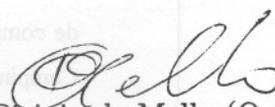


Decomposição Modular de Grafos Não Orientados

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

Campinas, 9 de março de 2007.

Este exemplar corresponde à redação final da Tese/Dissertação devidamente corrigida e defendida por: <u>Vagner Pedrotti</u>
e aprovada pela Banca Examinadora. Campinas, <u>11</u> de <u>Outubro</u> de <u>07</u>
<u>Redo de Azevedo</u> COORDENADOR DE PÓS-GRADUAÇÃO CPG-IC


Célia Picinin de Mello (Orientadora)

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

UNIDADE BC
Nº CHAMADA: _____
T/UNICAMP 7343d
V. _____ EX. _____
TOMBO BCCL 74890
PROC 16.145-07
C _____ D X
PREÇO 11,00
DATA 31/10/07
BIB-ID 415011

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Bibliotecária: Miriam Cristina Alves – CRB8a / 5094

Pedrotti, Vagner

P343d Decomposição modular de grafos não-orientados / Vagner Pedrotti
-- Campinas, [S.P. :s.n.], 2007.

Orientadora: Célia Picinin de Mello.

Dissertação (mestrado) - Universidade Estadual de Campinas,
Instituto de Computação.

1. Teoria dos grafos. 2. Representações dos grafos. 3. Algoritmos
de computador. I. Mello, Célia Picinin de. II. Universidade Estadual de
Campinas. Instituto de Computação. III. Título.

Título em inglês: Modular decomposition of undirected graphs

Palavras-chave em inglês (Keywords): 1. Graph theory. 2. Representation of graphs. 3.
Computer algorithms

Área de concentração: Teoria da computação

Titulação: Mestre em Ciência da Computação

Banca examinadora: Profa. Dra. Célia Picinin de Mello (IC-Unicamp)
Profa. Dra. Sulamita Klein (COPPE-UFRJ)
Prof. Dr. Orlando Lee (IC-Unicamp)
Profa. Dra. Christiane Neme Campos (IC-Unicamp)

Data da defesa: 09/03/2007

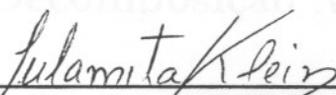
Programa de Pós-Graduação: Mestrado em Ciência da Computação

191254001

Instituto de Computação

TERMO DE APROVAÇÃO

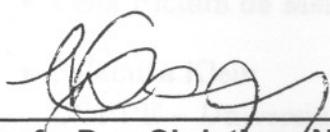
Tese defendida e aprovada em 09 de março de 2007, pela Banca examinadora composta pelos Professores Doutores:



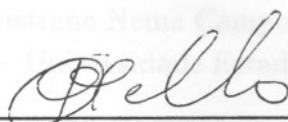
Profa. Dra. Sulamita Klein
UFRJ.



Prof. Dr. Orlando Lee
IC – UNICAMP.



Profa. Dra. Christiane Neme Campos
IC – UNICAMP.



Profa. Dra. Célia Pinicin de Mello
IC – UNICAMP.

Decomposição Modular de Grafos Não Orientados

Vagner Pedrotti¹

7 de fevereiro de 2007

Banca Examinadora:

- Célia Picinin de Mello (Orientadora)
- Sulamita Klein
COPPE – Universidade Federal do Rio de Janeiro
- Orlando Lee
IC – Universidade Estadual de Campinas
- Christiane Neme Campos
IC – Universidade Estadual de Campinas

¹Suporte financeiro de: Bolsa do CNPq (processo 131813/2005-6) 2005–2007

Resumo

Um módulo de um grafo é um subconjunto de seus vértices que não é diferenciado, em relação à adjacência, pelos demais vértices do mesmo grafo. Dado um módulo M de um grafo G , se todo módulo de G que intercepta M está contido nele ou o contém, M é denominado módulo forte. A decomposição modular de um grafo G é única e pode ser representada por uma árvore de decomposição definida pela relação de inclusão de todos os módulos fortes do grafo.

Esta dissertação apresenta um estudo sobre o tema da decomposição modular de grafos não-orientados, descrevendo seus principais conceitos, propriedades e algumas das técnicas e algoritmos mais simples e eficientes para a construção da árvore de decomposição modular de um grafo.

Adicionalmente, são descritas classes de grafos nas quais a decomposição modular atende a determinadas propriedades, o que possibilita o uso da mesma em diversas aplicações, destacando-se a solução eficiente de problemas de otimização através de técnicas de divisão e conquista.

Abstract

A module of a graph is a non-distinguishable subset of nodes, regarding the nodes adjacency. Let M denote any module of a graph G . If every module of G which overlaps M either contains M or is included in it, M is called a strong module. Modular decomposition of G is unique and may be represented by a decomposition tree defined by the inclusion relationship among all strong modules of G .

This dissertation presents a study on modular decomposition of undirected graphs, explaining its main principles and properties, among with a few simple and efficient techniques and algorithms used to produce the modular decomposition tree of a graph.

Furthermore, it describes a few graph classes with special properties, which allows the use of modular decomposition in several applications, specially to efficiently solve some optimization problems through conquer and divide techniques.

Agradecimentos

Gostaria de deixar meus sinceros agradecimentos à minha orientadora Célia Picinin de Mello por todo auxílio, direcionamento e apoio durante o desenvolvimento deste trabalho e pelas aulas ministradas; ao CNPq, pelo suporte financeiro através da bolsa de mestrado durante todo o período de estudos; e aos membros da banca, pelas considerações relevantes levantadas e várias sugestões que contribuíram positivamente para este trabalho.

Gratidão especial reservo à toda minha família, sempre presente e afetuosa, incluindo minha mãe, pelos incentivos e compreensão, e meu pai e minha avó, que embora não estejam mais presencialmente conosco, tiveram um importante papel na minha infância, que se reflete na minha formação como pessoa e cidadão.

Claro que também é impossível deixar de mencionar todo companherismo que encontrei em meus amigos, muitos garimpados no próprio instituto e vivendo em situação muito semelhante à minha. Dentre os quais, gostaria de fazer menção especial a Edna Ayako Hoshino, Marcelo Castilho Couto e Wellington da Silva Mariusso, sempre presentes e prestativos, me auxiliaram em diversos momentos importantes.

Muitas outras pessoas tiveram participação nesta conquista e em várias decisões da minha vida que permitiram que eu chegasse a ela. Embora seja impossível nomeá-los, não serão esquecidos.

Sumário

Resumo	v
Abstract	vi
Agradecimentos	vii
1 Introdução	1
2 Módulos	3
2.1 Propriedades	3
2.2 Quocientes e fatores	7
2.3 Representações	9
2.3.1 Decomposição modular	10
2.3.2 Conjuntos parcialmente ordenados modulares	14
3 Algoritmos para Decomposição Modular	16
3.1 Partição de congruência que isola um vértice	16
3.2 Recursão sobre adjacentes e não adjacentes	20
3.3 Permutações fatorizadas	26
3.3.1 Obter uma permutação fatorizada de um grafo	27
3.3.2 Obter a decomposição modular da permutação fatorizada	30
4 Classes com poucos P_4's	37
4.1 Algumas definições	37
4.2 Cografos	40
4.3 P_4 -redutível	43
4.3.1 P_4 -redutível estendida	46
4.4 P_4 -esparso	46
4.4.1 Aranhas	47
4.4.2 Caracterizações	48

4.4.3	Decomposição e aplicações	49
4.4.4	P_4 -esparso estendida	51
4.5	P_4 -extensível	51
4.6	P_4 -leve	54
4.7	P_4 -carregada	55
4.7.1	P_4 -carregada estendida	56
4.8	P_4 -arrumada	57
4.9	Conclusões	58
5	A classe P_4-arrumada	59
5.1	Propriedades	59
5.1.1	Caracterizações	63
5.2	Algoritmo de reconhecimento	66
5.2.1	Caracterização das classes reconhecidas	66
5.2.2	Algoritmo e análise	67
5.3	Aplicações	68
5.3.1	Número cromático e Clique máxima	68
5.3.2	Partição mínima em cliques e Conjunto independente máximo	70
6	Conclusão	71
A	Definições	72
A.1	Conjuntos	72
A.2	Grafos	72
A.2.1	Alguns grafos comuns	74
A.2.2	Operações e relações	74
A.2.3	Alguns parâmetros	74
A.2.4	Classes com características de interesse	75
A.3	Ordens	76
	Bibliografia	80

Lista de Tabelas

3.1	Construção da palavra P para o grafo da Figura 3.7.	31
4.1	Definição da operação $\textcircled{4}$	53
5.1	Valores de $\text{marca}(H)$ permitidos para cada $\text{tipo}(H)$ em algumas classes de grafos.	67

Lista de Figuras

2.1	Exemplos de módulos.	4
2.2	Exemplos de adjacência e sobreposição de módulos.	5
2.3	Exemplo de módulos fortes e não fortes num grafo.	7
2.4	Um grafo e duas de suas partições de congruência.	8
2.5	Exemplo de uma árvore de decomposição modular.	10
2.6	A árvore de decomposição modular do grafo da Figura 2.5a, com os nós internos rotulados.	13
2.7	A construção de um CPO modular de um grafo.	14
3.1	Um grafo G com os módulos de $\mathcal{M}(G, c)$ e seus módulos fortes maximais próprios destacados.	17
3.2	Um exemplo de decomposição modular segundo a técnica de Ehrenfeucht e outros.	19
3.3	Uma iteração do Algoritmo 3 sobre o grafo G da Figura 3.1.	21
3.4	Exemplo de construção da decomposição de $G/\mathcal{M}(G, c)$ pelo Algoritmo 5.	23
3.5	Exemplo de aplicação das regras usadas para obter a permutação fatorizada.	28
3.6	Exemplo de execução do algoritmo para criar uma permutação fatorizada do grafo da Figura 3.1.	30
3.7	Exemplo para técnica de decomposição por permutações fatorizadas.	31
3.8	Construção da árvore de fraturas.	32
3.9	Remoção de nós da árvore de fraturas que não são módulos.	35
4.1	Diagrama de Hasse das classes, adaptado de [GRT97].	38
4.2	Grafos de cinco vértices com um P_4 induzido e um vértice adicional.	39
4.3	Um cografo e suas decomposições.	41
4.4	Uma representação de uma multi-árvore cujo grafo de comparabilidade associado correspondente ao cografo da Figura 4.3a.	42
4.5	Dois subgrafos proibidos para a classe P_4 -reduzível (H e \overline{H}) e o touro.	44
4.6	Um grafo P_4 -reduzível e suas decomposições.	45
4.7	Exemplos de aranhas gorda e magra.	47

4.8	Decomposição de duas aranhas pela operação ③	50
4.9	A ps-árvore e a decomposição modular do grafo da Figura 4.8b.	50
4.10	Exemplos de extensões próprias.	52
4.11	Exemplo de px-árvore.	54
4.12	Exemplo de grafo da classe P_4 -leve e sua decomposição modular.	55
4.13	Exemplo de grafo da classe P_4 -carregada.	56
4.14	Exemplo de grafo da classe P_4 -arrumada.	58
5.1	Subgrafos nos quais o P_5 induzido por $\{a, b, c, d, e\}$ não seria um módulo. . .	61

Capítulo 1

Introdução

Nesta dissertação serão estudados os módulos de um grafo e sua decomposição modular. Para tal, faz-se uso de conceitos matemáticos e de teoria dos grafos que, em sua maioria, estão reunidos no Apêndice A. Definições ausentes deste apêndice são dadas no próprio texto ou são terminologias consagradas na área (veja [Gol80]).

Os módulos de um grafo e, principalmente, a decomposição modular de um grafo são assuntos extensivamente estudados na literatura mundial e aplicados na solução de variados problemas. A definição de módulo é atribuída a Gallai [Gal67], embora este conceito tenha sido redescoberto várias vezes, recebendo outros nomes, como `closed set`, `autonomous set`, `partitive set`, `clump` e `stable set`. O primeiro algoritmo que computa a decomposição modular é de James, Staton e Cowan, de 1972.

Com o contínuo interesse e pesquisa, mais propriedades da decomposição modular foram descobertas e surgiram vários outros algoritmos para computá-la, que melhoraram sucessivamente a complexidade do primeiro, de $O(n^4)$ até a linearidade, finalmente atingida em 1994 por McConnell e Spinrad, para o caso de grafos não orientados. Ao contrário de encerrar a pesquisa, esse trabalho se tornou um marco na pesquisa sobre a decomposição modular, pois além de aumentar a importância desta decomposição, direcionou a pesquisa na busca por algoritmos eficientes e mais simples de ser implementados e compreendidos.

Como fruto desta pesquisa, novos algoritmos de decomposição foram apresentados, alguns dos quais com complexidade pouco superior a linearidade, outros que apresentam bons resultados em arquiteturas paralelas e, ainda, algoritmos que decompõem estruturas mais abrangentes, como grafos orientados, 2-estruturas e hipergrafos.

A decomposição modular de um grafo tem o papel de guia para utilização de técnicas de divisão e conquista na solução de vários problemas em grafos. Suas aplicações vão desde os exemplos clássicos, que incluem a orientação transitiva de grafos de comparabilidade e o reconhecimento de grafos de permutação, até seu uso em algumas classes de grafos. Nestas,

a decomposição apresenta características que permitem seu uso na solução de problemas básicos e sabidamente difíceis para o caso geral, tais como coloração de vértices e clique máxima.

A primeira classe apresentada na qual a decomposição tem vastos usos foi a classe dos cografos, em 1971, por Lerchs. Posteriormente, observou-se que a decomposição tem aplicações em várias outras classes, em especial aquelas estudadas por Jamison e Olariu, como as classes P_4 -reduzível e P_4 -esparso, entre outras. A princípio, tanto para os cografos como para as demais classes definidas, a correlação com a decomposição modular não estava estabelecida e, a cada classe, correspondia uma decomposição particular usada em algumas aplicações. Ficou a cargo de Giakoumakis, Roussel e Thuillier a extensão de algumas de tais classes conhecidas, a determinação do seu relacionamento com a decomposição modular e a aplicação da mesma para a solução de problemas clássicos de otimização.

Esta dissertação foi motivada pela gama de aplicações da decomposição modular e pela ausência de um trabalho de investigação que abranja desde as propriedades da decomposição até suas aplicações.

O Capítulo 2 trata de módulos, suas propriedades fundamentais e representações, incluindo a decomposição modular. Muitas das afirmações são provadas de forma a deixar o leitor com toda informação pertinente. Em seguida, o Capítulo 3 detalha a estratégia de Ehrenfeucht, Gabow, McConnell e Sullivan para decomposição modular e dois algoritmos recentes de complexidade linear que a computam. Um de Dahlhaus, Gustedt e McConnell, facilmente paralelizável, que faz uso da técnica de Ehrenfeucht, e um com contribuições de Habib, Capelle, de Montgolfier e Paul, que se destaca por sua simplicidade de implementação. O Capítulo 4 expõe algumas classes nas quais a aplicação da decomposição modular permite algoritmos eficientes para problemas clássicos de otimização. Entre tais classes, destaca-se a classe P_4 -arrumada, para a qual suas propriedades e aplicações são detalhados no Capítulo 5.

O objetivo desta dissertação é estudar propriedades dos módulos e da decomposição modular e suas aplicações, limitada a grafos não orientados, provendo uma fonte de informação concisa e vasta sobre esta técnica.

Capítulo 2

Módulos

Este capítulo fornece as bases do assunto abordado, definindo os módulos de um grafo e agrupando suas principais propriedades, operações e representações.

O detalhamento das propriedades abrange desde as mais simples àquelas mais elaboradas. Descreve-se uma das operações mais comuns – o uso de módulos para particionar os vértices de um grafo, ferramenta básica para a representação dos módulos pela árvore de decomposição modular. É apresentada, também, uma representação dos módulos por meio de conjuntos parcialmente ordenados (CPOs, veja definição em A.3).

2.1 Propriedades

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 . Isto é, em um módulo, seus vértices são tratados da mesma forma pelos demais vértices do grafo e não o contrário, no qual os vértices do módulo tratariam da mesma os demais vértices. 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 $\{v : v \in V(G)\}$. 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)

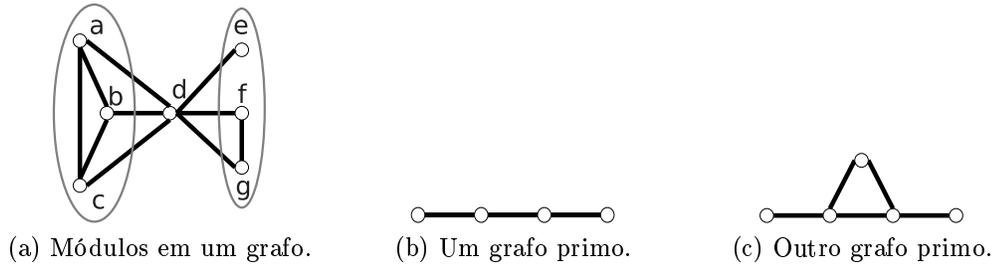


Figura 2.1: Exemplos de módulos.

da Figura 2.1. O leitor pode observar que grafos primos não são raros, dado que os grafos C_n para $n \geq 5$ e P_n para $n \geq 4$ são exemplares.

A propriedade abaixo garante que um grafo e seu complemento têm exatamente os mesmos módulos.

Proposição 2.1. *Se M é um módulo de um grafo G , então M é módulo de \overline{G} .*

Demonstração. Sejam G um grafo e M um módulo de G . Se M é trivial, então é módulo de \overline{G} . Considere M não trivial e suponha que não seja um módulo de \overline{G} . Logo, existe um vértice $v \in V(\overline{G}) \setminus M$ e dois vértices distintos a e b contidos em M tais que $\{a, v\}$ é uma aresta de \overline{G} e $\{b, v\}$ não é aresta de \overline{G} . Isto implica que $\{a, v\} \notin E(G)$ e $\{b, v\} \in E(G)$ e, portanto, M não seria módulo de G . \square

Como conseqüência, o complemento de um grafo primo também é primo. Dois módulos de um grafo podem um conter o outro, se interceptar, ou serem disjuntos. Em cada caso, apresentam propriedades de interesse.

Lema 2.2 ([Möh85]). *Se X é um módulo de um grafo G e $Y \subseteq X$, então Y é um módulo de $G[X]$ se e somente se Y é módulo de G .*

Demonstração. Sejam G um grafo e X e Y dois de seus módulos tais que $Y \subseteq X$. Se Y não fosse módulo de $G[X]$, então existiriam os vértices a e b em Y e v em $X \setminus Y$, de forma que $\{v, a\}$ seria aresta de G e $\{v, b\}$, não. Assim, v testemunharia que Y não seria módulo de G . Portanto, se Y é módulo de G , também é de $G[X]$.

Agora, suponha que Y é módulo de $G[X]$ e não é módulo de G , então, existiriam a e b em Y e v em $V(G) \setminus Y$ de forma que $\{a, v\}$ é aresta de G e $\{b, v\}$ não. Se $v \in X$, Y não seria módulo de $G[X]$ e se $v \in V(G) \setminus X$, X não seria módulo de G . Conclui-se que, neste caso, Y é módulo de G . \square

Lema 2.3 ([Möh85]). *Se dois módulos X e Y de G são tais que $X \cap Y = \emptyset$, então ou as arestas $\{\{x, y\} : x \in X, y \in Y\}$ pertencem a G ou G não contém nenhuma de tais arestas.*

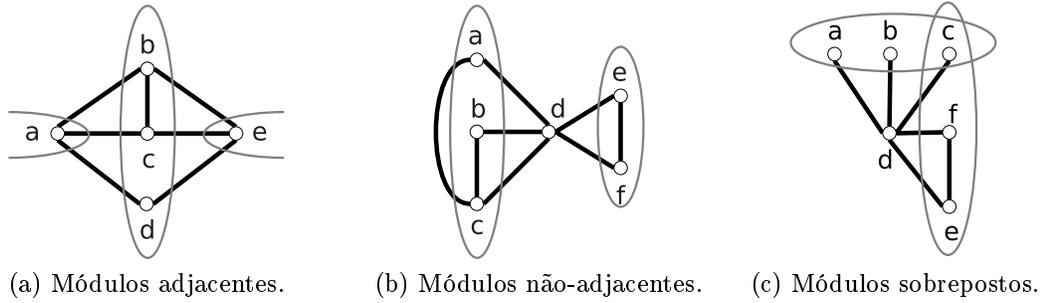


Figura 2.2: Exemplos de adjacência e sobreposição de módulos.

Demonstração. Sejam G um grafo e X, Y dois módulos de G com $X \cap Y = \emptyset$. Se ambos os módulos possuírem um vértice cada, a propriedade se verifica trivialmente. Suponha, então, que existam dois vértices a e b de X e dois vértices c e d de Y tais que $\{a, c\}$ é aresta de G e $\{b, d\}$ não é aresta de G . Se $a = b$, Y não seria módulo de G e se $c = d$, X não o seria. Portanto, $a \neq b$ e $c \neq d$. Como X é módulo, tem-se que $\{b, c\} \in E(G)$ e $\{a, d\} \notin E(G)$, contradizendo o fato de Y ser um módulo. Portanto, ou existem todas as arestas entre X e Y ou não existe nenhuma delas. \square

Dados dois módulos X e Y de um grafo G tal que $X \cap Y = \emptyset$, se as arestas $\{\{x, y\} : x \in X, y \in Y\}$ pertencem a G , então eles são chamados *adjacentes* e, caso contrário, os módulos são chamados *não-adjacentes*. Além disso, se dois módulos X e Y de G são tais que $X \setminus Y, Y \setminus X$ e $X \cap Y$ são todos não vazios, diz-se que os módulos se *sobrepoem*. Note que se X e Y são dois módulos que se sobrepoem, então $|X| \geq 2$ e $|Y| \geq 2$.

Para exemplos de módulos adjacentes, não-adjacentes e sobrepostos, veja a Figura 2.2. Note ainda que é possível encontrar outros módulos nestas condições para os grafos dados nesta figura, como os módulos $\{b, c\}$ e $\{a, e\}$, adjacentes em (a), os módulos $\{a\}$ e $\{e, f\}$, não-adjacentes em (b) e os módulos $\{a, b\}$ e $\{b, c\}$, sobrepostos em (c).

Lema 2.4 ([Möh85]). *Se X e Y são módulos de um grafo G que se sobrepoem, então $X \setminus Y, Y \setminus X, X \cap Y, X \cup Y$ e $X \Delta Y$ são módulos de G .*

Demonstração. Sejam G um grafo e X, Y dois módulos de G que se sobrepoem. Prova-se por contradição que cada um dos conjuntos abaixo é um módulo de G . Para tal, sejam a e b dois vértices distintos pertencentes ao conjunto considerado e seja v um vértice de G que testemunha que o dado conjunto não é um módulo, isto é, v não pertence ao conjunto e $\{v, a\}$ é aresta de G enquanto $\{v, b\}$ não é aresta de G .

- $X \cap Y$: Se $v \in V(G) \setminus X$, então v testemunha que X não é módulo de G , pois $\{a, b\} \subseteq X$. Por simetria, se $v \in V(G) \setminus Y$, Y não seria módulo de G .

- $X \setminus Y$: Se $v \notin X$, então X não seria módulo de G e, portanto, $v \in X \cap Y$. Seja $c \in Y \setminus X$. Como Y é módulo de G , $\{c, a\}$ é aresta de G . Por sua vez, como X é módulo de G , $\{c, b\}$ é aresta de G . Finalmente, sendo Y módulo, $\{b, v\}$ é aresta de G , contradição.
- $Y \setminus X$: Simétrico ao anterior.
- $X \cup Y$: Se a e b pertencerem ambos a X (ou Y), então este não seria um módulo. Portanto, sem perda de generalidade, considere $a \in X \setminus Y$ e $b \in Y \setminus X$. Seja $c \in X \cap Y$ (existe, pois X e Y se sobrepõem). Então $\{v, c\}$ deve ser uma aresta de G , caso contrário, X não seria módulo. No entanto, c , b e v atestam que Y não é módulo de G .
- $X \Delta Y$: Se a e b pertencerem ambos a $X \setminus Y$ (ou $Y \setminus X$), então X (ou Y) não seria um módulo. Escolhe-se um $c \in X \cap Y$ e aplica-se argumento semelhante ao do caso anterior. \square

Ainda na Figura 2.2c, pode-se observar dois módulos que sobrepõem-se e verificar a validade das propriedades anteriores.

As duas propriedades abaixo são de fácil verificação. A primeira atesta que ser módulo é propriedade hereditária nos subgrafos induzidos que o contém, e a segunda estabelece as condições para o caso inverso.

Lema 2.5. *Se M é um módulo de um grafo G , então M é módulo de todo subgrafo induzido, $G[X]$ tal que $M \subseteq X \subseteq V(G)$.*

Para verificar se um módulo de um subgrafo induzido é módulo do grafo original, basta procurar por vizinhos fracos. Dado um grafo e um subconjunto X de seus vértices, um vértice é *vizinho fraco* de X se não pertence a X e é vizinho de algum vértice de X , mas não de todos. Note que este conceito já foi usado anteriormente, por exemplo na prova do Lema 2.4.

Lema 2.6. *Seja G um grafo e $X \subseteq V(G)$. Se M é módulo de $G[X]$, então M é módulo de G se, e somente se, M não possui vizinhos fracos em $V(G) \setminus X$.*

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 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 2.3 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. A seguir, estendem-se algumas propriedades de módulos para módulos fortes.

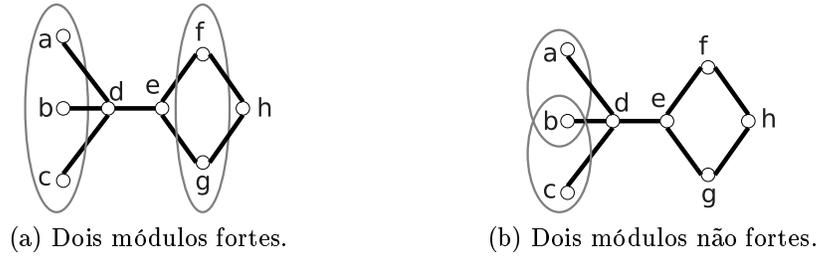


Figura 2.3: Exemplo de módulos fortes e não fortes num grafo.

Proposição 2.7. *Se M é um módulo forte de um grafo G , então é módulo forte de \overline{G} .*

Demonstração. Suponha que M seja módulo forte de um grafo G e não seja do grafo \overline{G} . Portanto, neste grafo, existe um módulo Y que se sobrepõe a X . Pela Proposição 2.1, Y é módulo forte de G e, como se sobrepõe a X , este não seria módulo forte de G . \square

Lema 2.8 ([Möh85]). *Se M é um módulo de um grafo G e $N \subset M$, então N é um módulo forte de G se, e somente se, N é módulo forte de $G[M]$.*

Demonstração. Seja $X \subset M$, um módulo forte de $G[M]$. Pelo Lema 2.2, X é módulo de G . Suponha que X não seja um módulo forte de G . Então existe um módulo Y em G que se sobrepõe a X . Note que Y não está contido em M pois X é módulo forte de $G[M]$. Todavia, pelo Lema 2.4, $M \setminus Y$ e $M \cap Y$ são módulos de G , propriamente contidos em M , e pelo Lema 2.2, são módulos de $G[M]$. Se $M \setminus Y \not\subseteq X$, $M \setminus Y$ sobrepõe-se a X , caso contrário, $M \cap Y$ sobrepõe-se a X . Em ambos os casos, contradiz-se que X seja módulo forte de $G[M]$.

Seja X um módulo forte de G propriamente contido em outro módulo, M , de G . Pelo Lema 2.2, X é módulo de $G[M]$. Caso X não seja módulo forte de $G[M]$, então existe um módulo Y , de $G[M]$, que se sobrepõe a X . Pelo Lema 2.2, Y é módulo de G e, portanto, X não é módulo forte de G , contradição. \square

2.2 Quocientes e fatores

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 todos as partes de \mathcal{P} são módulos dois a dois disjuntos, pelo Lema 2.3, 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 abaixo:

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 $V(G/\mathcal{P}) = \{M_1, M_2, \dots, M_k\}$

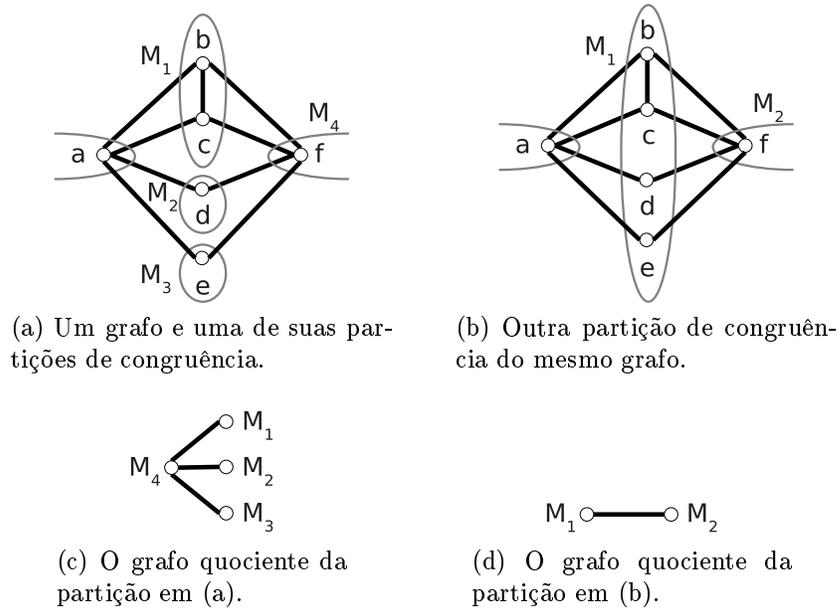


Figura 2.4: Um grafo e duas de suas partições de congruência.

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. Na Figura 2.4 estão representadas duas partições de congruência do mesmo grafo, com seus respectivos grafos quociente.

Denominam-se *fatores*, os subgrafos induzidos por cada membro de \mathcal{P} em G . A seguir, mostra-se que os módulos de G que não se sobrepõem a nenhuma parte de \mathcal{P} estão relacionados aos módulos do quociente e dos fatores de \mathcal{P} .

Lema 2.9 ([Möh85]). *Sejam G um grafo e \mathcal{P} uma partição de congruência de G . Então, $\mathcal{M} \subseteq \mathcal{P}$ é um módulo de G/\mathcal{P} se, e somente se, $\bigcup_{X \in \mathcal{M}} X$ é módulo de G .*

Demonstração. Sejam G um grafo, \mathcal{P} uma partição de congruência de G e $\mathcal{M} \subseteq \mathcal{P}$ um módulo de G/\mathcal{P} . Se $U = \bigcup_{X \in \mathcal{M}} X$ não for módulo de G , então existem a e b em U e $v \notin U$ tais que $\{a, v\}$ é aresta de G e $\{b, v\}$ não é aresta de G . Se a e b pertencerem a algum $Y \in \mathcal{M}$, então Y não seria módulo e \mathcal{P} não seria partição de congruência. Portanto $a \in A$, $A \in \mathcal{M}$ e $b \in B$, $B \in \mathcal{M}$ e $v \in V$, $V \in \mathcal{P} \setminus \mathcal{M}$ com $A \neq B$. Sendo A , B e V disjuntos, pelo Lema 2.3, em G/\mathcal{P} , $\{A, V\}$ é aresta e $\{B, V\}$ não é aresta, de forma que \mathcal{M} não é módulo.

Seja $U = \bigcup_{X \in \mathcal{M}} X$ um módulo de G e suponha que \mathcal{M} não seja um módulo de G/\mathcal{P} . Então, existem A e B em \mathcal{M} e V em $\mathcal{P} \setminus \mathcal{M}$, tais que, em G/\mathcal{P} , $\{A, V\}$ é aresta e $\{B, V\}$ não é aresta. Pela definição de grafo quociente e pelo Lema 2.3, existem $a \in A$, $b \in B$ e

$v \in V$ tais que, em G , $\{a, v\}$ é aresta e $\{b, v\}$ não é aresta. Como V é disjunto de U , este não é módulo de G . \square

Agora, é possível mostrar uma propriedade importante das partições de congruência formadas por módulos fortes de G :

Corolário 2.10. *Se G é um grafo e \mathcal{P} é uma partição de congruência de G formada por módulos fortes, então todos os módulos de G são dados pelos módulos de G/\mathcal{P} e pelos módulos dos fatores de G em relação a \mathcal{P} .*

Demonstração. Como as partes de \mathcal{P} são módulos fortes, cada módulo de G ou está contido em uma parte de \mathcal{P} ou é formado pela união de partes de \mathcal{P} . Então, pelos lemas 2.2 e 2.9, um módulo de G ou é módulo de algum fator de \mathcal{P} ou é dado por um módulo de G/\mathcal{P} . \square

O Corolário 2.10 torna interessante o estudo das partições de congruência cujas partes sejam módulos fortes do grafo, das quais pode-se observar dois exemplos na Figura 2.4, itens (a) e (b). Abaixo, mostra-se que as informações sobre os módulos fortes são preservadas pelas partições de congruência.

Lema 2.11. *Dado um grafo G e uma partição de congruência \mathcal{P} de G , então $M \notin \mathcal{P}$ é um módulo forte de G se, e somente se, ou M é módulo forte de $G[K]$ para um $K \in \mathcal{P}$ tal que $M \subset K$, ou existe $\mathcal{U} \subseteq \mathcal{P}$ tal que $M = \bigcup_{X \in \mathcal{U}} X$ e \mathcal{U} é módulo forte de G/\mathcal{P} .*

Demonstração. Quando M está contido em uma parte K de \mathcal{P} o Lema 2.8 é suficiente. Então, seja M um módulo forte de G , união dos elementos de um subconjunto \mathcal{U} das partes de \mathcal{P} . Pelo Lema 2.9, \mathcal{U} é um módulo forte de G/\mathcal{P} .

Seja \mathcal{U} módulo forte de G/\mathcal{P} e suponha que exista um módulo Y de G que se sobreponha a M . Considere um vértice $a \in M \setminus Y$ e o conjunto $\mathcal{Y} = \{Q : Q \in \mathcal{P}, Q \cap Y \neq \emptyset\} \setminus \{Z\}$, no qual $a \in Z$ e $Z \in \mathcal{P}$. Pelo Lema 2.4, $\bigcup_{X \in \mathcal{Y}} X$ é módulo de G e pelo Lema 2.9, \mathcal{Y} é módulo de G/\mathcal{P} , mas ele se sobrepõe a \mathcal{U} . \square

2.3 Representações

O número de módulos de um grafo pode ser exponencial no seu tamanho, por exemplo, qualquer subconjunto não vazio dos vértices de um K_n ou S_n é um módulo do grafo – e existem $2^n - 1$. Assim sendo, torna-se interessante encontrar um meio de representá-los que ocupe espaço polinomial e permita a solução eficiente de problemas como determinar se um dado conjunto de vértices é um módulo, contar os módulos do grafo ou mesmo listá-los em tempo compatível com sua quantidade.

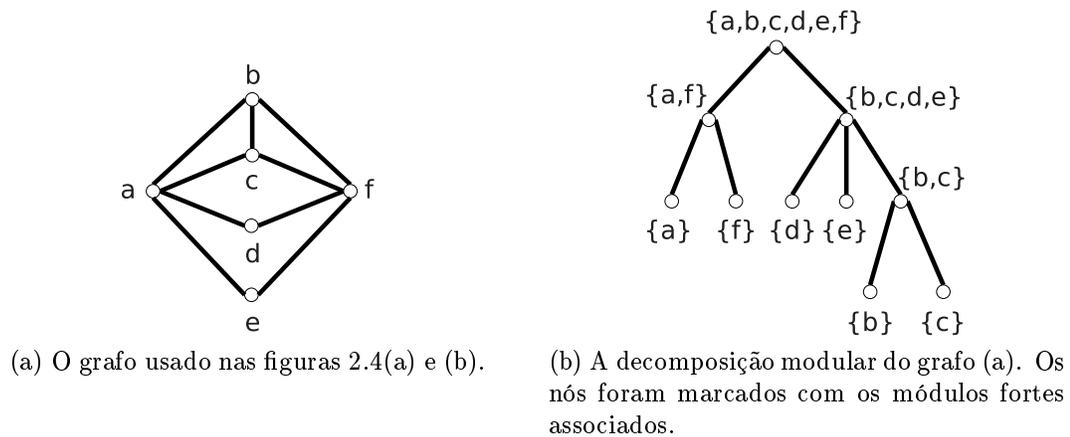


Figura 2.5: Exemplo de uma árvore de decomposição modular.

2.3.1 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 $V(G)$. 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 2.4 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* e está exemplificada na Figura 2.5.

Será visto que, usando tais partições de congruência para a decomposição, os grafos quocientes associados a cada nó da árvore de decomposição são restritos de forma que é possível identificar seus módulos de maneira trivial. Além disso, tais partições possuem outras propriedades interessantes, que são descritas a seguir.

Proposição 2.12. *A partição dos vértices de um grafo em módulos fortes maximais*

próprios é única.

Demonstração. Suponha que existam duas partições, \mathcal{P}_1 e \mathcal{P}_2 , distintas, dos vértices de um grafo G em módulos fortes maximais próprios. Então, existe $X \in \mathcal{P}_1$ tal que $X \notin \mathcal{P}_2$. Além disso, X não se sobrepõe a nenhum módulo pertencente a \mathcal{P}_2 , pois estes são fortes. Portanto, ou X contém uma parte Y de \mathcal{P}_2 , ou está contido em uma parte Y de \mathcal{P}_2 , ou $X \cap Y = \emptyset$ para todo $Y \in \mathcal{P}_2$. Então, X ou Y não são maximais ou \mathcal{P}_2 não é uma partição dos vértices de G . \square

Como consequência imediata da Proposição 2.12:

Corolário 2.13. *Dado um grafo G , a sua árvore de decomposição modular é única.*

Lema 2.14. *Os únicos grafos que não possuem módulos fortes não triviais são os grafos primos, os grafos completos e os grafos sem arestas.*

Demonstração. Será provado por indução no número de vértices que, se G é um grafo, não é primo, não é completo e possui alguma aresta, então ele contém um módulo forte não trivial. Sejam G um grafo que satisfaz as hipóteses e X um módulo não trivial de tamanho mínimo de G . Note que tal módulo existe, pois o grafo não é primo. Deste módulo, obtém-se um módulo forte não trivial para o grafo.

- Se $|X| \geq 3$, então X é um módulo forte não trivial, pois, se algum módulo Y de G sobrepusesse X , pelo Lema 2.4, $X \cap Y$ e $X \setminus Y$ seriam módulos, um dos quais conteria ao menos dois vértices e X não seria mínimo.
- Se $|X| = 2$, considere o grafo G' obtido de G pela contração de $X = \{a, b\}$ a um novo vértice x . Note que $\{X\} \cup (V(G) \setminus X)$ é uma partição de congruência de G e G' é isomorfo ao grafo quociente desta partição.
 - Se G' não possuir arestas, então $\{a, b\}$ é uma aresta de G (caso contrário G não possuiria arestas) e é um módulo forte não trivial de G . Qualquer módulo que se sobrepusesse a $\{a, b\}$, conteria a e não conteria b , sem perda de generalidade. Mas isso não acontece, pois b só é vizinho de a .
 - Se G' for completo, pelo caso anterior X é um módulo forte não trivial de \overline{G} e, pela Proposição 2.7, de G .
 - Se G' for primo, então X é um módulo forte não trivial de G . Caso contrário, haveria um módulo, Y , de G que se sobrepõe a X e, pelo Lema 2.4, $X \cup Y$ e $Y \setminus X$ são módulos de G . Logo, se $|X \cup Y| < |V(G)|$ ou $|Y \setminus X| \geq 2$, pelo Lema 2.9, G' não seria primo. Caso contrário, $|V(G)| = 3$ e atendem-se as condições de algum dos casos anteriores.

- Nos demais casos, por hipótese de indução, seja X' um módulo forte não trivial de G' . Pelo Lema 2.11, ou X' é módulo forte não trivial de G , se $x \notin X'$, ou $X \cup (X' \setminus \{x\})$ o é. \square

Corolário 2.15. *Dado um grafo G e sua única partição de congruência \mathcal{P} cujas partes são módulos fortes maximais próprios, então G/\mathcal{P} é primo, completo ou não contém arestas.*

Demonstração. O grafo quociente G/\mathcal{P} não contém módulos fortes não triviais, senão pelo Lema 2.11, os membros de \mathcal{P} não seriam maximais. Então, pelo Lema 2.14, G/\mathcal{P} é primo, completo ou não contém arestas. \square

Proposição 2.16. *Os nós da árvore de decomposição modular de um grafo correspondem exatamente a todos os seus módulos fortes.*

Demonstração. Por indução no número de vértices do grafo. O caso base é o grafo trivial.

Dado um grafo G , considere os filhos da raiz da árvore de decomposição modular, T , de G , que formam a partição de congruência de G em módulos fortes maximais próprios. Por indução, todo módulo forte de um dos fatores desta partição está representado em sua respectiva árvore de decomposição modular, que é o ramo de T a partir do nó correspondente ao filho. Pelo Corolário 2.15 e pelo Lema 2.11, G não contém outros módulos fortes. \square

Como conseqüência da Proposição 2.16, outra maneira 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.

Rotula-se cada nó interno da árvore de decomposição modular, segundo sua classificação pelo critério abaixo. Se o grafo quociente da partição de congruência usada na sua decomposição 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 2.6 exhibe a árvore de decomposição modular previamente apresentada na Figura 2.5, 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.

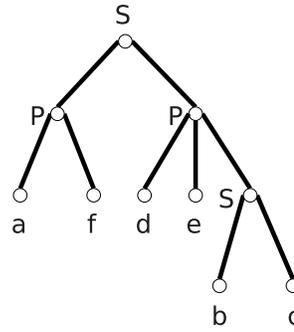


Figura 2.6: A árvore de decomposição modular do grafo da Figura 2.5a, com os nós internos rotulados.

Lema 2.17. *Se G é um grafo e T é a sua árvore de decomposição modular, então um subconjunto $M \subseteq V(G)$ é um módulo de G se, e somente se, uma das afirmativas vale:*

1. M é um nó de T ;
2. M é a união de filhos de um nó de T de rótulo serial ou paralelo.

Demonstração. Seja G um grafo e T sua árvore de decomposição modular, chame de \mathcal{P} a partição de congruência de G usada na raiz de T .

Prova-se que um módulo M de G atende às condições propostas, por indução no número de vértices de G . Se $M = V(G)$ ou se M for uma parte de \mathcal{P} , então é um nó de T . Caso contrário, pelo Corolário 2.10:

- $M = \bigcup_{X \in \mathcal{U}} X$ para algum \mathcal{U} , módulo de G/\mathcal{P} . Nota-se que $1 < |\mathcal{U}| < |\mathcal{P}|$ e, pelo Corolário 2.15, G/\mathcal{P} é completo ou sem arestas e, assim o nó raiz de T é rotulado como serial ou paralelo; ou
- M é módulo de $G[X]$ para alguma parte X de \mathcal{P} . Por hipótese de indução, M atende às condições propostas para a árvore de decomposição modular, T_X , de $G[X]$, mas T_X é um ramo de T .

Agora, mostra-se que se M atende às condições propostas, então é módulo de G . Um nó de T é um módulo de G segundo a Proposição 2.16. Prova-se por indução no número de vértices do grafo que, se M for uma união de filhos de um nó de T rotulado como serial ou paralelo, então é um módulo de G :

- Se $M = \bigcup_{X \in \mathcal{U}} X$ para algum \mathcal{U} , módulo de G/\mathcal{P} e a raiz de T tem rótulo serial ou paralelo, então G/\mathcal{P} é completo ou sem arestas e, portanto, \mathcal{U} é módulo de G/\mathcal{P} . Então pelo Lema 2.9, M é módulo de G .

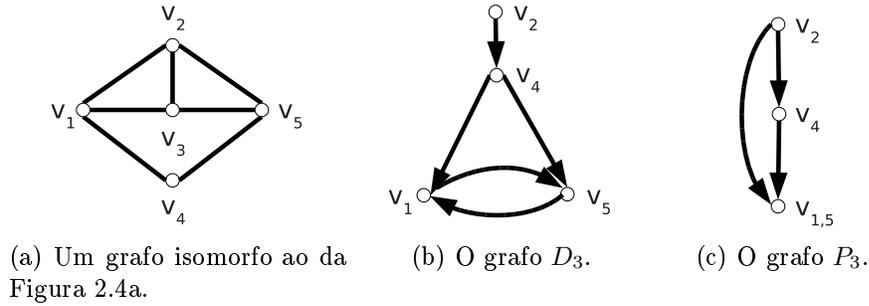


Figura 2.7: A construção de um CPO modular de um grafo.

- Caso contrário, M é a união dos filhos de algum nó de T que não a raiz e, portanto está contido em alguma parte X de \mathcal{P} . Por hipótese de indução M é módulo de $G[X]$ e, pelo Lema 2.2, é módulo de G . \square

2.3.2 Conjuntos parcialmente ordenados modulares

Em [KS03], Klein e Szwarcfiter propuseram uma representação para os módulos de um grafo que consiste em um conjunto de CPOs e que foi usada para resolver problemas relacionados aos módulos de um grafo. Especificamente, os problemas tratados foram: listar os módulos de um grafo, contar o número de módulos do grafo (sem enumerá-los), encontrar um módulo maximal de um grafo que satisfaz certas propriedades hereditárias e encontrar um módulo não trivial de um grafo cujo subgrafo induzido é conexo.

Dado um grafo G , a representação proposta para seus módulos consiste em um conjunto de CPOs, um para cada vértice de G . Para construir estes CPOs, primeiramente denote por v_i , $1 \leq i \leq |V(G)|$, cada um dos vértices de G e então considere o grafo orientado D_i , cujos vértices são dados por $V(G) \setminus \{v_i\}$ e cujo conjunto de arestas contém (v_j, v_k) se, e somente se:

- $\{v_j, v_k\} \in E(G)$ e $v_j \notin N_G(v_i)$; ou
- $\{v_j, v_k\} \notin E(G)$ e $v_j \in N_G(v_i)$.

Ao contrair as componentes fortemente conexas de D_i a um único vértice e, em seguida, adicionar os arcos para formar o fecho transitivo do grafo (veja Seção A.2), obtem-se o grafo P_i . Este grafo representa um CPO e é denominado *CPO modular* de G em relação a v_i (a Figura 2.7 exemplifica a construção dos grafos D_i e P_i). O conjunto dos grafos P_i para todo vértice v_i do grafo G forma sua representação.

Para expôr as propriedades dos CPOs modulares, serão necessárias algumas definições. Considere um CPO definido sobre um conjunto S e dado pelo grafo orientado P . O *ideal*

deste CPO é um subconjunto $I \subseteq S$ tal que para toda aresta $(x, y) \in E(P)$, se $y \in I$, então $x \in I$. A *decomposição em camadas* do mesmo CPO é uma seqüência (L_1, \dots, L_k) na qual todo $L_i \in S$ e $\bigcup_{1 \leq i \leq k} L_i = S$. Cada L_i contém os fontes do grafo $P[S \setminus \bigcup_{1 \leq j < i} L_j]$. Quando um CPO admite uma decomposição em camadas de forma que $(x, y) \in E(P)$ sempre que $x \in L_i$ e $y \in L_{i+1}$, diz-se que ele é um *CPO em camadas*.

O teorema abaixo estabelece a relação entre os CPOs modulares e os módulos de um grafo.

Teorema 2.18. *Dados um grafo G , um de seus vértices v_i e o CPO modular P_i de G relativo a v_i , então existe uma bijeção entre os ideais de P_i e os módulos de G que contém v_i .*

Pode-se observar, na Figura 2.7, que os ideais do CPO modular P_3 (\emptyset , $\{v_2\}$, $\{v_2, v_4\}$, $\{v_2, v_4, v_{1,5}\}$) estão em correspondência biunívoca com os módulos do grafo que contém v_3 ($\{v_3\}$, $\{v_3, v_2\}$, $\{v_3, v_2, v_4\}$ e $\{v_3, v_2, v_4, v_1, v_5\}$).

Além disso, os CPOs modulares apresentam importantes propriedades relacionadas a sua decomposição em camadas.

Teorema 2.19. *As seguintes afirmações são equivalentes:*

- P_i é um CPO modular de algum grafo G , relativo a um de seus vértices;
- P_i é um CPO em camadas;
- P_i é o fecho transitivo de um torneio bipartido acíclico.

O uso da representação e destas propriedades permitiu aos autores a solução dos problemas já mencionados no início desta seção. Uma desvantagem da representação é o seu espaço, $O(V(G)^3)$, em relação à árvore de decomposição modular, que ocupa apenas $O(V(G))$.

Capítulo 3

Algoritmos para Decomposição Modular

Neste capítulo são tratadas duas formas eficientes para construção da árvore de decomposição modular de um grafo, uma linear e outra mais complexa por um fator logarítmico. Muitos algoritmos que executam esta tarefa foram apresentados na literatura, dentre os quais, os algoritmos eficientes são de notável dificuldade para compreensão e implementação. Os algoritmos escolhidos se destacam por serem mais recentes e mais simples que seus predecessores de mesma complexidade.

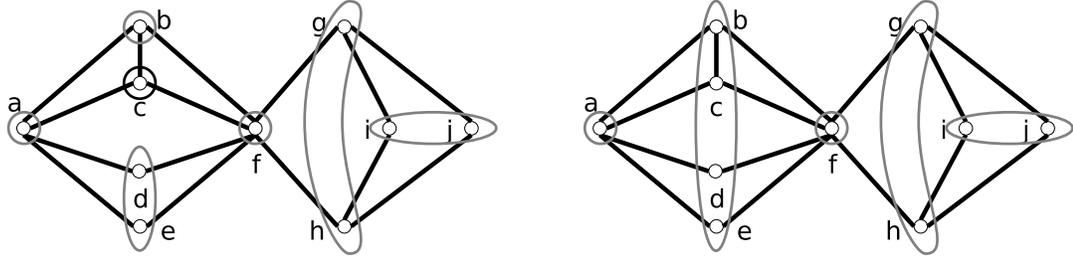
A Seção 3.1 trata da estratégia para decomposição apresentada por Ehrenfeucht e outros, posteriormente usada pelo algoritmo de Dahlhaus, Gustedt e McConnell, abordado na Seção 3.2. Termina o capítulo, a Seção 3.3, com dois algoritmos propostos por Habib e outros, que utilizam uma técnica completamente diferente.

3.1 Partição de congruência que isola um vértice

Dado um grafo G e um de seus vértices, v , define-se $\mathcal{M}(G, v)$ como o conjunto formado por $\{v\}$ e os módulos maximais entre os módulos de G que não contém v , ou seja, $\{v\}$ e todo módulo X tal que $v \notin X$ e qualquer $Y \supset X$, Y módulo de G , então $v \in Y$. A Figura 3.1a contém um exemplo destes conjuntos.

Lema 3.1. *Dado um grafo G e um de seus vértices, v , $\mathcal{M}(G, v)$ é uma partição de congruência de $V(G)$.*

Demonstração. Se algum $u \in V(G)$ não pertencesse a nenhum membro de $\mathcal{M}(G, v)$, o módulo maximal que contém u e não contém v não foi adicionado a $\mathcal{M}(G, v)$; por outro lado, se dois membros distintos, X e Y , de $\mathcal{M}(G, v)$ não forem disjuntos então, pelo Lema 2.4, $X \cup Y$ seria um módulo de G que não contém v , e contradiria a maximalidade



(a) As partes de $\mathcal{M}(G, c)$. Qualquer módulo de G não contido em uma destas partes, contém c .

(b) Os módulos máximos fortes próprios de G . Esta é a partição da raiz da decomposição modular do grafo.

Figura 3.1: Um grafo G com os módulos de $\mathcal{M}(G, c)$ e seus módulos fortes máximos próprios destacados.

de X e Y . Além disso, como todos os componentes de $\mathcal{M}(G, v)$ são módulos de G , $\mathcal{M}(G, v)$ é uma partição de congruência. \square

Além desta propriedade, Ehrenfeucht e outros, notaram que esta partição é útil para construção da decomposição modular:

Lema 3.2 ([EGMS94]). *Todos os módulos não unitários de $G/\mathcal{M}(G, v)$ contêm $\{v\}$ e são fortes.*

Demonstração. Se houvesse um módulo não unitário \mathcal{X} de $G/\mathcal{M}(G, v)$ que não contivesse $\{v\}$, então, pelo Lema 2.9, $\bigcup_{M \in \mathcal{X}} M$ seria um módulo de G que não contém v e, com isso, contradiria a maximalidade dos módulos contidos em $\mathcal{M}(G, v)$. Agora, suponha que existam dois módulos não triviais \mathcal{X} e \mathcal{Y} de $G/\mathcal{M}(G, v)$, que se sobreponham. Pelo Lema 2.4, $\mathcal{X} \Delta \mathcal{Y}$ seria módulo não trivial de $G/\mathcal{M}(G, v)$ e, como $\{v\} \in \mathcal{X} \cap \mathcal{Y}$, então $\mathcal{X} \Delta \mathcal{Y}$ não conteria $\{v\}$ e contradiria a maximalidade dos módulos contidos em $\mathcal{M}(G, v)$. \square

Lema 3.3 ([EGMS94]). *Um subconjunto \mathcal{X} de $\mathcal{M}(G, v)$ que contém $\{v\}$ é um módulo forte de $G/\mathcal{M}(G, v)$ se, e somente se, $\bigcup_{M \in \mathcal{X}} M$ for um módulo forte de G (que contém v). Todos demais módulos fortes de G – aqueles que não contem v – são subconjuntos de alguma parte de $\mathcal{M}(G, v)$.*

Demonstração. Seja M um módulo forte de G que contém v . Assim, os demais módulos de G ou estão contidos em M , ou o contêm, ou são disjuntos de M . Como os módulos de G que contém M não pertencem a $\mathcal{M}(G, v)$, $\mathcal{M}(G, v)$ é formado por módulos de G disjuntos de M e por módulos de G contidos em M e, então, existe um $\mathcal{X} \subset \mathcal{M}(G, v)$ tal que $\bigcup_{V \in \mathcal{X}} V = M$ e, pelo Lema 2.9, \mathcal{X} é módulo de $G/\mathcal{M}(G, v)$ e, portanto, forte.

Seja, agora, \mathcal{X} um módulo (forte) de $G/\mathcal{M}(G, v)$ e suponha que $\bigcup_{V \in \mathcal{X}} V = M$ não seja um módulo forte de G . Então, existe um módulo Y de G que se sobrepõe a M . Se

$v \notin Y$, escolha um vértice $u \in Y \cap M$ e seja $U \in \mathcal{M}(G, v)$ tal que $u \in U$, então, como $U \subset M$, ou $U \subset Y$, ou U e Y sobrepõem-se e, pelo Lema 2.4, $Y \cup U$ é módulo de G e não contém v , e, portanto, U não é maximal. Se $v \in Y$, escolha $u \in M \setminus Y$ e selecione U da mesma forma. Pelo Lema 2.4, $M\Delta Y$ é módulo de G , e então, ou $U \subset (M\Delta Y)$, ou U sobrepõe-se a $(M\Delta Y)$ e $U \cup (M\Delta Y)$ é um módulo de G que não contém v , mostrando que U não é maximal.

Todo módulo M de G que não contém v é um subconjunto de alguma parte de $\mathcal{M}(G, v)$. Caso não fosse, considere uma parte $X \in \mathcal{M}(G, v)$ tal que $M \cap X \neq \emptyset$. Então X não é maximal porque ou $M \supset X$, ou, pelo Lema 2.4, $M \cup X$ é um módulo. \square

Aplicando a Proposição 2.16 sobre os lemas 3.2 e 3.3, verifica-se que os ancestrais de v na árvore de decomposição modular de G são exatamente os nós internos da árvore de decomposição modular de $G/\mathcal{M}(G, v)$. Além disso, os demais nós da árvore de decomposição modular de G são v e os módulos fortes de G que não contém v , os quais, pelos lemas 3.3 e 2.8 são os nós das árvores de decomposição modular dos grafos $G[X]$ para todo $X \in \mathcal{M}(G, v)$ distinto de $\{v\}$.

E, com isto, tem-se uma estratégia geral para computar a decomposição modular de um grafo, esboçada no Algoritmo 1, abaixo, juntamente com um exemplo de seu uso na Figura 3.2. A condição interna valida que X é um módulo forte de G , removendo-o caso não seja.

Algoritmo 1 Estratégia para decomposição modular por Ehrenfeucht e outros.

Entrada: T e T_X , árvores de decomposição modular de $G/\mathcal{M}(G, v)$ e de todo $G[X]$: $X \in \mathcal{M}(G, v)$.

Saída: T é a árvore de decomposição modular de G .

Para toda folha X de T **faça**

 Seja T_X a árvore de decomposição modular de $G[X]$.

 Seja R_X a raiz de T_X .

 Seja R o pai de X em T .

Se R e R_X ambos estão rotulados como serial ou paralelo **então**

 Remova X de R e adicione os filhos de R_X com filhos de R .

Senão

 Remova X de R e adicione R_X como filho de R .

Fim Se

Fim Para

Retorne T .

Em seu trabalho, Ehrenfeucht e outros [EGMS94] apresentam um algoritmo recursivo, cujo tempo total de execução é $O(n^2)$, semelhante ao dado no Algoritmo 2. O tempo de execução é devido, principalmente, a técnica usada para computar da árvore de decom-

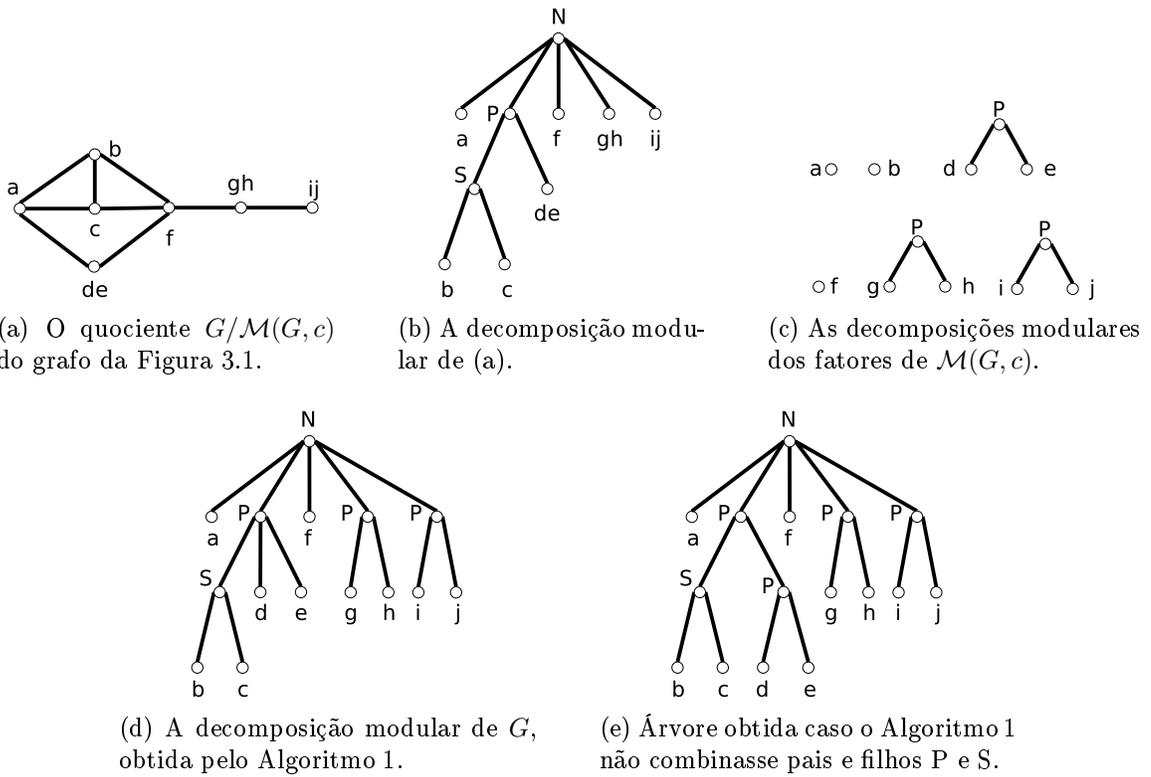


Figura 3.2: Um exemplo de decomposição modular segundo a técnica de Ehrenfeucht e outros.

posição modular de $G/\mathcal{M}(G, v)$, que, por este motivo, não será descrita. Ao invés, será exposto, na próxima seção, uma maneira de usar a técnica apresentada que resulta em um algoritmo linear.

Algoritmo 2 Decomposição modular por Ehrenfeucht e outros.

Entrada: G é um grafo simples.

Saída: T é árvore de decomposição modular de G .

Se G contém apenas um vértice **então**

Faça $T = G$.

Senão

Escolha um vértice v , qualquer, de G .

Compute a partição $\mathcal{P} = \mathcal{M}(G, v)$.

Compute a árvore de decomposição modular T_Q de G/\mathcal{P} .

Para toda folha X de T_Q **faça**

 Compute recursivamente a decomposição modular de $G[X]$ em T_X .

Fim Para

Use o Algoritmo 1 com T_Q e todos os $T_X : X \in \mathcal{M}(G, v)$ para obter a árvore T .

Fim Se

3.2 Recursão sobre adjacentes e não adjacentes

Usando estratégia semelhante, Dahlhaus e outros [DGM01] apresentaram um algoritmo linear para decomposição modular. Seu diferencial é a forma usada para computar $G/\mathcal{M}(G, v)$, sua decomposição modular e as árvores de decomposição modular de seus membros, que são detalhadas abaixo.

O Algoritmo 3 esboça os passos básicos para obter as árvores de decomposição modular de $G/\mathcal{M}(G, v)$ e dos grafos fatores da partição $\mathcal{M}(G, v)$. Um exemplo da aplicação de uma iteração é dado na Figura 3.3, na qual as decomposições em (b) e (d) são obtidas recursivamente.

O primeiro passo para detalhar o algoritmo é a descrição do cálculo das restrições das árvores de decomposição modular no passo 2, tarefa desempenhada pelo Algoritmo 4. Este algoritmo calcula apenas a restrição da árvore de decomposição modular de G em relação aos vizinhos de v . O algoritmo para obter a restrição em relação aos não vizinhos é essencialmente o mesmo, apenas substituindo $N_G(v)$ por $\overline{N}_G(v)$.

A aplicação do Algoritmo 4 sobre as árvores (b) e (d) da Figura 3.3 produz exatamente as árvores de decomposição modular dos fatores de G em relação a $\mathcal{M}(G, c)$, vistos na Figura 3.2c. Note que, embora os vértices d e e tenham sido separados no primeiro passo do algoritmo, quando aplicado sobre a árvore da Figura 3.3d, eles são unidos novamente no

Algoritmo 3 Decomposição modular por Dahlhaus e outros.**Entrada:** G é um grafo.**Saída:** Para algum vértice v de G , \mathcal{M} é $\mathcal{M}(G, v)$, T é árvore de decomposição modular de $G/\mathcal{M}(G, v)$ e T_X é a árvore de decomposição modular de $G[X]$ para todo $X \in \mathcal{M}$.

- 1: Encontre as árvores de decomposição modular T_N e $T_{\overline{N}}$, de $G[N_G(v)]$ e de $G[\overline{N_G(v)}]$, recursivamente.
- 2: Calcule as restrições da árvore de decomposição modular de G em relação a $N_G(v)$ e a $\overline{N_G(v)}$ usando T_N e $T_{\overline{N}}$.
 {Tais restrições fornecem T_X para todo $X \in \mathcal{M}(G, v)$ }
- 3: Compute \mathcal{M} , agrupando os vértices de cada árvore obtida no passo anterior.
- 4: Compute T , a árvore de decomposição modular de $G/\mathcal{M}(G, v)$.

Algoritmo 4 Cálculo das restrições da decomposição modular.**Entrada:** G é um grafo e T_N é a árvore de decomposição modular de $G[N_G(v)]$.**Saída:** T_N^* é a restrição da árvore de decomposição modular de G a $N_G(v)$.

Remova de T_N todos os nós que não são módulos de G . Os filhos dos nós removidos se tornam novas árvores. Chame essa floresta de T_N^* .

Agrupe novamente dois ou mais filhos de um mesmo nó serial ou paralelo, removido de T_N , desde que seus vértices possuam os mesmos vizinhos em $V(G) \setminus N_G(v)$.

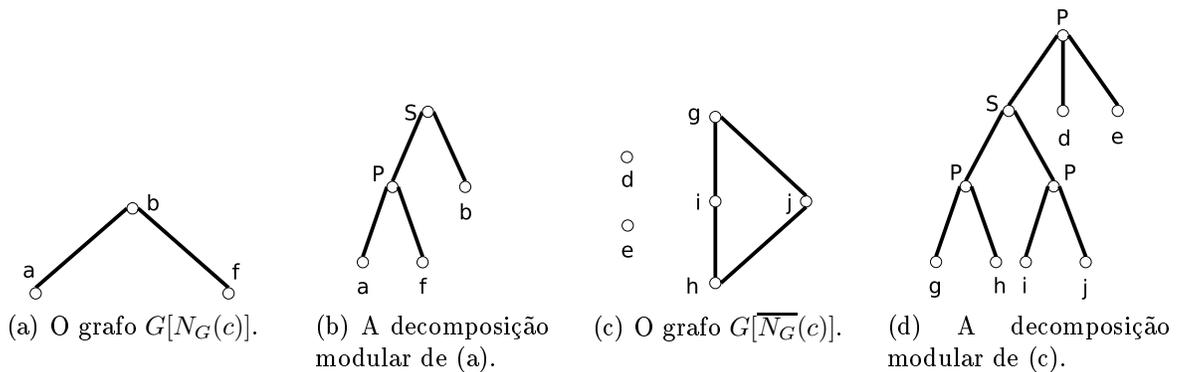


Figura 3.3: Uma iteração do Algoritmo 3 sobre o grafo G da Figura 3.1.

segundo passo, pois d e e têm o mesmo conjunto de vizinhos em $V(G) \setminus \overline{N_G}(c)$, que é $\{a, f\}$. Em seguida, mostra-se que este passo realmente obtém as decomposições modulares dos grafos induzidos pelas partes de $\mathcal{M}(G, v)$.

Lema 3.4. *As decomposições modulares de $G[X]$ para $X \in \mathcal{M}(G, v)$ são dadas pelas restrições das decomposições modulares de G a $G[N_G(v)]$ e a $G[\overline{N_G}(v)]$.*

Demonstração. Seja $X \in \mathcal{M}(G, v)$, $X \neq \{v\}$. Como X é um módulo de G disjunto de $\{v\}$, está totalmente contido em $N_G(v)$ ou em $\overline{N_G}(v)$ – chame de N o conjunto dentre estes que contém X . Pelo Lema 2.5, X é módulo de $G[N]$ e pelo Lema 2.17, ou é um nó da árvore de decomposição modular, T , desse grafo ou é a união de alguns filhos de um nó serial ou paralelo de T . Como X é módulo de G , o processo de restrição de T a $G[X]$, que produz T' , não remove de T nenhum dos submódulos de X . Se X era um nó de T , continua sendo em T' ; se era a união de dois ou mais filhos de um nó serial ou paralelo de T , tais filhos foram mantidos em T' e, caso seu pai tenha sido excluído, um novo foi adicionado contendo todos os filhos que formam X . Para finalizar, X não possui um pai em T' , pois neste caso tal pai seria um módulo de G (já que não foi removido durante a restrição) que não contém v e, portanto, X não seria maximal. \square

O último passo é a construção da árvore de decomposição modular de $G/\mathcal{M}(G, v)$ no passo 4 do Algoritmo 3. Este passo se resume a encontrar todos os módulos fortes de G que contém v e determinar sua relação de inclusão.

Primeiramente, o conjunto de vértices $V(G) \setminus \{v\}$ é particionado em classes de equivalência determinadas pela relação definida de forma que dois vértices x e y se relacionam se satisfizerem uma das seguintes condições:

- Se x e y pertencem à $N_G(v)$:
 - x e y pertencem a mesma componente conexa de $\overline{G[N_G(v)]}$; ou
 - x e y pertencem a componentes conexas distintas de $\overline{G[N_G(v)]}$, ambas contidas em um módulo de G totalmente contido em $N_G(v)$.
- Se x e y pertencem à $\overline{N_G}(v)$:
 - x e y pertencem a mesma componente conexa de $G[\overline{N_G}(v)]$; ou
 - x e y pertencem a componentes conexas distintas de $G[\overline{N_G}(v)]$, ambas contidas em um módulo de G totalmente contido em $\overline{N_G}(v)$.

A transitividade da relação vem do fato de que a união de dois módulos que se sobrepõem em G também é módulo de G . As classes definidas por esta relação são chamadas de *blocos básicos*. Denota-se por \mathcal{B} e $\overline{\mathcal{B}}$ os conjuntos daqueles blocos básicos contidos em

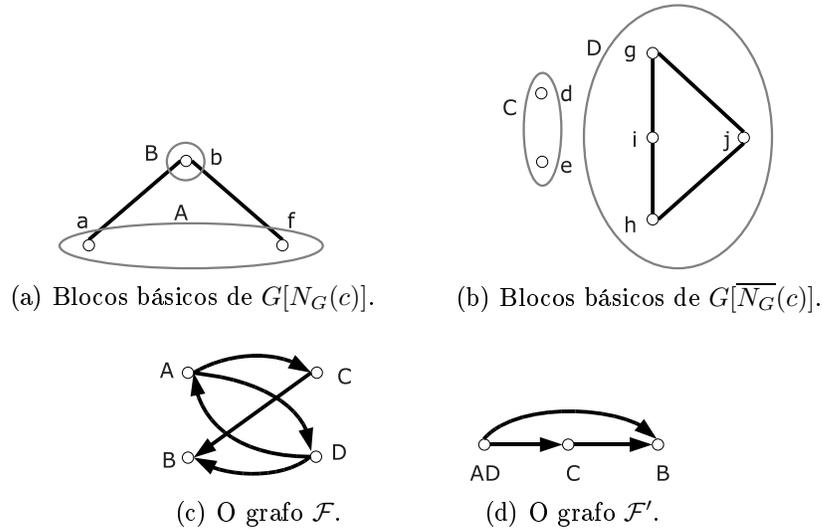


Figura 3.4: Exemplo de construção da decomposição de $G/\mathcal{M}(G, c)$ pelo Algoritmo 5.

$N_G(v)$ e em $\overline{N_G}(v)$, respectivamente. Os blocos básicos são módulos de $G[N_G(v)]$ ou de $G[\overline{N_G}(v)]$. Note que, na Figura 3.4b, os vértices d e e foram unidos no mesmo bloco básico pois formam um módulo em G .

Constrói-se, agora, um grafo orientado bipartido \mathcal{F} , de bipartição $\{\mathcal{B}, \overline{\mathcal{B}}\}$. As arestas do grafo são definidas de forma que $(X, Y) \in E(\mathcal{F})$ se e somente se $X \in \mathcal{B}$, $Y \in \overline{\mathcal{B}}$ e existem $x \in X$ e $y \in Y$ tais que $\{x, y\} \in E(G)$, ou $X \in \overline{\mathcal{B}}$, $Y \in \mathcal{B}$ e existem $x \in X$ e $y \in Y$ tais que $\{x, y\} \notin E(G)$.

O objetivo do grafo \mathcal{F} é mostrar as relações de inclusão forçadas por módulos contendo v , ou seja, se $(X, Y) \in E(\mathcal{F})$, então qualquer módulo de G que contenha $\{v\} \cup X$, conterá Y . Assim sendo, uma componente fortemente conexa neste grafo significa que qualquer módulo de G que contenha v e um dos blocos básicos da componente obrigatoriamente contém toda ela.

Toma-se, então, o grafo \mathcal{F}' obtido de \mathcal{F} substituindo suas componentes fortemente conexas por novos vértices que contém a união dos blocos básicos de cada componente. O grafo \mathcal{F}' é acíclico e admite uma ordenação topológica única (veja Seção A.3). Um *sufixo* de uma ordem O sobre um conjunto S é um conjunto $U \subseteq S$ tal que, se $a \in U$, então $b \in U$ para todo $b \in S$ tal que $b \geq a$ em O . Os sufixos da ordenação topológica dada pelo grafo \mathcal{F}' da Figura 3.4d são \emptyset , $\{B\}$, $\{C, B\}$ e $\{AD, C, B\}$ (AD representa um único elemento). A união de $\{v\}$ com os vértices contidos nos elementos dos sufixos da ordenação topológica, fornece os módulos fortes de G que contém v . Para o grafo \mathcal{F}' da mesma figura, tais módulos são $\{c\}$, $\{c, b\}$, $\{c, b, d, e\}$ e $\{c, b, d, e, a, f, g, h, i, j\}$.

O Algoritmo 5 resume os passos descritos. Um exemplo do procedimento é dado na Figura 3.4.

Algoritmo 5 Cálculo de $G/\mathcal{M}(G, v)$ por Dahlhaus e outros

Entrada: G é um grafo simples, v é um de seus vértices e \mathcal{M} é $\mathcal{M}(G, v)$.

Saída: T é árvore de decomposição modular de $G/\mathcal{M}(G, v)$.

Encontre os blocos básicos de G em relação a v .

Compute o grafo orientado de implicação \mathcal{F} a partir dos blocos básicos.

Compute o grafo fortemente conexo \mathcal{F}' de \mathcal{F} .

Obtenha a ordenação topológica O de \mathcal{F}' .

Seja T tal que $V(T) = \{v\}$ e $E(T) = \emptyset$.

Para todo vértice x de \mathcal{F}' , visitado na ordem inversa de O **faça**

Adicione uma nova raiz u a árvore a T .

Ligue a nova raiz a antiga.

Adicione como filho de u um vértice para cada elemento de $\mathcal{M} \setminus \{v\}$ contido em x .

Rotule o vértice u segundo o grafo quociente da partição que o nó representa.

Fim Para

Primeiramente, serão mostradas a relação entre os módulos de G que contém v e as estruturas dos grafos \mathcal{F} e \mathcal{F}' .

Lema 3.5. *Um conjunto X formado por v e a união de blocos básicos é um módulo de G se, e somente se, não existe uma aresta (A, B) no grafo \mathcal{F} tal que $A \subseteq X$ e $B \not\subseteq X$.*

Demonstração. Suponha que X não é um módulo de G , então existe um vértice $r \notin X$ que distingüe dois vértices contidos em X . Se $r \in N_G(v)$, então r deve ser não adjacente a algum vértice $s \in X$ (note que $s \neq v$). Se $s \in N_G(v)$, r e s estariam no mesmo bloco básico (justamente porque r e s são não vizinhos), contradição. Assim, $s \in \overline{N_G}(v)$ e existe uma aresta (A, B) no grafo \mathcal{F} do bloco básico A que contém r ao bloco básico B que contém s . Com raciocínio análogo (trocando-se adjacência por não adjacência) trata-se do caso em que $r \in \overline{N_G}(v)$. Portanto, pela contrapositiva, se não existe aresta no grafo \mathcal{F} de um bloco básico contido em X a outro, não contido em X , então X é módulo de G .

Para completar a prova, suponha que existe uma aresta no grafo \mathcal{F} de um bloco básico A contido em X a outro, B , não contido em X . Suponha que $A \in \mathcal{B}$ e, portanto, $B \in \overline{\mathcal{B}}$. A aresta no grafo \mathcal{F} implica que existe uma aresta $\{a, b\}$ em G , tal que $a \in A$ e $b \in B$ e, assim, b distingüe a de v , fazendo com que X não seja módulo. O caso em que $A \in \overline{\mathcal{B}}$ é simétrico, invertendo o papel das adjacências e não-adjacências. \square

Note que o Lema 3.5 não implica que todo módulo de G que contém v pode ser formado pela união de v com vértices de blocos básicos, o que, de fato, é falso. Em contrapartida, a afirmação vale para módulos fortes de G que contém v , como demonstrado no Lema 3.6.

Lema 3.6. *Um conjunto X de $V(G)$ que contém v é um módulo forte de G se, e somente se, $X \setminus \{v\}$ é a união de blocos básicos e não existe uma aresta (A, B) no grafo \mathcal{F} tal que $A \subseteq X$ e $B \not\subseteq X$.*

Demonstração. Seja X um módulo forte de G que contém v . Suponha que $X \setminus \{v\}$ não seja a união de blocos básicos. Então, existe um bloco básico Y tal que $Y \cap X \neq \emptyset$ e $Y \not\subset X$ (inclui-se aqui o caso em que $X \subset Y$).

Considere o caso em que $Y \in \overline{\mathcal{B}}$ e, portanto, é composto por vértices de componentes conexas de $G[\overline{N_G(v)}]$. Se $Y \cap X$ sobrepõe-se aos vértices de alguma de tais componentes conexas, então existem dois vértices a e b , adjacentes em G , tais que $a \in X \cap Y$ e $b \in Y \setminus X$. Desta forma, b testemunha que X não é módulo de G (pois b é vizinho de a e não é de v). No outro caso, $Y \cap X$ e $Y \setminus X$ são formados ambos por vértices de componentes conexas de $G[\overline{N_G(v)}]$. Como seus vértices foram agrupados em um único bloco básico, existe um módulo M de G que contém dois vértices a e b tais que $a \in X \cap Y$ e $b \in Y \setminus X$. Logo, M sobrepõe-se a X e este não seria forte.

O caso em que $Y \in \mathcal{B}$ é simétrico a este em relação à adjacência. Pelo Lema 3.5 não existe aresta em \mathcal{F} de um bloco básico contido em X para outro não contido em X .

Seja $X \subset V(G)$ tal que $v \in X$, $X \setminus \{v\}$ seja uma união de blocos básicos e que não exista aresta em \mathcal{F} de um bloco básico contido em X para outro, não contido em X . Pelo Lema 3.5, X é um módulo de G . Se X não for um módulo forte, então existe um módulo Y que sobrepõe-se a X . Caso $v \notin Y$, Y está completamente contido em $N_G(v)$ ou em $\overline{N_G(v)}$ (de outro modo, v testemunharia que Y não é módulo) e estaria contido num único bloco básico, não sobrepondo-se a X . Caso $v \in Y$, $X \Delta Y$ é um módulo de G que não contém v e sobrepõe-se a X , contradição ao caso anterior. Logo, X é um módulo forte de G . \square

Em seguida, é justificado o uso do grafo \mathcal{F}' para obter os módulos fortes de G que contém v . A iniciar, prova-se que \mathcal{F}' admite uma ordenação topológica única e, portanto, \mathcal{F} e \mathcal{F}' são conexos.

Lema 3.7. *Os grafos \mathcal{F} e \mathcal{F}' são conexos e \mathcal{F}' admite uma ordenação topológica única.*

Demonstração. Pode-se obter uma ordenação topológica única para um grafo, removendo cada um de seus vértices sorvedouros e tomando a ordem inversa. Caso a ordenação seja única, em cada passo, o subgrafo induzido conterá apenas um vértice sorvedouro.

Da contração dos componentes fortemente conexos de \mathcal{F} , resulta que \mathcal{F}' é acíclico e, portanto admite uma ordenação topológica. Suponha, então, que, na aplicação deste processo ao grafo \mathcal{F}' , em um passo qualquer, o subgrafo induzido contém ao menos dois sorvedouros, A e B . Seja X o conjunto de vértices de \mathcal{F}' já removidos antes deste passo (X pode ser vazio). Pelo Lema 3.6, a união dos vértices de G contidos nos elementos de A e de X com $\{v\}$ é um módulo forte de G . Da mesma forma, a união dos vértices de G contidos nos elementos de B e de X com $\{v\}$ também é módulo forte de G . Como estes módulos se sobrepõem (v é comum a ambos), esta contradição mostra que \mathcal{F}' tem uma ordenação topológica única.

Caso \mathcal{F}' possuísse mais de uma componente conexa, cada uma delas seria uma árvore e possuiria ao menos um vértice sorvedouro. Sendo \mathcal{F}' conexo, claramente \mathcal{F} o é também. \square

Por fim, mostra-se que o uso dos sufixos da ordenação topológica de \mathcal{F}' produz os módulos fortes de G que contém v .

Teorema 3.8. *Um subconjunto M dos vértices de G que contém v é módulo forte de G se, e somente se, $M \setminus \{v\}$ é a união de vértices contidos em elementos de um sufixo da ordenação topológica de \mathcal{F}' .*

Demonstração. Pelo Lema 3.7, \mathcal{F}' possui uma única ordenação topológica, portanto seus sufixos são todos bem definidos.

Seja X um módulo forte de G que contém v . Pelo Lema 3.6, $X \setminus \{v\}$ é união de vértices contidos em blocos básicos e não existe aresta em \mathcal{F} de um destes blocos básicos para outro bloco básico, não contido em X . Com isso, não existe componente fortemente conexa em \mathcal{F} que contenha um bloco básico contido em X a outro, não contido em X . Portanto, $X \setminus \{v\}$ é formado pela união de vértices de G contidos nos conjuntos representados pelos vértices de \mathcal{F}' . Estes vértices de \mathcal{F}' formam um conjunto \mathcal{M} que é um sufixo da sua ordenação topológica. Caso não seja, existe ao menos um vértice de \mathcal{F}' não contido em \mathcal{M} após o primeiro vértice de \mathcal{M} , na ordenação topológica. Seja u o primeiro vértice que atende estas condições. Como não há ordenação topológica do grafo em que u fique antes do primeiro vértice de \mathcal{M} , existe alguma aresta de um vértice em \mathcal{M} para u , não contido em \mathcal{M} . Esta aresta corresponde, em \mathcal{F} , a uma aresta de um bloco básico contido em X para outro, não contido em X , contradição.

Um sufixo da ordenação topológica de \mathcal{F}' corresponde, em \mathcal{F} , a um conjunto de blocos básicos e não existe aresta de um destes blocos básico para outro, fora do conjunto. Então, pelo Lema 3.6, a união dos vértices de G que correspondem aos vértices de \mathcal{F}' com v é um módulo forte de G . \square

Pelos lemas 3.2 e 3.3, os módulos fortes não triviais de $G/\mathcal{M}(G, v)$ correspondem aos módulos fortes não triviais de G que contém v . Assim, esta etapa é suficiente para construir a decomposição modular de $G/\mathcal{M}(G, v)$.

3.3 Permutações fatorizadas

Uma outra técnica, mais recente e um pouco mais simples para obter a decomposição modular de um grafo, envolve duas fases: obter uma ordem dos vértices do grafo que seja uma permutação fatorizada e, então, construir a árvore de decomposição modular do grafo.

Uma *permutação fatorizada* é uma ordem dos vértices do grafo de forma que os vértices de todo módulo forte do grafo estão consecutivos. É fácil obter uma ordem dessa natureza a partir da árvore de decomposição modular do grafo – basta percorrer a árvore numa busca em profundidade, produzindo na saída as folhas da árvore.

Nota-se que um mesmo grafo possui várias permutações fatorizadas. Por exemplo, o grafo da Figura 3.1 tem como permutação fatorizada a seqüência $(a, b, c, d, e, f, g, h, i, j)$, obtida numa simples busca em profundidade por sua árvore de decomposição modular, que está na Figura 3.2(d). Contudo, alternando a ordem dos filhos de alguns nós da árvore e repetindo a busca, obtem-se outra permutação fatorizada, $(h, g, d, b, c, e, a, i, j, f)$.

Em seguida, os dois passos envolvidos no algoritmo são tratados.

3.3.1 Obter uma permutação fatorizada de um grafo

O algoritmo que será descrito foi apresentado por Habib, Paul e Viennot [HPV99] e executa a tarefa de construir uma permutação fatorizada para os vértices de um grafo G em tempo $O(|V(G)| + |E(G)| \log |V(G)|)$. Este algoritmo foi escolhido pela sua simplicidade e pela existência de uma extensão proposta por Habib, de Montgolfier e Paul [HdMP04] cuja complexidade é linear.

Em cada momento, os vértices do grafo são mantidos numa estrutura lógica que corresponde a um CPO (veja A.3) de vértices do grafo (veja exemplo na Figura 3.5). Cada conjunto de vértices incomparáveis do grafo está contido numa parte, representada por uma caixa retangular. As partes estão totalmente ordenadas entre si e formam uma partição dos vértices do grafo.

O algoritmo começa com todos os vértices contidos dentro de uma única parte, de forma que não há nenhuma relação de ordem, e termina quando toda parte contém apenas um vértice, formando uma ordem total. Para atingir o objetivo a estrutura é alterada de acordo com as duas regras descritas abaixo e exemplificadas na Figura 3.5.

- A regra do centro: É aplicada a uma parte P e a divide em três novas partes consecutivas, que ficam na mesma posição de P . Escolhe-se um vértice c , chamado de *centro*, contido em P e, então, coloca-se na primeira das novas partes os vértices de P não adjacentes a c , na segunda, apenas o vértice c e, na terceira, os demais vértices de P , adjacentes a c .
- A regra do pivô: Escolhe-se um vértice *pivô* p de uma parte P e então, toda outra parte que contiver vértices adjacentes a p e vértices não adjacentes a p é dividida em duas, separando seus vértices nestes dois casos. Se a parte considerada estava entre a parte que contém o centro c e a que contém o pivô p , então a nova parte com os vértices adjacentes a p precede a outra (Figura 3.5b), caso contrário (Figura 3.5c) a ordem é a inversa.

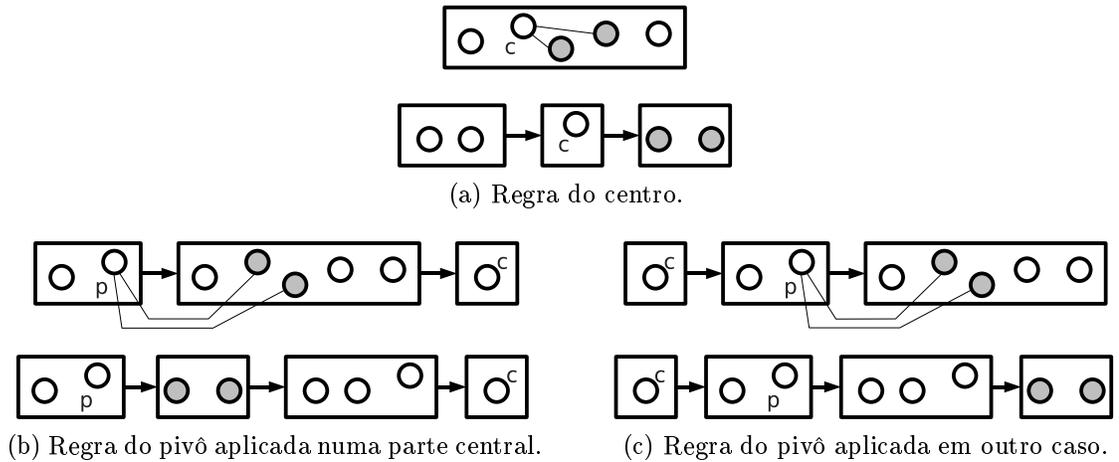


Figura 3.5: Exemplo de aplicação das regras usadas para obter a permutação fatorizada. Os vértices marcados com c e p são o centro e o pivô. Linhas conectando os círculos representam arestas do grafo (quando relevantes).

Ao aplicar a regra do centro sobre uma parte, as duas partes criadas (excluindo a parte que contém o centro) serão usadas como fontes de novos pivôs. Porém, para atingir a complexidade desejada, a menor parte criada deve ser usada antes da maior. Esta característica é implementada no algoritmo através de duas filas, *Pivôs*, à qual é adicionada a menor parte, e *Módulos*, que recebe a maior parte. Se as partes possuírem o mesmo tamanho, não importa qual é colocada em cada fila.

Além disso, quando uma parte é removida da fila *Pivôs*, todos os seus vértices são usados para a aplicação da regra do pivô. Já quando uma parte é removida da fila *Módulos*, apenas um de seus vértices é usado na aplicação da regra do pivô. Se uma parte que foi removida da fila *Módulos* é selecionada para aplicação da regra do centro, o mesmo vértice usado como pivô deve ser usado como centro. Isto é implementado através de um campo existente em cada parte, chamado *PrimeiroPivô*, inicialmente indefinido.

O Algoritmo 6 descreve o método proposto. No seu início, a regra do centro é aplicada a um vértice qualquer da única parte existente, adicionando conjuntos de pivôs às duas filas. Os pivôs são removidos da fila *Pivôs* e são usados na aplicação da regra do pivô. Se a fila *Pivôs* esvazia sem que a partição esteja pronta (formada por partes unitárias), conjuntos de pivôs são retirados de *Módulos* e usados. Quando as duas filas ficam vazias e ainda existem partes com mais de um vértice, uma de tais partes é escolhida e a regra do centro é aplicada a ela, reiniciando o processo.

A regra do pivô também adiciona elementos às filas. Suponha que uma parte P é dividida em duas por esta regra. Se P fazia parte da fila *Pivôs*, as duas novas partes devem estar contidas na fila, para manter todos os pivôs previamente selecionados. Caso

contrário a menor das novas partes é adicionada a fila Pivôs e a maior é adicionada a Módulos, substituindo P , caso estivesse contida em Módulos.

Algoritmo 6 Construção da permutação fatorizada.

Entrada: G é um grafo simples.

Saída: O é uma permutação fatorizada de seus vértices.

Pivôs $\leftarrow \emptyset$, Módulos $\leftarrow \emptyset$, $O \leftarrow (V(G))$.

Enquanto Algum elemento de O não for unitário **faça**

Se Módulos = \emptyset **então**

Escolha um elemento qualquer $X \in O$ tal que $|X| > 1$.

Seja $x = X$.PrimeiroPivô, se existir, ou um elemento qualquer de X , caso contrário.

Aplice a regra do centro usando x .

Adicione a menor parte entre $N_G(x) \cap X$ e $\overline{N_G(x)} \cap X$ a Pivôs.

Adicione a maior parte entre $N_G(x) \cap X$ e $\overline{N_G(x)} \cap X$ a Módulos.

Senão

Remova o primeiro elemento, X , de Módulos.

Escolha um vértice x , qualquer, de X e adicione $\{x\}$ à Pivôs.

X .PrimeiroPivô $\leftarrow x$.

Fim Se

Enquanto Pivôs $\neq \emptyset$ **faça**

Remova um elemento E qualquer de Pivôs.

Para todo $p \in E$ **faça**

Aplice a regra do pivô usando p .

Atualize Pivôs e Módulos.

Fim Para

Fim Enquanto

Fim Enquanto

Um exemplo de execução está na Figura 3.6. As partes são representadas por seus vértices em uma ordem qualquer, delimitados por $[$ e $]$. O sobrescrito P indica que uma parte pertence à fila Pivôs e M , à fila Módulos. Um vértice pode possuir o sobrescrito P quando é adicionado isoladamente à fila Pivôs (sem os demais de sua parte). A aplicação da regra do centro com um vértice v é denotadas por $C : v$ e a aplicação da regra do pivô, por $P : v$. Sucessivas aplicações da regra do pivô são resumidas da forma $P : a, b, c, \dots$, sendo que somente a aplicação com o último vértice pode provocar alguma alteração da estrutura. A remoção de um elemento da fila Módulos é representada por M .

A correção do algoritmo baseia-se nas seguintes invariantes. Note que quando o algoritmo esvazia as filas Pivôs e Módulos e ainda existem partes com mais de um vértice, estas são todas módulos do grafo.

$$\begin{array}{l}
[dacbfgeijh] \quad \rightarrow C:c \rightarrow [dgeijh]^M[c][abf]^P \quad \rightarrow P:a \rightarrow \\
[gijh]^M[de]^P[c][abf] \quad \rightarrow P:b, f \rightarrow [ij]^P[gh]^M[de]^P[c][abf] \quad \rightarrow P:d \rightarrow \\
[ij]^P[gh]^M[de][c][b]^P[af]^M \quad \rightarrow P:e, i, j, b \rightarrow [ij][gh]^M[de][c][b][af]^M
\end{array}$$

(a) Início até o esvaziamento de Pivôs.

$$\begin{array}{l}
\rightarrow M \rightarrow [ij][g^Ph][de][c][b][af]^M \quad \rightarrow P:g \rightarrow [ij][gh][de][c][b][a]^P[f]^M \\
\rightarrow P:a \rightarrow [ij][gh][de][c][b][a][f]^M \quad \rightarrow M \rightarrow [ij][gh][de][c][b][a][f^P] \\
\rightarrow P:f \rightarrow [ij][gh][de][c][b][a][f]
\end{array}$$

(b) Processamento de Módulos.

$$\begin{array}{l}
\rightarrow C:i \rightarrow [j]^M[i][gh][de][c][b][a][f] \quad \rightarrow M \rightarrow [j^P][i][gh][de][c][b][a][f] \\
\dots \quad \rightarrow C:g \rightarrow \dots \rightarrow C:e \rightarrow \dots \quad [j][i][h][g][d][e][c][b][a][f]
\end{array}$$

(c) Quebra das partições remanescentes.

Figura 3.6: Exemplo de execução do algoritmo para criar uma permutação fatorizada do grafo da Figura 3.1.

Lema 3.9. *Dado um módulo forte M de um grafo G , em qualquer momento da execução do algoritmo, as partes do CPO armazenado pelo algoritmo que interceptam M são contíguas e, caso sejam mais de duas, todas, exceto talvez a primeira e a última, estão contidas em M .*

Lema 3.10. *Se, em qualquer momento da execução do algoritmo, alguma parte do CPO armazenado pelo algoritmo não é um módulo de G , então existe algum pivô em Pivôs ou em Módulos que refina a partição.*

3.3.2 Obter a decomposição modular da permutação fatorizada

Em seguida, apresenta-se uma simples adaptação do algoritmo de Capelle, Habib e de Montgolfier [CHdM02], que originalmente também é capaz de construir a decomposição modular de grafos orientados, porém seu estudo será limitado ao caso não orientado.

Para facilidade de entendimento, o algoritmo será visto como composto por cinco fases¹ sucessivamente aplicadas à entrada, todas executando em tempo linear, como esboçado pelo Algoritmo 7.

Para a construção da seqüência com parênteses P a partir da permutação fatorizada de entrada, esta é percorrida da esquerda para direita a fim de cercar com parênteses os vértices da fratura esquerda (se houver) e da fratura direita (se houver) de cada par de

¹A sexta fase presente no algoritmo original só é necessária para grafos orientados.

Algoritmo 7 Construção da árvore de decomposição modular.

Entrada: G é um grafo simples e O é uma permutação fatorizada de seus vértices.

Saída: T é árvore de decomposição modular de G .

Produza uma seqüência com parênteses P a partir de O e G .

Construa a árvore de fraturas T a partir de P .

Identifique nós falsos de T .

Remova os nós falsos de T .

Separe os nós agregados de T .

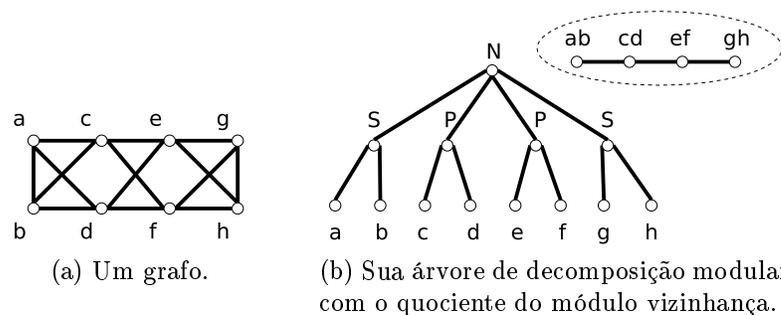


Figura 3.7: Exemplo para técnica de decomposição por permutações fatorizadas.

vértices consecutivos em O . Dado um par de vértices (x, y) consecutivos em O , a sua *fratura esquerda* é a seqüência de vértices de a a x em O na qual a é o vértice mais à esquerda (em O) que distingüe o par (isto é, o vértice é vizinho de um dos vértices do par e não é do outro), se houver. Simetricamente, a *fratura direita* do par é a seqüência de y a b em O na qual b é o vértice mais à direita que distingüe x de y . Por fim, a palavra é cercada por um par de parênteses adicional. Note que o tamanho de P é $O(n)$, pois para cada par, introduz-se no máximo quatro parênteses. O grafo da Figura 3.7, que será usado como exemplo de decomposição, admite a permutação fatorizada $abcdefgh$ e a construção de sua palavra com parênteses é exemplificada na Tabela 3.1.

Uma possível implementação linear para esta fase começa preenchendo dois vetores, A e F , com o número de parênteses que abrem e fecham, respectivamente, após cada vértice, na ordem dada (o índice zero de A indica parênteses antes do primeiro vértice).

Par considerado	Palavra formada	Par considerado	Palavra formada
ab	abcdefgh	bc	ab(cdef)gh
cd	ab(cdef)gh	de	(ab(cd)(ef)gh)
ef	(ab(cd)(ef)gh)	fg	(ab((cd)(ef))gh)
gh	(ab((cd)(ef))gh)	<i>resultado</i>	((ab((cd)(ef))gh))

Tabela 3.1: Construção da palavra P para o grafo da Figura 3.7.

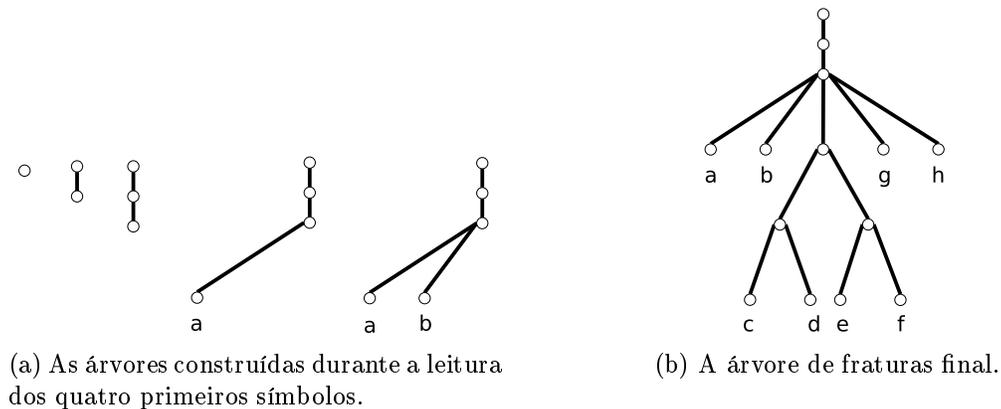


Figura 3.8: Construção da árvore de fraturas.

Em seguida, percorre-se tais vetores para produzir a seqüência com parênteses P .

Cada par de vértices é analisado, da esquerda para direita, procurando-se o menor e o maior vértice que distingüe os vértices do par, a partir das listas de adjacências dos mesmos. Mantendo o tempo tomado para analisar um par de vértices proporcional ao tamanho das listas de adjacências de ambos os vértices, o tempo total da fase (a análise de todos os pares) será proporcional ao número de arestas do grafo e o algoritmo será linear. Para tal, usa-se um vetor de bits, B , com uma posição para cada vértice do grafo, na mesma ordem da seqüência original. Com todos os bits de B inicialmente com valor 0, percorre-se a lista de adjacências do primeiro vértice do par, marcando em B com 1 os vértices adjacentes. Então percorre-se a lista de adjacências do segundo vértice do par, invertendo o bit correspondente em B aos vértices adjacentes. Agora basta procurar o primeiro e último bit em B com valor 1 (apenas entre os vértices adjacentes a um dos vértices do par) e atualizar, por fim, os vetores A e F , quando necessário.

O passo seguinte constrói a árvore de fraturas T a partir de P , processando a seqüência da esquerda para a direita. A partir de um nó raiz inicial N , lê-se um símbolo por vez e executa-se um dos seguintes passos (um exemplo é dado na Figura 3.8):

- '(': Cria um novo nó filho de N e o atribui a N .
- ')': Atribui a N o seu nó pai.
- x : Cria um filho de N e rotula-o por x .

Note que o tamanho de T é $O(n)$ pois cada vértice de T corresponde a um vértice ou um '(' em P . Daqui em diante, identifica-se um nó de T pelo conjunto das folhas da subárvore induzida em T pelo nó, notando que estas folhas são consecutivas em O .

A árvore de fraturas T , formada na segunda fase do Algoritmo 7, é uma aproximação da decomposição modular do grafo, pois suas folhas correspondem exatamente aos vértices

do grafo e alguns dos módulos fortes do grafo correspondem às folhas de alguma subárvore de T , fatos garantidos pelas seguintes propriedades.

Lema 3.11 ([CHdM02]). *Sejam G um grafo, M um de seus módulos fortes e G' o grafo obtido da contração de M a um vértice. Para qualquer permutação fatorizada O de G , a árvore de fraturas correspondente pode ser obtida a partir da árvore de fraturas de G' (associada a ordem O' , obtida de O substituindo os vértices de M por um novo vértice x), substituindo o vértice x por uma floresta, na qual as raízes das árvores são conectadas ao pai de x .*

Lema 3.12 ([CHdM02]). *Sejam G um grafo, O uma de suas permutações fatorizadas e M um nó da árvore de decomposição modular de G . Se M é um nó vizinhança ou se seu pai é um nó serial ou paralelo, então existe um nó que representa M na árvore de fraturas de G correspondente a O .*

Porém, a árvore de fraturas T apresenta algumas imperfeições que são corrigidas nas próximas três fases, das quais a primeira separa os nós da árvore de fraturas que são módulos do grafo daqueles que não são. Estes últimos são chamados de *nós falsos* de T . Para tal, a seguinte propriedade é valiosa:

Lema 3.13 ([CHdM02]). *Sejam O uma permutação fatorizada de um grafo G , T a sua árvore de fraturas e M um dos nós da árvore. Se, para todo par de vértices consecutivos em O , ambos contidos em M , os vértices que distinguem o par pertencerem a M , então M é um módulo de G .*

Sendo assim, associa-se a cada nó da árvore o primeiro e o último entre os vértices que distinguem os pares (consecutivos em O) contidos no nó (caso não existam, tomam-se o primeiro e o último vértice do nó, respectivamente). Para cada um destes pares, o primeiro e último vértices que distinguem seus vértices foram calculados na primeira fase e podem ser armazenados num vetor (disponível em $O(1)$). Para computar tais parâmetros para todos os nós da árvore, faz-se uma busca em profundidade que calcula o primeiro e o último vértices entre os que distinguem os pares de cada filho, considerando, adicionalmente, os vértices que distinguem pares formados pelo último vértice de um filho e o primeiro vértice do próximo filho.

O Algoritmo 8 é um esboço do que é pretendido e a Figura 3.9a é o resultado da aplicação do mesmo sobre a árvore da Figura 3.8. Neste algoritmo, cada nó da árvore contém três campos, *primero*, *ultimo* e *vertice*. Inicialmente apenas *vertice* está preenchido em cada folha, contendo o vértice correspondente. Os outros dois campos são preenchidos pelo algoritmo. Note que o nó pai dos vértices marcados com $c-d$ e $e-f$ está marcado com $a-h$, isto porque a e h são o primeiro e último vértices que distinguem o par (d, e) .

Algoritmo 8 Computa as bordas que distinguem os nós da árvore de fraturas - Terceira Fase.

Entrada: T é a raiz de uma subárvore da árvore de fraturas. Pri e Ult são vetores que contêm o primeiro e último vértices que distinguem cada par de O (o índice destes vetores é o primeiro elemento do par).

Saída: Todo nó x de T contém os campos $x.primeiro$ e $x.ultimo$ preenchidos com o primeiro e último vértice que distinguem os pares do nó.

Se T não tem filhos **então**

$T.primeiro \leftarrow T.vertice, T.ultimo \leftarrow T.vertice.$

Senão

Chame recursivamente para cada filho $\{F_1, \dots, F_k\}$.

$T.primeiro \leftarrow$ o menor entre os valores de $x.primeiro$ para x de F_1 e F_k e de $Pri[y]$ para y sendo o último vértice contido em cada um dos filhos de F_1 a F_{k-1} .

$T.ultimo \leftarrow$ o maior entre os valores de $x.ultimo$ para x de F_1 e F_k e de $Ult[y]$ para y sendo o último vértice contido em cada um dos filhos de F_1 a F_{k-1} .

Fim Se

A próxima fase consiste em simplesmente remover os nós da árvore de fraturas nos quais o primeiro ou último vértices que o distinguem, calculados na fase anterior, não estiverem entre os vértices contidos no nó. Além disso, são removidos também nós que possuem um único filho. Os filhos de um nó removido são adotados pelo seu pai. É fácil notar que este passo pode ser feito ao mesmo tempo que seu predecessor, no qual tais nós foram identificados, e toma tempo linear pois é apenas uma busca em profundidade. Agora, todos os nós da árvore de fraturas são módulos e não há nós falsos ou inúteis. Pode-se ver o resultado sobre o exemplo em uso na Figura 3.9b.

A última etapa consiste em identificar os filhos de um módulo vizinhança da árvore de decomposição modular do grafo que não estão representados na árvore de fraturas, que constituem exatamente o caso não coberto pelo Lema 3.12. Estes foram agregados ao nó correspondente ao módulo vizinhança como é o caso dos pares $\{a, b\}$ e $\{g, h\}$, que, na Figura 3.9b não estão representados como nós da árvore e que devem ser separados de modo que se conclua o algoritmo, obtendo a árvore da Figura 3.7b. Esta tarefa baseia-se em encontrar *vértices gêmeos* – pares de vértices que não são distingüidos por nenhum outro vértice – no grafo quociente dos nós da árvore T . Os gêmeos x e y são ditos *verdadeiros* se $\{x, y\} \in E(G)$ e *falsos*, caso contrário. A propriedade abaixo fundamenta o uso dos gêmeos, tomando como O_M , para um nó M da árvore de fraturas, a seqüência formada por um vértice de cada filho de M , na mesma ordem que O .

Lema 3.14 ([CHdM02]). *Os módulos seriais e paralelos agregados em um nó M de T são exatamente as classes de equivalência não triviais das relações de gêmeos verdadeiros e falsos sobre a ordem O restrita ao grafo quociente associado a M em T . Estas classes*

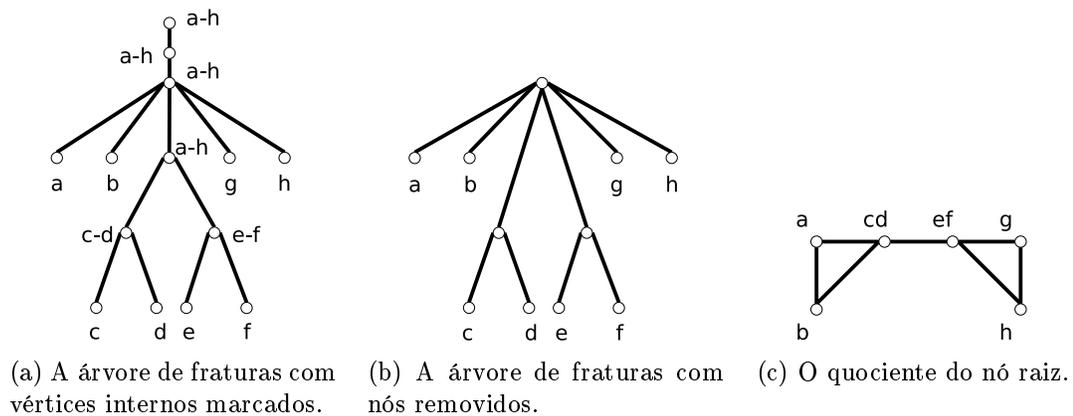


Figura 3.9: Remoção de nós da árvore de fraturas que não são módulos.

correspondem a subsequências maximais de gêmeos verdadeiros e de falsos.

A implementação apenas testa, para cada par (A, B) de filhos consecutivos de todo nó da árvore de fraturas, se o primeiro e último vértices que distinguem o par formado pelo último vértice de A e o primeiro vértice de B estão (se existirem) entre o primeiro vértice de A e o último vértice de B . Note que, sendo A e B módulos, um par formado por qualquer vértice de A e qualquer vértice de B tem o mesmo conjunto de vértices que o distingue. A escolha do par formado pelo último vértice de A e o primeiro de B se justifica pois estes são consecutivos e, assim, o primeiro e último dos vértices que o distingue já foram computados na primeira fase e, como cada par é usado apenas uma vez, a busca é linear. Quando for encontrado um bloco de gêmeos consecutivos, filhos de um mesmo nó M , um novo nó é criado, adotando-os como filhos e tendo como pai o próprio nó M . Desta forma, os módulos fortes foram desagregados.

O resultado da aplicação desta fase sobre a árvore da Figura 3.9b é a própria árvore de decomposição modular vista na Figura 3.7b, exceto pelos rótulos dos nós e o grafo quociente do módulo vizinhança, parâmetros que, por sua vez, podem ser computados, para qualquer grafo, em tempo linear (veja o Algoritmo 9). Note que os pares (a, b) e (g, h) formam os blocos de vértices gêmeos no grafo quociente, visto na Figura 3.9c.

Algoritmo 9 Computa os quocientes e rótulos dos nós da árvore de decomposição modular.

Entrada: G é um grafo, T é sua árvore de decomposição modular e N é um de seus nós.

Saída: Todo nó descendente de N contém rótulo e quociente e v é um vértice descendente de N (cujas arestas não foram percorridas).

Se T tem filhos **então**

Chame recursivamente para cada filho de N .

Seja V o conjunto dos vértices retornados.

Seja v um vértice qualquer de V .

Construa o grafo quociente de N , percorrendo as arestas de G que possuam um extremo entre os vértices em $V \setminus \{v\}$.

Defina o rótulo de N testando se seu quociente é uma clique, conjunto independente ou outro qualquer.

Senão

Faça v igual ao vértice de G correspondente ao nó N de T .

Fim Se

Capítulo 4

Classes com poucos P_4 's

Neste capítulo serão estudadas algumas classes que podem ser definidas em termos de sua decomposição modular, 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.

A Figura 4.1 exibe as classes que serão detalhadas e a relação de continência entre elas. Em seguida, algumas notações são introduzidas e, então, cada classe é definida, suas caracterizações equivalentes são listadas em conjunto com demais propriedades relevantes e algumas de suas aplicações.

4.1 Algumas definições

As restrições que versam sobre estas classes de grafos envolvem a ocorrência de subgrafos induzidos isomorfos ao P_4 e, quando permitidos, sobre o relacionamento que mantêm entre si ou com os demais vértices do grafo. Assim, de certa forma, as classes restringem a densidade local¹ de subgrafos induzidos isomorfos ao P_4 . Dada a importância deste grafo nesse capítulo, 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

¹Número de ocorrências em subgrafos induzidos com número máximo de vértices limitado.

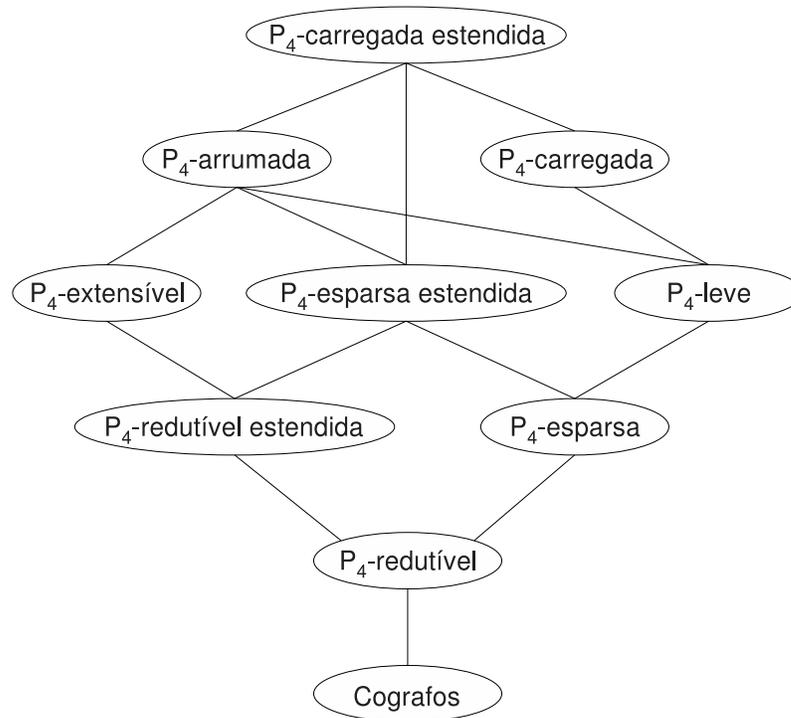


Figura 4.1: Diagrama de Hasse das classes, adaptado de [GRT97].

G . Note que, neste caso, H é isomorfo a um dos grafos Z_1 a Z_7 da Figura 4.2. Caso contrário, H é isomorfo a algum dos grafos dos casos I , U e B da mesma figura, nos quais $|P_4(H)| = 1$. São denotados por $I(G, P)$, $U(G, P)$ e $B(G, P)$, os conjuntos formados por todos os vértices de G que, no papel de v , produzem um subgrafo H isomorfo aos três casos citados, respectivamente.

A seguir, destaca-se um resultado de Giakoumakis e Vanherpe que é usado como base para caracterização, através da decomposição modular, das classes que serão estudadas neste capítulo. Este resultado dá uma nova caracterização a classes definidas por subgrafos primos proibidos, através de restrições aos grafos quocientes dos nós vizinhança da árvore de decomposição modular de seus grafos.

Teorema 4.1 ([GV97]). *Dado um grafo primo Z com mais de três vértices, um grafo G não possui subgrafos induzidos isomorfos a Z se, e somente se, os grafos quocientes de todos os nós rotulados como vizinhança da árvore de decomposição modular de G não possuem subgrafos induzidos isomorfos a Z .*

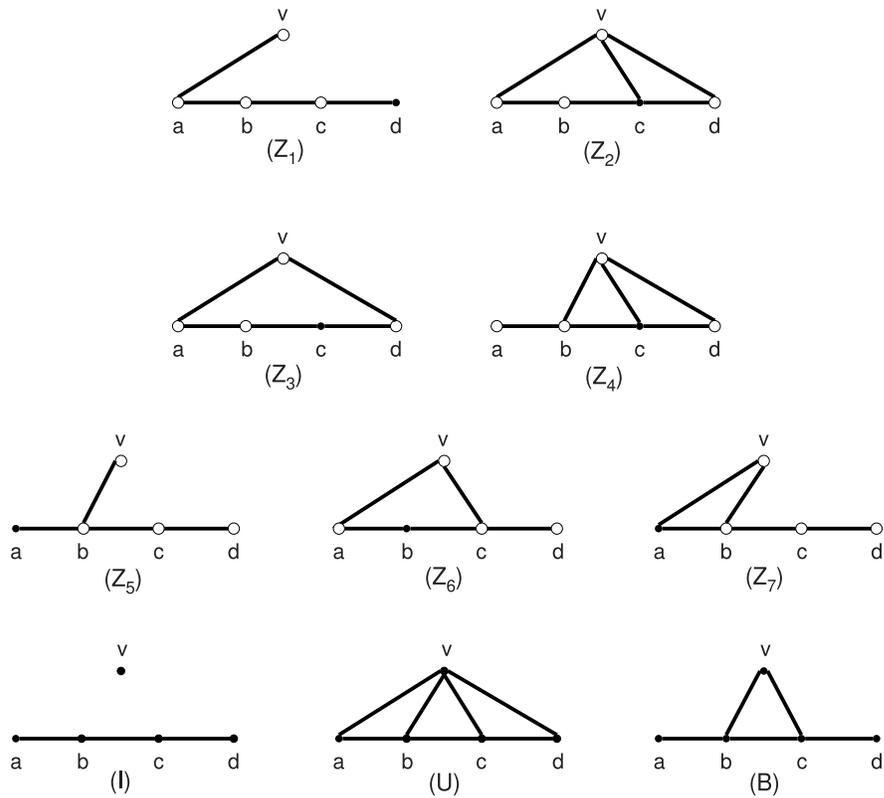


Figura 4.2: Grafos de cinco vértices com um P_4 induzido (por $\{a, b, c, d\}$) e um vértice adicional (v). Os grafos Z_1 a Z_7 possuem ao menos outro P_4 induzido no mesmo grafo, destacado por vértices vazados, de forma que, nestes grafos, v é parceiro do P_4 induzido por $\{a, b, c, d\}$.

4.2 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 [Ler71] 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 A.2.2).

Da própria definição segue que pode-se decompor qualquer cografo até seus vértices isolados, alternadamente tomando seus componentes conexos e os complementando. 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 (operação θ) ou pelos vértices dos componentes conexos do complemento do grafo (operação ι). Um exemplo de um cografo e sua co-árvore estão na Figura 4.3.

A co-árvore e a árvore de decomposição modular de um cografo são isomorfas e seus rótulos são correspondentes, ou seja, os rótulos θ e ι da co-árvore correspondem aos rótulos paralelo e serial da decomposição modular, respectivamente (esta relação pode ser vista na Figura 4.3). Com isso, os cografos são os únicos grafos cuja decomposição modular não contém nós do tipo vizinhança.

Invertendo o sentido das operações usadas para decompor os cografos, pode-se reconstruí-los a partir de seus vértices usando a co-árvore. Considere, desta forma, as operações:

$$G_1 \textcircled{\theta} G_2 = (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$$

$$G_1 \textcircled{\iota} G_2 = (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2) \cup \{\{x, y\} : x \in V(G_1), y \in V(G_2)\})$$

Lema 4.2. *Todo cografo pode ser construído a partir de seus vértices pela aplicação recursiva e finita das operações $\textcircled{\theta}$ e $\textcircled{\iota}$.*

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

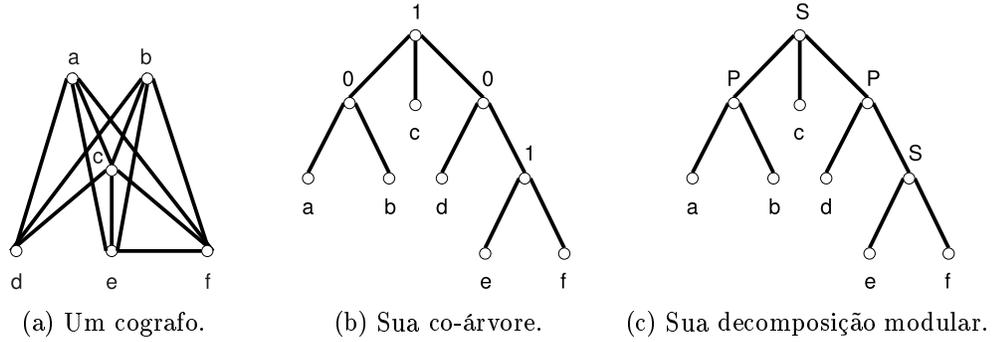


Figura 4.3: Um cografo e suas decomposições.

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 A.2.4) 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 4.3 ([CLB81]). *Todo subgrafo induzido de um cografo é um cografo.*

Apesar dos cografos terem sido introduzidos por Lerchs [Ler71, Ler72], também foram estudados independentemente por vários pesquisadores, como já mencionado, resultando em uma riqueza de caracterizações equivalentes. A seguir, são apresentadas ao leitor algumas definições que serão usadas para relacionar as caracterizações.

Um *grafo Dacey* é um grafo G tal que para toda clique maximal C e todo par de vértices distintos u e v , se $C \subseteq N_G(u) \cup N_G(v)$, então $\{u, v\} \in E(G)$. Um grafo G cujos subgrafos induzidos são todos grafos Dacey é um *HD-grafo* (grafo Dacey hereditário).

Um CPO (V, \leq) é uma *multi-árvore* se, para todo $v, v' \in V$, (1) $v \leq v'$ ou então (2), definindo $S(v) = \{e \in V : v \leq e\}$, para todo $x \in S(v) \setminus S(v')$ e todo $y \in S(v) \cap S(v')$, tem-se $x \leq y$.

Dado um CPO (V, \leq) , associa-se um *grafo de comparabilidade* (veja A.2), cujos vértices são o conjunto V e cujas arestas são os pares $\{u, v\}$ tais que $u \leq v$ ou $v \leq u$.

Teorema 4.4 ([CLB81]). *Dado um grafo G , as seguintes afirmações são equivalentes:*

1. G é um cografo.
2. Todo subgrafo de G não trivial possui dois vértices gêmeos (possuem a mesma vizinhança).
3. Para qualquer subgrafo H de G , toda clique maximal de H tem exatamente um vértice em comum com todo conjunto independente maximal de H (CK-property).

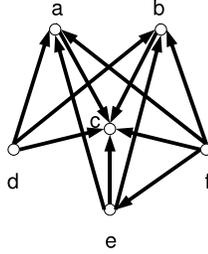


Figura 4.4: Uma representação de uma multi-árvore cujo grafo de comparabilidade associado correspondente ao cografo da Figura 4.3a.

4. G não contém o P_4 como subgrafo induzido ($P_4(G) = \emptyset$).
5. O complemento de qualquer subgrafo conexo de G é desconexo.
6. G é um HD-grafo.
7. Todo subgrafo conexo de G tem diâmetro menor ou igual a dois.
8. G é o grafo de comparabilidade de uma multi-árvore.

Dentre tais equivalências, as implicações $(1) \Rightarrow (2) \Rightarrow (3) \Rightarrow (4)$ e $(5) \Rightarrow (1)$ são dadas em [CLB81]. Além disso, $(4) \Leftrightarrow (6) \Leftrightarrow (7)$ estão em [Sum74], $(4) \Leftrightarrow (5)$ está em [Ler72, Sei74] e $(4) \Leftrightarrow (8)$ está em [Jun78].

Como ilustração da validade das caracterizações dadas sobre o grafo da Figura 4.3a, nota-se que (1) ele é obtido por meio de uniões e complementos de cografos; (2) possui dois vértices gêmeos, a e b ; (3) suas cliques maximais são $\{a, c, d\}$, $\{b, c, d\}$, $\{a, c, e, f\}$ e $\{b, c, e, f\}$, cada uma intercepta em um vértice cada um dos seus conjuntos independentes maximais, $\{a, b\}$, $\{d, e\}$ e $\{d, f\}$; (4) não possui subgrafo induzido isomorfo ao P_4 ; (5) seu complemento é formado por três componentes conexos, $\{a, b\}$, $\{c\}$ e $\{d, e, f\}$; (6) a clique $\{a, d, c\}$ está contida na união da vizinhança de b e e , portanto a aresta $\{b, e\}$ pertence ao grafo; e (7) a distância máxima entre todo par de vértices é dois.

Por fim (8), na Figura 4.4 é apresentada uma representação de um CPO que é uma multi-árvore e cujo grafo de comparabilidade associado é isomorfo ao cografo exemplo. Aceitando a convenção que quando $u \leq v$ para dois elementos u e v do CPO, existe uma aresta orientada (u, v) entre os correspondentes vértices de sua representação, pode-se observar que d e e são incomparáveis e que $S(d) = \{a, b, c, d\}$ e $S(e) = \{a, b, c, e\}$, exigindo que $d \leq a$, $d \leq b$ e $d \leq c$ (uma vez que $S(d) \setminus S(e) = \{d\}$ e $S(d) \cap S(e) = \{a, b, c\}$), afirmações satisfeitas pelas arestas orientadas (d, a) , (d, b) e (d, c) da representação.

Uma vez que grafos de comparabilidade são perfeitos, segue que:

Corolário 4.5. *Cografos são grafos perfeitos.*

Os mesmos autores publicaram um algoritmo linear para construção da co-árvore e reconhecimento da classe [CLB85], anterior aos algoritmos lineares de decomposição modular (para grafos gerais), que também podem ser usados para este fim, sendo, porém, muito mais complexos.

Como principais aplicações desta classe, destacam-se as contribuições abaixo:

Teorema 4.6 ([CLB81]). *Dado um cografo G , determinam-se os seguintes parâmetros em tempo linear: $\chi(G), \omega(G), \theta(G), \alpha(G), s(G)$, além do número de cliques, número de cliques máximas e número de orientações transitivas de G (veja A.2 e A.2.3).*

Golumbic, Kaplan e Shamir [GKS95] mostraram que encontrar um subgrafo de um grafo que contém um conjunto obrigatório de arestas e é um cografo (ou detectar que tal subgrafo não existe) pode ser feito em tempo polinomial. Este resultado abre caminho para algoritmos que aproximem os parâmetros acima quando um grafo não é cografo, obtendo um cografo com conjunto de arestas semelhante ao do grafo original.

Outra aplicação notável dos cografos se refere ao isomorfismo, que pode ser determinado em tempo linear, bastando testar o isomorfismo de suas co-árvores. Além disso, esta classe continua recebendo atenção de pesquisadores e, assim, muitos problemas tem sido resolvidos eficientemente para estes grafos, como, por exemplo, emparelhamento máximo em tempo linear [YY93], decomposição no menor número de caminhos disjuntos nos vértices [LOP06, NOZ03], particionamento em cliques e conjuntos independentes [DEdW05], determinação do comportamento da aplicação repetida do operador clique sobre os cografos e grafos seriais [LdMM⁺04] e ainda novos algoritmos de reconhecimento paralelo [NP04a] e dinâmico [SS04].

4.3 P_4 -reduzível

Jamison e Olariu introduziram a classe P_4 -reduzível (do inglês, P_4 -reducible), que pode ser vista como uma generalização dos cografos [JO89b]. Um grafo G é P_4 -reduzível se, e somente se, cada vértice de G pertence a no máximo um P_4 induzido em G . Mais formalmente, para qualquer $v \in V(G)$, vale $|\{X \in P_4(G) : v \in X\}| \leq 1$, ou então, não existe $\{X, Y\} \subseteq P_4(G)$ tal que $X \cap Y \neq \emptyset$.

Giakoumakis e Vanherpe caracterizaram a classe pela decomposição modular de seus grafos:

Teorema 4.7 ([GV97]). *Um grafo G pertence à classe P_4 -reduzível se, e somente se, para todo nó vizinhança de sua árvore de decomposição modular, seu grafo quociente é isomorfo*

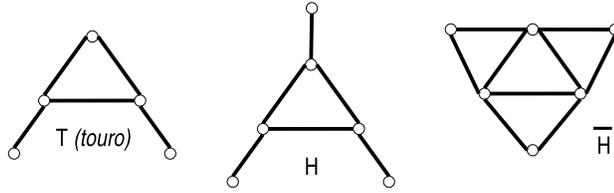


Figura 4.5: Dois subgrafos proibidos para a classe P_4 -reduzível (H e \overline{H}) e o touro.

ao P_4 ou ao touro (grafo T da Figura 4.5) e seus fatores são grafos triviais, exceto pelo fator correspondente ao vértice de grau dois do grafo quociente, quando este for isomorfo ao touro.

Aplicando a caracterização acima aos resultados do Teorema 4.1, os autores também propuseram uma caracterização por subgrafos proibidos.

Teorema 4.8 ([GV97]). *Um grafo G pertence a classe P_4 -reduzível se, e somente se, não possuir nenhum subgrafo induzido isomorfo a qualquer dos grafos $\{Z_1, Z_2, \dots, Z_7\}$ da Figura 4.2 e aos grafos H e \overline{H} da Figura 4.5.*

Claramente, todo cografo é P_4 -reduzível, mas não é possível representar todo grafo P_4 -reduzível por uma co-árvore. Anteriormente às caracterizações citadas acima, Jamison e Olariu propuseram outra árvore de decomposição para esta classe, baseada na seguinte caracterização.

Teorema 4.9 ([JO89b]). *Um grafo G é P_4 -reduzível se, e somente se, para todo subgrafo induzido H de G , uma das seguintes afirmações vale:*

- H é desconexo;
- \overline{H} é desconexo;
- Existe um único P_4 induzido em H com conjunto P de vértices, tal que todo vértice em $V(H) \setminus P$ não é adjacente aos extremos de $H[P]$ e é adjacente aos demais vértices de $H[P]$ (vértices internos).

Note que um grafo que se enquadre no terceiro caso do Teorema 4.9 terá como raiz de sua decomposição modular um nó vizinhança.

Assim sendo, os autores definiram uma nova operação que reflete a terceira condição do Teorema 4.9. Dados dois grafos G_1 e G_2 tais que $V(G_1) = \{a, d\}$, $E(G_1) = \emptyset$ e dois vértices adjacentes b e c de $V(G_2)$, vizinhos a todos os demais vértices de G_2 . Então:

$$G_1 \textcircled{2} G_2 = (V(G_1) \cup V(G_2), E(G_2) \cup \{\{a, b\}, \{c, d\}\})$$

Esta nova operação permite outra caracterização para os grafos da classe P_4 -reduzível:

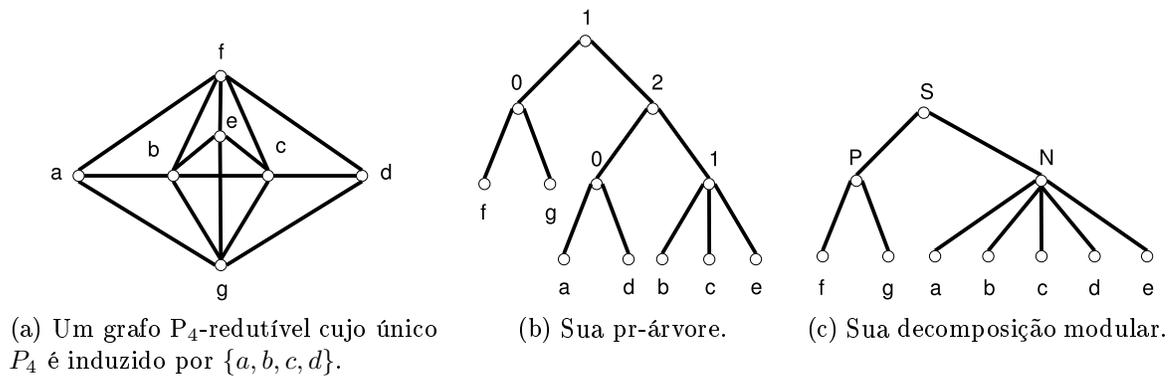


Figura 4.6: Um grafo P_4 -reduzível e suas decomposições.

Lema 4.10 ([JO89b]). *Todo grafo P_4 -reduzível pode ser construído a partir de seus vértices pela aplicação recursiva e finita das operações ①, ② e ③.*

Como consequência, pode-se decompor o grafo em uma árvore (única) rotulada de acordo com a operação aplicada a cada vértice, a *pr-árvore*. Um exemplo é dado na Figura 4.6, onde também está a árvore de decomposição modular do mesmo grafo.

Pelo fato da operação ② não ser comutativa, é necessário reconhecer entre os dois nós filhos da pr-árvore, qual induz o primeiro e o segundo operandos. Para tal, pode-se tanto marcar os nós como identificá-los pelas regras da operação. Também observa-se que pode-se escolher quaisquer dois vértices do segundo operando que satisfaçam as condições necessárias para reconstruir o grafo. Na pr-árvore da Figura 4.6, o filho esquerdo do nó rotulado com ② induz seu primeiro operando e pode-se escolher quaisquer dois entre entre $\{b, c, e\}$ para o papel dos vértices b e c da definição da operação, resultando em grafos isomorfos.

Embora não sejam isomorfas como no caso dos cografos, a decomposição modular e a pr-árvore podem ser obtidas uma da outra explorando a relação entre os nós vizinhança da primeira e os correspondentes a aplicação da operação ② da segunda.

O reconhecimento desta classe tem complexidade linear no tamanho do grafo, pelo algoritmo proposto pelos mesmos autores que a definiram [JO95b]. Este estende o algoritmo de reconhecimento de cografos de [CLB85] e também é capaz de construir a pr-árvore dos grafos da classe. Outro algoritmo linear de reconhecimento, que faz uso da decomposição modular, foi apresentado por Giakoumakis e Vanherpe [GV97].

Uma característica interessante desta classe é sua relação estreita com os cografos. Ao tomar cada P_4 induzido em um grafo P_4 -reduzível e remover um de seus extremos do grafo, obtem-se o *cografo canônico*. Apesar da remoção ser arbitrária, Jamison e Olariu notaram que os grafos resultantes são isomorfos:

Teorema 4.11 ([JO95b]). *Dado um grafo P_4 -reduzível, seu cografo canônico é único, exceto por isomorfismos.*

Além disso, o cografo canônico é o maior cografo induzido em um grafo P_4 -reduzível e preserva muitos parâmetros do grafo (veja abaixo). Note que, encontrar o maior cografo induzido em um grafo qualquer é um problema intratável [CLB85].

Teorema 4.12 ([JO95b]). *Dado um grafo P_4 -reduzível G e seu cografo canônico C :*

$$\omega(G) = \omega(C) \quad \chi(G) = \chi(C) \quad \alpha(G) = \omega(\overline{C}) \quad \theta(G) = \chi(\overline{C})$$

Aliando este resultado ao Teorema 4.6, os parâmetros acima citados podem ser calculados em tempo linear para os grafos P_4 -reduzível. Adicionalmente, Giakoumakis e Vanherpe [GV97] propuseram versões lineares para os mesmos problemas, considerando pesos nos vértices e o fato de que esta classe é uma subclasse dos grafos P_4 -esparso (veja Seção 4.4) e, assim, os algoritmos aplicáveis àquela também funcionam nesta. Também foi proposto um algoritmo para calcular o número de árvores geradoras de um grafo da classe [NP04b].

4.3.1 P_4 -reduzível estendida

Esta classe foi introduzida por Giakoumakis e Vanherpe [GV97] como uma pequena extensão da classe P_4 -reduzível, permitindo a presença de ciclos de cinco vértices induzidos no grafo, como um adendo à caracterização dada pelo Teorema 4.7. Portanto, um grafo pertence a classe *P_4 -reduzível estendida* (do inglês, **extended P_4 -reducible**) se, e somente se, para todo nó vizinhança de sua árvore de decomposição modular, seu grafo quociente é isomorfo ao P_4 , ao touro (veja Figura 4.5) ou ao C_5 e seus fatores são grafos triviais, à exceção do fator correspondente ao vértice de grau dois do grafo quociente, quando isomorfo ao touro.

No mesmo trabalho, os autores adaptaram para esta nova classe o algoritmo de reconhecimento linear e alguns algoritmos de otimização aplicáveis a classe P_4 -esparso estendida (veja Seção 4.4.4), uma superclasse própria desta.

4.4 P_4 -esparso

Esta classe de grafos foi introduzida por Hoàng em sua dissertação de mestrado [Hoà83], na qual forneceu algumas caracterizações equivalentes e mostrou que estes grafos são perfeitos (veja A.2.4) e perfeitamente ordenáveis (veja A.2.4).

Um grafo G pertence à classe *P_4 -esparso* (do inglês **P_4 -sparse**) se, todo subgrafo H induzido em G , com cinco vértices, for tal que $|P_4(H)| \leq 1$, ou, de forma equivalente, se

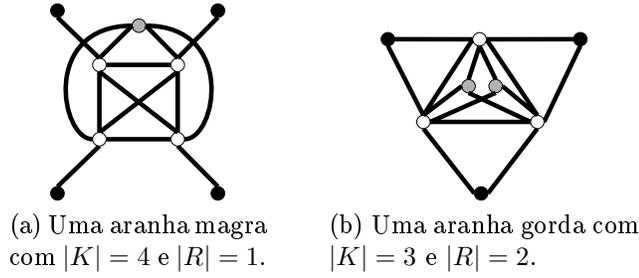


Figura 4.7: Exemplos de aranhas gorda e magra.

para todo $P \in P_4(G)$ tem-se que $R(G, P) = \emptyset$. Abaixo, examina-se uma família de grafos especiais para esta classe e então introduz-se algumas caracterizações equivalentes.

4.4.1 Aranhas

Um grafo G é uma *aranha* 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 $f : K \rightarrow S$ tal que:
 - $\forall k \in K, N_G(k) \cap S = \{f(k)\}$, caso em que G é chamado de *aranha magra*; ou
 - $\forall k \in K, N_G(k) \cap S = S \setminus \{f(k)\}$, caso em que G é chamado de *aranha gorda*.
- $\forall r \in R, (K \subseteq N_G(r))$ 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*. Esta partição está bem definida, uma vez que é única. A Figura 4.7 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 .

Proposição 4.13 ([JO92b]). *Seja G uma aranha. A partição canônica dos vértices de G em (K, S, R) é única.*

Além disso, a disposição de subgrafos isomorfos ao P_4 em uma aranha obedece algumas regras:

Proposição 4.14 ([JO92b]). *Seja G uma aranha e (K, S, R) a partição canônica de seus vértices. Podemos afirmar que:*

- *Todo P_4 em G está inteiramente contido em R ou em $K \cup S$.*

- *Todo P_4 induzido em $G[K \cup S]$ possui seus extremos em S e vértices internos em K .*

A decomposição modular das aranhas também apresenta características notáveis:

Proposição 4.15. *Seja H uma aranha de partição canônica (K, S, R) com R não vazio, então R é um módulo forte maximal próprio de H .*

4.4.2 Caracterizações

Giakoumakis e Vanherpe também propuseram para esta classe uma caracterização relacionada a decomposição modular dos grafos da classe:

Teorema 4.16 ([GV97]). *Um grafo G pertence a classe P_4 -esparso 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 aquele correspondente ao conjunto R , da partição canônica (K, S, R) do grafo quociente.*

A classe possui, também, uma caracterização por subgrafos proibidos semelhante à caracterização dos grafos P_4 -reduzível.

Lema 4.17 ([JO92b]). *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 4.2.*

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

Teorema 4.18 ([JO92b]). *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.*

No último caso do Teorema 4.18, H e \overline{H} são conexos, de forma que o nó raiz da decomposição modular de H é rotulado como vizinhança. Note que este teorema caracteriza construtivamente a classe de maneira muito similar ao Teorema 4.9 para a classe P_4 -reduzível.

Novamente, Jamison e Olariu buscaram uma árvore de decomposição para esta classe de grafos e, para tal, definiram uma nova operação. Sejam dois grafos G_1 e G_2 tais que $E(G_1) = \emptyset$ e $V(G_2)$ pode ser particionado em três conjuntos: $\{v\}, K, R$ de forma que:

- $|K| = |V(G_1)| + 1 \geq 2$;
- K é uma clique (em G_2);
- Todo vértice de R é adjacente a todo vértice de K e não é adjacente a v .
- Existe um vértice v' em K tal que $N_{G_2}(v) = \{v'\}$ ou $N_{G_2}(v) = K \setminus \{v'\}$.

Então, escolha qualquer bijeção $f : V(G_1) \rightarrow K \setminus \{v'\}$, para a qual define-se $G = G_1 \textcircled{3} G_2$, sendo $V(G) = V(G_1) \cup V(G_2)$ e $E(G) = E(G_2) \cup E'$, com E' dado por:

$$E' = \begin{cases} \{\{x, f(x)\} : x \in V(G_1)\} & \text{quando } N_{G_2}(v) = \{v'\} \\ \{\{x, z\} : x \in V(G_1), z \in K \setminus \{f(x)\}\} & \text{quando } N_{G_2}(v) = K \setminus \{v'\} \end{cases}$$

Com uso desta nova operação, pode-se decompor uma aranha G de partição canônica (K, S, R) em dois grafos $G_1 = (S \setminus \{v\}, \emptyset)$ e $G_2 = (\{v\} \cup K \cup R, E(G) \setminus \{\{x, y\} : x \in S \setminus \{v\}, y \in K\})$, sendo v um vértice qualquer de S . A Figura 4.8 exibe dois exemplos da decomposição. Em ambos, o grafo é decomposto em G_1 e G_2 , grafos induzidos pelos vértices $\{b, c\}$ e $\{a, d, e, f, g\}$, respectivamente. Nos grafos sendo decompostos, os vértices vazados pertencem à K , os vértices negros, à S e os acinzentados, à R .

Lema 4.19 ([JO92b]). *O grafo G é uma aranha se, e somente se, para dois subgrafos induzidos G_1 e G_2 de G , tem-se $G = G_1 \textcircled{3} G_2$. Além disso, sejam os subgrafos G'_1 e G'_2 , induzidos em G , para os quais $G = G'_1 \textcircled{3} G'_2$, então G'_1 é isomorfo a G_1 e G'_2 é isomorfo a G_2 .*

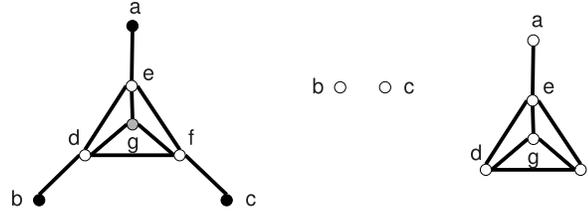
Ao conjugar este e o Teorema 4.18, chega-se a outra caracterização para os grafos da classe P_4 -esparso:

Teorema 4.20 ([JO92b]). *Um grafo G é P_4 -esparso se, e somente se, pode ser obtido a partir de seus vértices aplicando recursivamente as operações $\textcircled{0}$, $\textcircled{1}$ e $\textcircled{3}$.*

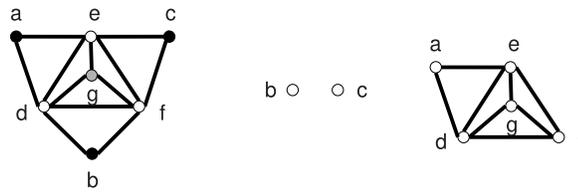
4.4.3 Decomposição e aplicações

Dada a árvore de decomposição modular de um grafo da classe P_4 -esparso, algoritmos apresentados por Giakoumakis e Vanherpe [GV97] determinam, em tempo linear no número de vértices do grafo, o tamanho da maior clique e do maior conjunto independente do grafo, considerando pesos nos seus vértices.

Jamison e Olariu também propuseram uma árvore de decomposição para os grafos P_4 -esparso, que é construída de maneira análoga a pr-árvore (apenas as operações usadas mudam para $\textcircled{0}$, $\textcircled{1}$ e $\textcircled{3}$, conforme o Teorema 4.20), e foi denominada *ps-árvore*. Na Figura 4.9, vê-se a ps-árvore e a decomposição modular do grafo da Figura 4.8b. Os



(a) Uma aranha magra com $|R| = 1$.



(b) Uma aranha gorda com $|R| = 1$.

Figura 4.8: Decomposição de duas aranhas pela operação ③.

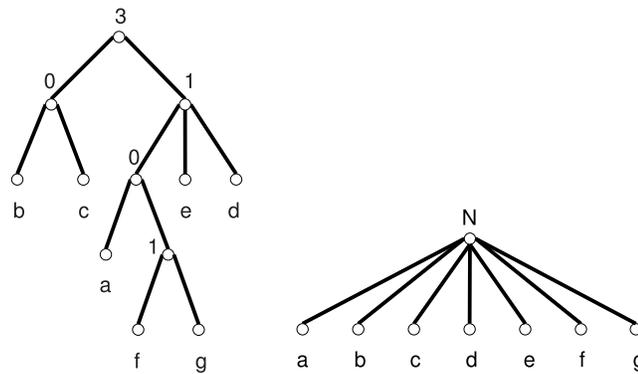


Figura 4.9: A ps-árvore e a decomposição modular do grafo da Figura 4.8b.

autores também apresentaram um algoritmo linear para o reconhecimento da classe e construção da ps-árvore [JO92a].

No mesmo trabalho, os autores estenderam o uso do cografo canônico da classe P_4 -reduzível para os grafos P_4 -esparso, obtendo algoritmos lineares no tamanho do grafo para os mesmos problemas resolvidos naquela classe (veja Teorema 4.21). Posteriormente [JO95a], mostraram que estes parâmetros e $\Phi(G)$ (veja A.2.4) de um grafo G da classe P_4 -esparso podem ser resolvidos em tempo linear no número de seus vértices, uma vez calculada a co-árvore do seu cografo canônico e uma partição especial de seus vértices, também usada no algoritmo de reconhecimento [JO92a].

Teorema 4.21 ([JO92b]). *Dado um grafo P_4 -esparso G e seu cografo canônico C :*

$$\omega(G) = \omega(C) \quad \chi(G) = \chi(C) \quad \alpha(G) = \omega(\overline{C}) \quad \theta(G) = \chi(\overline{C})$$

Entre outras aplicações para a classe também destacam-se a contagem de separadores de tamanho mínimo [NP06] e o seu reconhecimento por algoritmo paralelo [LO98].

4.4.4 P_4 -esparso estendida

Esta é outra extensão proposta por Giakoumakis e Vanherpe [GV97], a semelhança da classe P_4 -reduzível estendida, devida a caracterização dada pelo Teorema 4.16. Um grafo pertence a classe *P_4 -esparso estendida* (do inglês **extended P_4 -sparse**) se, e somente se, para todo nó vizinhança de sua árvore de decomposição modular, seu grafo quociente é isomorfo a uma aranha ou ao C_5 e seus fatores são grafos triviais, a exceção do fator correspondente ao conjunto R (se não vazio), da partição canônica do grafo quociente, quando isomorfo a uma aranha.

O algoritmo de reconhecimento linear proposto pelos autores para a classe P_4 -esparso foi adaptado a esta extensão, bem como dois algoritmos de otimização, também aplicáveis a classe P_4 -reduzível estendida: encontrar a maior clique e o maior conjunto independente, com pesos nos vértices.

4.5 P_4 -extensível

Esta é outra classe introduzida por Jamison e Olariu [JO91], cuja interseção com a classe P_4 -esparso forma a classe P_4 -reduzível. Para defini-la, introduz-se a noção de extensão própria de um grafo G . Para um subconjunto $X \subset V(G)$, denote por $S(X)$ a união de $P \setminus X$ para todo $P \in P_4(G)$ tal que $P \cap X \neq \emptyset$. Isto é, a união dos vértices de todos os P_4 induzidos em G que interceptem X , retirados os vértices de X . Diz-se que um conjunto X possui uma *extensão própria* se $|S(X)| \leq 1$.

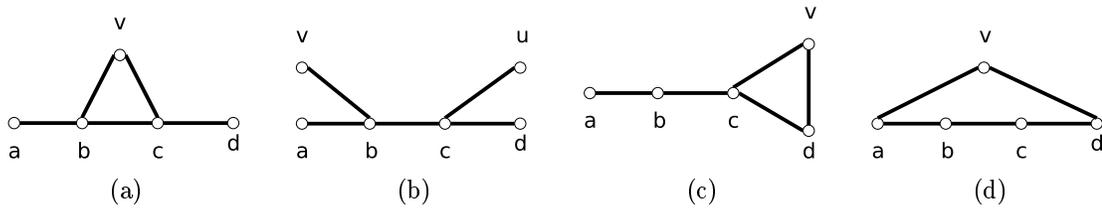


Figura 4.10: Exemplos de extensões próprias. Em todos os grafos, $P = \{a, b, c, d\}$ induz um P_4 . Em (a), $S(\{b, d\}) = \{a, c\}$, $S(\{v\}) = \emptyset$ e $S(P) = \emptyset$; em (b), $S(P) = \{u, v\}$; em (c) e em (d), $S(P) = \{v\}$. Logo P possui uma extensão própria em (a), (c) e (d).

Um grafo pertence à classe P_4 -extensível (do inglês P_4 -extendible) se todo conjunto P de seus vértices que induz um subgrafo isomorfo ao P_4 possui uma extensão própria. Estes conceitos estão exemplificados na Figura 4.10.

Novamente, Giakoumakis em um trabalho conjunto com Roussel e Thuillier, forneceram uma caracterização para a classe baseada na decomposição modular de seus grafos:

Teorema 4.22 ([GRT97]). *Um grafo G pertence a classe P_4 -extensível se, e somente se, o grafo quociente de todo nó vizinhança de sua árvore de decomposição modular é isomorfo ao P_4 , P_5 , $\overline{P_5}$, C_5 ou touro. Os grafos fatores devem ser todos grafos triviais, a exceção do correspondente ao vértice de grau dois do touro e a um único fator que pode conter dois vértices se o quociente for isomorfo ao touro ou ao P_4 .*

Jamison e Olariu apresentaram uma caracterização construtiva que, para ser apresentada, requer uma nova definição: se um subconjunto $Y \subset V(G)$ for tal que $Y = P \cup S(P)$, para algum $P \in P_4(G)$ que possui uma extensão própria, então ele é chamado de *conjunto extensível* e é *separável* se nenhum de seus vértices fizer parte do extremo de um P_4 induzido em $G[Y]$ e do interior de outro P_4 induzido no mesmo. Na Figura 4.10, apenas o grafo em (c) possui um conjunto extensível separável, em (d), a é interno ao P_4 induzido por $\{v, a, b, c\}$ e é extremo do induzido por P .

Agora, apresenta-se a caracterização proposta.

Teorema 4.23 ([JO91]). *Um grafo G pertence a classe P_4 -extensível se, e somente se, todo subgrafo H induzido por G satisfizer uma das seguintes restrições:*

- H é desconexo;
- \overline{H} é desconexo;
- $V(H)$ é um conjunto extensível do próprio H .

G_1	G_2	$G_1 \textcircled{4} G_2$	G_1	G_2	$G_1 \textcircled{4} G_2$

Tabela 4.1: Definição da operação $\textcircled{4}$.

- Existe um único conjunto extensível separável $Y \in V(H)$ tal que todo vértice de $V(H) \setminus Y$ é, ao mesmo tempo, adjacente aos pontos interiores e não adjacente aos extremos de todo $P \in P_4(H[Y])$.

Os dois últimos casos do Teorema 4.23 causam o aparecimento dos nós vizinhança na árvore de decomposição modular dos grafos desta classe. Em ambos os casos, o grafo quociente dos nós vizinhança produzidos são limitados a um número pequeno de possibilidades.

Com a definição das operações inversas aos itens da caracterização acima, somos levados a uma decomposição dos grafos da classe em uma árvore conhecida por *px-árvore*. As operações que compõem as duas primeiras alternativas são $\textcircled{0}$ e $\textcircled{1}$, vide Seção 4.2. A operação $\textcircled{4}$ é definida na Tabela 4.1 e corresponde a terceira alternativa da decomposição. Já a última operação é definida de forma que $G_1 \textcircled{5} G_2 = (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2) \cup X)$, sendo $V(G_1)$ um conjunto extensível separável de G_1 e $X = \{\{x, y\} : x \in V(G_1), y \in V(G_2)\}$ para todo vértice interno x de um $P_4(G_1)$.

A Figura 4.11 contém a decomposição de um grafo da classe em sua *px-árvore* e em sua árvore de decomposição modular. Note que os operandos para a operação $\textcircled{5}$ são tais que $V(G_1) = \{a, b, c, d, f\}$ (formando um conjunto extensível separável) e $V(G_2) = \{e\}$.

Hochstättler e Tinhofer [HT95] mostraram ainda outra caracterização para a classe, baseada em *p*-componentes. Um grafo G é dito *p-conexo* se o grafo $G' = (V(G), E')$ é

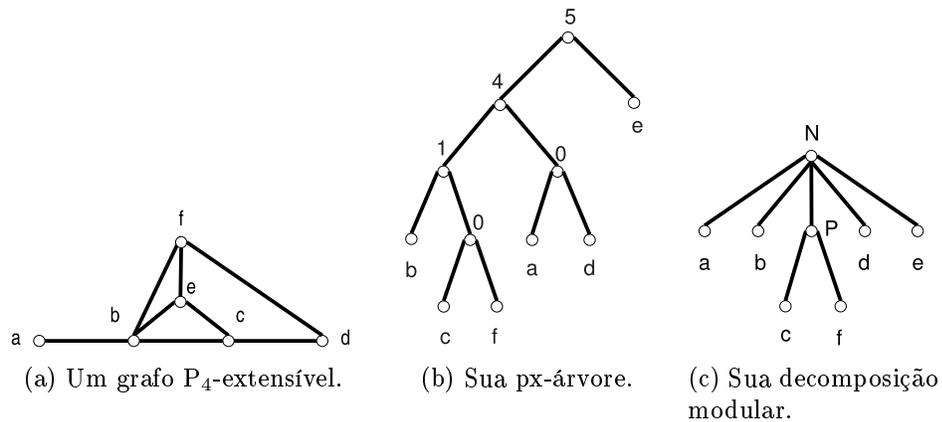


Figura 4.11: Exemplo de px-árvore.

conexo para $E' = \{e \in E(G[P]) : P \in P_4(G)\}$. Um subgrafo induzido maximal e p -conexo de um grafo é chamado p -componente.

Lema 4.24 ([HT95]). *Um grafo G pertence a classe P_4 -extensivo se, e somente se, todas as suas p -componentes tem no máximo cinco vértices, ou seja, são isomorfas a um dos grafos produzidos pela operação ④ na terceira e sexta colunas da Tabela 4.1.*

A maior aplicação desta classe é o teste de isomorfismo, que pode ser resolvido em tempo polinomial pela comparação das px-árvores, uma vez que podemos construí-las em tempo linear pelo algoritmo de Hochstättler e Schindler [HS95].

4.6 P_4 -leve

Esta classe foi introduzida por Jamison e Olariu ([JO89a]) (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 4.5 (que possuem seis vértices e três subgrafos isomorfos ao P_4). Um exemplo de grafo da classe é dado na Figura 4.12a, embora tal afirmação fique mais clara apenas após a caracterização dada abaixo.

Embora para esta classe os autores não tenham definido uma árvore de decomposição, Giakoumakis, Roussel e Thuillier [GRT97] apresentaram um algoritmo de reconhecimento linear baseado na árvore de decomposição modular, do qual obtém-se outra caracterização para a classe:

Teorema 4.25. *Um grafo G pertence a 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 ,*

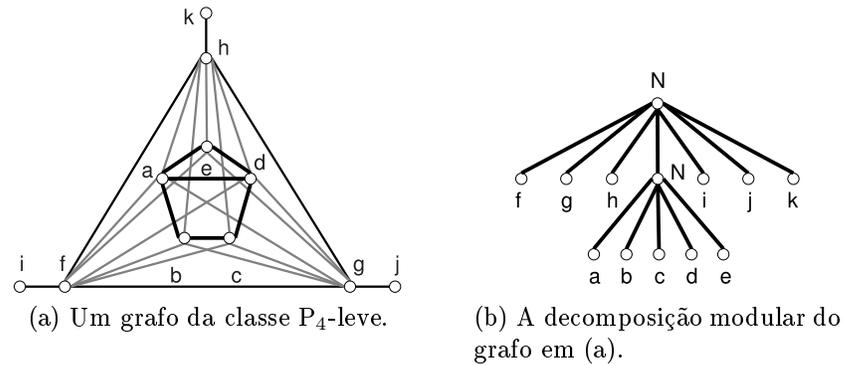


Figura 4.12: Exemplo de grafo da classe P_4 -leve e sua decomposição modular.

ao $\overline{P_5}$ ou a uma aranha (vide Seção 4.4.1) 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 pode possuir dois vértices.

O grafo da Figura 4.12a é uma aranha magra cuja parte R de sua partição canônica (K, S, R) é $\{a, b, c, d, e\}$. O grafo induzido por R é um fator da decomposição modular da aranha (dada na Figura 4.12b) e é isomorfo ao $\overline{P_5}$, que é primo.

É fácil notar que ela contém todos os grafos P_4 -esparso. Ainda, mostra-se [JO89a] que os grafos da classe são **brittle** (veja A.2.4) e, portanto, perfeitamente ordenáveis.

4.7 P_4 -carregada

Giakoumakis introduziu a classe P_4 -carregada (do inglês P_4 -laden [Gia96]) como contendo todo grafo cujos subgrafos induzidos de até seis vértices contém no máximo dois P_4 induzidos distintos ou são grafos **split** (veja A.2.4).

Uma vez que os grafos H e \overline{H} da Figura 4.5 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 clique e 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, $S(G) = \{s \in S : \exists k \in K \wedge \{s, k\} \notin E(G)\}$, e por $K(G)$ a vizinhança de $S(G)$, que está contida em K (isto é, $K(G) = N_G(S(G))$).

Teorema 4.26 ([Gia96]). *Um grafo pertence a classe P_4 -carregada se, e somente se, o grafo quociente de cada nó vizinhança de sua árvore de decomposição modular:*

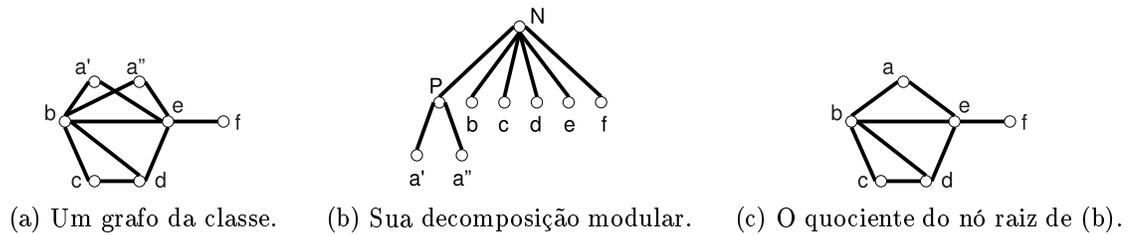


Figura 4.13: Exemplo de grafo da classe P_4 -carregada.

- For isomorfo ao P_5 ou ao $\overline{P_5}$ e seus fatores forem grafos triviais; ou
- For isomorfo a uma aranha e seus fatores forem todos grafos de um único vértice, 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
- For isomorfo 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.

O autor apresentou um algoritmo linear de reconhecimento da classe que valida os critérios definidos no Teorema 4.26. Outra característica interessante da classe é que seus grafos são *brittle*, a exemplo de sua subclasse P_4 -leve.

Na Figura 4.13 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 $S(G)$.

4.7.1 P_4 -carregada estendida

Outra classe definida por Giakoumakis em [Gia96]. É uma simples extensão da classe P_4 -carregada para admitir C_5 induzidos, mas que perde sua característica de ser perfeitamente ordenável. Para defini-la, nota-se que um grafo é *split* se, e somente se, não possui C_5 , C_4 e $\overline{C_4}$ como subgrafos induzidos. Ao relaxar esta restrição, define-se um grafo *pseudo-split* como sendo livre de C_4 e $\overline{C_4}$ como subgrafos induzidos.

Um grafo pertence a classe *P_4 -carregada estendida* (do inglês, *extended P_4 -laden*) se todo seu subgrafo induzido de até seis vértices contém no máximo dois P_4 induzidos ou é um grafo *pseudo-split*.

Uma vez que esta classe é a união das classes P_4 -carregada e P_4 -arrumada (definida na Seção 4.8), os grafos das figuras 4.12a e 4.14 são exemplos de grafos da classe. Note que nenhum dos dois exemplos apresentados pertence a ambas as classes.

Basta adicionar a possibilidade de isomorfismo com o C_5 à primeira opção do Teorema 4.26 para uma caracterização da classe.

4.8 P_4 -arrumada

A classe P_4 -arrumada (do inglês, *P_4 -tidy*) foi definida por I. Rusu em comunicados não publicados e, posteriormente, foi apresentada num trabalho de V. Giakoumakis, F. Roussel e H. Thuiller [GRT97], no qual a caracterizaram por conjuntos permitidos de grafos quocientes para os módulos vizinhança e disso derivam um algoritmo de reconhecimento em tempo linear.

Um grafo G pertence à classe P_4 -arrumada se $\forall P \in P_4(G)$, $|R(G, P)| \leq 1$. Vendo de outra forma, um grafo G pertence a classe P_4 -arrumada se, e somente se, não existem dois subconjuntos distintos, X e Y , de $V(G)$, com $X \cap Y \in P_4(G)$ e tais que $G[X]$ e $G[Y]$ são isomorfos a qualquer dos grafos $\{Z_1, \dots, Z_7\}$ da Figura 4.2 (note que X e Y não são necessariamente isomorfos). Abaixo está a caracterização proposta pelos autores:

Teorema 4.27. *Um grafo G pertence a classe P_4 -arrumada se, e somente se, todo grafo quociente de um nó rotulado como vizinhança em sua árvore de decomposição modular for isomorfo a um C_5 , P_5 ou $\overline{P_5}$ cujos fatores são grafos triviais, ou for isomorfo a uma aranha, caso em que exceto pelo fator correspondente ao conjunto R de sua partição canônica, se não vazio, e um outro fator, que pode conter dois vértices, os demais devem ser todos grafos triviais.*

Um exemplo de grafo da classe está na Figura 4.14. Sua árvore de decomposição modular é isomorfa a da Figura 4.12b, contudo o grafo quociente do fator $\{a, b, c, d, e\}$ é um C_5 . De fato, a única diferença entre os o grafo apresentado e o da Figura 4.12a é a ausência da aresta $\{a, d\}$. Note que este grafo não pertence a nenhuma das classes vistas anteriormente, exceto a classe P_4 -carregada estendida, e que o único grafo apresentado anteriormente neste Capítulo que não pertence a classe P_4 -arrumada é o exemplo da Figura 4.13a.

Além disso, no mesmo trabalho, apresentam algoritmos lineares para vários problemas de otimização aplicados a grafos da classe, como clique máxima, número cromático entre outros. Uma vez que esta classe engloba as classes P_4 -esparso estendida, P_4 -extensível e P_4 -leve, e, assim, todas as classes estudadas neste Capítulo, à exceção de P_4 -carregada e P_4 -carregada estendida, tais algoritmos são diretamente aplicáveis a todas estas. Levando em consideração, ainda, que o algoritmo de reconhecimento é capaz de reconhecer todas as suas subclasses, a mesma merece um estudo mais detalhado, que por não se adequar ao intuito deste Capítulo foi deixado para o Capítulo 5.

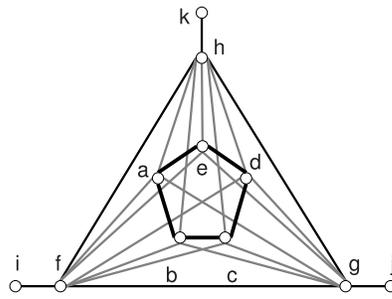


Figura 4.14: Exemplo de grafo da classe P_4 -arrumada.

4.9 Conclusões

Todas as classes aqui estudadas apresentam características comuns, entre as quais destaca-se que são hereditárias, isto é, subgrafos induzidos de um grafo pertencente a qualquer destas classes também pertence a mesma classe, e que são fechadas pelas operações de complemento e pelas operações ① e ②, definidas na Seção 4.2.

A importância das mesmas reside em suas aplicações, das quais foram listadas as que pareceram mais relevantes. Além das quais, sabe-se que modelos em grafos para muitos problemas (tais como agendamento de cursos e agrupamento semântico de itens [JO91]) recaem em uma destas classes.

Capítulo 5

A classe P_4 -arrumada

Esta classe foi introduzida na Seção 4.8 e neste capítulo será detalhada com o objetivo de provar algumas de suas propriedades, incluindo sua caracterização em termos da árvore de decomposição modular. Também é apresentado o algoritmo de reconhecimento proposto em [GRT97], bem como algumas aplicações da classe em problemas de otimização.

Para começar, recorda-se a definição da classe: um grafo G pertence a classe P_4 -arrumada se, e somente se, para todo $P \in P_4(G)$, $R(G, P) \leq 1$. Isto é, cada subgrafo induzido isomorfo ao P_4 tem no máximo um parceiro. Em seguida serão estudadas algumas de suas propriedades (algumas definições usadas são dadas na Seção 4.1).

Uma *quase-aranha* é um grafo obtido de uma aranha de partição canônica (K, S, R) , opcionalmente substituindo um dos vértices contidos em $K \cup S$ por um par de novos vértices e fazendo ambos adjacentes aos mesmos vértices aos quais o removido era. Os novos vértices podem ser ou não adjacentes. A partição canônica da quase-aranha é obtida da partição canônica da aranha, removendo o vértice substituído de seu conjunto e adicionando, ao mesmo, o par de novos vértices, caso a substituição tenha ocorrido.

5.1 Propriedades

Um grafo da classe P_4 -arrumada apresenta várias propriedades de interesse. Abaixo, algumas delas são descritas:

Proposição 5.1. *A classe P_4 -arrumada é fechada por complemento.*

Demonstração. Seja G um grafo pertencente a classe P_4 -arrumada e suponha que \overline{G} não pertença a classe. Portanto existe algum $A \in P_4(\overline{G})$, tal que $|R(\overline{G}, A)| \geq 2$. Sejam $\{b, c\} \in R(\overline{G}, A)$. Como b e c são parceiros de A , existem $\{B, C\} \subseteq P_4(\overline{G})$ tais que $B \subset A \cup \{b\}$, $C \subset A \cup \{c\}$ e $B \neq A \neq C$.

Como $P_4 \simeq \overline{P_4}$, $\{A, B, C\} \subseteq P_4(G)$. Então $\{b, c\} \in R(G, A)$, contradição. \square

Proposição 5.2. *Se um grafo pertence a classe P_4 -arrumada, todo subgrafo induzido de G pertence a classe P_4 -arrumada.*

Demonstração. Seja G um grafo pertencente a classe P_4 -arrumada e H um subgrafo induzido de G . Se $P \in P_4(H)$, então $P \in P_4(G)$. Ainda, $R(H, P) \subseteq R(G, P)$ e, como $|R(G, P)| \leq 1$ então $|R(H, P)| \leq 1$. Com isto, H pertence a classe P_4 -arrumada. \square

Proposição 5.3 ([GRT97]). *Seja G um grafo pertencente a classe P_4 -arrumada. Então todo subconjunto M de $V(G)$ tal que $G[M]$ é isomorfo a um C_5 , P_5 ou $\overline{P_5}$ é um módulo de G .*

Demonstração. Considere os seguintes casos:

1. $G[M]$ é isomorfo a um P_5

Sejam a e e os extremos de $G[M]$ e considere $A = M \setminus \{e\}$ e $B = M \setminus \{a\}$. Tem-se que $\{A, B\} \subseteq P_4(G)$, $\{e\} = R(G, A)$ e $\{a\} = R(G, B)$. Considere qualquer vértice $v \in V(G) \setminus M$. Note que $v \notin R(G, A)$, caso contrário $|R(G, A)| \geq 2$. Portanto, v precisa pertencer a um dos seguintes conjuntos (definidos na Seção 4.1). Veja na Figura 5.1 os três casos, associados com as situações não permitidas.

- (a) $I(G, A)$: v não é adjacente a e em G , pois, caso fosse, $v \in R(G, B)$. Portanto $N(v) \cap M = \emptyset$.
- (b) $U(G, A)$: v é adjacente a e em G , pois, caso não fosse, $v \in R(G, B)$. Portanto $M \subseteq N(v)$.
- (c) $B(G, A)$: esta situação não ocorre pois, neste caso, $v \in R(G, B)$.

Portanto, todo vértice de G ou pertence a M ou, é adjacente a todos os vértices de M , ou a nenhum deles. Com isso, conclui-se que M é um módulo de G .

2. $G[M]$ é isomorfo a um $\overline{P_5}$

Considere o grafo \overline{G} que, pela Proposição 5.1, pertence à classe P_4 -arrumada. Tem-se que $\overline{G}[M]$ é isomorfo ao P_5 e, pelo Caso 1, M é um módulo de \overline{G} . A Proposição 2.1 garante que M é um módulo de G .

3. $G[M]$ é isomorfo a um C_5

Seja a qualquer vértice de M e e um vértice de M adjacente a a . Com argumentos semelhantes ao Caso 1, prova-se que M é módulo de G .

Note que é preciso ajustar os grafos da Figura 5.1 de forma que a seja adjacente a e em todos eles. Isto, contudo, não altera a situação do vértice v em relação aos conjuntos A e B . \square

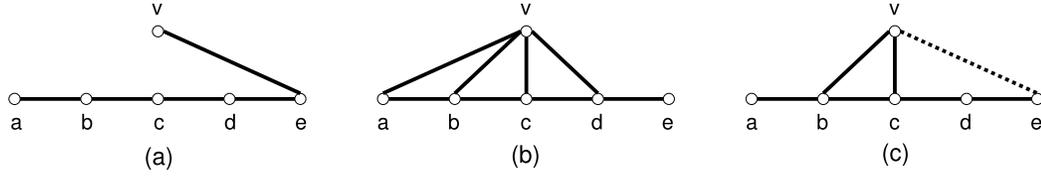


Figura 5.1: Subgrafos nos quais o P_5 induzido por $\{a, b, c, d, e\}$ não seria um módulo.

Lema 5.4 ([GRT97]). *Sejam G um grafo da classe P_4 -arrumada, H um subgrafo induzido de G isomorfo a algum dos quatro grafos Z_4, Z_5, Z_6, Z_7 da Figura 4.2 e v um vértice de $V(G) \setminus V(H)$. Então uma das seguintes afirmativas vale: $V(H) \subseteq N_G(v)$, $V(H) \cap N_G(v) = \emptyset$ ou $V(H) \cap N_G(v) = \{v : \exists P \in P_4(H) \wedge v \in B(G, P)\}$ (neste último caso, $V(H) \cap N_G(v)$ são os vértices de G que estão contidos em $B(G, P)$ para algum P , cujo grafo induzido em H é um P_4).*

Demonstração. Sejam $\{A, B\} = P_4(H)$, $\{t\} = V(H) \setminus B$ e $\{u\} = V(H) \setminus A$. Note que $\{t\} = R(H, B)$ e $\{u\} = R(H, A)$. A relação de $v \in V(G) \setminus V(H)$ com A será tratada em três casos:

1. $v \in U(G, A)$: se v não fosse adjacente a u , seguiria que $\{t, v\} \in R(G, B)$. Portanto, $V(H) \subseteq N_G(v)$.
2. $v \in I(G, A)$: se v fosse adjacente a u , seguiria que $\{t, v\} \in R(G, B)$. Portanto, $V(H) \cap N_G(v) = \emptyset$.
3. $v \in B(G, A)$: se H for isomorfo a Z_5 ou a Z_7 , v não pode ser adjacente a u , caso contrário, $\{t, v\} \in R(G, B)$. Então $N_H(v)$ são exatamente os vértices internos de $G[A]$. De outra forma, H é isomorfo a Z_4 ou a Z_6 e, se v não for adjacente a u , $\{t, v\} \in B(G, B)$. Logo, neste caso, $N_H(v)$ são os dois vértices internos de $G[A]$, que coincidem com os vértices internos de $G[B]$. \square

Teorema 5.5. *Seja G um grafo primo da classe P_4 -arrumada. Então G é isomorfo a um dos seguintes grafos: $P_5, \overline{P_5}, C_5$ ou uma aranha.*

Demonstração. Suponha que G contenha um subgrafo induzido H isomorfo a $P_5, \overline{P_5}$ ou C_5 , então pela Proposição 5.3, $V(H)$ é um módulo de G . Como G é primo, $G \simeq H$.

Então, seja G não isomorfo a $P_5, \overline{P_5}$ e C_5 . Suponha que G contenha um subgrafo induzido H isomorfo a um dos grafos Z_4, Z_5, Z_6, Z_7 da Figura 4.2. Todos eles possuem

um módulo de dois vértices que, pelo Lema 5.4, é um módulo de G , contradição. Uma vez que $Z_1 \simeq P_5$, $Z_2 \simeq \overline{P_5}$ e $Z_3 \simeq C_5$, G é livre de todos os grafos de Z_1 a Z_7 da figura. Pelo Lema 4.17, G pertence a classe P_4 -esparso e, pelo Teorema 4.18, G é isomorfo a uma aranha. \square

Lema 5.6. *Seja G um grafo da classe P_4 -arrumada e M um módulo de G contido na árvore de decomposição modular do grafo e cujo nó tem rótulo vizinhança. Se o grafo quociente associado ao nó é uma aranha H cuja partição canônica é (K, S, R) então $G[M]$ é uma quase-aranha obtida a partir de H , trocando no máximo um vértice de H , não contido em R , por um K_2 ou S_2 e substituindo $G[R]$, se R não for vazio, pelo grafo induzido por um módulo de G contido em M .*

Demonstração. Seja (K, S, R) a partição canônica dos vértices da aranha H . Denote os vértices de K por $\{k_1, \dots, k_l\}$ e os vértices de S por $\{s_1, \dots, s_l\}$ de modo que a bijeção entre K e S é dada pelo índice dos vértices. Então para cada $i \in \{1, \dots, l\}$, ou $N_G(s_i) = \{k_i\}$ ou $N_G(s_i) = K \setminus (R \cup \{k_i\})$. Da decomposição modular, sabe-se que $G[M]$ é obtido substituindo cada k_i por um grafo A_i , cada s_j por um grafo B_j e o único vértice de R , se existir (pois H é primo), por um grafo C , tais que $V(A_i)$, $V(B_j)$ e $V(C)$, se existirem, são módulos fortes de $G[M]$.

Agora, suponha que para algum A_i (B_j), $|V(A_i)| \geq 2$ ($|V(B_j)| \geq 2$). Seja $P \in P_4(H)$ tal que $P \cap A_i \neq \emptyset$ ($P \cap B_j \neq \emptyset$). É fácil de ver que $P \cap R = \emptyset$ e que, em G , cada vértice de A_i (B_j), exceto k_i (s_j), é parceiro de P . Como G pertence a classe P_4 -arrumada, $|V(A_i)| = 2$ ($|V(B_j)| = 2$) e $G[A_i]$ ($G[B_j]$) ou é isomorfo a um K_2 ou a um S_2 .

Suponha que existam dois vértices a e b em $K \cup S$ cujos correspondentes módulos A e B possuam dois vértices. Seja $P \in P_4(H)$, tal que $\{a, b\} \subset P$. Vê-se que os demais vértices de $(A \cup B) \setminus \{a, b\}$ são parceiros de P , contradizendo G pertencer a classe P_4 -arrumada. Logo, só existe um módulo não trivial em $K \cup S$. \square

Lema 5.7. *Seja H um grafo primo que é uma aranha de partição canônica (K, S, R) e G uma quase-aranha. Se G puder ser formado a partir de H , trocando $H[R]$, se R não for vazio, por um grafo da classe P_4 -arrumada e no máximo um vértice de $K \cup S$ por um K_2 ou S_2 . Então G pertence a classe P_4 -arrumada.*

Demonstração. O caso simples ocorre quando nenhum vértice foi adicionado a K ou S . Pela Proposição 4.14, nenhum $P \in P_4(G)$ contém vértices em ambos R e $K \cup S$ e se $P \subseteq K \cup S$, seus extremos estão em S e seus vértices internos em K .

Então ou $P \subseteq R$ ou $P \subseteq K \cup S$. Todo $P \subseteq R$ não possui parceiros em $K \cup S$, pois estes vértices respectivamente pertencem a $U(G, P)$ e a $I(G, P)$ e, assim, seus parceiros, se existirem, estão em R . Como $H[R]$ pertence à classe P_4 -arrumada, P possui no máximo um parceiro em R . De outro modo, todo $P \subseteq K \cup S$ não possui parceiros, pois $R \subseteq$

$B(G, P)$ e todo vértice em $(K \cup S) \setminus P$ pertence a $B(G, P)$, $I(G, P)$ ou a $U(G, P)$. Portanto, todo P em $P_4(G)$ tem no máximo um parceiro em G .

Suponha, então, o caso em que um vértice $v \in K \cup S$ foi trocado por um par v, v' (adjacentes ou não). Note que nenhum $P \in P_4(G)$ contém o par de vértices adicionados, pois qualquer vértice restante de G é ou não é adjacente a ambos, contudo, para formar um P_4 , é necessário ao menos um vértice adjacente a apenas um dos vértices do par.

Os resultados da Proposição 4.14 serão estendidos para esta configuração. Não existe $P \in P_4(G)$ com vértices em ambos R e $K \cup S$. Se P não contiver nem v nem v' , ao remover v' de G chega-se a uma contradição ao caso anterior. Se P contiver apenas um entre v e v' remove-se de G o outro vértice e chega-se a uma nova contradição. Por fim, P não pode conter ambos v e v' .

Seja agora um $P \in P_4(G[K \cup S])$. Como P não pode conter ambos os vértices v e v' , sempre é possível remover um deles e, pelo caso anterior, P tem seus extremos em S e seus vértices internos em K .

Analogamente ao caso anterior, todo $P \in R$ não possui parceiros em $K \cup S$ e todo $P \subseteq K \cup S$ não possui parceiros em R . Contudo, seja $P \subseteq K \cup S$. Se P não contiver nem v nem v' , pode-se remover um deles e, pelo caso anterior, P não tem parceiros. Se P contiver um entre v e v' , seja v o vértice que P contém, então v' é o único parceiro de P , pois v tem as mesmas adjacências que v' , e se for removido v' de G , pelo caso anterior, P não tem outros parceiros. \square

Lema 5.8. *Seja G um grafo que pertence a classe P_4 -arrumada e M um módulo presente na árvore de decomposição modular de G tal que o grafo quociente associado ao nó é uma aranha de partição canônica (K, S, R) . Então $G[M \setminus R]$ é um p -componente de G .*

Demonstração. Decorre imediatamente da extensão da Proposição 4.14 para o caso das quase aranhas, já realizada na prova do Lema 5.7 e do fato de M ser um módulo, tornando impossível que $G[K \cup S]$ esteja contido em um p -componente maior (veja definição na Seção 4.5). \square

5.1.1 Caracterizações

Esta seção aborda algumas caracterizações equivalentes para a classe. Para tal, são necessárias novas definições e resultados, dados a seguir.

Dado um grafo G qualquer, o conjunto dos grafos quociente associados aos nós vizinhança de sua árvore de decomposição modular será representado por $\pi(G)$. Note que se dois nós vizinhança da árvore de decomposição modular de G possuírem grafos quocientes isomorfos, ambos estarão contidos em $\pi(G)$, pois não se tratam do mesmo grafo, já que seu conjunto de vértices é distinto.

Teorema 5.9 ([JO95c]). *Um grafo G ou é p -conexo, ou pode ser obtido de seus p -componentes e vértices fracos por uma seqüência finita de operações ①, ② e ③.*

A operação ③ é definida de forma que $G = G_1 \textcircled{3} G_2$ está definida quando G_2 é um grafo p -conexo e separável. Neste contexto, um grafo é *separável* se todo vértice que é interno a um P_4 induzido no grafo não é extremo de nenhum outro P_4 induzido no mesmo grafo. Note que está definição abrange os grafos formados por conjuntos extensíveis separáveis, vistos na Seção 4.5. O grafo G resultante contém a união dos vértices e arestas de G_1 e G_2 além de todas arestas ligando um vértice qualquer de G_1 com qualquer vértice de G_2 que não seja extremo de um P_4 induzido no mesmo.

Lema 5.10. *Um grafo G pertence a classe P_4 -arrumada se, e somente se, todo p -componente de G for isomorfo a um dos seguintes grafos: C_5 , P_5 , $\overline{P_5}$ ou uma quase-aranha.*

Demonstração. Sejam G um grafo da classe P_4 -arrumada e B um conjunto de seus vértices que induz uma de suas p -componentes. Suponha que $G[B]$ não seja isomorfo ao C_5 , ao P_5 e ao $\overline{P_5}$. Denote por M o menor módulo forte de G que contém B . O rótulo de M na árvore de decomposição modular de G não é serial nem paralelo, pois $G[B]$ é conexo e seu complemento também. Assim sendo, pelo Teorema 5.5, o grafo quociente, Q , do módulo M é uma aranha. Tomando a partição canônica (K, S, R) de Q , sabe-se que $B \subset K \cup S$, pois não existe nenhum P_4 induzido com vértices em ambos R e $K \cup S$ e B se estivesse contido em R , M não seria mínimo. Todavia, pelo Lema 5.8, $G[K \cup S]$ é um p -componente de G e $B = K \cup S$. Então, pelo Lema 5.6, B induz uma quase-aranha.

Prova-se por indução no número de vértices que, se todos os p -componente de G forem isomorfos ao C_5 , P_5 , $\overline{P_5}$ ou a uma quase-aranha, G é um grafo P_4 -arrumada. A base é formada pelo grafo trivial e pelo C_5 , P_5 , $\overline{P_5}$ e quase-aranhas, todos são sabidamente grafos da classe.

Para o passo indutivo, G é decomposto usando uma das operações disponíveis e, por hipótese de indução, os grafos obtidos (G_1 e G_2) pertencem a classe P_4 -arrumada. É de fácil constatação que não existe P_4 induzido com vértices de ambos operandos e que os parceiros de todo P_4 induzido em um deles está contido no mesmo, portanto o grafo resultante é P_4 -arrumada. Isso ocorre com as operações ① ou ② pois os vértices de G_1 e G_2 estão em componentes conexas distintas de G ou de \overline{G} . No caso da operação ③, o motivo é que cada vértice de G_1 pertence a $B(P_i)$, para todo $P_i \in P_4(G_2)$. \square

Uma caracterização semelhante para a classe, envolvendo os nós vizinhança da árvore de decomposição modular também ser enunciada:

Teorema 5.11. *Um grafo G pertence a classe P_4 -arrumada se, e somente se, todo grafo induzido por um nó vizinhança da árvore de decomposição modular de G for isomorfo a um C_5 , P_5 , $\overline{P_5}$ ou uma quase-aranha.*

Demonstração. Seja G um grafo da classe P_4 -arrumada. Todo grafo em $\pi(G)$ é um subgrafo primo induzido de G e, pela Proposição 5.2, é um grafo da classe P_4 -arrumada. Então, pelo Teorema 5.5, é isomorfo a um C_5 , P_5 , $\overline{P_5}$ ou uma aranha prima. Seja T a árvore de decomposição modular de G , r um nó de T e X , o grafo quociente do módulo correspondente.

Se X for isomorfo a um C_5 , P_5 , $\overline{P_5}$, os filhos de r em T correspondem a módulos triviais (são vértices de G). Caso contrário, existe um filho a de r com mais de um vértice. Seja Y um conjunto formado por um vértice pertencente a cada módulo correspondente aos filhos de r , $G[Y]$ é isomorfo a um C_5 , P_5 ou $\overline{P_5}$ e, pela Proposição 5.3, é um módulo de G . Como Y e o módulo correspondente a a interceptam-se e nenhum deles contém o outro, então nenhum deles é um módulo forte, contradição ao fato de a pertencer a T .

Se X for uma aranha prima, então, pelo Lema 5.6, o grafo induzido pelo módulo correspondente a r é obtido substituindo a cabeça de X , se houver, por um módulo e no máximo um vértice de X , distinto da possível cabeça, por um K_2 ou S_2 .

Para provar a suficiência da hipótese, seja G um grafo tal que todo grafo induzido por um nó vizinhança da árvore de decomposição modular de G é isomorfo a um C_5 , P_5 ou $\overline{P_5}$ ou a uma quase-aranha. Enfim, prova-se por indução que G pertence a classe P_4 -arrumada.

Hipótese: todo grafo H que satisfaz as condições do teorema e tem menos vértices que G , pertence a classe P_4 -arrumada.

Base: G contém um vértice e, trivialmente, pertence a classe P_4 -arrumada.

Passo: Considere o módulo trivial $M = V(G)$. Como G tem mais de um vértice, sua árvore de decomposição modular possui uma raiz r e ao menos dois filhos. Pela hipótese de indução, os grafos fatores correspondentes pertencem a classe P_4 -arrumada. Três casos serão analisados, de acordo com o tipo de M .

- Vizinhança: Ou G é isomorfo a um C_5 , P_5 ou $\overline{P_5}$ e pertence a classe P_4 -arrumada ou é uma quase-aranha de partição (K, S, R) que atende as condições do teorema. Neste caso, é fácil estender a Proposição 4.15 para quase-aranhas e então, R , se não vazio, é um filho de r e, por hipótese de indução, pertence a classe P_4 -arrumada. Então, pelo Lema 5.7, G pertence a classe P_4 -arrumada.
- Paralelo: Todos os filhos de r são desconexos dois a dois. Claramente todo $P \in P_4(G)$ está contido em um dos filhos de r e todos seus parceiros estão contidos no mesmo filho, portanto G pertence a classe P_4 -arrumada.
- Serial: Note que todo $P \in P_4(G)$ está contido em um dos filhos de r , pois, no complemento de G , os filhos de r são desconexos, e que todos os parceiros de P estão contidos no mesmo filho, portanto G pertence à classe P_4 -arrumada. \square

5.2 Algoritmo de reconhecimento

O algoritmo de reconhecimento que será descrito foi apresentado em [GRT97] e tem complexidade linear em relação ao tamanho do grafo. Ele utiliza diretamente a árvore de decomposição modular, realizando um conjunto de verificações sobre os nós da mesma que possuam rótulo vizinhança, a fim de certificar que um grafo pertence a classe P_4 -arrumada ou algumas de suas subclasses.

A exatidão do algoritmo baseia-se na caracterização da classe P_4 -arrumada dada pelo Teorema 5.11 e nas caracterizações equivalentes das demais subclasses, que estão revistas na próxima seção.

5.2.1 Caracterização das classes reconhecidas

Com objetivo de utilizar no mesmo algoritmo as caracterizações dadas pelos teoremas 4.7, 4.16, 4.22 e 4.25 para as classes P_4 -reduzível, P_4 -esparso, P_4 -extensível e P_4 -leve, bem como duas extensões para contemplar as classes P_4 -reduzível estendida e P_4 -esparso estendida, considere a seguinte notação.

Para identificar entre os vários grafos especiais considerados nas caracterizações acima citadas, toma-se uma função $tipo(H)$ que mapeia grafos de interesse em inteiros.

$$tipo(H) = \begin{cases} 1, & \text{se } H \text{ for isomorfo ao } P_4; \\ 2, & \text{se } H \text{ for isomorfo ao touro}; \\ 3, & \text{se } H \text{ for isomorfo a uma aranha (exceto } P_4 \text{ e touro)}; \\ 4, & \text{se } H \text{ for isomorfo ao } C_5; \\ 5, & \text{se } H \text{ for isomorfo ao } P_5; \\ 6, & \text{se } H \text{ for isomorfo ao } \overline{P_5}; \text{ ou} \\ 7, & \text{caso contrário.} \end{cases}$$

Um grafo H em $\pi(G)$ é dito *marcado* se pelo menos um dos fatores do nó N , da árvore de decomposição modular, cujo grafo quociente é H , possuir mais de um vértice. A esta regra se aplicam duas exceções, uma quando o grafo induzido pelo módulo correspondente a N for isomorfo a uma aranha, caso em que H é tido como não marcado, e outra quando for isomorfo a uma quase-aranha, sendo o H tido como *fracamente marcado*. Para auxiliar o uso desta notação seja a função $marca(H)$, que mapeia as possíveis marcas em inteiros conforme as condições abaixo:

$$marca(H) = \begin{cases} 1, & \text{se } H \text{ não estiver marcado;} \\ 2, & \text{se } H \text{ for uma aranha fracamente marcada; ou} \\ 3, & \text{se } H \text{ estiver marcado.} \end{cases}$$

A Tabela 5.1 relaciona os valores de $tipo(H)$ com os valores máximos permitidos de $marca(H)$ para os grafos induzidos pelos nós vizinhança da decomposição modular da classe P_4 -arrumada e de algumas de suas subclasses. O valor 0 na tabela significa que nenhum grafo do tipo correspondente é permitido para a classe em questão.

P_4	touro	aranha	C_5	P_5	$\overline{P_5}$	outros	Classe
0	0	0	0	0	0	0	cografo
≤ 1	≤ 1	0	0	0	0	0	P_4 -reduzível
≤ 1	≤ 1	0	1	0	0	0	P_4 -reduzível estendida
≤ 1	≤ 1	≤ 1	0	0	0	0	P_4 -esparso
≤ 1	≤ 1	≤ 1	1	0	0	0	P_4 -esparso estendida
≤ 2	≤ 2	0	≤ 1	≤ 1	≤ 1	0	P_4 -extensível
≤ 2	≤ 2	≤ 2	0	≤ 1	≤ 1	0	P_4 -leve
≤ 2	≤ 2	≤ 2	≤ 1	≤ 1	≤ 1	0	P_4 -arrumada

Tabela 5.1: Valores de $marca(H)$ permitidos para cada $tipo(H)$ em algumas classes de grafos.

5.2.2 Algoritmo e análise

O Algoritmo 10 constrói a decomposição modular do grafo e, então, verifica o tipo e a marcação dos grafos induzidos pelos nós vizinhança. Após isso, basta verificar a classe mais restrita da Tabela 5.1 cujas condições foram atendidas.

Para mostrar que o Algoritmo 10 executa em tempo linear no tamanho do grafo ($|V(G)| + |E(G)|$), seus passos serão analisados. Os dois laços entre as linhas 2 e 4 e entre as linhas 12 e 16 são executados em tempo constante. A construção da árvore de decomposição modular no passo 1 leva tempo linear em G , como exposto no Capítulo 3. Durante a construção da árvore, pode-se preencher um vetor com cada um dos grafos em $\pi(G)$. Resta mostrar que o laço principal, entre os passos 5 e 11, no qual as funções $tipo$ e $marca$ são computadas para cada grafo G_i em $\pi(G)$ toma tempo linear.

A função $tipo$ testa isomorfismo de G_i com C_5 , P_5 e $\overline{P_5}$, em tempo constante, e, se falharem, verifica se G_i é uma aranha (o P_4 e o touro são as aranhas de 4 e 5 vértices), em tempo linear no tamanho de G_i , da seguinte forma:

- Verifica se G_i é um grafo **split**. Isto é feito em tempo linear [Gol80].

Algoritmo 10 Reconhecimento de grafos da classe P_4 -arrumada e subclasses.

```

1: Construir a árvore de decomposição modular  $T(G)$  de  $G$ .
2: Para  $i$  de 1 a 7 faça
3:    $mascara[i] \leftarrow 0$ 
4: Fim Para
5: Para  $G_i \in \pi(G)$  faça
6:   Se  $tipo(G_i) \leq 6$  então
7:      $mascara[tipo(G_i)] \leftarrow \max(mascara[tipo(G_i)], marca(G_i))$ 
8:   Senão
9:      $mascara[7] \leftarrow 1$ 
10:  Fim Se
11: Fim Para
12: Para  $i$  de 1 até 6 faça
13:   Se  $mascara$  é compatível com a linha  $i$  da Tabela 5.1 então
14:     Retorne o nome da classe da linha  $i$  da Tabela 5.1
15:   Fim Se
16: Fim Para
17: Retorne “nenhuma”

```

- Ordena os vértices de G_i pelos seus graus através de algum algoritmo linear como *bucket sort*. Verifica se os primeiros $|S_i|$ vértices possuem todos grau 1, $|K_i| - 1$ ou $|K_i| - 2$ (respectivamente, aranha magra, gorda sem cabeça e com cabeça).

Já a função *marca* verifica se todos os vértices de G_i são folhas na árvore de decomposição modular. Caso contrário, se G_i for uma aranha, verifica se há apenas um vértice além da possível cabeça que não é folha da árvore e, caso exista, se possui exatamente dois filhos que são folhas.

Pode-se ver que o tempo que o laço principal toma é $O(\sum_{i=1}^p |G_i|)$. Como cada vértice de cada G_i corresponde a um nó da árvore de decomposição modular, a soma é limitada pelo número de nós da árvore, que é sempre menor que $2|V(G)|$.

5.3 Aplicações

5.3.1 Número cromático e Clique máxima

Determinar $\chi(G)$ e $\omega(G)$ (veja A.2.3) para um grafo G qualquer são dois problemas sabidamente NP-difíceis. Contudo, eles podem ser resolvidos em tempo linear para grafos da classe P_4 -arrumada com o uso da árvore de decomposição modular.

Ambos os parâmetros podem ser calculados para um grafo G pertencente à classe P_4 -arrumada de forma recursiva sobre sua árvore de decomposição modular T . Para cada nó

r de T determina-se $\omega(M)$ e $\chi(M)$, sendo M o grafo induzido pelo módulo representado por r , dispondo dos valores $\omega(X_i)$ e $\chi(X_i)$ de todos os nós filhos de r . O cálculo depende do tipo do módulo r .

Primeiramente, observa-se o efeito das operações $\textcircled{0}$ e $\textcircled{1}$ (definidas na Seção 4.2) sobre os parâmetros em questão. Dados dois grafos quaisquer, G_1 e G_2 , tais que $V(G_1) \cap V(G_2) = \emptyset$:

$$\begin{aligned}\chi(G_1 \textcircled{0} G_2) &= \max\{\chi(G_1), \chi(G_2)\} \\ \omega(G_1 \textcircled{0} G_2) &= \max\{\omega(G_1), \omega(G_2)\} \\ \chi(G_1 \textcircled{1} G_2) &= \chi(G_1) + \chi(G_2) \\ \omega(G_1 \textcircled{1} G_2) &= \omega(G_1) + \omega(G_2)\end{aligned}$$

Aplicando as regras acima, calcula-se os parâmetros para todos os módulos seriais e paralelos de T . Resta lidar com os módulos vizinhança. Pelo Teorema 5.11, um módulo vizinhança de T ou é isomorfo a um P_5 , C_5 ou $\overline{P_5}$, ou é uma quase-aranha. Exceto no último caso, os parâmetros desejados são conhecidos de antemão:

$$\begin{aligned}\chi(C_5) &= 3 & \chi(P_5) &= 2 & \chi(\overline{P_5}) &= 3 \\ \omega(C_5) &= 2 & \omega(P_5) &= 2 & \omega(\overline{P_5}) &= 3\end{aligned}$$

Caso seja uma quase-aranha, o grafo induzido pelo módulo pode ser construído por $G_1 \textcircled{3} G_2$ (veja definição na Seção 4.4.2), no qual G_2 é a cabeça da quase-aranha, se houver, e G_1 é o grafo induzido pelo restante dos vértices da quase-aranha.

Os parâmetros calculados dependem da natureza de G_1 . Caso G_1 seja uma quase-aranha, este provém de uma aranha H com partição (K, S, \emptyset) , da qual se substituiu um vértice por um K_2 ou S_2 . Caso G_1 seja uma aranha, considere (K, S, \emptyset) a partição de G_1 .

Seja $\epsilon = 1$ quando G_1 for uma quase-aranha obtida substituindo um vértice de K por um K_2 e $\epsilon = 0$, caso contrário. Define-se, também, $\epsilon' = 1$ quando G_1 for uma quase-aranha obtida substituindo um vértice de S por um K_2 e H for uma aranha gorda, e $\epsilon' = 0$, caso contrário. Então, tem-se:

$$\begin{aligned}\chi(G_1 \textcircled{3} G_2) &= |K| + \epsilon + \max\{\chi(G_2), \epsilon'\} \\ \omega(G_1 \textcircled{3} G_2) &= |K| + \epsilon + \max\{\omega(G_2), \epsilon'\}\end{aligned}$$

A árvore de decomposição modular de G pode ser construída em tempo linear e os testes para determinar se um módulo vizinhança é um dos permitidos para um grafo da classe P_4 -arrumada também são lineares (veja a Seção 5.2.2). Com uma única travessia da árvore em pós-ordem o algoritmo é capaz de determinar $\chi(G)$ e $\omega(G)$.

5.3.2 Partição mínima em cliques e Conjunto independente máximo

Estes dois problemas são os duais dos problemas da seção anterior. Uma partição em cliques de um grafo é uma partição de seus vértices cujas partes induzem subgrafos completos. O menor entre os tamanhos de tais partições é representado por $\theta(G)$.

Como $\theta(G) = \chi(\overline{G})$, $\alpha(G) = \omega(\overline{G})$ e a classe P_4 -arrumada é fechada por complemento (Proposição 5.1), pode-se usar a árvore de decomposição modular de \overline{G} , t' para calcular tais parâmetros usando o método definido na seção anterior para calcular $\chi(\overline{G})$ e $\alpha(G)$ de um grafo da classe.

Todavia, para preservar a linearidade daqueles algoritmos, não é possível complementar o grafo para computar T' , pois $|E(\overline{G})| = O(|E(G)|^2)$. Então, obtém-se T' a partir da árvore de decomposição modular de G , T , alternando os rótulos dos módulos seriais e paralelos. Quanto aos módulos vizinhança, para determinar seu grafo quociente em T' a partir do grafo quociente em T , nota-se que o C_5 permanece inalterado, o P_5 e o $\overline{P_5}$ alternam e quase-aranhas magras e gordas também alternam (neste último caso, o número de arestas de um grafo quociente em T' é função linear do número de arestas do correspondente grafo quociente em T).

Capítulo 6

Conclusão

Neste trabalho, foram descritas muitas das propriedades dos módulos de um grafo e de sua decomposição modular, a partir das mais intuitivas até algumas mais elaboradas, que estão distribuídas na literatura científica por vários anos. Em seguida, a técnica de Ehrenfeucht e os algoritmos propostos por Dahlhaus, Gustedt e McConnell, e por Habib, de Montgolfier, Paul e Capelle para construção da árvore de decomposição modular foram expostos, descritos e, em algumas situações, provados de forma mais detalhada que nos trabalhos originias.

Além de estudar a decomposição em si, foram apresentadas aplicações da mesma na caracterização de classes de grafos e solução de problemas de otimização e isomorfismo nos grafos dessas classes. Uma das quais, a classe P_4 -arrumada, foi detalhada em particular, por abranger boa parte das demais, juntamente com a descrição de algoritmos para solução de muitos problemas clássicos em grafos da classe.

Como um todo, o trabalho objetivou introduzir ao leitor na teoria dos módulos e da decomposição modular, expondo suas aplicações atuais. Como futuros trabalhos de pesquisa nesta área, destacam-se o uso da decomposição para resolver novos problemas, notoriamente nas classes de grafos estudadas, ou a extensão de técnicas utilizadas em um problema para outra classe, caminhos que vem sendo trilhados por vários pesquisadores.

Apêndice A

Definições

Um par ordenado é um ente matemático composto por dois elementos e representado da forma (a, b) . Uma seqüência é uma lista ordenada de elementos, representada da forma $(a_1, a_2, a_3, \dots, a_n)$.

A.1 Conjuntos

Os conjuntos neste trabalho seguem a noção matemática de conjuntos. Apenas para clarificação, 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, diferença e diferença simétrica¹ entre conjuntos serão representadas por \cup , \cap , \setminus e Δ . A união e intersecção de vários conjuntos pode ser representada por \bigcup_{C_i} e \bigcap_{C_i} .

A cardinalidade do conjunto A (número de elementos do mesmo) é expressa como $|A|$.

O produto cartesiano de conjuntos será representado pelo operador \times .

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 $\bigcup_{1 \leq i \leq n} P_i = S$. Os elementos do conjunto que forma a partição P são chamados *partes*.

A.2 Grafos

Um grafo G não orientado é representado por um conjunto de vértices $V(G)$ e uma família de arestas $E(G)$. Cada elemento de $E(G)$ é um par não ordenado contendo dois elementos de $V(G)$.

¹ $A \Delta B = (A \setminus B) \cup (B \setminus A)$

Um grafo G orientado é formado por um conjunto de vértices $V(G)$ e uma família de arcos (ou arestas orientadas) $E(G)$. Cada elemento de $E(G)$ é um par ordenado contendo dois elementos de $V(G)$.

Dados um conjunto V de vértices e uma família A de arestas, que satisfazem as condições para formarem um grafo não orientado, então o grafo correspondente pode ser denotado por (V, A) . A mesma representação é usada para grafos orientados, sendo A uma família de arcos.

Um grafo é *trivial* se contém apenas um vértice e nenhuma aresta.

Os vértices que formam uma aresta são chamados extremos da aresta. Um grafo é dito *simples* se não possui *laços* (arestas cujos extremos coincidem) nem *arestas múltiplas* (mais de uma aresta com os mesmos extremos, e mesma orientação no caso de grafos orientados). Neste caso, as arestas do grafo formam um conjunto. Exceto se explicitamente observado, os grafos deste documento são simples. Dois vértices que são extremos de uma aresta são ditos *adjacentes* ou *vizinhos*.

Um *subgrafo* H de um grafo G é um grafo tal que $V(H) \subseteq V(G)$ e $E(H) \subseteq E(G)$. Como H é um grafo, além das condições acima, ambos os extremos de cada uma de suas arestas devem pertencem ao seu conjunto de vértices. Um grafo F é um *subgrafo induzido* de G se é subgrafo de G e toda aresta de G cujos extremos pertencem ambos aos vértices de F , também é aresta de F . O sugrafo 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 para os quais existe uma aresta $\{v, u\}$ ((v, u) se o grafo for orientado) no grafo e é representada por $N_G(v)$. O complemento da vizinhança de v , ou seja, $V(G) \setminus N_G(v) \setminus \{v\}$ é representado por $\overline{N_G(v)}$. O *grau* de um vértice v , por vezes denotado por $\delta_G(v)$ é o número de arestas que o tem como extremo (laços contados duas vezes).

Um grafo orientado G é dito *transitivo* se a relação binária formada pelo seu conjunto de arcos é transitiva, isto é, sempre que (x, y) e (y, z) são arestas de G , (x, z) , para x, y e z distintos. Dado um grafo orientado G , seu *fecho transitivo* é o grafo obtido adicionando a G os arcos necessários para que este se torne transitivo.

Uma *orientação transitiva* de um grafo não orientado G é uma orientação de suas arestas de forma que o grafo orientado correspondente seja transitivo. Um grafo não orientado que admite ao menos uma orientação transitiva é chamado de *grafo de comparabilidade*.

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

Um grafo não orientado é *conexo* se para todo par de vértices, x e y , distintos 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 grafo orientado G é dito *fortemente conexo* se, para todo par de vértices u e v , distintos, de G , existe um caminho orientado de u para v e outro de v para u .

Um *corte* é uma partição dos vértices de um grafo em dois conjuntos. O tamanho de um corte é o número de arestas do grafo que possuem um extremo em cada parte do corte.

Um *separador* S de um grafo G é um subconjunto de seus vértices de modo que $G[V(G) \setminus S]$ é desconexo.

A.2.1 Alguns grafos comuns

O *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 $\{\{v_i, v_{i+1}\} : 1 \leq i < n\}$.

O *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 .

A.2.2 Operações e relações

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 o grafo $G = (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$. 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 G_1 e, então, toma-se a união dos grafos.

A *contração* de vértices de um grafo G não orientado é uma operação que toma um subconjunto X dos vértices de G e produz um novo grafo G' , no qual os vértices de X são substituídos por um novo vértice. Seja x o novo vértice, então $V(G') = \{x\} \cup V(G) \setminus X$ e $E(G') = \{\{x, y\} : \forall \{z, y\} \in E(G), z \in X, y \notin X\} \cup \{\{z, y\} : \forall \{z, y\} \in E(G), z \notin X, y \notin X\}$. Para grafos orientados, $E(G') = \{(x, y) : \forall (z, y) \in E(G) \text{ se } z \in X \text{ e } y \notin X \text{ ou se } z \notin X \text{ e } y \in X\} \cup \{(z, y) : \forall (z, y) \in E(G), z \notin X, y \notin X\}$.

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

A.2.3 Alguns parâmetros

O tamanho de um caminho é seu número de vértices menos um (equivalente ao número de arestas a serem percorridas). A *distância* de um vértice a a um vértice b num grafo G

é menor entre os tamanhos de todos os caminhos de a a b em G , se existir algum (caso contrário a distância não está definida). Dado um grafo qualquer G , seu *diâmetro* é a maior distância entre todos os seus pares de vértices (desconsiderando aqueles para os quais a distância não existe).

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.

Chama-se de *número cromático* de um grafo G , o menor número de cores necessário para colorir seus vértices de forma que vértices adjacentes sejam coloridos distintamente. Representa-se este número por $\chi(G)$. O problema oposto (em relação ao complemento do grafo) é encontrar a menor partição dos vértices do grafo em cliques, cujo tamanho é representado por $\theta(G)$.

Um *scattering set* de um grafo G é um subconjunto S de $V(G)$ tal que $G[V(G) \setminus S]$ contém mais de um componente conexo. O *scattering number* do grafo G , representado por $s(G)$ é dado por $\max\{c(G[V(G) \setminus S]) - |S|\}$ para todo *scattering set* S de G e sendo $c(H)$ o número de componentes conexos do grafo H .

A.2.4 Classes com características de interesse

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*. Opcionalmente, 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, excluído 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 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.

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 *menor preenchimento* e é denotado por $\Phi(G)$.

Os grafos **split** são aqueles cujo conjunto de vértices pode ser particionado em um dois conjuntos K e S que induzem um grafo completo e um grafo sem arestas, respectivamente.

Um *torneio* é um grafo orientado obtido atribuindo uma orientação a cada aresta de um grafo não orientado e completo. Um *torneio bipartido* é obtido de modo semelhante, a partir de um grafo não orientado e bipartido completo.

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, perfeitos e perfeitamente ordenáveis.

A.3 Ordens

Uma ordem é uma relação binária (denotada por \leq) nos elementos de um conjunto S . Esta relação satisfaz as seguintes propriedades, para três elementos quaisquer a, b e c , de S :

- Reflexividade: $a \leq a$;
- Antisimetria: se $a \leq b$ e $b \leq a$, então $a = b$;
- Transitividade: se $a \leq b$ e $b \leq c$, então $a \leq c$;

Uma ordem é total se, para todo par de elementos a e b de S , ou $a \leq b$ ou $b \leq a$. Caso contrário a ordem é dita parcial.

Um par (S, \leq) formado por um conjunto S e uma ordem parcial \leq sobre S é um *conjunto parcialmente ordenado*, abreviado como CPO (em inglês, **poset**). Pode-se representar um conjunto parcialmente ordenado por um grafo orientado acíclico P cujos vértices são os elementos de S . Também é possível produzir um CPO a partir de um grafo orientado acíclico, embora grafos distintos possam produzir o mesmo CPO.

Um *conjunto totalmente ordenado* é um CPO no qual a ordem dada é total.

Uma *ordenação topológica* T de um CPO $P = (S, \leq)$ ou, equivalentemente, de um grafo orientado acíclico, é uma ordem total dos elementos de S tal que a precede b em T apenas se, em P , $a \leq b$ ou a e b são incomparáveis. Isto é, a ordenação topológica é

compatível com a ordem imposta pelo CPO, não violando nenhuma de suas restrições. Note que um CPO pode possuir várias ordenações topológicas distintas.

Índice Remissivo

- aranha, 47–49, 51, 55–57, 59, 61
 - gorda, 47
 - magra, 47
 - partição canônica, 47, 48, 55, 57
- árvore, 75
- brittle, 55, 56, 75
- classe hereditária, 58, 76
- co-árvore, 40, 51
- cografo, 43
 - canônico, 45, 51
- componente
 - conexa, 22
 - fortemente conexa, 14, 23
- conjunto homogêneo, 3
- conjunto parcialmente ordenado, *veja* CPO
- conjunto totalmente ordenado, 27, 76
- corte, 74
- CPO, 14, 27, 41, 76
 - decomposição em camadas, 15
 - em camadas, 15
 - ideal, 14, 15
 - modular, 14
 - multi-árvore, 41
- decomposição modular, 10, 30, 32, 38, 48, 55
 - árvore, 10, 13, 18, 20, 35, 40, 49, 54
 - nó degenerado, 12
 - nó paralelo, 12
 - nó primo, 12
 - nó serial, 12
 - nó vizinhança, 12, 34, 40, 43, 44, 46, 48, 51–55, 57
- extensão própria, 51
- conjunto extensível, 52
- fecho transitivo, 14, 15
- fratura, 31
 - árvore, 32
- grafo
 - bipartido, 75
 - bipartido completo, 75
 - comparabilidade, 41, 73
 - cordal, 76
 - Dacey, 41
 - fator, 8, 44, 46, 48, 51, 52, 55–57
 - HD*, 41
 - perfeitamente ordenável, 46, 55, 75
 - perfeito, 42, 46, 75
 - primo, 3, 11, 38
 - pseudo-split, 56
 - quociente, 7, 10, 12, 43, 46, 48, 51–55, 57
 - separável, 64
 - split, 55, 56, 67, 76
 - torneio, 76
 - torneio bipartido, 15, 76
 - trivial, 73
- módulo, 3, 13–15, 33
 - adjacente, 5

- forte, 6, 9–12, 17, 22, 33
 - forte maximal próprio, 10, 48
 - não-adjacente, 5
 - sobreposição, 5, 22
 - trivial, 3, 11
- orientação transitiva, 73
- p-componente, 54, 63
- p-conexo, 53, 64
- P_4 -arrumada, 56, 57
- P_4 -carregada, 55–57
- P_4 -carregada estendida, 56, 57
- P_4 -esparso, 46, 51, 55
- P_4 -esparso estendida, 51, 57
- P_4 -extensível, 52, 57
- P_4 -leve, 54–57
- P_4 -reduzível, 46, 48, 51
- P_4 -reduzível estendida, 46
- parceiro, 37
- partição de congruência, 7, 10, 16
- permutação fatorizada, 27, 30, 33
- poset, *veja* CPO
- pr-árvore, 45, 49
- ps-árvore, 49
- px-árvore, 53
- quase-aranha, 59
- scattering number, 75
- scattering set, 75
- vértices gêmeos, 34

Referências Bibliográficas

- [CHdM02] C. Capelle, M. Habib, e F. de Montgolfier. Graph decompositions and factorizing permutations. *Discrete Mathematics and Theoretical Computer Science*, 5:55–70, 2002.
- [CLB81] D.G. Corneil, H. Lerchs, e L. Stewart Burlingham. Complement reducible graphs. *Discrete Applied Mathematics*, 3(3):163–174, 1981.
- [CLB85] D.G. Corneil, H. Lerchs, e L. Stewart Burlingham. A linear recognition algorithm for cographs. *SIAM J. Computing*, 14(4):926–934, 1985.
- [DEdW05] Marc Demange, Tinaz Ekim, e Dominique de Werra. Partitioning cographs into cliques and stable sets. *Discrete Optimization*, 2:145–153, 2005.
- [DGM01] E. Dahlhaus, J. Gustedt, e R.M. McConnell. Efficient and practical algorithms for sequential modular decomposition. *Journal of Algorithms*, 41(2):360–387, 2001. ISSN 0196-6774.
- [EGMS94] 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. *Journal of Algorithms*, 16(2):283–294, 1994. ISSN 0196-6774.
- [Gal67] T. Gallai. Transitiv orientierbare graphen. *Acta Math. Acad. Sci. Hungar.*, 18:25–66, 1967.
- [Gia96] V. Giakoumakis. P_4 -laden graphs: A new class of brittle graphs. *Information Processing Letters*, 80:29–36, 1996.
- [GKS95] M.C. Golumbic, H. Kaplan, e R. Shamir. Graph sandwich problems. *Journal of Algorithms*, 19(3):449–473, 1995. ISSN 0196-6774.
- [Gol80] M.C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, 1980. ISBN 0122892607.

- [GRT97] V. Giakoumakis, F. Roussel, e H. Thuillier. On P_4 -tidy graphs. *Discrete Mathematics & Theoretical Computer Science*, 1(1):17–41, 1997.
- [GV97] V. Giakoumakis e J.M. Vanherpe. On extended P_4 -reducible and extended P_4 -sparse graphs. *Theoretical Computer Science*, 180(1-2):269–286, 1997.
- [HdMP04] M. Habib, F. de Montgolfier, e C. Paul. A simple linear-time modular decomposition algorithm for graphs, using order extending. *Lecture Notes in Computer Science*, 3111:187–198, 2004.
- [Hoà83] C.T. Hoàng. *A class of perfect graphs*. Tese de Mestrado, School of Computer Science, Montreal, 1983.
- [HPV99] Michel Habib, Christophe Paul, e Laurent Viennot. Partition refinement techniques: An interesting algorithmic tool kit. *International Journal of Foundations of Computer Science*, 10(2):147–170, 1999.
- [HS95] W. Hochstättler e H. Schindler. Recognizing P_4 -extendible graphs in linear time. Relatório Técnico 95188, Universität zu Köln, 1995.
- [HT95] W. Hochstättler e G. Tinhofer. Hamiltonicity in graphs with few P_4 's. *Computing*, 54(3):213–225, Setembro 1995. ISSN 0010-485X.
- [JO89a] B. Jamison e S. Olariu. A new class of brittle graphs. *Studies in Applied Mathematics*, 81:89–92, 1989.
- [JO89b] B. Jamison e S. Olariu. P_4 -reducible graphs, a class of uniquely tree representable graphs. *Studies in Applied Mathematics*, 81:79–87, 1989.
- [JO91] B. Jamison e S. Olariu. On a unique tree representation for P_4 -extendible graphs. *Discrete Applied Mathematics*, 34(1-3):151–164, 1991. ISSN 0166-218X.
- [JO92a] B. Jamison e S. Olariu. Recognizing P_4 -sparse graphs in linear time. *SIAM Journal on Computing*, 21(2):381–406, 1992.
- [JO92b] B. Jamison e S. Olariu. A tree representation for P_4 -sparse graphs. *Discrete Applied Mathematics*, 35(2):115–129, 1992. ISSN 0166-218X.
- [JO95a] B. Jamison e S. Olariu. Linear time optimization algorithms for P_4 -sparse graphs. *Discrete Applied Mathematics*, 61:155–175, 1995.

- [JO95b] B. Jamison e S. Olariu. A linear-time recognition algorithm for P_4 -reducible graphs. *Theoretical Computer Science*, 145(1-2):329–344, 1995. ISSN 0304-3975.
- [JO95c] B. Jamison e S. Olariu. P-components and the homogeneous decomposition of graphs. *SIAM J. Discrete Math.*, 8(3):448–463, 1995. ISSN 0895-4801.
- [Jun78] H.A. Jung. On a class of posets and the corresponding comparability graphs. *J. Combin. Theory*, 24(Series B):125–133, 1978.
- [KS03] Sulamita Klein e Jaime L Szwarcfiter. A representation for the modules of a graph and applications. *Journal of the Brazilian Computer Society*, 9(1):9–16, 2003.
- [LdMM⁺04] F. Larrión, C.P. de Mello, A. Morgana, V. Neumann-Lara, e M.A. Pizaña. The clique operator on cographs and serial graphs. *Discrete Mathematics*, 282(1-3):183–191, 2004.
- [Ler71] H. Lerchs. On cliques and kernels. Relatório técnico, Department of Computer Science, University of Toronto, Março 1971.
- [Ler72] H. Lerchs. On the clique-kernel structure of graphs. Relatório técnico, Department of Computer Science, University of Toronto, Outubro 1972.
- [LO98] Rong Lin e Stephan Olariu. A fast parallel algorithm to recognize P_4 -sparse graphs. *Discrete Applied Mathematics*, 81:191–215, 1998.
- [LOP06] R. Lin, S. Olariu, e G. Pruesse. An optimal path cover algorithm for cographs. *Computers & Mathematics with Applications*, 51:75–83, 2006.
- [Möh85] R. H. Möhring. Algorithmic aspects of the substitution decomposition in optimization over relations, set systems and boolean functions. *Annals of Operations Research*, 4(1):195–225, Dezembro 1985.
- [NOZ03] Koji Nakano, Stephan Olariu, e Albert Y. Zomaya. A time-optimal solution for the path cover problem on cographs. *Theor. Comput. Sci.*, 290(3):1541–1556, 2003. ISSN 0304-3975.
- [NP04a] Stavros D. Nikolopoulos e Leonidas Palios. Efficient parallel recognition of cographs. *Discrete Applied Mathematics*, 136:329–340, 2004.
- [NP04b] Stavros D. Nikolopoulos e Charis Papadopoulos. The number of spanning trees in p_4 -reducible graphs. *Colloquium on Math. and Comput. Science (CMCS '04)*, 2004.

- [NP06] Stavros D. Nikolopoulos e Leonidas Palios. Minimal separators in P_4 -sparse graphs. *Discrete Mathematics*, 306:381–392, 2006.
- [Sei74] D. Seinsche. On a property of the class of n -colorable graphs. *Journal of Combinatorial Theory*, 16:191–193, 1974.
- [SS04] Ron Shamir e Roded Sharan. A fully dynamic algorithm for modular decomposition and recognition of cographs. *Discrete Applied Mathematics*, 136:329–340, 2004.
- [Sum74] D.P. Summer. Dacey graphs. *J. Austral. Math. Soc.*, 18:492–502, 1974.
- [YY93] Ming-Shing Yu e Cheng-Hsing Yang. An $o(n)$ time algorithm for maximum matching on cographs. *Inf. Process. Lett.*, 47(2):89–93, 1993. ISSN 0020-0190.