlhaus e M. Karpinski. Matching and multidimensional matching in chordal and strongly chordal graphs. *Discrete Appl. Math.*, 84(1-3):79–91, 1998.
- [25] S. Dantas, S. Klein, C. P. de Mello e A. Morgana. The graph sandwich problem for P_4 -sparse graphs. *Discrete Math.*, 309(11):3664–3673, 2009.
- [26] C. M. de Figueiredo, J. Meidanis e C. P. de Mello. A linear-time algorithm for proper interval graph recognition. *Inform. Process. Lett.*, 56:179–184, 1995.
- [27] C. M. H. de Figueiredo, J. Meidanis e C. P. de Mello. Total-chromatic number and chromatic index of dually chordal graphs. *Inform. Process. Lett.*, 70:147–152, 1999.
- [28] C. P. de Mello, A. Morgana e M. Liverani. The clique operator on graphs with few P_4 's. *Discrete Appl. Math.*, 154(3):485–492, 2006.
- [29] A. Ehrenfeucht, H. N. Gabow, R. M. McConnell e S. J. Sullivan. An $O(n^2)$ divide-and-conquer algorithm for the prime tree decomposition of two-structures and modular decomposition of graphs. *J. Algorithms*, 16(2):283–294, 1994.
- [30] J. L. Fouquet, I. Parfenoff e H. Thuillier. An $O(n)$ time algorithm for maximum matching in P_4 -tidy graphs. *Inf. Process. Lett.*, 62(6):281–287, 1997.
- [31] V. Giakoumakis. P_4 -laden graphs: A new class of brittle graphs. *Inf. Process. Lett.*, 80:29–36, 1996.
- [32] V. Giakoumakis, F. Roussel e H. Thuillier. On P_4 -tidy graphs. *Discrete Math. Theor. Comput. Sci.*, 1(1):17–41, 1997.
- [33] V. Giakoumakis e J. M. Vanherpe. On extended P_4 -reducible and extended P_4 -sparse graphs. *Theor. Comput. Sci.*, 180(1-2):269–286, 1997.
- [34] P. C. Gilmore e A. J. Hoffman. A characterization of comparability graphs and of interval graphs. *The Canadian Journal of Mathematics*, 16:539–548, 1964.
- [35] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York, 1980.
- [36] V. Guruswami, C. Pandu Rangan, M. S. Chang, G. J. Chang e C. K. Wong. The K_r -Packing Problem. *Computing*, 66:79–89, 2001.
- [37] D. Gusfield. The multi-state perfect phylogeny problem with missing and removable data: Solutions via integer-programming and chordal graph theory. *J. Comput. Biol.*, 17, 2010.

- [38] R. Gysel e D. Gusfield. Extensions and improvements to the chordal graph approach to the multi-state perfect phylogeny problem. Em *Bioinformatics Research and Applications*, volume 6053 de *Lect. Notes Comput. Sci.*, páginas 52–60. Springer Berlin / Heidelberg, 2010.
- [39] M. Habib e C. Paul. A survey on algorithmic aspects of modular decomposition. *Computer Science Review*, 4(1):41–59, 2010.
- [40] P. L. Hammer e B. Simeone. The splittance of a graph. *Combinatorica*, 1(3):275–284, 1981.
- [41] P. Heggernes. Minimal triangulations of graphs: A survey. *Discrete Math.*, 306(3):297–317, 2006.
- [42] C. T. Hoàng. A class of perfect graphs. Dissertação de mestrado, School Comput. Sci., Montreal, 1983.
- [43] J. Hulgan. Concise proofs for adjacent vertex-distinguishing total colorings. *Discrete Math.*, 309:2548–2550, 2009.
- [44] B. Jamison e S. Olariu. A new class of brittle graphs. *Stud. Appl. Math.*, 81:89–92, 1989.
- [45] B. Jamison e S. Olariu. On a unique tree representation for P_4 -extendible graphs. *Discrete Appl. Math.*, 34(1-3):151–164, 1991.
- [46] B. Jamison e S. Olariu. A tree representation for P_4 -sparse graphs. *Discrete Appl. Math.*, 35(2):115–129, 1992.
- [47] B. Jamison e S. Olariu. Linear time optimization algorithms for P_4 -sparse graphs. *Discrete Appl. Math.*, 61:155–175, 1995.
- [48] B. Jamison e S. Olariu. A linear-time recognition algorithm for P_4 -reducible graphs. *Theor. Comput. Sci.*, 145(1-2):329–344, 1995.
- [49] D. G. Kirkpatrick e P. Hell. On the completeness of a generalized matching problem. Em *STOC '78: Proceedings of the tenth annual ACM symposium on Theory of computing*, páginas 240–245, New York, NY, USA, 1978. ACM.
- [50] D. G. Kirkpatrick e P. Hell. On the complexity of general graph factor problems. *SIAM J. Comput.*, 12:601–609, 1983.
- [51] T. Kloks e D. Kratsch. Listing all minimal separators of a graph. *SIAM J. Comput.*, 27(3):605–613, 1998.

- [52] P. S. Kumar e C. E. V. Madhavan. Minimal vertex separators of chordal graphs. *Discrete Appl. Math.*, 89:155–168, 1998.
- [53] H. Lerchs. On cliques and kernels. Relatório técnico, Departament of Computer Science, University of Toronto, Março de 1971.
- [54] G. Maniç e Y. Wakabayashi. Packing triangles in low degree graphs and indifference graphs. *Discrete Math.*, 308:1455–1471, 2008.
- [55] F. Mazoit. Listing all the minimal separators of a 3-connected planar graph. *Discrete Math.*, 306:372–380, 2006.
- [56] R. M. McConnell e J. P. Spinrad. Linear-time modular decomposition and efficient transitive orientation of comparability graphs. Em *SODA '94: Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms*, páginas 536–545, Philadelphia, PA, USA, 1994. Society for Industrial and Applied Mathematics.
- [57] J. H. Muller e J. P. Spinrad. Incremental modular decomposition. *J. ACM*, 36(1):1–19, 1989.
- [58] S. D. Nikolopoulos e L. Palios. Minimal separators in P_4 -sparse graphs. *Discrete Math.*, 306(3):381–392, 2006.
- [59] A. Parra e P. Scheffler. How to use the minimal separators of a graph for its chordal triangulation. Em *Automata, Languages and Programming*, volume 944 de *Lect. Notes Comput. Sci.*, páginas 123–134. Springer Berlin / Heidelberg, 1995.
- [60] V. Pedrotti. Decomposição modular de grafos não orientados. Dissertação de mestrado, Universidade Estadual de Campinas, 2007.
- [61] V. Pedrotti e C. P. de Mello. Adjacent-vertex-distinguishing total coloring of indifference graphs. *Matemática Contemporânea*. Aceito para publicação.
- [62] V. Pedrotti e C. P. de Mello. K_r -packing of P_4 -tidy graphs. Relatório Técnico IC-09-19, Instituto de Computação, Universidade de Campinas, Maio de 2009.
- [63] V. Pedrotti e C. P. de Mello. Minimal separators in P_4 -tidy graphs. *Electronic Notes in Discrete Math.*, 35:71–76, 2009.
- [64] V. Pedrotti e C. P. de Mello. Minimal separators in extended P_4 -laden graphs. Relatório Técnico IC-10-05, Instituto de Computação, Universidade de Campinas, Fevereiro de 2010.

- [65] F. S. Roberts. On the compatibility between a graph and a simple order. *J. Comb. Theory B*, 11:28–38, 1971.
- [66] D. J. Rose. Triangulated graphs and the elimination process. *J. Math. Anal. Appl.*, 32(3):597–609, 1970.
- [67] A. Sánchez-Arroyo. Determining the total colouring number is NP-Hard. *Discrete Math.*, 78(3):315–319, 1989.
- [68] M. Tedder, C. Corneil, M. Habib e C. Paul. Simpler linear-time modular decomposition via recursive factorizing permutations. Em *Proceedings of the 35th international colloquium on Automata, Languages and Programming, Part I*, páginas 634–645, Berlin, Heidelberg, 2008. Springer-Verlag.
- [69] V. G. Vizing. On an estimate of chromatic class of a p -graph. *Diskrete Analiz*, 3:25–30, 1964. (em Russo).
- [70] V. G. Vizing. The chromatic class of a multigraph. *Kibernetika (Kiev)*, 3:29–39, 1965. Versão em inglês na *Cybernetics* 1, 32–41.
- [71] H. Wang. On the adjacent vertex-distinguishing total chromatic numbers of the graphs with $\Delta(G) = 3$. *J. Comb. Optim.*, 14:87–109, 2007.
- [72] Y. Wang e W. Wang. Adjacent vertex distinguishing total colorings of outerplanar graphs. *J. Comb. Optim.*, 19:123–133, 2010.
- [73] M. Yannakakis. Complexity the minimum fill-in is NP-Complete. *SIAM J. Algebra. Discr. Methods*, 2(1):77–79, 1981.
- [74] Z. Zhang, X. Chen, J. Li, B. Yao, X. Lu e J. Wang. On adjacent-vertex-distinguishing total coloring of graphs. *Science in China Series A: Mathematics*, 48:289–299, 2005.
- [75] Z. Zhang, L. Liu e J. Wang. Adjacent strong edge coloring of graphs. *Appl. Math. Lett.*, 15:623–626, 2002.

Índice Remissivo

- aranha, 15–19
 - gorda, 15
 - magra, 15
 - partição canônica, 16, 18
- árvore, 5
- brittle, 6, 18, 19
- classe hereditária, 6
- co-árvore, 14, 52
 - binária, 53
- cografo, 14
- coloração
 - número cromático, 70
 - número cromático total, 70
 - número cromático total forte, 70, 71
 - total, 69
 - total distinguidora de vértices adjacentes, 69
 - total forte, 70
- conjunto homogêneo, 7
- decomposição modular, 9, 16, 18
 - árvore, 9, 14, 18, 52, 54
 - nó degenerado, 10, 11
 - nó paralelo, 10
 - nó primo, 10
 - nó serial, 10
 - nó vizinhança, 10, 14, 16, 18–20
 - quase binária, 55
- empacotamento do tipo (n_2, \dots, n_r) , 48
- grafo
 - bipartido, 6
 - bipartido completo, 6
 - cordal, 6
 - dualmente cordal, 6
 - fator, 8, 16, 18, 19
 - fortemente cordal, 6
 - fracamente cordal, 6
 - indiferença, 71
 - join, 5
 - perfeitamente ordenável, 6, 15, 18
 - perfeito, 6, 15
 - primo, 7
 - pseudo-split, 19
 - quociente, 8–10, 16, 18–20
 - split, 6, 18, 19, 29, 47
 - trivial, 4
 - união, 5
- módulo, 7
 - adjacente, 7
 - forte, 8, 9
 - forte maximal próprio, 9
 - não-adjacente, 7
 - sobrepostos, 7
 - trivial, 7
- ordem indiferença, 71
- P_4 -arrumada, 20
- P_4 -carregada, 18
- P_4 -carregada estendida, 20
- P_4 -esparsa, 15, 18

- P₄-esparsa estendida, 20
- P₄-leve, 17–19
- parceiro, 12
- partição de congruência, 8, 9
- preenchimento mínimo, 6
- Problema de Empacotamento de Subgrafos em Grafos, 46
- Problema do Empacotamento de Cliques, 46
- Problema do Emparelhamento Máximo, 47
- Problema do Emparelhamento Multidimensional, 47
- Problema dos Separadores Minimais, 21
- pseudo-aranha, 17–19
 - partição canônica, 17
- pullback, 72
- separador, 21
 - ab*-separador, 21
 - ab*-separador minimal, 21
 - minimal, 21, 25, 27