

INSTITUTO FEDERAL DO NORTE DE MINAS GERAIS  
CAMPUS MONTES CLAROS  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**IMPLEMENTAÇÃO EFICIENTE DE ALGORITMO  
PARA RECONHECIMENTO DE  
GRAFOS BIPARTIDOS PFAFFIANOS**

ÁLVARO ERNÂNI FONSECA MOTA  
ORIENTADOR: ALBERTO ALEXANDRE ASSIS MIRANDA

Montes Claros  
Novembro de 2022



ÁLVARO ERNÂNI FONSECA MOTA

IMPLEMENTAÇÃO EFICIENTE DE ALGORITMO  
PARA RECONHECIMENTO DE  
GRAFOS BIPARTIDOS PFAFFIANOS

Monografia apresentada ao Curso de Graduação em Ciência da Computação do Instituto Federal do Norte de Minas Gerais – Campus Montes Claros, como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

ORIENTADOR: ALBERTO ALEXANDRE ASSIS MIRANDA

Montes Claros  
Novembro de 2022



# Agradecimentos

Eu gostaria de agradecer a todas as pessoas que me ajudaram a alcançar essa conquista, todos os professores que tive no IFNMG Campus Montes Claros e a própria instituição. Agradeço a minha família que me deu o apoio para eu conseguir me manter dedicado aos meus estudos e aos meus objetivos. Agradeço ao meu orientador, Alberto Alexandre Assis Miranda por todos os ensinamentos, pela atenção e paciência durante este projeto. Agradeço também aos amigos que fiz durante meu curso pelos momentos de diversão que me proporcionaram. Agradeço a toda a sociedade por viabilizar uma ótima qualidade de ensino em uma instituição pública.



*“Tudo fica mais interessante quando  
se presta atenção nos detalhes.”*  
(Autor desconhecido)





# Resumo

O Problema de se decidir se um grafo bipartido é pfaffiano está em P. Mas o problema de se decidir se um grafo qualquer é pfaffiano ainda não foi bem caracterizado e nem sabemos se esse problema está em NP. Neste projeto, implementamos o algoritmo descoberto por MRST (McCuaig, Robertson, Seymour, Thomas), que chamaremos de Algoritmo MRST. Ele tem tempo de execução  $O(nm^3)$  e decide se um grafo bipartido é pfaffiano. Usando sucessivas reduções em um grafo bipartido podemos decidir se este grafo bipartido é pfaffiano apenas checando se todos os grafos obtidos das suas reduções são pfaffianos. Dentre as operações realizadas no Algoritmo MRST, podemos destacar a Decomposição em Cortes Justos e a Decomposição 4-Soma. Neste projeto, além de apresentarmos os resultados de MRST sobre os quais o funcionamento do algoritmo se baseia, também descrevemos os passos do Algoritmo MRST. Neste projeto, criamos uma implementação (com complexidade  $O(nm^3)$ ) do algoritmo MRST que reconhece grafos bipartidos pfaffianos .

**Palavras-chave:** Algoritmo, Grafo Bipartido, Pfaffiano.



# Abstract

The problem of deciding whether a bipartite graph is pfaffian is in P. But the problem of deciding whether any graph is pfaffian has not yet been well characterized and we do not even know if this problem is in NP. In this project, we implemented the algorithm discovered by MRST (McCuaig, Robertson, Seymour, Thomas), which we will call MRST Algorithm. It has runtime  $O(nm^3)$  and decides whether a bipartite graph is pfaffian. Using successive reductions on a bipartite graph we can decide whether this bipartite graph is pfaffian just by checking if all graphs obtained from its reductions are pfaffian. Among the operations performed in the MRST Algorithm, we can highlight the Tight Cut Decomposition and the 4-Sum Decomposition. In this project, in addition to presenting the MRST results which are the basis for the algorithm, we also describe the steps of the MRST Algorithm. During this project's execution, we have implemented (with  $O(nm^3)$  complexity) the MRST algorithm that recognizes pfaffian bipartite graphs.

**Keywords:** Algorithm, Bipartite Graph, Pfaffian.



# Sumário

Agradecimentos	ix
Resumo	xiii
Abstract	xv
Lista de Figuras	xix
Lista de Tabelas	xxiii
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	2
1.1.1 Sistemas de Sinais Sinais-Solucionáveis (Sign-Solvable Sign Sys- tems) . . . . .	2
1.1.2 O Problema do Dímero (The Dimer Problem) . . . . .	5
1.2 Contextualização do Problema . . . . .	5
1.3 Algoritmo . . . . .	7
1.4 Objetivo . . . . .	7
1.4.1 Objetivos Específicos . . . . .	7
1.5 Organização do Texto . . . . .	7
<b>2 Conceitos básicos</b>	<b>9</b>
2.1 Grafos . . . . .	9
2.1.1 Grafo orientado . . . . .	11
2.1.2 Famílias Especiais de Grafos . . . . .	13
2.1.3 Emparelhamento . . . . .	13
2.2 Grafos Cobertos por Emparelhamentos . . . . .	15
2.2.1 Cortes . . . . .	16
2.2.2 Cortes Justos . . . . .	16
2.2.3 Decomposição em Cortes Justos . . . . .	17
2.3 Grafo Redutível . . . . .	17

2.4	Permutações . . . . .	18
2.5	Grafo Pfaffiano . . . . .	21
2.5.1	Pfaffiano de uma matriz . . . . .	21
2.5.2	Pfaffiano de um grafo . . . . .	23
<b>3</b>	<b>Revisão da Literatura</b>	<b>27</b>
<b>4</b>	<b>Metodologia</b>	<b>31</b>
4.1	Algoritmo MRST e Por Que Funciona . . . . .	31
4.1.1	Obtendo Grafo(s) Coberto(s) por Emparelhamentos . . . . .	31
4.1.2	Decomposição em Cortes Justos . . . . .	34
4.1.3	4-Somas . . . . .	41
4.1.4	Presilhas Pfaffianas e grafo de Heawood . . . . .	44
4.2	Descrição do Algoritmo MRST . . . . .	46
4.2.1	Algoritmo para encontrar emparelhamento máximo em grafo bi-partido . . . . .	46
4.2.2	Obtendo grafo(s) coberto(s) por emparelhamentos . . . . .	51
4.2.3	Decompondo em cortes justos . . . . .	53
4.2.4	Aplicando a Decomposição 4-soma . . . . .	57
4.2.5	Verificando a Propriedade Pfaffiana . . . . .	62
4.2.6	Tempo de Execução do Algoritmo MRST . . . . .	62
4.3	Testes e Tecnologias . . . . .	62
4.3.1	Testes . . . . .	62
4.3.2	Tecnologias . . . . .	63
4.4	Atividades . . . . .	64
<b>5</b>	<b>Resultados Obtidos</b>	<b>65</b>
5.1	Remoção das arestas inadmissíveis . . . . .	65
5.2	Decomposição em cortes justos . . . . .	66
5.3	Decomposição 4-soma . . . . .	67
5.4	Verificando presilhas irreduzíveis . . . . .	68
5.5	Verificação da propriedade pfaffiana . . . . .	70
5.6	Análise da complexidade . . . . .	70
5.7	Conclusão . . . . .	72
	<b>Referências Bibliográficas</b>	<b>77</b>

# Lista de Figuras

1.1	Aplicação em química orgânica McCuaig [2004][Fig 7]. . . . .	6
1.2	Redução a presilhas irreduzíveis. As arestas seccionadas se ligam à quádrupla de vértices que reduzem o grafo [Miranda, 2006][página 6. Fig 1.1]). . .	8
2.1	Diagrama do grafo $G$ . . . . .	10
2.2	$a)$ Um grafo simples sem laços e sem arestas paralelas. $b)$ Um grafo não simples com arestas paralelas $e_3$ e $e_4$ e um laço $e_5$ . . . . .	11
2.3	Exemplo de grafo orientado. . . . .	12
2.4	$(a)$ Um grafo conexo, e $(b)$ um grafo desconexo. . . . .	13
2.5	$(a)$ O grafo planar $K_4$ , $(b)$ o desenho planar do grafo $K_4$ . . . . .	13
2.6	Um emparelhamento perfeito no grafo de Petersen representado pelas arestas duplas. . . . .	14
2.7	Um caminho $M$ -aumentante. . . . .	14
2.8	$(a)$ O grafo de Petersen $G$ com as arestas do emparelhamento $M$ representada pelas arestas duplas. $(b)$ O emparelhamento $M$ . $(c)$ Um caminho $M$ -aumentante $P$ . $(d)$ O emparelhamento $M' := M \Delta E(P)$ . $(e)$ O grafo de Petersen $G$ com as arestas do emparelhamento $M'$ representadas pelas arestas duplas. . . . .	15
2.9	Os itens $(a)$ , $(b)$ , $(c)$ , $(d)$ e $(e)$ representam, respectivamente, os grafos cobertos por emparelhamentos $K_{3,3}$ , $K_4$ , $\overline{C_6}$ , o grafo de Petersen e o grafo $K_2$ . . . . .	15
2.10	Exemplo de um corte justo não trivial $C := \partial(X)$ composto pelas arestas $e_1$ e $e_2$ . . . . .	16
2.11	Os grafos $G_1 := G\{X \rightarrow x\}$ e $G_2 := G\{\overline{X} \rightarrow \overline{x}\}$ são as $C$ -contrações do corte justo não trivial $C := \partial(X)$ de $G$ . . . . .	17
2.12	Grafo redutível com três componentes conexas e o conjunto $Z$ que o reduz. . . . .	18
2.13	As inversões da permutação $\pi$ estão representadas pelas letras $a$ , $b$ e $c$ . Ao todo $\pi$ tem três inversões. . . . .	19
2.14	Na imagem temos um grafo orientado $D$ , um emparelhamento perfeito $M = \{(4, 1), (3, 2), (5, 6)\}$ de $D$ e uma permutação $\pi(M, D)$ . Temos 3 inversões na permutação $\pi(M, D)$ . . . . .	20

2.15	Exemplo de construção da matriz de um grafo orientado. A matriz $A$ é gerada conforme as orientações das arestas em $D$ . . . . .	23
2.16	Um grafo pfaffiano e o valor dos sinais de cada um dos emparelhamentos perfeitos do grafo. Miranda [2009][Figura 1.1] . . . . .	24
4.1	Um fluxograma do Algoritmo MRST. A entrada deste algoritmo é um grafo $G$ . . . . .	32
4.2	O grafo orientado $D$ é composto pelas componentes conexas e pfaffianas $D_1$ e $D_2$ . . . . .	34
4.3	Um subconjunto $X$ da parte $U$ da bipartição e seus vizinhos $N(X)$ (da parte $W$ da bipartição) e um subgrafo formado por $G - (N(X) \cup X)$ . . . . .	36
4.4	O grafo $H := G - u - w$ tem emparelhamento perfeito. . . . .	36
4.5	Não existe aresta que liga $U'$ e $W''$ . . . . .	37
4.6	Um subconjunto $X$ da parte $U$ da bipartição e seus vizinhos $N(X)$ (da parte $W$ da bipartição), um corte justo $C$ com praias $X \cup N(X)$ e $V(G) - (N(X) \cup X)$ . . . . .	39
4.7	Um grafo $G$ com bipartição $\{U, W\}$ com $ U  =  W $ e o conjunto $X' \subseteq (U)$ . Um grafo $J := G - u_1 - u_2 - w_1 - w_2$ e o subconjunto $X''$ de $J$ onde $X'' \subseteq U$ . . . . .	39
4.8	Um grafo $G$ com um corte justo $C = \partial(X)$ . . . . .	40
4.9	O grafo $H_{10}$ , a 4-soma fina de três $K_{3,3}$ 's. (o quadrilátero $Q$ é mostrado em linhas pontilhadas) (Miranda [2006]). . . . .	42
4.10	(a) O desenho planar $L_1$ da parcela $G_1$ com o quadrilátero $Q_1$ , (b) O desenho planar $L_2$ da parcela $G_2$ com o quadrilátero $Q_2$ , (c) É o desenho planar $L'_1$ e mostra o quadrilátero $Q_1$ como a face externa, (d) O grafo $G$ depois da substituição do quadrilátero $Q_2$ de $L_2$ por $L'_1$ . . . . .	44
4.11	O grafo de Heawood. . . . .	45
4.12	Representação de uma iteração do algoritmo baseado no teorema de Berge para encontrar emparelhamentos máximos. (a) O grafo $G$ com as arestas que estão no emparelhamento $M$ sendo representadas pelas arestas duplas. (b) Um emparelhamento $M$ de $G$ . (c) Um caminho $M$ -aumentante $P$ . (d) O novo emparelhamento $M \Delta P$ . . . . .	47
4.13	(a) Um grafo $G$ com as arestas do emparelhamento $M$ sendo representadas pelas arestas duplas, (b) O emparelhamento $M$ e (c) Representam o grafo $D(G, U, W, M)$ e os conjuntos $S = \{s_1\}$ e $T = \{t_1\}$ . . . . .	48
4.14	Representação do grafo $A$ obtido a partir de uma busca em largura dos vértices de $S$ até encontrar um vértice de $T$ (Hopcroft & Karp [1973]). . . . .	50
4.15	. . . . .	50



4.16	(a) um grafo bipartido $G$ com bipartições $U$ e $W$ . (b) Um emparelhamento $M$ vazio de $G$ . (c) o grafo orientado $D(G, U, W, M)$ . (d) um conjunto $P$ de caminhos mais curtos maximais e vértice disjunto de $S = U$ para $T = W$ . (e) o grafo $D'$ gerado pela inversão dos arcos por onde passa os caminhos $P$ . As arestas de $D'$ correspondentes ao novo emparelhamento encontrado está sendo representado pelos arcos seccionados. . . . .	51
4.17	(a) é o grafo $D'$ do item (e) da Figura 4.16. (b) É o emparelhamento $M$ obtido pelos arcos de $D'$ com orientação de $W$ para $U$ . (c) Um conjunto $P_2$ de caminhos (que possui apenas um caminho) mais curtos maximais e vértice disjunto de $S$ para $T$ em $D'$ . (d) o grafo $D''$ obtido através da inversão dos arcos por onde passa os caminhos $P_2$ , e assim encontrando um novo emparelhamento maior que o anterior. As arestas seccionadas representam as arestas dos emparelhamentos. . . . .	52
4.18	Representação do grafo $G'$ com um corte justo $C$ com praias $X = X^+ \cup X^-$ e $Y = Y^+ \cup Y^-$ . O emparelhamento máximo $M'$ é representada pelas arestas duplas. Os vértices $u_1, w_1, u_2, w_2$ e todas as arestas incidentes a estes vértices não fazem parte do grafo $G'$ . . . . .	55
4.19	(a) Árvore binária do caso base. (b) Árvore binária do passo indutivo. . . .	56
4.20	Aplicação de busca em profundidade adaptada em $G$ para encontrar o grafo $D$ . As arestas da árvore $T$ é representada pelas arestas contínuas e as arestas de $A \setminus T$ são representadas pelas arestas pontilhadas. . . . .	60
4.21	o item (a) nos mostra as componentes fortemente conexas de $D$ que são representadas pelos conjuntos $J_i$ . O item (b) nos mostra os conjuntos $J'_i$ e os vértices de corte $z_{J_1}, z_{J_2}$ e $z_{J_3}$ que estão nas interseção dos conjuntos $J'_i$ tomados de dois em dois. . . . .	60
5.1	O item (a) corresponde a entrada da função que remove arestas inadmissíveis e o item (b) são os resultados que são as componentes obtidas resultante da remoção das arestas inadmissíveis $\{1, 7\}, \{4, 7\}, \{6, 13\}$ e $\{2, 5\}$ do grafo do item (a). . . . .	66
5.2	O item (a) corresponde a entrada da função que remove arestas inadmissíveis e o item (b) são os resultados que são as componentes obtidas resultante da remoção das arestas inadmissíveis $\{7, 8\}, \{3, 5\}$ e $\{11, 12\}$ do grafo do item (a). . . . .	66
5.3	Aplicação da decomposição em cortes justos. (a) grafo usado como entrada que possui um corte justo descrito pelo conjunto de arestas $\{\{0, 1\}, \{6, 7\}, \{12, 13\}\}$ . (b) resultado da decomposição em cortes justos do item (a). . . . .	67

5.4	Aplicação da decomposição em cortes justos. O item (a) é o grafo usado como entrada. Os grafos dos itens (c), (d), (f) e (g) são as presilhas obtidas como resultados. . . . .	68
5.5	Aplicação da decomposição 4-soma. O item (a) representa a presilha redutível usada como entrada para o algoritmo. Os grafos (b), (d), (e), (f) e (g) são as presilhas irredutíveis obtidas desta decomposição 4-soma. . . . .	69
5.6	(a) Um grafo isomorfo ao grafo de Heawood. (b) Um grafo com 14 vértices e 21 arestas, 3 regular, bipartido e não isomorfo ao grafo de Heawood. (c) Um outro grafo com 14 vértices e 21 arestas, 3 regular, bipartido e não isomorfo ao grafo de Heawood. . . . .	70
5.7	Aplicação da verificação da propriedade pfaffiana do grafo descrito no item (a), onde podemos observar as etapas de remoção das arestas inadmissíveis e a decomposição em cortes justos. . . . .	71
5.8	Continuação da verificação da propriedade pfaffiana da Figura 5.7. Podemos observar a decomposição 4-soma realizada pelo algoritmo na presilha redutível do item (d) gerando as presilhas irredutíveis (f), (g) e (h). . . . .	72
5.9	presilhas irredutíveis obtidas através da aplicação do algoritmo que verifica a propriedade pfaffiana. Como todas essas presilhas irredutíveis são pfaffianas, logo o grafo de entrada representada pelo item (a) da Figura 5.7 é pfaffiana. . . . .	73
5.10	Gráfico dos tempos de execuções para grafos obtidos com densidade constante $m = 1.5n$ em função do número de vértices. . . . .	74
5.11	Gráfico dos tempos de execuções para grafos obtidos com densidade constante $m = 2n$ em função do número de vértices. . . . .	74
5.12	Gráfico dos tempos de execuções para grafos obtidos com densidade constante $m = 4n$ em função do número de vértices. . . . .	75
5.13	Gráfico dos tempos de execuções para grafos obtidos com densidade constante $m = 10n$ em função do número de vértices. . . . .	75
5.14	Gráfico dos tempos de execuções para grafos obtidos com densidade constante $m = n^{1.5}$ em função do número de vértices. . . . .	76
5.15	Gráfico dos tempos de execuções para grafos obtidos com densidade constante $m = 0.1n^2$ em função do número de vértices. . . . .	76



# Capítulo 1

## Introdução

A ideia de usar pfaffiano na teoria de emparelhamentos é devida a Tutte. Ele descreve em seu livro Tutte [1998] como esperava encontrar uma fórmula para determinar o número de emparelhamentos de um grafo. Apesar de ele não ter sido bem sucedido em encontrar essa fórmula, ele conseguiu utilizar identidades envolvendo pfaffianos para demonstrar o seu famoso teorema que caracteriza grafos que possuem emparelhamentos perfeitos.

O Little [1975] conseguiu caracterizar os grafos bipartidos Pfaffianos. No entanto, somente em 1999, McCuaig [2004] e independentemente Robertson et al. [1999] criaram um algoritmo polinomial para o reconhecimento de grafos bipartidos Pfaffianos.

De acordo com McCuaig [2004], orientações pfaffianas foram examinadas primeiro por Kasteleyn (em seus três artigos Kasteleyn [1961], Kasteleyn [1963] e Kasteleyn [1967]), Fisher [1961] e Temperley & Fisher [1961]. Eles estavam interessados em aplicações em química como o problema do dímero. Decidir se um grafo bipartido é pfaffiano resolve vários outros problemas equivalentes como o *problema do permanente* e mais outros problemas descritos no artigo do McCuaig [2004] [sessão 4]. Dentre os problemas equivalentes de se decidir se um grafo bipartido é pfaffiano, podemos citar:

- O problema de se decidir se um dado grafo orientado tem ou não um circuito orientado de tamanho par.
- O problema da contagem das possíveis ligações duplas de carbono-carbono em hidrocarbonetos.
- Sistemas de sinais sinal-solucionável (Sign-Solvable Sign Systems).

Um problema em aberto de longa data é determinar a complexidade de decidir se um dado grafo tem orientação pfaffiana, ou até mesmo decidir se uma dada orientação de um grafo é pfaffiana. De acordo com Lovász & Plummer [1986] [página 326], apesar

de muitas condições equivalentes para a existência de uma orientação pfaffiana de um grafo, esse problema não é bem caracterizado. Nenhum algoritmo polinomial é conhecido para checar se uma dada orientação de um grafo é pfaffiana (de um grafo qualquer). Nem sabemos se esse problema está em NP. É trivial mostrar que esse problema está em CO-NP. Para provar que uma dada orientação não é pfaffiana, basta exibir dois emparelhamentos perfeitos com sinais diferentes.

Neste projeto, estudamos os grafos bipartidos. Os autores McCuaig [2004] e independentemente Robertson et al. [1999] criaram um algoritmo polinomial para verificar se um dado grafo bipartido é pfaffiano, ou seja, que possui uma orientação pfaffiana. Portanto, decidir se um grafo bipartido possui uma orientação pfaffiana está em P.

## 1.1 Motivação

A motivação é a viabilidade de solução de vários problemas equivalentes ao de se decidir se um grafo bipartido é pfaffiano, assim como McCuaig [2004] cita em seu artigo. Como estes problemas são equivalentes, a existência de um algoritmo que resolva um deles, resolve também todos os outros, bastando apenas uma redução polinomial do problema. Entre os problemas que podem ser resolvidos com o algoritmo MRST, podemos citar o problema dos Sistemas de Sinais Sinais-Solucionáveis e o problema do Dímero.

### 1.1.1 Sistemas de Sinais Sinais-Solucionáveis (Sign-Solvable Sign Systems)

Chamamos de matriz- $\{-, 0, +\}$  uma matriz cujos elementos pertencem ao conjunto  $\{-, 0, +\}$ . Sejam  $S$  uma matriz- $\{-, 0, +\}_{n \times n}$  e  $\bar{t}$  uma matriz- $\{-, 0, +\}_{n \times 1}$ . A equação  $S\bar{x} = \bar{t}$  representa o conjunto de todas as equações matriciais  $A\bar{x} = \bar{b}$  de forma que  $A$  tem o padrão de sinal de  $S$  e  $\bar{b}$  tem o padrão de sinal de  $\bar{t}$ . Dizemos que  $S\bar{x} = \bar{t}$  é um *sistema de sinais* e as equações matriciais em  $S\bar{x} = \bar{t}$  são suas *instâncias*. A equação  $S\bar{x} = \bar{t}$  é *sinal-solucionável* se todas as suas instâncias têm uma única solução e todas as soluções para as instâncias têm o mesmo padrão de sinal [McCuaig, 2004]. Vejamos o exemplo abaixo.

$$A\bar{x} = \begin{bmatrix} 3 & -2 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ -5 \end{bmatrix} = \bar{b} \Rightarrow \bar{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

$$S\bar{x} = \begin{bmatrix} + & - \\ - & - \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ - \end{bmatrix} = \bar{t} \Rightarrow \bar{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} + \\ + \end{bmatrix}$$

Para o sistema de sinal do exemplo, dada uma instância arbitrária

$$\begin{bmatrix} a & -b \\ -c & -d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ -e \end{bmatrix}$$

onde  $a, b, c, d$  e  $e$  são números positivos. Podemos utilizar a regra de Cramer para encontrar a solução, como mostra o exemplo abaixo:

$$x = \frac{\begin{vmatrix} 0 & -b \\ -e & -d \end{vmatrix}}{\begin{vmatrix} a & -b \\ -c & -d \end{vmatrix}} = \frac{-be}{-ad - bc} > 0 \quad y = \frac{\begin{vmatrix} a & 0 \\ -c & -e \end{vmatrix}}{\begin{vmatrix} a & -b \\ -c & -d \end{vmatrix}} = \frac{-ae}{-ad - bc} > 0$$

Note que todos os determinantes são negativos para todas as escolhas de  $a, b, c, d$  e  $e$ . Consequentemente, toda instância tem solução única, e ambas as entradas da solução são positivas. Portanto, o sistema de sinal é *sinal-solucionável*. Todos esses determinantes negativos são devidos ao fato de que as matrizes em questão são matrizes *sinais-não-singulares*. Determinar se uma matriz é sinal-não-singular é uma versão do problema de se decidir se um grafo bipartido é pfaffiano.

#### 1.1.1.1 Relações Entre Variáveis Econômicas

Dadas duas variáveis econômicas, nós podemos não saber exatamente como uma afeta a outra, mas podemos saber a relação qualitativa entre elas, ou seja, se o aumento de uma variável implica no aumento, decaimento ou na independência de outra. Com essas situações podemos chegar na pergunta principal (que foi feita pelo autor Samuelson [1971], em seu livro “*Foundations of Economic Analysis*” enquanto pesquisava por este tipo de problema em economia): Dadas algumas relações qualitativas entre as variáveis, quais outras relações qualitativas podem ser derivadas? O exemplo abaixo mostra um caso de análise de relações qualitativas para o comércio de café. Para o comércio do café temos cinco parâmetros

- $t$  é uma medida do gosto das pessoas por café.
- $p(t)$  é o preço por saca em função de  $t$ .
- $V(t)$  é o volume de sacas vendidas por dia em função de  $t$ .
- $S(p)$  é o suprimento de sacas por dia em função de  $p$ .
- $D(p, t)$  é a demanda de sacas por dia em função de  $p$  e  $t$ .

Suponha o seguinte.

- $\frac{\partial S}{\partial p} > 0$ , indica que o suprimento de café aumenta à medida que o preço aumenta.
- $\frac{\partial D}{\partial p} < 0$ , indica que a demanda por café diminui à medida que o preço do café aumenta.
- $\frac{\partial D}{\partial t} > 0$  indica que a demanda aumenta à medida que aumenta o gosto das pessoas por café.

Mostraremos que essas suposições implicam em  $\frac{\partial p}{\partial t} > 0$  e  $\frac{\partial V}{\partial t} > 0$ , portanto o preço  $p$  e o volume  $V$  aumenta à medida que o gosto das pessoas pelo café aumenta. O volume é sempre menor ou igual ao suprimento e a demanda, ou seja  $V(t) \leq S(p)$  e  $V(t) \leq D(p, t)$ . Suponha que o volume é (estritamente) menor que o suprimento. Então o preço irá cair, causando o aumento de  $D(p, t)$  por suposição; e o volume será também menor que a demanda. Agora com o volume menor que ambos o suprimento e a demanda, o volume irá crescer. Similarmente, podemos mostrar que o volume aumenta quando é menor que a demanda. O volume para de crescer e o mercado fica em equilíbrio quando o suprimento, demanda e o volume são todos iguais, ou seja, quando

$$S(p) = V(t) = D(p, t).$$

Tomando derivadas parciais em relação a  $t$ , obtemos

$$\frac{\partial S}{\partial p} \frac{\partial p}{\partial t} = \frac{\partial V}{\partial t} = \frac{\partial D}{\partial p} \frac{\partial p}{\partial t} + \frac{\partial D}{\partial t}.$$

Equivalentemente,

$$\begin{bmatrix} \frac{\partial S}{\partial p} & -1 \\ \frac{\partial D}{\partial p} & -1 \end{bmatrix} \begin{bmatrix} \frac{\partial p}{\partial t} \\ \frac{\partial V}{\partial t} \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{\partial D}{\partial t} \end{bmatrix}.$$

Pela nossa suposição  $\frac{\partial S}{\partial p} > 0$ ,  $\frac{\partial D}{\partial p} < 0$  e  $\frac{\partial D}{\partial t} > 0$ , temos uma instância para um sistema de sinais

$$\begin{bmatrix} + & - \\ - & - \end{bmatrix} \begin{bmatrix} \frac{\partial p}{\partial t} \\ \frac{\partial V}{\partial t} \end{bmatrix} = \begin{bmatrix} 0 \\ - \end{bmatrix}.$$

Como no problema anterior, este também é um sistema sinal-solucionável, e para cada instância, ambas as entradas da solução são positivas. Consequentemente o preço e o volume aumentam à medida em que o gosto das pessoas pelo café aumenta.

### 1.1.2 O Problema do Dímero (The Dimer Problem)

Um exemplo do problema do dímero foi descrito por McCuaig no seu artigo McCuaig [2004]. Uma ligação simples em uma molécula consiste em dois elétrons compartilhados por dois átomos, e uma ligação dupla consiste em quatro elétrons compartilhados por dois átomos. Ambos os hidrocarbonetos no exemplo da Figura 1.1 têm 16 ligações simples carbono-carbono e 10 ligações simples carbono-hidrogênio. Estes hidrocarbonetos são chamados de aromáticos porque cada um tem uma ligação  $\pi$  que consiste em 14 elétrons compartilhados por 14 átomos de carbono. Cada hidrocarboneto tem um grafo associado onde cada átomo de carbono do hidrocarboneto é representado por um vértice e cada ligação simples carbono-carbono, por uma aresta. A forma ressonante de qualquer hidrocarboneto aromático é definido da seguinte maneira. Escolha um emparelhamento perfeito do grafo associado. Para cada uma das 7 ligações simples carbono-carbono correspondentes, adiciona-se dois elétrons da ligação  $\pi$  para obter uma ligação dupla carbono-carbono. Um hidrocarboneto aromático é dito ser um híbrido da sua forma ressonante. Devemos notar que a forma ressonante são moléculas teóricas, ou seja, elas não podem ser encontradas na natureza. Nós estamos descrevendo algo que existe como um híbrido de coisas que não existem, ou como John D. Roberts [Morrison & Boyd, 1973] coloca, estamos descrevendo um rinoceronte como um cruzamento entre um dragão e um unicórnio.

A soma das forças das ligações é maior para um hidrocarboneto aromático do que para qualquer um das suas formas ressonantes. A diferença é chamada de energia ressonante da ligação  $\pi$ . A energia ressonante é uma função crescente no número de formas ressonantes, que é igual ao número  $m$  de emparelhamentos perfeitos do grafo associado. O cálculo de  $m$  pode ser feito de forma eficiente se o grafo tem uma orientação pfaffiana. Um exemplo deste problema é mostrado na Figura 1.1.

A natureza das ligações  $\pi$  faz com que os hidrocarbonetos aromáticos sejam fisicamente planares, resultando em grafos associados planares. Convenientemente, todos os grafos planares têm orientações pfaffianas, isso foi mostrado por Kasteleyn [1963].

## 1.2 Contextualização do Problema

O problema resolvido neste projeto é a implementação do Algoritmo MRST que dado um grafo bipartido  $G$  como entrada, devemos decidir se  $G$  é pfaffiano. Em outras palavras, o algoritmo MRST responde sim caso o grafo bipartido  $G$  seja pfaffiano, ou não caso contrário.



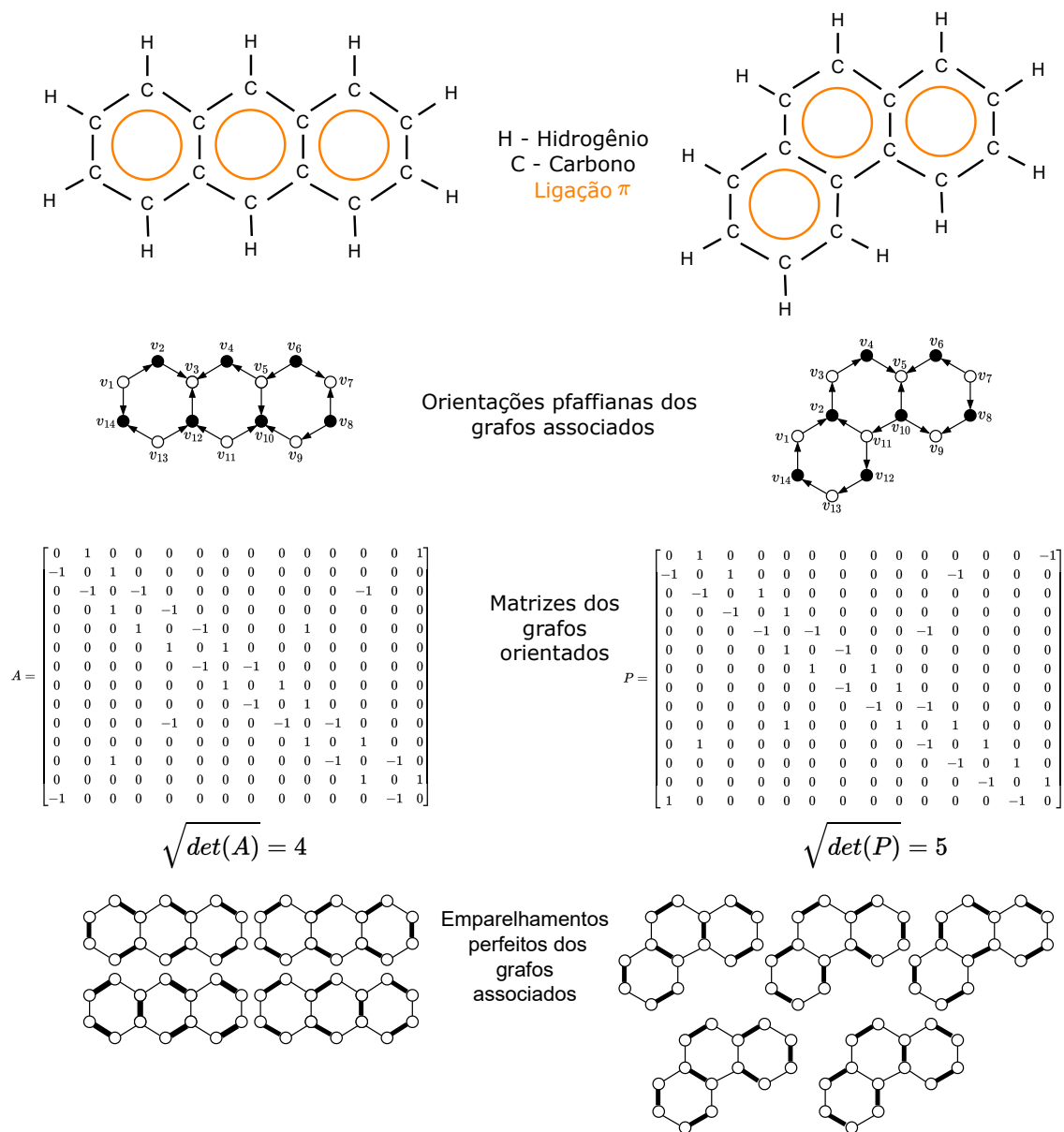


Figura 1.1. Aplicação em química orgânica McCuaig [2004][Fig 7].

## 1.3 Algoritmo

O algoritmo estudado que chamaremos de MRST por causa de seus criadores McCuaig, Robertson, Seymour e Thomas, recebe um grafo bipartido  $G$  como entrada e determina se  $G$  é pfaffiano. O algoritmo reduz o problema de se decidir se um grafo bipartido  $G$  é pfaffiano ao problema de decidir se os grafos obtidos nas decomposições de  $G$  são pfaffianos. Por exemplo, o algoritmo reduz o problema de se decidir se o grafo mostrado na Figura 1.2 é pfaffiano ao problema de se decidir se os cinco grafos menores mostrados são pfaffianos.

O Algoritmo será detalhado no Capítulo 4. Definiremos o que são grafos pfaffianos, as decomposições e os grafos que obtemos em cada uma das decomposições nas próximas seções.

## 1.4 Objetivo

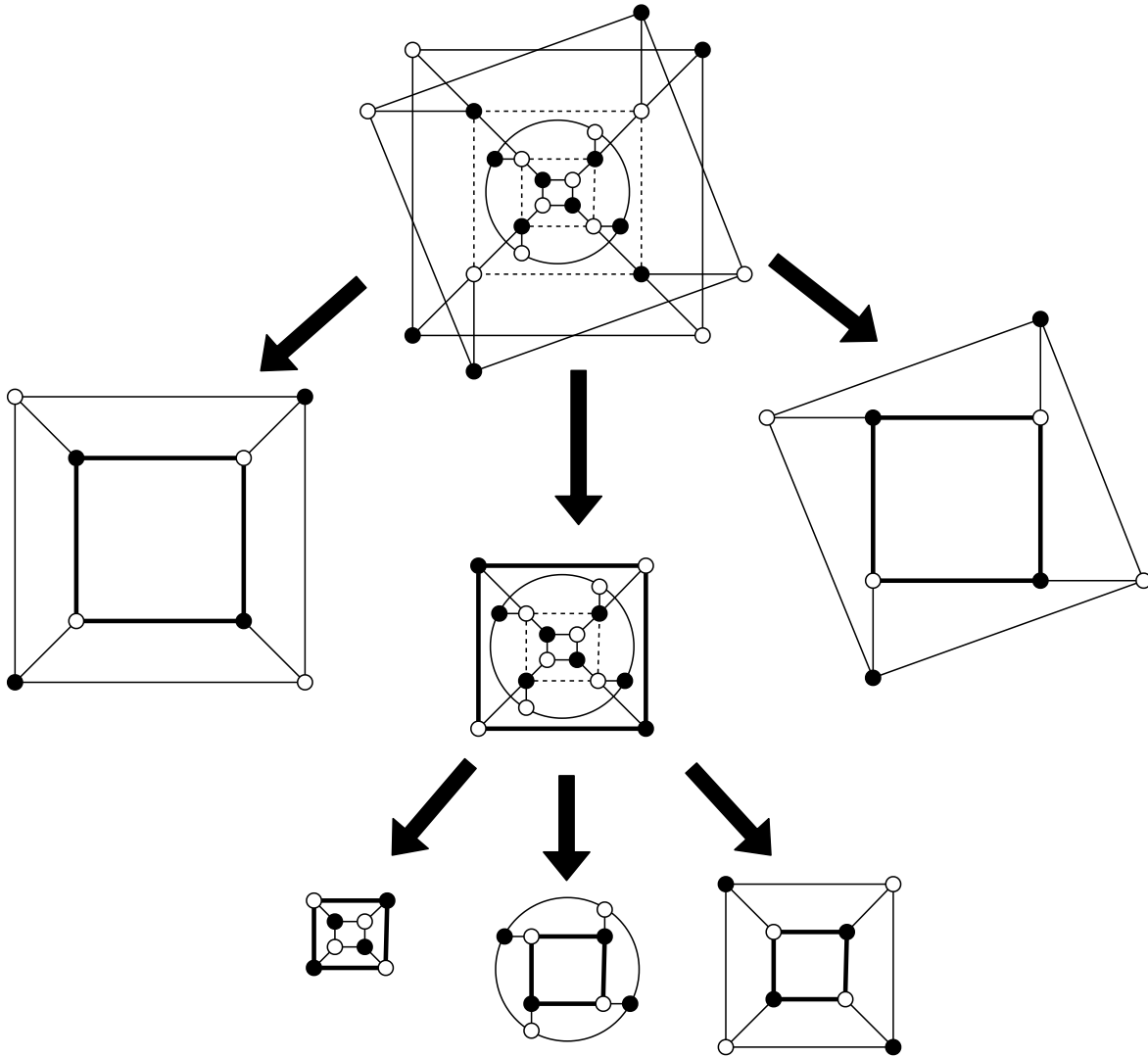
Um algoritmo que decide se um grafo é bipartido pfaffiano é uma ferramenta que tem várias aplicações em diversas áreas, e com um grande potencial para auxiliar pesquisas e aplicações de forma a trazer mais conhecimento para a sociedade. Dada a alta complexidade do tema relativo aos grafos bipartidos pfaffianos e a ausência de uma implementação disponível, os objetivos são fornecer para a sociedade uma explicação acessível e com exemplos do tema de grafos bipartidos pfaffianos e uma implementação eficiente do algoritmo MRST.

### 1.4.1 Objetivos Específicos

- Implementação do algoritmo MRST que é capaz de reconhecer grafos bipartidos pfaffianos de forma eficiente.
- Explicação do funcionamento do algoritmo MRST e seu tempo de execução.
- Explicação de teoremas sobre algumas propriedades dos grafos bipartidos pfaffianos.

## 1.5 Organização do Texto

Este projeto está dividido em 4 capítulos. O Capítulo 1 é a introdução deste projeto, onde são apresentadas as motivações, o problema, os objetivos e uma visão muito superficial do algoritmo para a solução do problema de se decidir se um grafo bipartido é pfaffiano. No Capítulo 2 são apresentados os conceitos básicos que foram usados



**Figura 1.2.** Redução a presilhas irreduzíveis. As arestas seccionadas se ligam à quádrupla de vértices que reduzem o grafo [Miranda, 2006][página 6. Fig 1.1]).

no desenvolvimento do Algoritmo MRST. No Capítulo 3 é apresentado uma revisão da literatura, onde são descritos alguns resultados conhecidos sobre o tema referente à implementação do algoritmo MRST. O Capítulo 4 descreve a metodologia usada, a descrição do porquê o algoritmo MRST funciona, a descrição da implementação do algoritmo MRST, como foram feitos os testes. O Capítulo 5 contém os resultados obtidos.

# Capítulo 2

## Conceitos básicos

Muitas das definições descritas neste capítulo foram baseadas no livro do Bondy & Murty [2008]. Caso você esteja familiarizado com este livro, você pode continuar a leitura a partir da seção de cortes justos, onde serão apresentados novos conceitos.

### 2.1 Grafos

Com o intuito de solucionar problemas, precisamos de ferramentas matemáticas que facilitem a sua análise. Para problemas que consistem em elementos e as relações entre estes elementos, podemos usar os grafos. Os grafos nos permitem abstrair estes tipos de problemas e nos concentrar nas suas soluções. Alguns exemplos de problemas nos quais podemos aplicar o conceito de grafos são: situações envolvendo cidades e estradas, como por exemplo, encontrar o caminho mínimo entre duas cidades quaisquer; situações envolvendo pessoas e amizades em uma rede social; ou situações envolvendo quartos de hotéis, pessoas e as preferências das pessoas pelos quartos. Em todos estes exemplos, podemos representar os elementos destes problemas como sendo os vértices de um grafo e as relações entre os elementos como sendo as arestas de um grafo. Para o problema de se encontrar o caminho mínimo entre duas cidades, as cidades são os vértices e as estradas as arestas de um grafo. Nos *diagramas de grafos*, representamos um vértice com um pequeno círculo e representamos uma aresta com uma linha que se conecta com dois círculos. Podemos ver um exemplo de um diagrama de um grafo na Figura 2.1.

Um *grafo*  $G$  é um par ordenado  $(V(G), E(G))$  que consiste em um conjunto  $V(G)$  de *vértices* e um conjunto  $E(G)$  de *arestas*, disjunto de  $V(G)$ , e com uma função de incidência  $\psi_G$  que associa cada aresta de  $G$  com um par de vértices de  $G$ . Seja  $e$  uma aresta, e  $u$  e  $v$  vértices de forma que  $\psi_G(e) = \{u, v\}$ . Adotamos as convenções

para os conceitos de adjacente, aresta paralela, múltipla, laços, incidente, vizinhos, grafo simples, grafo nulo, grafo trivial e componentes conexas do livro Bondy & Murty [2008] e sugerimos aos leitores que consultem este livro.

Podemos ver um exemplo de um grafo cuja descrição está abaixo e cujo diagrama está na Figura 2.1.

$$G = (V(G), E(G))$$

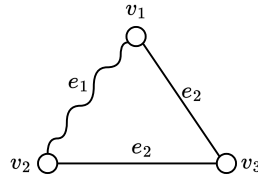
onde  $V(G)$  e  $E(G)$  são definidos da seguinte forma:

$$V(G) = \{v_1, v_2, v_3\}$$

$$E(G) = \{e_1, e_2, e_3\}$$

e a função  $\psi_G$  é definida como:

$$\psi_G(e_1) = \{v_1, v_2\} \quad \psi_G(e_2) = \{v_2, v_3\} \quad \psi_G(e_3) = \{v_3, v_2\}$$

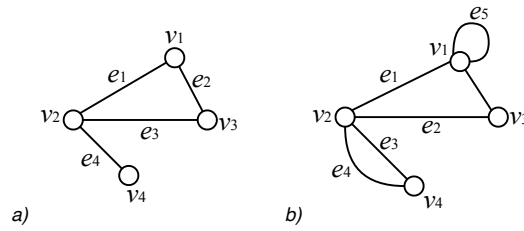


**Figura 2.1.** Diagrama do grafo  $G$ .

Os grafos simples podem ter a sua notação simplificada. Para essa simplificação podemos definir o conjunto das arestas do grafo  $G$  do exemplo acima como sendo o conjunto dos pares de vértices correspondentes às arestas de  $G$ , de forma que  $E(G)$  é substituído por  $\{\psi_G(e_1), \psi_G(e_2), \psi_G(e_3)\}$ . Neste caso podemos omitir a função  $\psi$  para descrever um grafo, assim cada aresta pode ser descrita pelos seus dois extremos. Usaremos como exemplo o grafo da Figura 2.1. Desta forma a notação será da seguinte maneira:

$$V(G) = \{v_1, v_2, v_3\}$$

$$E(G) = \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_2\}\}$$



**Figura 2.2.** a) Um grafo simples sem laços e sem arestas paralelas. b) Um grafo não simples com arestas paralelas  $e_3$  e  $e_4$  e um laço  $e_5$ .

### 2.1.1 Grafo orientado

Para alguns tipos de problemas, o conceito de grafos como vimos até agora não é o suficiente para a sua modelagem. Alguns casos, precisamos da informação do sentido das arestas. Por exemplo, se quisermos um grafo que nos permita otimizar o trânsito de uma cidade, precisamos de informações do sentido de uma via, podendo ser mão dupla ou mão única. Uma forma de representar este problema é usar as esquinas (ligação de 3 ou mais ruas) como sendo os vértices e as ruas como sendo as arestas. Para modelarmos este tipo de problema, utilizamos os grafos orientados. Diferente dos grafos não orientados, os grafos orientados possuem orientação nas suas arestas.

Um *grafo orientado*  $D$  é formado por um par ordenado  $(V(D), E(D))$  e uma função  $\psi_D$ , onde  $V(D)$  representa o conjunto dos vértices e  $E(D)$  disjuncto de  $V(D)$  representa o conjunto das arestas orientadas, também chamadas de *arcos*, junto com uma *função de incidência*  $\psi_D$  que associa cada aresta de  $E(D)$  com um par ordenado de vértices de  $V(D)$  não necessariamente distintos. Usamos as definições de aresta orientada, dominar, vizinhos de entrada, vizinhos de saída e extremos de arestas do livro do Bondy & Murty [2008]. A partir de um grafo não orientado  $G$  podemos criar um grafo orientado  $D$  usando os mesmos conjuntos de vértices e arestas de  $G$ , substituindo cada aresta de  $G$  por um arco com os mesmos extremos em  $D$ , usando uma das duas possíveis orientações para o arco, criando assim uma *orientação*  $D$  do grafo  $G$ .

Um *grafo orientado estrito* é um grafo orientado sem arestas múltiplas. Quando usamos a notação de par ordenado  $(u, v)$  temos o sentido em que a aresta está orientada. Diferente da notação de conjunto  $\{u, v\}$  em que a ordem dos vértices que descrevem as arestas não importam.

Podemos ver um exemplo de um grafo orientado cuja descrição está abaixo e cujo diagrama está na Figura 2.3.

$$D = (V(D), E(D))$$

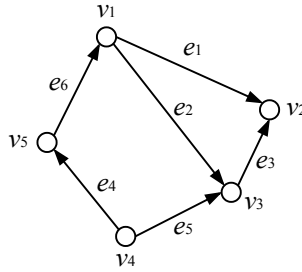
onde  $V(D)$  e  $E(D)$  são definidos como:

$$V(D) = \{v_1, v_2, v_3, v_4, v_5\}$$

$$E(D) = \{e_1, e_2, e_3, e_4, e_5, e_6\}$$

e a função  $\psi_D$  é definida da seguinte maneira:

$$\begin{aligned} \psi_D(e_1) &= (v_1, v_2) & \psi_D(e_2) &= (v_1, v_3) & \psi_D(e_3) &= (v_3, v_2) & \psi_D(e_4) &= (v_4, v_5) \\ \psi_D(e_5) &= (v_4, v_3) & \psi_D(e_6) &= (v_5, v_1) \end{aligned}$$



**Figura 2.3.** Exemplo de grafo orientado.

Para simplificar a notação dos grafos orientados estritos podemos omitir a função  $\psi_D$ , substituindo as arestas de  $E(D)$  representadas pelos seus elementos por imagens da função  $\psi_D$  aplicadas em cada uma das arestas de  $E(D)$ , gerando um conjunto de arestas  $E(D)$  de pares ordenados de vértices. Com isso a definição do grafo orientado da Figura 2.3 será:

$$V(D) = \{v_1, v_2, v_3, v_4, v_5\}$$

$$E(D) = \{(v_1, v_2), (v_1, v_3), (v_3, v_2), (v_5, v_1), (v_4, v_5), (v_4, v_3)\}$$

Podemos fazer ainda mais simplificações usando a notação  $v_i v_j$  para representar uma aresta (orientada ou não) com os extremos  $v_i$  e  $v_j$ . A descrição do grafo  $D$  acima ficará com a seguinte representação:

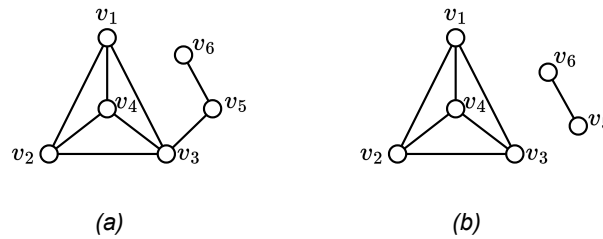
$$V(D) = \{v_1, v_2, v_3, v_4, v_5\}$$

$$E(D) = \{v_1 v_2, v_1 v_3, v_3 v_2, v_5 v_1, v_4 v_5, v_4 v_3\}$$

### 2.1.2 Famílias Especiais de Grafos

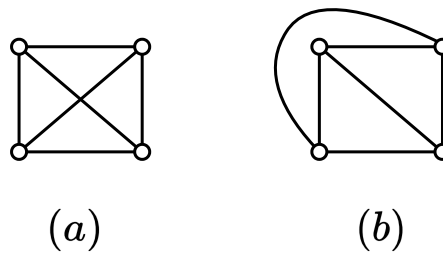
Alguns tipos de grafos desempenham papéis importantes na teoria de grafos. Um grafo  $G$  é bipartido se seu conjunto de vértices pode ser particionado em dois subconjuntos  $X$  e  $Y$  de modo que cada aresta tenha um extremo em  $X$  e outro em  $Y$ . Essa partição  $\{X, Y\}$  é chamada de *bipartição* do grafo  $G$ , e  $X$  e  $Y$  são suas *partes*. Chamamos um grafo  $G$  com bipartição  $\{X, Y\}$  de  $G[X, Y]$ .

Um grafo  $G$  é *conexo* se, e somente se, existe um caminho entre quaisquer pares de vértices, caso contrário o grafo é *desconexo*. A Figura 2.4 nos mostra dois grafos, o grafo do item (a) é um grafo conexo, e o grafo do item (b) é desconexo e suas componentes conexas são os subgrafos representados pelos vértices  $\{v_1, v_2, v_3, v_4\}$  e  $\{v_5, v_6\}$ . Note que não existe qualquer caminho entre os vértices  $\{v_1, v_2, v_3, v_4\}$  e os vértices  $\{v_5, v_6\}$  do grafo do item (b) da Figura 2.4.



**Figura 2.4.** (a) Um grafo conexo, e (b) um grafo desconexo.

Um grafo é *planar* se, e somente se, ele pode ser desenhado no plano de forma que suas arestas não se cruzem. Por exemplo, o grafo da Figura 2.5 é planar. Tal desenho de um grafo no plano é chamado de *desenho planar*.



**Figura 2.5.** (a) O grafo planar  $K_4$ , (b) o desenho planar do grafo  $K_4$ .

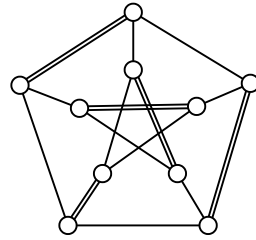
### 2.1.3 Emparelhamento

Chamamos um conjunto  $M$  de arestas de um grafo  $G$  de *emparelhamento* se  $M$  é um conjunto de arestas não adjacentes, ou seja, não existem duas arestas de  $M$  com um dos



extremos em comum. Cada vértice que incide em uma aresta de  $M$  é dito ser *coberto* ou *emparelhado* por  $M$ . Um *emparelhamento perfeito* é aquele que cobre todos os vértices de um grafo e um *emparelhamento máximo* é aquele que cobre tantos vértices quanto possível. O conjunto  $\mathcal{M}(G)$  é o conjunto de todos os emparelhamentos perfeitos de  $G$ .

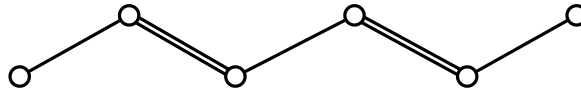
Um exemplo de emparelhamento perfeito no grafo de Petersen é mostrados na Figura 2.6 respectivamente.



**Figura 2.6.** Um emparelhamento perfeito no grafo de Petersen representado pelas arestas duplas.

### 2.1.3.1 Caminhos $M$ -alternados

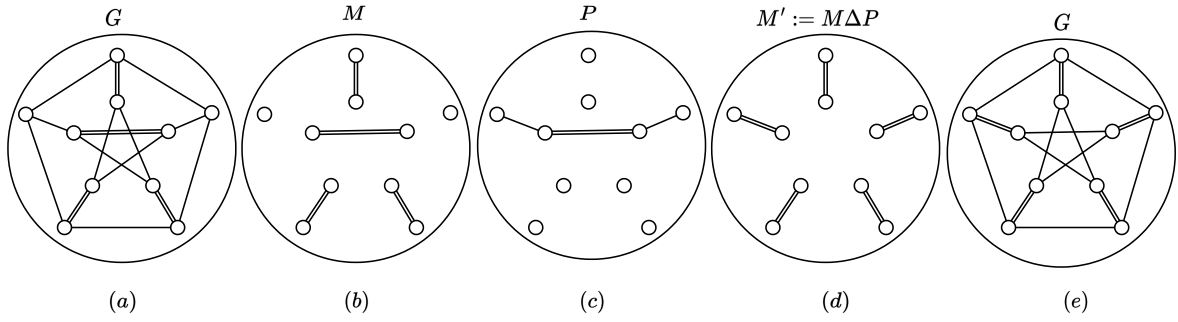
Seja  $M$  um emparelhamento em um grafo  $G$ . Um *caminho  $M$ -alternado* ou um *ciclo  $M$ -alternado* em  $G$  é um caminho ou ciclo tal que um percurso dele encontra alternadamente arestas de  $M$  e de  $E(G) \setminus M$ .



**Figura 2.7.** Um caminho  $M$ -aumentante.

Um caminho  $M$ -alternado que começa e termina com vértices não cobertos por  $M$  é um caminho  $M$ -aumentante, assim como é mostrado na Figura 2.7. Um caminho  $M$ -aumentante  $P$  nos permite aumentar o emparelhamento  $M$  deste caminho para um emparelhamento maior, trocando as arestas de  $P \cap M$  pelas arestas de  $P - M$ . A Figura 2.8 nos mostra o grafo de Petersen  $G$ , um emparelhamento  $M$  de  $G$ , o caminho  $M$ -aumentante  $P$ , e um emparelhamento maior  $M'$  obtido através do emparelhamento  $M$  e o caminho  $M$ -aumentante  $P$ .

**Definição 2.1.1.** Sejam  $M_1$  e  $M_2$  conjuntos de arestas. O símbolo  $\Delta$  é o operador de diferença simétrica, O conjunto  $M_1 \Delta M_2 = (M_1 \setminus M_2) \cup (M_2 \setminus M_1)$ . Chamamos  $M_1 \Delta M_2$  de *diferença simétrica* entre  $M_1$  e  $M_2$ .



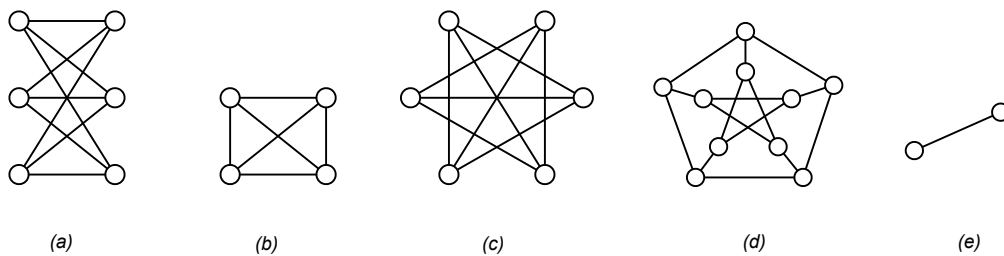
**Figura 2.8.** (a) O grafo de Petersen  $G$  com as arestas do emparelhamento  $M$  representada pelas arestas duplas. (b) O emparelhamento  $M$ . (c) Um caminho  $M$ -aumentante  $P$ . (d) O emparelhamento  $M' := M \Delta E(P)$ . (e) O grafo de Petersen  $G$  com as arestas do emparelhamento  $M'$  representadas pelas arestas duplas.

## 2.2 Grafos Cobertos por Emparelhamentos

Seja  $G$  um grafo que possui algum emparelhamento perfeito. Uma aresta  $e \in E(G)$  é *admissível*, se existe um emparelhamento perfeito  $M \in \mathcal{M}(G)$  que contém  $e$ , caso contrário  $e$  é *inadmissível*.

Um grafo  $G$  é *coberto por emparelhamentos* se  $G$  é um grafo conexo, tem pelo menos dois vértices e todas as suas arestas são admissíveis. Por exemplo, os grafos  $K_{3,3}$ ,  $K_4$ ,  $\overline{C_6}$ , o grafo de Petersen e o grafo  $K_2$ , vistos na Figura 2.9, são cobertos por emparelhamentos.

**Definição 2.2.1.** Sejam  $G = (V, E)$  um grafo e  $X$  um subconjunto de  $V(G)$ . O conjunto  $\overline{X}$  representa o conjunto  $V(G) \setminus X$ . O grafo  $\overline{G} := (V(G), R \setminus E(G))$  é formado pelo par  $(V(G), R \setminus E(G))$ , onde  $R$  é o conjunto das arestas de um grafo completo com conjunto de vértices igual a  $V(G)$ .



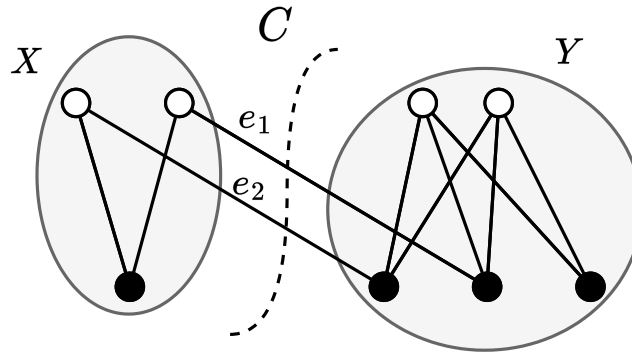
**Figura 2.9.** Os itens (a), (b), (c), (d) e (e) representam, respectivamente, os grafos cobertos por emparelhamentos  $K_{3,3}$ ,  $K_4$ ,  $\overline{C_6}$ , o grafo de Petersen e o grafo  $K_2$ .

### 2.2.1 Cortes

Sejam  $G = (V, E)$  um grafo e  $X$  um subconjunto não vazio de  $V(G)$ . Um corte  $C = \partial(X)$  é composto por todas as arestas com extremos em  $X$  e em  $Y := V(G) \setminus X$ . Um corte é definido por um conjunto de vértices, e ele divide os vértices de um grafo em dois conjuntos, que são os conjuntos  $X$  e  $Y$ . Os conjuntos  $X$  e  $Y$  são as praias do corte  $C$ . Podemos definir um corte usando qualquer uma das suas praias. Um corte é *trivial* se qualquer uma das suas praias tiverem apenas um vértice.

### 2.2.2 Cortes Justos

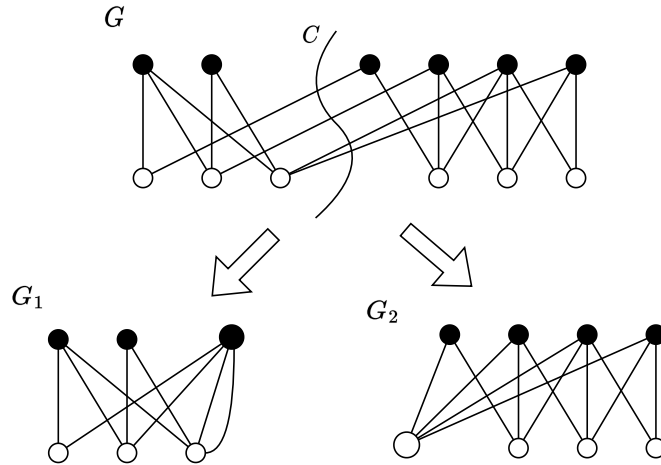
Sejam  $G = (V, E)$  um grafo coberto por emparelhamentos e  $X$  um subconjunto de  $V$ . Seja  $C := \partial(X)$  um corte de  $G$ . Dizemos que  $C$  é um *corte justo* de  $G$ , se e somente se, para cada emparelhamento perfeito  $M$  de  $G$ , o emparelhamento  $M$  tem precisamente uma aresta em comum com  $C$ . A Figura 2.10 nos mostra um exemplo de um corte justo não trivial.



**Figura 2.10.** Exemplo de um corte justo não trivial  $C := \partial(X)$  composto pelas arestas  $e_1$  e  $e_2$ .

Um grafo  $H$  obtido a partir de  $G$  pela contração de todos os vértices em  $X$  a um único vértice  $x$  e subsequente remoção dos laços criados é chamado de *C-contração*, e essa *C-contração*  $H$  é representada por  $H := G\{X \rightarrow x\}$ . As duas *C-contrações* de  $G$  são  $G\{X \rightarrow x\}$  e  $G\{\bar{X} \rightarrow \bar{x}\}$ , assim como é mostrado na Figura 2.11.

Uma *presilha* é um grafo bipartido, coberto por emparelhamentos e livre de cortes justos não triviais, e um *tijolo* é um grafo não bipartido, coberto por emparelhamentos e livre de cortes justos. As *presilhas simples* são presilhas que não possuem arestas múltiplas, ou seja, não possuem laços e nem arestas paralelas. De acordo com de Carvalho et al. [2005], os grafos  $K_4$ ,  $\overline{C_6}$  e o grafo de Petersen são exemplos de tijolos, e o grafo  $K_{3,3}$  é um exemplo de presilha (Figura 2.9).



**Figura 2.11.** Os grafos  $G_1 := G\{X \rightarrow x\}$  e  $G_2 := G\{\bar{X} \rightarrow \bar{x}\}$  são as  $C$ -contrações do corte justo não trivial  $C := \partial(X)$  de  $G$ .

### 2.2.3 Decomposição em Cortes Justos

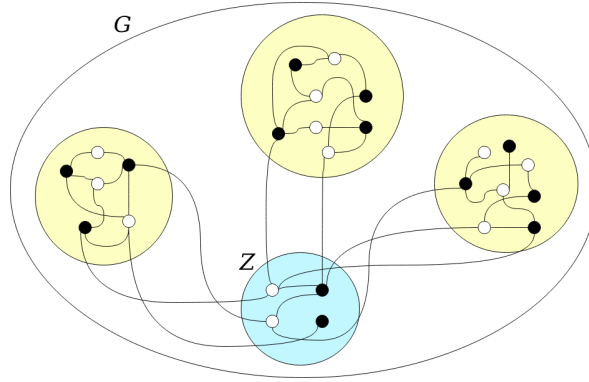
Sejam  $G$  um grafo bipartido coberto por emparelhamentos e  $C$  um corte justo não trivial de  $G$ . Sejam  $G_1$  e  $G_2$  as duas  $C$ -contrações de  $G$  a menos de arestas múltiplas. Temos que, tanto  $G_1$  quanto  $G_2$  são cobertos por emparelhamentos. Se  $G_1$  e/ou  $G_2$  tem um corte justo não trivial, então repetimos este procedimento. No fim, teremos uma lista de presilhas. No caso de grafos não bipartidos, teríamos uma lista de presilhas e tijolos. Chamamos este procedimento de *Decomposição em cortes justos* e a lista de presilhas obtidas é o resultado deste procedimento [Lovász, 1987]. Decomposições em cortes justos de grafos bipartidos resultam somente em presilhas (bipartidos). Decomposições em cortes justos de grafos não bipartidos resultam em presilhas e tijolos (não bipartidos).

## 2.3 Grafo Redutível

Seja  $G$  um grafo coberto por emparelhamentos com bipartição  $\{U, W\}$ . Uma quádrupla  $Z$  de quatro vértices de  $G$  *reduz*  $G$  se:

- $Z$  contém dois vértices em  $U$  e dois vértices em  $W$ .
- $G - Z$  consiste em três ou mais componentes conexas.

Um grafo bipartido coberto por emparelhamentos  $G$  é *reduzível* se alguma quádrupla de vértices de  $G$  reduz  $G$  e é *irreduzível* caso contrário, assim como é mostrado na Figura 2.12.



**Figura 2.12.** Grafo redutível com três componentes conexas e o conjunto  $Z$  que o reduz.

## 2.4 Permutações

Uma *enumeração* ou *listagem* dos elementos de um conjunto é uma sequência destes elementos nomeados e sem repetições. Para um conjunto com seis elementos, uma enumeração destes elementos poderia ser a sequência  $(a, b, c, d, e, f)$ ,  $(1, 2, 3, 4, 5, 6)$  ou  $(e_1, e_2, e_3, e_4, e_5, e_6)$ . Uma *permutação* mapeia um conjunto nele mesmo. Ao permutar a enumeração de um conjunto, geramos uma nova ordem dos elementos, sem repetições. Uma permutação da enumeração  $(e_1, e_2, e_3, e_4, e_5, e_6)$  poderia ser a enumeração  $(e_1, e_3, e_2, e_4, e_6, e_5)$ . Note que apenas a ordem dos elementos é alterada.

Seja  $S$  um conjunto. Usamos a função bijetora  $\pi : S \rightarrow S$  para descrevermos uma permutação  $\pi$ . Uma permutação  $\pi$  é representada na notação abaixo:

$$\pi := \begin{pmatrix} e_1 & e_2 & e_3 & e_4 & \dots & e_n \\ \pi(e_1) & \pi(e_2) & \pi(e_3) & \pi(e_4) & \dots & \pi(e_n) \end{pmatrix}$$

onde a primeira linha descreve uma enumeração e a segunda linha representa a imagem da função  $\pi$  calculada sobre os respectivos elementos da primeira linha. Vemos que a permutação  $\pi$  mapeia uma enumeração em outra. A permutação  $\pi$  é uma das possíveis permutações dos elementos do conjunto  $S$ .

Dada uma enumeração base  $(a_1, a_2, a_3, \dots, a_n)$ , dois elementos  $a_i$  e  $a_j$  são uma *inversão* em uma permutação  $\pi$  se  $i < j$  e se  $a_j$  precede  $a_i$  na sequência  $(\pi(a_1), \pi(a_2), \pi(a_3), \dots, \pi(a_n))$ . O *signal*  $\text{sgn}(\pi)$  de uma permutação  $\pi$  é igual a  $(-1)^t$ , onde  $t$  é o número de inversões existentes na permutação (ver a Figura 2.13). Observe que, como a primeira linha de uma permutação descreve uma enumeração base de um conjunto, a permutação descrita por:

$$\pi = \begin{pmatrix} 1 & 5 & 2 & 3 & 4 \\ 1 & 5 & 2 & 4 & 3 \end{pmatrix}$$

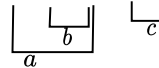
apresenta apenas uma inversão nos elementos 3 e 4 na sequência  $(1, 5, 2, 4, 3)$ . Os elementos 5 e 2 apesar de intuitivamente parecer estarem invertidos, eles não estão, pois, a enumeração base  $(1, 5, 2, 3, 4)$  nos mostra a ordem correta. Para podermos visualizar melhor essas inversões, podemos chamar o elemento 1 de  $a$ , 5 de  $b$ , 2 de  $c$ , 3 de  $d$  e o 4 de  $e$ , assim como é mostrado abaixo:

$$\pi = \begin{pmatrix} a & b & c & d & e \\ a & b & c & e & d \end{pmatrix}.$$

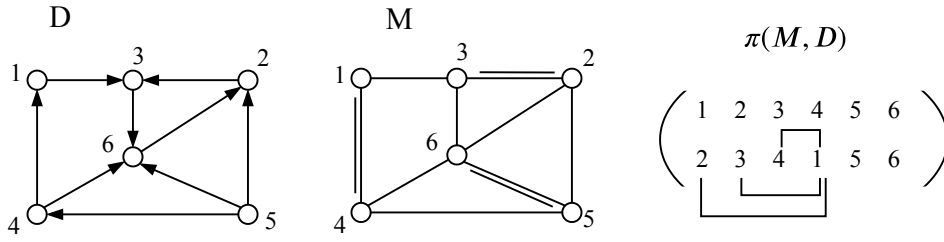
Seja  $D$  um grafo orientado com emparelhamento perfeito. Fixe uma enumeração  $(u_1, u_2, u_3, \dots, u_n)$  de  $V(D)$ . Seja  $M := \{(v_1, w_1), (v_2, w_2), \dots, (v_{n/2}, w_{n/2})\}$  um emparelhamento perfeito de  $D$ . Fixando uma enumeração arbitrária das arestas de  $M$  como sendo  $((v_1, w_1), (v_2, w_2), \dots, (v_{n/2}, w_{n/2}))$ , podemos descrever uma permutação  $\pi(M, D)$ :

$$\pi(M, D) := \begin{pmatrix} u_1 & u_2 & u_3 & u_4 & \dots & u_{n-1} & u_n \\ v_1 & w_1 & v_2 & w_2 & \dots & v_{n/2} & w_{n/2} \end{pmatrix}$$

Definimos o sinal do emparelhamento  $M$  em  $D$ , chamado de  $\text{sgn}(M, D)$ , como sendo o sinal da permutação  $\pi(M, D)$ . Se  $D$  está subentendido, podemos chamar o  $\text{sgn}(M, D)$  de  *sinal do emparelhamento  $M$*  ou  $\text{sgn}(M)$ . Um exemplo de  $\pi(M, D)$  é mostrado na Figura 2.14. O sinal da permutação não depende da ordem em que as arestas de  $M$  são listadas. Essa propriedade é demonstrada no Teorema 2.4.1.

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & 4 & 2 & 6 & 5 \end{pmatrix}$$


**Figura 2.13.** As inversões da permutação  $\pi$  estão representadas pelas letras  $a$ ,  $b$  e  $c$ . Ao todo  $\pi$  tem três inversões.



**Figura 2.14.** Na imagem temos um grafo orientado  $D$ , um emparelhamento perfeito  $M = \{(4, 1), (3, 2), (5, 6)\}$  de  $D$  e uma permutação  $\pi(M, D)$ . Temos 3 inversões na permutação  $\pi(M, D)$ .

**Teorema 2.4.1.** *Seja  $D$  um grafo orientado. Seja  $M$  um emparelhamento perfeito de  $D$ . O sinal da permutação  $\pi(M, D)$  não se altera quando se muda a ordem de listagem das arestas de  $M$  na permutação.*

*Demonstração.* Seja  $M = \{(v_1, v_2), (v_3, v_4), \dots, (v_{n-1}, v_n)\}$  um emparelhamento de  $D$ . A permutação  $\pi(M, D)$  pode ser descrita como:

$$\pi(M, D) = \begin{pmatrix} 1 & 2 & 3 & 4 & \dots & n-1 & n \\ v_1 & v_2 & v_3 & v_4 & \dots & v_{n-1} & v_n \end{pmatrix}$$

Mostraremos inicialmente que a troca de posição de duas arestas consecutivas em uma enumeração das arestas de  $M$  não altera o sinal de  $M$ . Qualquer permutação das arestas de  $M$  pode ser obtida com sucessivas trocas de arestas consecutivas.

Sejam  $e_p = (v_i, v_{i+1})$  e  $e_q = (v_{i+2}, v_{i+3})$  duas arestas de  $M$  em  $\pi(M, D)$ , tal que  $i$  é ímpar e  $1 \leq i \leq n-3$ . Se trocarmos a ordem de  $e_p$  e  $e_q$  na permutação  $\pi(M, D)$  gerando uma permutação  $\pi'(M, D)$ , poderemos observar algumas propriedades.

Os vértices de  $e_p$  e os vértices de  $e_q$  permanecem na mesma ordem, pois trocamos apenas as arestas de posição, ou seja, o número de inversões entre os vértices de cada aresta não é alterado, não impactando na quantidade de inversões das permutações  $\pi(M, D)$  e  $\pi'(M, D)$ .

As outras possibilidades de inversões são dos pares de vértices  $(v_i, v_{i+2})$ ,  $(v_i, v_{i+3})$ ,  $(v_{i+1}, v_{i+2})$  e  $(v_{i+1}, v_{i+3})$  de  $e_p$  e  $e_q$  na permutação  $\pi(M, D)$ . Seja  $k$  o número de pares de vértices invertidos de  $e_p$  e  $e_q$  na permutação  $\pi(M, D)$ . Sabemos que  $k$  pode ser no máximo 4. Se existem  $k$  pares invertidos, então existem  $4-k$  pares não invertidos. Na permutação  $\pi'(M, D)$ , as  $k$  inversões de  $\pi(M, D)$  passam a não ser inversões e as  $4-k$  não inversões de  $\pi(M, D)$  passam a ser inversões. A paridade do número total de inversões de  $\pi(M, D)$  e  $\pi'(M, D)$  não são alterados, pois trocamos  $k$  inversões em  $\pi(M, D)$  por  $4-k$  inversões em  $\pi'(M, D)$ , e para todos os valores de  $k$ , o número  $4-k$  terá sempre a mesma paridade de  $k$ .

Portanto, como a quantidade de inversões em  $\pi(M, D)$  e em  $\pi'(M, D)$  tem a mesma paridade, então ambos têm o mesmo sinal, pois o sinal é igual à  $(-1)^t$ , onde  $t$  é o número de inversões, e para  $t$  ímpar, o valor de  $(-1)^t = -1$ , e para  $t$  par, o valor de  $(-1)^t = 1$ .  $\square$

**Teorema 2.4.2.** *Seja  $\sigma = (\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_n)$  uma enumeração. Seja  $\pi_\sigma$  a permutação descrita por*

$$\pi_\sigma := \begin{pmatrix} 1 & 2 & 3 & \dots & n \\ \sigma_1 & \sigma_2 & \sigma_3 & \dots & \sigma_n \end{pmatrix}$$

*Sejam  $u$  e  $v$  dois elementos da enumeração  $\sigma$ . Seja  $k$  a quantidade de elementos entre os elementos  $u$  e  $v$  na enumeração  $\sigma$ . A troca da posição dos elementos  $u$  e  $v$  em  $\sigma$  impactará em  $2k+1$  pares de elementos de  $\sigma$ , trocando uma inversão para não inversão e vice-versa em  $\pi_\sigma$ .*

*Demonstração.* A demonstração será feita utilizando indução em  $k$ .

- *Caso Base,  $k = 0$ :* ao trocar a ordem de  $u$  e  $v$ , com  $u$  e  $v$  sendo dois elementos adjacentes na enumeração  $\sigma$ , temos um par de elementos impactados em  $\pi_\sigma$ , ou seja,  $2 \cdot 0 + 1$ .
- *Hipótese de Indução:* para  $k$  elementos entre os elementos  $u$  e  $v$  na enumeração  $\sigma$ , temos  $2k + 1$  mudanças no número de inversões em  $\pi_\sigma$ .
- *Passo Indutivo:* para  $k + 1$  elementos entre os elementos  $u$  e  $v$  em  $\sigma$ . Seja  $w$  o elemento anterior a  $v$  na enumeração. Realizamos sucessivamente a troca de  $v$  com  $w$ , a troca de  $v$  com  $u$  e finalmente a troca de  $w$  com  $u$ . A primeira troca é de 2 elementos consecutivos e muda em 1 o número de inversões. A segunda troca, por hipótese de indução, impacta  $2k + 1$  pares de elementos. Finalmente, a troca de  $u$  com  $w$  muda em 1 o número de inversões. De fato, o número total de pares de elementos que tem seu status de inversão trocado e de inversões impactadas é de  $2k + 1 + 2 = 2k + 2 + 1 = 2(k + 1) + 1$ .

$\square$

## 2.5 Grafo Pfaffiano

### 2.5.1 Pfaffiano de uma matriz

Uma matriz quadrada  $A = (a_{ij})$  é dita ser *antissimétrica* se  $a_{ij} = -a_{ji}$ , para todo par de índices  $(i, j)$ . Seja  $A = (a_{ij})_{n \times n}$  uma matriz antissimétrica  $n \times n$ . Quando  $n$



é ímpar,  $\det(A) = 0$ . Quando  $n$  é par, então existe um polinômio  $P := P(A)$  nos  $a_{ij}$  tal que  $\det(A) = P(A)^2$ . O polinômio  $P(A)$  é chamado de *pfaffiano* da matriz  $A$ . Seja  $S_n$  o conjunto das partições em pares do conjunto  $\{1, 2, 3, \dots, n\}$ . O polinômio  $P(A)$  é caracterizado como:

$$P(A) := \sum_{\sigma \in S_n} \left( \text{sgn}(\pi_\sigma) \cdot \prod_{i=1}^{n/2} (a_{\sigma_{2i-1}, \sigma_{2i}}) \right)$$

onde

$$A := (a_{ij})_{n \times n}$$

$$\pi_\sigma := \begin{pmatrix} 1 & 2 & 3 & \dots & n \\ \sigma_1 & \sigma_2 & \sigma_3 & \dots & \sigma_n \end{pmatrix}.$$

O somatório é calculado sobre o conjunto  $S_n$ , tal que  $S_n$  é o conjunto das partições de  $\{1, 2, \dots, n\}$  em pares [Lovász & Plummer, 1986]. Por exemplo:

$$S_2 = \{\{1, 2\}\},$$

$$S_4 = \{\{1, 2\}, \{3, 4\}\}, \{\{1, 3\}, \{2, 4\}\}, \{\{1, 4\}, \{2, 3\}\}.$$

Cada elemento  $\sigma$  de  $S_n$  é composto por pares de inteiros, chamaremos os elementos dos pares de  $\sigma$  de  $\{\{\sigma_1, \sigma_2\}\{\sigma_3, \sigma_4\}, \dots, \{\sigma_{n-1}, \sigma_n\}\}$ , assim  $\sigma = \{\{\sigma_1, \sigma_2\}\{\sigma_3, \sigma_4\}, \dots, \{\sigma_{n-1}, \sigma_n\}\}$ . A ordem dos elementos nos pares de  $\sigma$  e a ordem dos pares de  $\sigma$  são irrelevantes para o cálculo do sinal da permutação  $\pi_\sigma$  (veja o Teorema 2.4.1 e o Teorema 2.4.2). A notação  $\pi_\sigma$  representa a permutação da enumeração  $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \dots, \sigma_{n-1}, \sigma_n)$  em relação à enumeração base  $(1, 2, 3, \dots, n)$ , assim podemos calcular o sinal da permutação  $\pi_\sigma$ . Para exemplificarmos como calculamos o pfaffiano de uma matriz, seja  $A$  uma matriz antissimétrica dada por:

$$A = \begin{pmatrix} 0 & -1 & 1 & 1 \\ 1 & 0 & 1 & -1 \\ -1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \end{pmatrix}.$$

O conjunto  $S_4 = \{\{1, 2\}, \{3, 4\}\}, \{\{1, 3\}, \{2, 4\}\}, \{\{1, 4\}, \{2, 3\}\}$ . O pfaffiano  $P(A)$  é dado como

$$P(A) = \text{sgn} \left( \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix} \right) a_{1,2} \cdot a_{3,4}$$

$$+sgn \left( \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & 2 & 4 \end{pmatrix} \right) a_{1,3} \cdot a_{2,4}$$

$$+sgn \left( \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 2 & 3 \end{pmatrix} \right) a_{1,4} \cdot a_{2,3}$$

$$P(A) = 1 \cdot (-1) \cdot 0 + (-1) \cdot 1 \cdot (-1) + 1 \cdot 1 \cdot 1$$

$$P(A) = 2$$

Podemos chegar no mesmo valor do pfaffiano da matriz  $A$  usando a propriedade  $P(A)^2 = \det(A)$  demonstrada no artigo Kasteleyn [1961] para matrizes antissimétricas de ordem par. Calculando o determinante da matriz  $A$  temos que  $\det(A) = 4$ . Como  $P(A) = \sqrt{\det(A)}$ , portanto  $P(A) = 2$ .

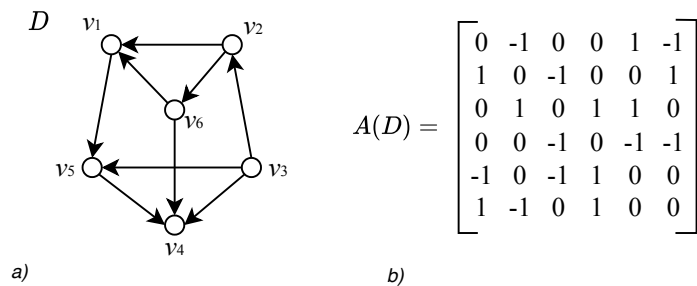
## 2.5.2 Pfaffiano de um grafo

Seja  $D$  um grafo orientado e seja  $n = |V(D)|$ . Uma matriz  $A$  do grafo  $D$  é descrita da seguinte maneira:

$$A(D) = (a_{ij})_{n \times n}$$

$$\text{onde } a_{ij} = \begin{cases} 1, & \text{se } (v_i, v_j) \in E(D) \\ -1, & \text{se } (v_j, v_i) \in E(D) \\ 0, & \text{caso contrário.} \end{cases}$$

Um exemplo da matriz de um grafo é mostrado na Figura 2.15.



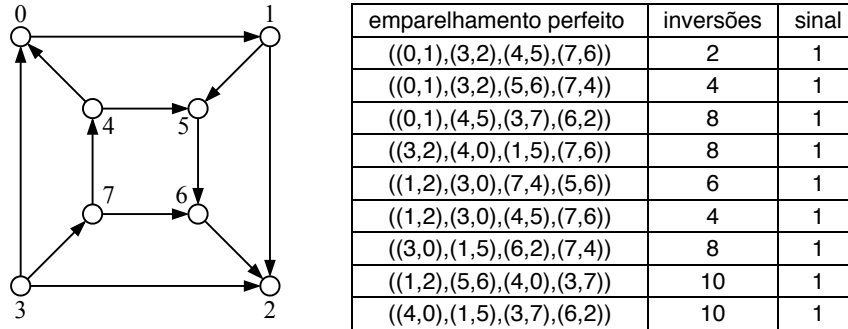
**Figura 2.15.** Exemplo de construção da matriz de um grafo orientado. A matriz  $A$  é gerada conforme as orientações das arestas em  $D$ .

**Definição 2.5.1.** Um grafo orientado é *pfaffiano* se todos os seus emparelhamentos perfeitos têm o mesmo sinal.

**Definição 2.5.2.** Um grafo não orientado é *pfaffiano* se admite uma orientação pfaffiana.

Se  $D$  for um grafo pfaffiano, o pfaffiano da matriz  $A(D)$  é igual em módulo ao número de emparelhamentos perfeitos existentes em  $D$ .

A Figura 2.16 mostra um grafo pfaffiano  $G$  e os sinais referentes a cada emparelhamento perfeito de  $G$ . Um exemplo de um grafo não pfaffiano é o grafo  $K_{3,3}$ .



**Figura 2.16.** Um grafo pfaffiano e o valor dos sinais de cada um dos emparelhamentos perfeitos do grafo. Miranda [2009][Figura 1.1]

**Proposição 2.5.3.** Um grafo sem emparelhamento perfeito é sempre pfaffiano.

*Demonstração.* Considere um grafo que não tenha emparelhamento perfeito. Podemos afirmar por vacuidade que todos seus emparelhamentos perfeitos têm o mesmo sinal, portanto este grafo é pfaffiano.  $\square$

**Teorema 2.5.4.** Seja  $D$  um grafo orientado coberto por emparelhamentos. A mudança da enumeração dos vértices de  $D$  ou mudam os sinais de todos os emparelhamentos perfeitos de  $D$  simultaneamente, ou os sinais se mantêm inalterados.

*Demonstração.* Seja  $\pi$  uma permutação. De acordo com o Teorema 2.4.2, a troca da enumeração de dois vértices de um grafo impacta um número ímpar de inversões, independentemente de onde se encontram os vértices na permutação  $\pi$ . Com isso, a troca da enumeração de quaisquer dois vértices do grafo  $D$  inverterá os sinais de todos os emparelhamentos perfeitos de  $D$  simultaneamente.

Como sabemos que a troca da enumeração de dois vértices inverte os sinais de todos os emparelhamentos perfeitos, com  $k$  trocas da enumeração de dois em dois vértices, teremos o sinal trocado  $k$  vezes ou  $\pi(M, D) \cdot (-1)^k$ . Qualquer outra enumeração pode ser obtida a partir de sucessivas trocas da enumeração de dois em dois vértices.  $\square$

**Corolário 2.5.5.** Seja  $D$  um grafo pfaffiano. A troca da enumeração dos vértices de  $D$  não altera a propriedade de  $D$  ser pfaffiano.

*Demonstração.* De acordo com o Teorema 2.5.4, com a troca da enumeração dos vértices de um grafo, os sinais de todos os emparelhamentos perfeitos ou são trocados simultaneamente, ou continuam inalterados. Como  $D$  é pfaffiano, então todos os emparelhamentos de  $D$  têm o mesmo sinal. A mudança da enumeração dos vértices de  $D$  mantém todos os sinais dos seus emparelhamentos perfeitos iguais. Então  $D$  continua sendo pfaffiano após a troca da enumeração de seus vértices.  $\square$

Seja  $C$  um ciclo não orientado qualquer de tamanho par de um grafo orientado  $D$ . O ciclo  $C$  tem *orientação par* se o ciclo  $C$  em  $D$  possui um número par de arestas orientadas no mesmo sentido do ciclo (não importa qual sentido, horário ou anti-horário). O ciclo  $C$  tem *orientação ímpar* se o ciclo  $C$  em  $D$  possui um número ímpar de arestas orientadas no mesmo sentido do ciclo.

O Lema 8.3.1 do livro do Lovász & Plummer [1986] apresenta uma demonstração para o Teorema 2.5.6.

**Teorema 2.5.6.** *Seja  $G$  um grafo não orientado. Seja  $D$  uma orientação de  $G$ . Sejam  $M_1$  e  $M_2$  dois emparelhamentos perfeitos de  $G$ , e  $k$  o número de ciclos alternados com orientação par formados por  $M_1 \Delta M_2$ . Então vale que  $\text{sgn}(M_1) \cdot \text{sgn}(M_2) = (-1)^k$ .*

O Teorema 2.5.7 nos mostra outra forma equivalente de definir um grafo pfaffiano. Seja  $G$  um grafo com emparelhamento perfeito (tem um número par de vértices) e  $D$  uma orientação das arestas do grafo  $G$ , e seja  $C$  um ciclo de  $G$  de tamanho par (sabendo que caminhos e ciclos não tem vértices repetidos). Um subgrafo  $H$  de  $G$  é *conforme*, se  $G - V(H)$  tem emparelhamento perfeito. Dizemos que  $D$  é uma orientação Pfaffiana de  $G$  se todo ciclo conforme de  $G$  de tamanho par tem orientação ímpar em  $D$  [Robertson et al., 1999][página 3-4].

**Teorema 2.5.7.** *Seja  $G$  um grafo,  $D$  uma orientação de  $G$  e  $M$  um emparelhamento perfeito de  $G$ . Então as seguintes propriedades são equivalentes.*

1.  $D$  é uma orientação pfaffiana de  $G$ .
2. Todo emparelhamento perfeito de  $G$  tem o mesmo sinal de  $M$  em  $D$ .
3. Todo ciclo conforme em  $G$  tem orientação ímpar em  $D$ .
4. Todo ciclo  $M$ -alternado tem orientação ímpar em  $D$ .



# Capítulo 3

## Revisão da Literatura

Neste capítulo são apresentados alguns resultados conhecidos. Estes resultados são de vital importância para o desenvolvimento do Algoritmo MRST que reconhece grafos bipartidos pfaffianos.

Tutte demonstrou seu importante teorema que caracteriza os grafos que possuem emparelhamentos perfeitos utilizando Pfaffianos. Sejam  $G$  um grafo e  $M$  um emparelhamento perfeito de  $G$  e  $o(G)$  a quantidade de componentes (conexas) ímpares (componentes com número ímpar de vértices) de  $G$ . O Teorema de Tutte [1947] nos diz que:

**Teorema 3.0.1** (Teorema de Tutte). *Um grafo  $G = (V, E)$  tem um emparelhamento perfeito, se e somente se,  $o(G - S) \leq |S|$  para todo  $S \subseteq V$ .*

Outro importante teorema é o Teorema de Hall [1935], que nos mostra uma propriedade muito relevante para o estudo de emparelhamentos perfeitos em grafos bipartidos. Seja  $S$  um subconjunto de vértices de  $V(G)$ , o conjunto  $N(S)$  são os vértices vizinhos de  $S$ .

**Teorema 3.0.2** (Teorema de Hall). *Seja  $G[X, Y]$  um grafo bipartido com bipartições  $X$  e  $Y$ , tal que  $|X| = |Y|$ . O grafo  $G$  tem emparelhamento perfeito, se e somente se, para todo  $S$  subconjunto de  $X$ ,  $|N(S)| \geq |S|$ .*

O seguinte teorema desenvolvido por Berge [1957] foi muito importante para o estudo de emparelhamentos máximos. Podemos encontrar uma demonstração para este teorema no livro do Bondy & Murty [2008][Teorema 16.3].

**Teorema 3.0.3** (Teorema de Berge). *Um emparelhamento  $M$  em um grafo  $G$  é um emparelhamento máximo, se e somente se, o grafo  $G$  não contém um caminho  $M$ -aumentante.*

Com o Teorema de Berge podemos encontrar um emparelhamento máximo em um grafo encontrando os caminhos  $M$ -aumentantes e aumentando o emparelhamento de um grafo até que não haja mais caminhos  $M$ -aumentantes, e este é o algoritmo de Kuhn [1955]. O algoritmo de Hopcroft & Karp [1973] usa o teorema de Berge para encontrar um emparelhamento máximo em grafos bipartidos de uma forma mais eficiente. No entanto, não utilizamos este algoritmo mais eficiente, mas sim o algoritmo de Kuhn, pois a complexidade total do algoritmo para reconhecimento de grafos bipartidos pfaffianos é dominada por outras fases de sua execução, sendo que a mudança do algoritmo de Kuhn para o algoritmo de Hopcroft Karp adicionaria complexidade à implementação sem ganho de diminuição do tempo de execução.

Em emparelhamentos em grafos bipartidos tivemos muitas contribuições de vários autores. Frobenius [1917] e Hall caracterizam os grafos bipartidos que têm emparelhamento perfeito. Apesar de Frobenius ter feito essa descoberta primeiro, essa descoberta ficou conhecido como o Teorema de Hall. Muitos outros autores fizeram grandes contribuições para o estudo de emparelhamentos, mas citamos aqui apenas os de maior relevância para este projeto.

Kasteleyn, ao pesquisar por problemas relacionados à química em seu artigo Kasteleyn [1963], provou seu importante teorema que diz respeito aos grafos planares terem orientação pfaffiana.

**Teorema 3.0.4** (Teorema de Kasteleyn). *Todo grafo planar tem uma orientação pfaffiana. Tal orientação pode ser construída em tempo polinomial.*

Neste texto não será demonstrado este teorema, o mesmo pode ser consultado no artigo Kasteleyn [1963][Teorema 8.3.4]. Um famoso Teorema de Kuratowski [1930] diz que um grafo é planar, se e somente se, ele não tem uma subdivisão do  $K_5$  e do  $K_{3,3}$ . Uma *subdivisão*  $H$  de um grafo  $G$  é um grafo obtido de  $G$  pela substituição de arestas de  $G$  por caminhos. Uma *bissubdivisão*  $H$  de um grafo  $G$  é uma subdivisão onde todos os caminhos da substituição tem um número par de vértices internos. Isso nos leva em um resultado obtido por Little [1974] que é uma generalização do Teorema de Kasteleyn no caso bipartido.

**Teorema 3.0.5.** *Um grafo bipartido  $G$  é pfaffiano, se e somente se, o grafo  $G$  não tem bissubdivisão conforme de  $K_{3,3}$ .*

Quando temos um grafo com uma orientação pfaffiana, podemos encontrar a quantidade de emparelhamentos perfeitos usando o Teorema 3.0.6, demonstrado por Kasteleyn [1961], na sua matriz de adjacência.

**Teorema 3.0.6** (Teorema de Kasteleyn). *Seja  $m$  o número de emparelhamentos perfeitos de um grafo  $G$ . Suponha que  $S$  é uma matriz antissimétrica de uma orientação  $D$  de  $G$ . Se  $D$  é pfaffiano, então  $m^2 = \det(S)$ .*

Na etapa da decomposição em cortes justos do Algoritmo MRST, fazemos o uso de sucessivas reduções do grafo bipartido e coberto por emparelhamentos para as suas  $C$ -contrações, e o que prova que estas reduções mantêm a propriedade pfaffiana é o Teorema de Little & Rendl [1991] (Teorema 3.0.7).

**Teorema 3.0.7** (Little & Rendl [1991]). *Sejam  $G$  um grafo coberto por emparelhamentos e  $C$  um corte justo de  $G$ . Então, o grafo  $G$  é pfaffiano, se e somente se, ambas as  $C$ -contrações de  $G$  são pfaffianas.*

O Teorema de McCuaig [2004] (Teorema 4.1.8) prova que podemos reduzir as presilhas redutíveis para presilhas irredutíveis, como será melhor explicado na Seção 4.1.3 (4-soma).

O Teorema 4.1.13 e o algoritmo MRST foram obtidos por McCuaig e separadamente por Robertson, Seymour e Thomas em seus artigos McCuaig [2004] e Robertson et al. [1999] respectivamente. O Teorema 4.1.13 nos mostra que uma presilha irredutível não planar é pfaffiana, se e somente se, ela for isomorfa ao grafo de Heawood. Assim, junto com o Teorema 3.0.4, podemos decidir se uma presilha irredutível é ou não pfaffiana, bastando verificar se ela é planar ou é isomorfa ao grafo de Heawood. Caso uma presilha irredutível não seja planar e nem isomorfa ao grafo de Heawood, então esta presilha não é pfaffiana.





# Capítulo 4

## Metodologia

Podemos dividir o algoritmo MRST em várias partes, onde cada parte recebe entradas e saídas específicas no seu processamento. Podemos chamar essas partes de: remoção das arestas inadmissíveis; decomposição em cortes justos; decomposição 4-soma; verificação da propriedade pfaffiana. A Figura 4.1 nos mostra, de forma resumida, o fluxograma do Algoritmo MRST.

Provamos que os passos do algoritmo funcionam através dos seguintes teoremas: Teorema 4.1.1, Teorema 4.1.2, Teorema 3.0.7, Teorema 4.1.8, Teorema 4.1.9, Teorema 4.1.13 e do Teorema 3.0.4 na Seção 4.1.

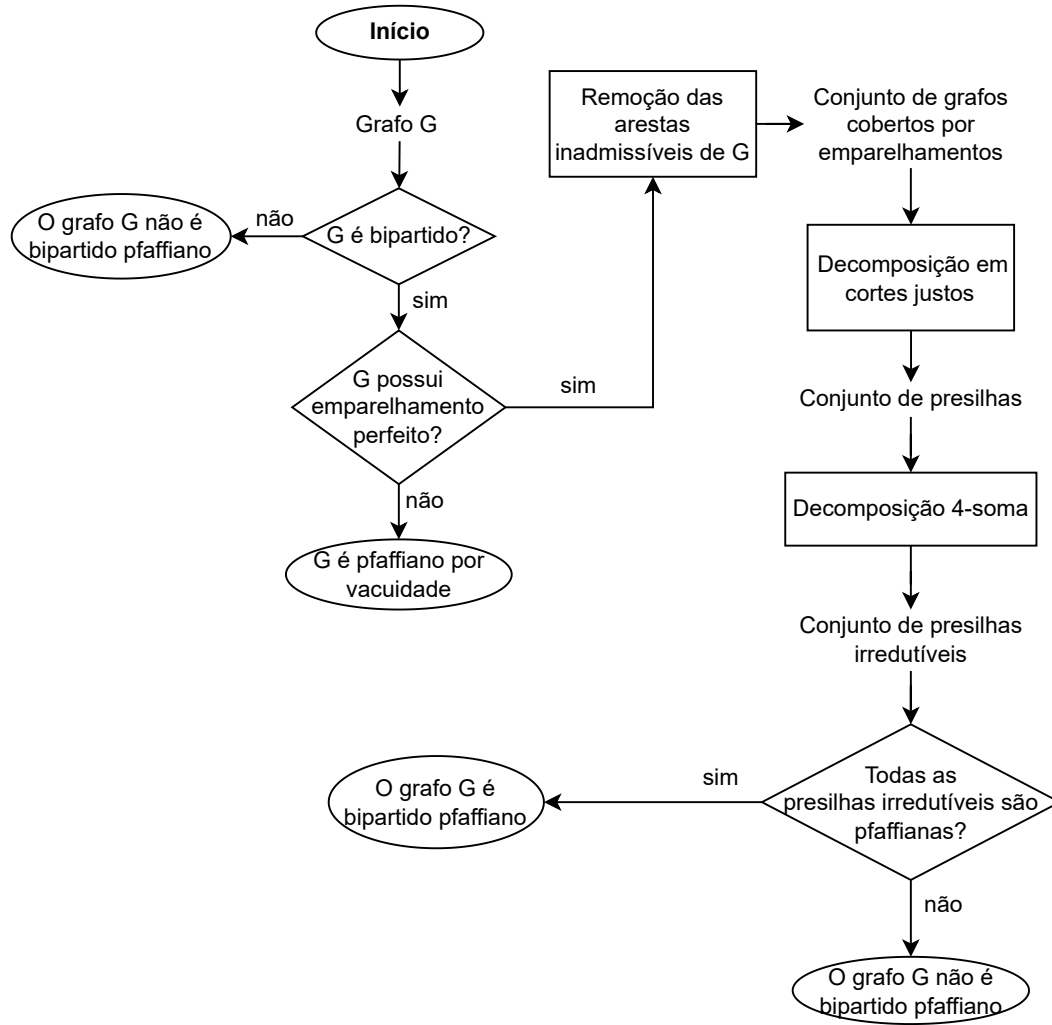
Na Seção 4.2 detalhamos o Algoritmo MRST com a apresentação dos algoritmos de cada etapa vista na Seção 4.1 bem como a apresentação do tempo de execução de cada etapa.

### 4.1 Algoritmo MRST e Por Que Funciona

Nesta seção apresentaremos alguns teoremas e provas que mostram porque o Algoritmo MRST que reconhece grafos bipartidos pfaffianos realmente funciona.

#### 4.1.1 Obtendo Grafo(s) Coberto(s) por Emparelhamentos

Inicialmente  $G$  pode não ser coberto por emparelhamentos. Então devemos remover todas as arestas inadmissíveis de  $G$ . Depois da remoção das arestas inadmissíveis,  $G$  se torna coberto por emparelhamentos ou  $G$  é desconexo e tem componentes que são cobertas por emparelhamentos. Essas remoções geram um conjunto de componentes  $\{G_1, \dots, G_n\}$  de grafos a partir de  $G$ . O processo de remoção das arestas inadmissíveis não muda a propriedade de um grafo ser pfaffiano, como é mostrado no Teorema 4.1.1 e no Teorema 4.1.2.



**Figura 4.1.** Um fluxograma do Algoritmo MRST. A entrada deste algoritmo é um grafo  $G$ .

**Teorema 4.1.1.** *Seja  $G = (V, E)$  um grafo. Seja  $H$  o grafo gerado pela remoção das arestas inadmissíveis de  $G$ . Temos que  $G$  é pfaffiano, se e somente se,  $H$  é pfaffiano.*

*Demonstração.* “ $\rightarrow$ ”: Como descrito anteriormente, se  $G$  é pfaffiano, então  $G$  possui uma orientação pfaffiana  $D$ . Seja  $\mathcal{M}(D)$  o conjunto de todos os emparelhamentos perfeitos de  $D$ . Por  $D$  ser pfaffiano, todos os emparelhamentos perfeitos de  $D$  têm o mesmo sinal. Como as arestas inadmissíveis não estão em nenhum emparelhamento perfeito de  $\mathcal{M}(D)$  então a sua remoção não altera os sinais dos emparelhamentos perfeitos em  $\mathcal{M}(D)$ . Seja  $D_H$  um grafo orientado gerado pela remoção das arestas inadmissíveis de  $G$  em  $D$ , portanto  $D_H$  é uma orientação de  $H$  e  $\mathcal{M}(D) = \mathcal{M}(D_H)$ . Além disso, para todo  $M \in \mathcal{M}(D_H)$ , temos que  $\text{sgn}(M, D_H) = \text{sgn}(M, D)$ . Como  $D$  é uma orientação pfaffiana de  $G$  e  $\mathcal{M}(D) = \mathcal{M}(D_H)$ , concluímos que  $H$  é pfaffiano, pois a orientação  $D_H$  de  $H$  é pfaffiana.

“ $\leftarrow$ ”: Como  $H$  é pfaffiano, temos que existe um  $D_H$  que é uma orientação pfaffiana de  $H$ . Sejam  $G$  um super-grafo de  $H$  formado pela a adição de um conjunto de arestas inadmissíveis no grafo resultante e  $D$  um super-grafo de  $D_H$  com a adição das arestas  $E(G) \setminus E(H)$  orientadas em qualquer sentido. Como  $D_H$  é pfaffiano, então  $D$  é pfaffiano, visto que  $\mathcal{M}(D) = \mathcal{M}(D_H)$ , e como  $D$  é uma orientação de  $G$  e  $D$  é pfaffiano, então  $G$  é pfaffiano.  $\square$

**Teorema 4.1.2.** *Sejam  $G$  um grafo bipartido e  $G_1, \dots, G_n$  as componentes geradas pela remoção das arestas inadmissíveis de  $G$ . Então,  $G$  é pfaffiano, se e somente se,  $G_1, \dots, G_n$  forem pfaffianos.*

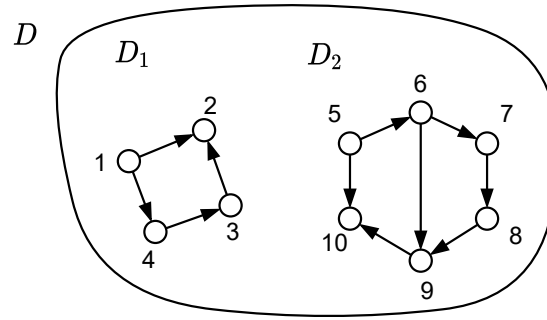
*Demonstração.* “ $\rightarrow$ ”

Assim como é mostrado no Teorema 4.1.1, a remoção das arestas inadmissíveis não altera a propriedade de  $G$  ser pfaffiano, pois as arestas inadmissíveis não alteram o sinal de nenhum emparelhamento perfeito de  $G$ . Sejam  $D$  uma orientação de  $G$ ,  $M$  um emparelhamento perfeito de  $D$ . Seja  $D_i$  a restrição de  $D$  ao grafo  $G_i$  para  $i = 1, 2, \dots, n$ . Seja  $M_i$  a restrição de  $M$  a  $G_i$  para  $i = 1, 2, \dots, n$ . De acordo com o Corolário 2.5.5 podemos mudar a nomenclatura de todos os vértices de  $G$  para que os vértices de uma componente conexa de  $G$  tenham uma nomenclatura de forma sequencial sem alterar a propriedade de  $G$  ser pfaffiano, assim como mostra a Figura 4.2. Em sequência, o Teorema 2.4.1 nos permite rearranjar as arestas dos emparelhamentos no cálculo do seu sinal de forma que as arestas estejam rearranjadas e agrupadas em relação à sua componente conexa em  $D$ , nos permitindo fazer a seguinte mudança, por exemplo:

$$\pi(M) := \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 2 & 4 & 3 & 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix}$$

$$\pi(M_1) := \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 4 & 3 \end{pmatrix} \quad \pi(M_2) := \begin{pmatrix} 5 & 6 & 7 & 8 & 9 & 10 \\ 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix}.$$

Assim sabemos que não haverá inversões entre vértices de distintas componentes  $G_1, \dots, G_n$  e que  $\text{sgn}(M) = \prod_{i=1}^n \text{sgn}(M_i)$ . Cada componente  $D_i$  deve ser pfaffiana para que  $D$  seja pfaffiano. Suponha por absurdo que  $D$  é pfaffiano e uma componente  $D_k$  de  $D$  não seja pfaffiana. Então  $D_k$  tem dois emparelhamentos  $M_p$  e  $M_q$  com sinais diferentes. Seja  $M^*$  um emparelhamento perfeito qualquer de  $G - V(G_k)$ . Seja  $M_p^* = M_p \cup M^*$  e  $M_q^* = M_q \cup M^*$ . Então,  $M_p^*$  e  $M_q^*$  diferem somente nas arestas de  $D_k$ . Então,  $\text{sgn}(M_p^*) \neq \text{sgn}(M_q^*)$ . Isso implica em dizer que  $G$  possui dois emparelhamentos perfeitos com sinais diferentes. Contradizendo a premissa de  $D$  ser pfaffiano.



**Figura 4.2.** O grafo orientado  $D$  é composto pelas componentes conexas e pfaffianas  $D_1$  e  $D_2$ .

“ $\leftarrow$ ”

Se os grafos  $G_1, \dots, G_n$  são pfaffianos, então existem orientações  $D_1, \dots, D_n$  de  $G_1, \dots, G_n$  que são pfaffianas. O super grafo  $D = D_1 \cup \dots \cup D_n$  é pfaffiano.

□

### 4.1.2 Decomposição em Cortes Justos

Depois de obtermos o(s) grafo(s) cobertos por emparelhamentos, fazemos sucessivamente a decomposição em cortes justos deste(s) grafo(s) até que os grafos resultantes da decomposição sejam livres de cortes justos não triviais. Não importa a ordem que executamos a decomposição em cortes justos, os grafos resultantes são sempre os mesmos a menos de diferenças nas multiplicidades das arestas. O seguinte teorema de Lovász [1987] nos mostra essa propriedade.

**Teorema 4.1.3.** *O resultado de qualquer decomposição em cortes justos de um mesmo grafo produz a mesma lista de presilhas e tijolos a menos de arestas múltiplas.*

Os tijolos são grafos não bipartidos, cobertos por emparelhamentos e livres de cortes justos não triviais. Os tijolos não são importantes para o algoritmo em questão, visto que a decomposição em cortes justos de grafos bipartidos tem como resultado apenas grafos bipartidos.

O Teorema 3.0.7 demonstrado por Little & Rendl [1991] em seu artigo nos permite reduzir o problema de decidir se um grafo bipartido é pfaffiano para o problema de decidir se as presilhas resultantes da decomposição em cortes justos do grafo são pfaffianas.

Para encontrarmos um corte justo em um grafo bipartido precisamos fazer algumas análises sobre os grafos bipartidos cobertos por emparelhamentos. Os resultados obtidos por Lovász & Plummer [1986][Teorema 4.1.1] relacionam algumas importantes

propriedades necessárias para se encontrar cortes justos de um grafo coberto por emparelhamentos. A apresentação abaixo utiliza elementos da demonstração presente na dissertação de Miranda [2006][Teorema 2.3.1].

**Teorema 4.1.4.** *Seja  $G$  um grafo bipartido com bipartição  $\{U, W\}$ . Se  $|U| = |W|$  e  $G$  tem pelo menos quatro vértices, então as seguintes afirmações são equivalentes.*

- (i) *O grafo  $G$  é coberto por emparelhamentos.*
- (ii) *Para cada subconjunto não vazio  $X$  de  $U$ , se  $|X| \leq |U| - 1$  então  $|N(X)| \geq |X| + 1$ .*
- (iii) *Para todo  $u \in U$ ,  $w \in W$ , o grafo  $G - u - w$  tem emparelhamento perfeito.*
- (iv) *O grafo  $G$  tem emparelhamento perfeito, e para toda partição  $\{U', U''\}$  de  $U$  e toda partição  $\{W', W''\}$  de  $W$ , tal que  $|U'| = |W'|$  o grafo  $G$  tem pelo menos uma aresta que liga  $U'$  a  $W''$ .*

*Demonstração.* (i)  $\rightarrow$  (ii)

Vamos demonstrar que (i)  $\rightarrow$  (ii) usando a afirmação equivalente (i)  $\wedge \neg(ii) \rightarrow F$ , ou seja, vamos demonstrar que: “O grafo  $G$  ser coberto por emparelhamentos e existir um conjunto não vazio  $X$  de  $U$ , tal que  $|X| \leq |U| - 1$  e  $|N(X)| \leq |X|$  implica em uma falsidade.”

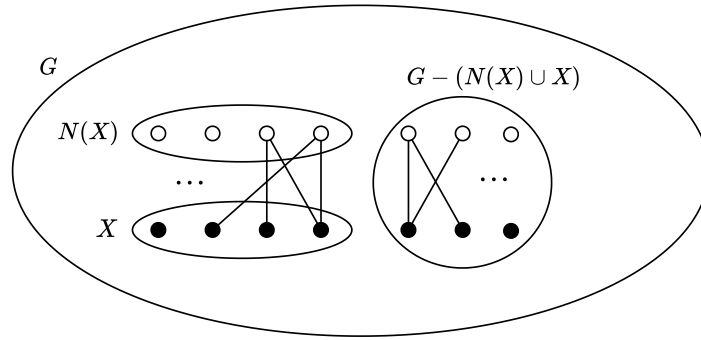
Suponha que o grafo  $G$  é coberto por emparelhamentos e existe um subconjunto não vazio  $X$  de  $U$ , tal que  $|X| \leq |U| - 1$ , e suponha também que  $|N(X)| \leq |X|$ .

Para o caso em que  $|N(X)| < |X|$ , sabemos que o grafo  $G$  não possui nenhum emparelhamento perfeito de acordo com o Teorema de Hall (Teorema 3.0.2). Assim chegamos em uma contradição.

Para o caso em que  $|N(X)| = |X|$  temos que, em todo emparelhamento perfeito de  $G$ , os vértices do conjunto  $X$  obrigatoriamente irão se emparelhar com os vértices de  $N(X)$ . Assim sabemos que não existem arestas admissíveis de  $N(X)$  para  $V(G) - (N(X) \cup X)$  (Figura 4.3). Como  $G$  é coberto por emparelhamentos, não há aresta de  $N(X)$  para  $V(G) - (N(X) \cup X)$ . Pela definição de  $N(X)$ , também não há aresta de  $X$  para  $V(G) - (N(X) \cup X)$ . Desta forma, como  $|X| \leq |U| - 1$ , os conjuntos  $V(G) - (N(X) \cup X)$  e  $(X \cup N(X))$  estão em componentes conexas distintas de  $G$ . Isto é uma contradição a  $G$  ser coberto por emparelhamentos.

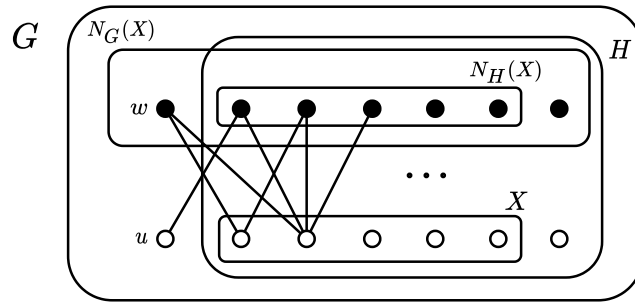
(ii)  $\rightarrow$  (iii)

Sejam  $u \in U$  e  $w \in W$ . Seja  $H := G - u - w$ . Seja  $X$  um subconjunto não vazio de  $U - u$ , assim como é mostrado na Figura 4.4. Pela hipótese sabemos que  $|N_G(X)| \geq |X| + 1$ . O conjunto  $|N_H(X)| \geq |N_G(X)| - 1$ , pois, ou  $|N_H(X)| = |N_G(X)|$ , ou  $|N_H(X)| = |N_G(X)| - 1$ . Como, pela hipótese  $|N_G(X)| \geq |X| + 1$ , então  $|N_H(X)| - 1 \geq |X|$ . Assim,



**Figura 4.3.** Um subconjunto  $X$  da parte  $U$  da bipartição e seus vizinhos  $N(X)$  (da parte  $W$  da bipartição) e um subgrafo formado por  $G - (N(X) \cup X)$ .

temos que  $|N_H(X)| \geq |N_G(X)| - 1 \geq |X|$ , e isso vale para todos os subconjuntos  $X$  de  $U - u$ . Os conjuntos  $\{U - u, W - w\}$  são uma bipartição de  $H$ . Portanto, pelo Teorema de Hall, o grafo  $G - u - w$  tem emparelhamento perfeito. Esta conclusão vale para todo  $u \in U$  e  $w \in W$ .



**Figura 4.4.** O grafo  $H := G - u - w$  tem emparelhamento perfeito.

(iii)  $\rightarrow$  (i)

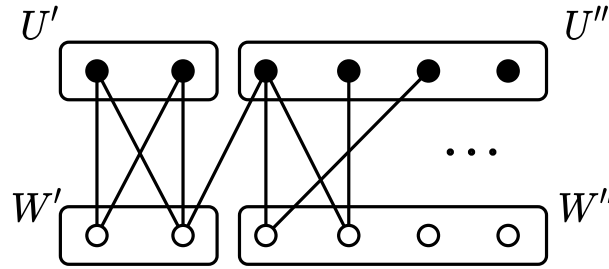
Provaremos que todas as arestas de  $G$  são admissíveis, e que  $G$  é conexo. Sabemos pela hipótese que vale o item (iii). Sejam  $e = uw$  qualquer aresta de  $G$  e  $M$  um emparelhamento perfeito de  $G - u - w$ . Logo  $M \cup \{e\}$  é um emparelhamento perfeito de  $G$ . Isto vale para toda aresta  $e$  de  $G$ . Portanto, todas as arestas de  $G$  são admissíveis.

Para mostrar que  $G$  é conexo suponha, por absurdo, que  $K$  e  $L$  sejam duas componentes conexas de  $G$ . Primeiramente mostraremos que  $K$  tem vértices nas duas partes da bipartição de  $G$ . Suponha, por absurdo, que  $V(K) \cap U$  é vazio. Seja  $k_w$  um vértice de  $V(K) \cap W$ , e  $u$  um vértice em  $U$ . Então,  $G - u - k_w$  tem emparelhamento perfeito  $M(u, k_w)$ . Sendo assim,  $M(u, k_w)$  restrito a  $K - k_w$  é emparelhamento perfeito de  $K - k_w$ . No entanto,  $K$  não tem vértices em  $U$ . Portanto,  $V(K) = \{k_w\}$ . Por hipótese temos que  $V(G) \geq 4$  e  $|U| = |W|$ . Portanto, existe vértice  $w' \in W - k_w$ . Também por hipótese, existe emparelhamento perfeito  $M(u, w')$  de  $G - u - w'$ . No

entanto,  $k_w$  é um vértice isolado de  $G - u - w'$ , uma contradição. Sendo assim,  $V(K) \cap U$  e  $V(K) \cap W$  não são vazios. Analogamente,  $V(L) \cap U$  e  $V(L) \cap W$  não são vazios. Ajuste a notação de forma que  $|V(K) \cap U| \leq |V(K) \cap W|$ . Sejam  $u \in V(K) \cap U$ , e  $w \in V(L) \cap W$ . No entanto,  $G - u - w$  não tem emparelhamento perfeito, pois  $|V(K) \cap U - u| < |V(K) \cap W|$ , uma contradição. De fato,  $G$  é conexo.

(ii)  $\rightarrow$  (iv)

Vamos provar (ii)  $\rightarrow$  (iv) usando a sua contra-positiva  $\neg(iv) \rightarrow \neg(ii)$ . Suponha que vale  $\neg(iv)$ . Assim, existe uma partição  $\{U', U''\}$  de  $U$  e uma partição  $\{W', W''\}$  de  $W$ , tal que  $|U'| = |W'|$ , e não existe aresta que liga  $U'$  a  $W''$  (Figura 4.5). Sendo assim,  $N(U') \subseteq W'$ . Portanto temos que  $|N(U')| \leq |W'| = |U'|$ . Isto implica  $\neg(ii)$ . Portanto (ii)  $\rightarrow$  (iv).



**Figura 4.5.** Não existe aresta que liga  $U'$  e  $W''$ .

(iv)  $\rightarrow$  (ii)

Vamos provar (iv)  $\rightarrow$  (ii) usando a sua contra-positiva  $\neg(ii) \rightarrow \neg(iv)$ , ou seja, provaremos que “Existir um subconjunto não vazio  $X$  de  $U$  com  $|X| \leq |U| - 1$  e  $|N(X)| \leq |X|$  implica em  $G$  não ter emparelhamento perfeito, ou que existe partição  $\{U', U''\}$  de  $U$  e uma partição  $\{W', W''\}$  de  $W$ , com  $|U'| = |W'|$ , de forma que o grafo  $G$  não possui uma aresta que liga  $U'$  a  $W''$ ”. Por  $\neg(ii)$  sabemos que existe um subconjunto não vazio  $X$  de  $U$ , com  $|X| < |U| - 1$ , tal que  $|N(X)| \leq |X|$ . Para o caso em que  $|N(X)| < |X|$ , de acordo com o Teorema de Hall (Teorema 3.0.2), sabemos que o grafo  $G$  não possui um emparelhamento perfeito.

Para o caso em que  $|N(X)| = |X|$ , temos que  $U$  pode ser particionado em  $\{X, U - X\}$  e  $W$  em  $\{N(X), W - N(X)\}$ . Vamos chamar  $\{X, U - X\}$  de  $\{U', U''\}$  e  $\{N(X), W - N(X)\}$  de  $\{W', W''\}$ , assim como mostra a Figura 4.5. Como não existe aresta de  $U'$  para  $W''$  (de  $X$  para  $W - N(X)$ ) então vale  $\neg(iv)$ . Portanto (iv)  $\rightarrow$  (ii).  $\square$

O seguinte teorema pode ser encontrado em de Carvalho et al. [2005][Proposição 2.1].



**Teorema 4.1.5.** *Seja  $G$  um grafo bipartido coberto por emparelhamentos com bipartição  $\{U, W\}$ . Um corte  $C$  de  $G$  é justo, se e somente se, há uma partição  $(U_1, U_2)$  de  $U$  e outra partição  $(W_1, W_2)$  de  $W$ , onde  $|W_1| = |U_1| + 1$  e  $|U_2| = |W_2| + 1$ , tal que cada aresta de  $C$  tem como extremos um vértice de  $W_1$  e outro de  $U_2$ .*

*Notação.* Sejam  $G$  um grafo bipartido com bipartição  $\{U, W\}$ , e  $X$  um subconjunto dos vértices de  $G$ , tal que  $|X \cap U|$  e  $|X \cap W|$  sejam distintos. Então, definimos como  $X_+$  ou  $X^+$  o conjunto dentre  $X \cap U$  e  $X \cap W$  que contém mais vértices. Analogamente,  $X_-$  ou  $X^-$  é definido como o conjunto dentre  $X \cap U$  e  $X \cap W$  que contém menos vértices. Dizemos que  $X_+$  é a parte *majoritária* de  $X$  e que  $X_-$  é a parte *minoritária* de  $X$ .

**Corolário 4.1.6.** *Seja  $G$  um grafo bipartido e coberto por emparelhamentos com bipartição  $\{U, W\}$ . Um corte  $C$  com praia  $X$  de  $G$  é justo, se e somente se, (i):  $|X \cap U|$  e  $|X \cap W|$  diferem de exatamente um (o mesmo vale para  $|\overline{X} \cap U|$  e  $|\overline{X} \cap W|$ ), e (ii) toda aresta de  $C$  é incidente em um vértice da parte majoritária de  $X$  e em um vértice da parte majoritária de  $\overline{X}$ .*

As presilhas são caracterizadas da seguinte maneira [Lovász, 1987][Lema 1.4]:

**Teorema 4.1.7.** *Seja  $G$  um grafo bipartido com bipartição  $\{U, W\}$  e coberto por emparelhamentos. Se o grafo  $G$  tem pelo menos seis vértices, então as seguintes afirmações são equivalentes:*

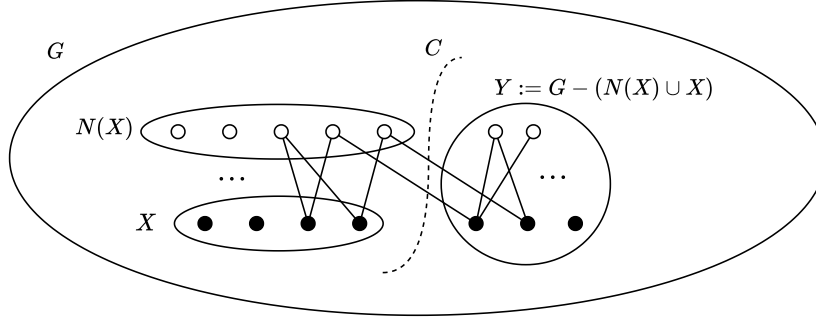
- (i) *O grafo  $G$  é uma presilha, ou seja, o grafo  $G$  é bipartido, coberto por emparelhamentos e livre de cortes justos não triviais.*
- (ii) *Para cada subconjunto não vazio  $X$  de  $U$ , se  $|X| \leq |U| - 2$  então  $|N(X)| \geq |X| + 2$ .*
- (iii) *Para quaisquer dois vértices  $u_1$  e  $u_2$  de  $U$  e quaisquer dois vértices  $w_1$  e  $w_2$  de  $W$ , o grafo  $G - u_1 - u_2 - w_1 - w_2$  tem emparelhamento perfeito.*
- (iv) *Para quaisquer duas arestas não adjacentes (disjuntas)  $e_1$  e  $e_2$  de  $E(G)$ ,  $G$  tem um emparelhamento perfeito contendo  $e_1$  e  $e_2$ .*

*Demonstração.* Vamos provar que (i)  $\rightarrow$  (ii) utilizando a sua contra-positiva.

$\neg(ii) \rightarrow \neg(i)$ . Ou seja, demonstraremos que: “Existir um subconjunto não vazio  $X$  de  $U$  tal que  $|X| \leq |U| - 2$  e  $|N(X)| \leq |X| + 1$ , implica que o grafo  $G$  não é uma presilha.”

Analisaremos o caso em que  $|N(X)| = |X| + 1$ . Em todos os emparelhamentos perfeitos em  $G$ , os vértices do conjunto  $X$  se emparelham com todos os vértices de  $N(X)$  menos um, onde este vértice se emparelhará com um vértice de  $Y := V(G) - (N(X) \cup X)$ , veja a Figura 4.6. Seja  $C = \partial(Y)$  um corte. Como todo emparelhamento  $M$  de  $G$  tem precisamente uma aresta com um extremo em  $X \cup N(X)$  e outro extremo

fora de  $X \cup N(X)$  (no conjunto  $Y$ ), temos que  $C = \partial(Y)$  é um corte justo. Como  $|X| \leq |U| - 2$  e  $|U| = |W|$ , o corte não é trivial. Assim, como o grafo  $G$  tem um corte justo não trivial  $C$ , então  $G$  não é uma presilha.

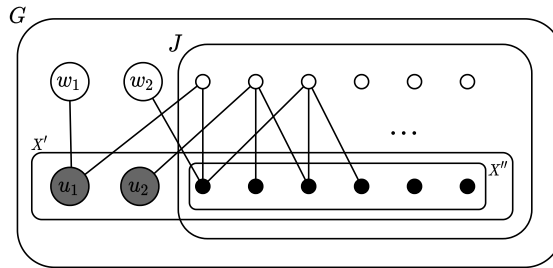


**Figura 4.6.** Um subconjunto  $X$  da parte  $U$  da bipartição e seus vizinhos  $N(X)$  (da parte  $W$  da bipartição), um corte justo  $C$  com praias  $X \cup N(X)$  e  $V(G) - (N(X) \cup X)$ .

Para o caso em que  $|N(X)| \leq |X|$  sabemos que o grafo  $G$  não é coberto por emparelhamentos como é mostrado no Teorema 4.1.4.

Demonstraremos que  $(ii) \rightarrow (iii)$ .

Seja  $J := G - u_1 - u_2 - w_1 - w_2$  e sejam  $X''$  um subconjunto de  $V(J) \cap U$  e  $X' := X'' \cup \{u_1, u_2\}$  um subconjunto de  $V(G)$ . Assim como é mostrado na Figura 4.7. Pelo item  $(ii)$ , sabemos que para cada subconjunto não vazio  $X$  de  $U$ , tal que  $|X| \leq |U| - 2$ , temos que  $|N(X)| \geq |X| + 2$ . Sendo assim  $|N_G(X'')| \geq |X''| + 2$ . Portanto, a remoção de  $u_1, u_2, w_1, w_2$  de  $G$  fará com que  $|N_J(X'')| \geq |X''|$ . Isto vale para todo subconjunto  $X''$  de  $V(J) \cap U$ . Portanto, pelo Teorema de Hall,  $J$  tem emparelhamento perfeito.

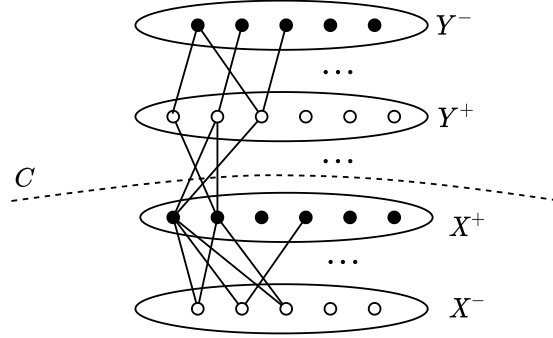


**Figura 4.7.** Um grafo  $G$  com bipartição  $\{U, W\}$  com  $|U| = |W|$  e o conjunto  $X' \subseteq (U)$ . Um grafo  $J := G - u_1 - u_2 - w_1 - w_2$  e o subconjunto  $X''$  de  $J$  onde  $X'' \subseteq U$ .

Vamos demonstrar que  $(iii) \rightarrow (i)$  usando a sua contra-positiva  $\neg(i) \rightarrow \neg(iii)$ . Ou

seja: “Se o grafo  $G$  não é uma presilha, então existem dois vértices  $u_1$  e  $u_2$  de  $U$  e dois vértices  $w_1$  e  $w_2$  de  $W$ , tal que o grafo  $G - u_1 - u_2 - w_1 - w_2$  não tem um emparelhamento perfeito”.

Vamos supor que o grafo  $G$  não é uma presilha. Então  $G$  possui um corte justo  $C$ . Como o corte  $C$  é justo, para cada emparelhamento perfeito  $M$  de  $G$ , temos que  $|C \cap M| = 1$ . Seja  $X$  uma praia do corte  $C$ . Sabemos que, pelo Corolário 4.1.6 que  $X$  possui dois conjuntos de vértices  $X^+$  e  $X^-$ , onde os vértices de  $X^+$  são vizinhos dos vértices de  $X^-$ . Analogamente chamamos a outra praia do corte  $C$  de  $Y$  composto pelos conjuntos  $Y^+$  e  $Y^-$ . Veja a Figura 4.8. Assim sabemos que, em qualquer emparelhamento perfeito de  $G$ , um vértice de  $X^+$  emparelha-se com um vértice de  $Y^+$ . Seja  $J := G - u_1 - u_2 - w_1 - w_2$ . Ao removermos os vértices  $u_1, u_2, w_1, w_2$  de  $G$  de forma que  $u_1$  e  $u_2$  pertencem a  $X^+$ , e  $w_1$  e  $w_2$  pertencem ao conjunto  $Y$ , o conjunto  $X^+$  terá menos vértices que o conjunto  $X^-$ . Assim, temos que  $|N_J(X^+)| < |X^+|$ . De acordo com o Teorema de Hall (Teorema 3.0.2), o grafo  $J$  não tem emparelhamento perfeito.



**Figura 4.8.** Um grafo  $G$  com um corte justo  $C = \partial(X)$

(iii)  $\rightarrow$  (iv)

Para qualquer aresta de  $E(G)$ , sabemos que seus extremos estão em partes diferentes da bipartição de  $G$ . Sejam  $e_1 = u_1w_1$  e  $e_2 = u_2w_2$  quaisquer duas arestas disjuntas de  $E(G)$ . Pela hipótese, o grafo  $G - u_1 - u_2 - w_1 - w_2$  tem emparelhamento perfeito. Portanto  $G$  tem emparelhamento perfeito contendo  $e_1$  e  $e_2$ , com  $e_1$  emparelhando os vértices  $u_1$  e  $w_1$ , e  $e_2$  emparelhando os vértices  $u_2$  e  $w_2$ . Assim, temos que (iii)  $\rightarrow$  (iv).

(iv)  $\rightarrow$  (i)

Vamos provar que (iv)  $\rightarrow$  (i) usando a sua contra-positiva  $\neg(i) \rightarrow \neg(iv)$ . Ou seja, “O grafo  $G$  não ser uma presilha implica que existem duas arestas não adjacentes  $e_1$  e  $e_2$  tal que  $G$  não tem emparelhamento perfeito contendo  $e_1$  e  $e_2$ ”. Suponha que vale  $\neg(i)$ . Isto é, o grafo  $G$  possui um corte justo não trivial  $C$ . Sejam  $X$  e  $Y$  as praias do corte justo  $C$ . Pelo Corolário 4.1.6, o conjunto  $X$  possui dois subconjuntos de vértices  $X^+$

e  $X^-$ , e o conjunto  $Y$  possui dois subconjuntos de vértices  $Y^+$  e  $Y^-$ , assim como é mostrada na Figura 4.8.

Para continuar a nossa demonstração, precisamos antes mostrar que o corte  $C$  tem duas arestas disjuntas. Suponha por absurdo que  $C$  não tem duas arestas disjuntas. Assim, para todo par de arestas  $e_1$  e  $e_2$  de  $C$ , um de seus extremos é comum. Caso  $C$  tenha pelo menos 3 arestas, a única forma de 3 ou mais arestas compartilharem extremos quando comparadas duas a duas é existir um único vértice incidente em todas as arestas de  $C$ . Seja  $v$  tal vértice. Ajuste a notação para que  $v$  esteja em  $Y^+$ . Então, o único vizinho de vértices de  $X$  em  $Y$  é  $v$ . Dessa forma,  $|N(X^+)| = |X^-| + |\{v\}| = |X^+|$ , uma contradição a  $G$  ser coberto por emparelhamentos, de acordo com o item (ii) do Teorema 4.1.4. De fato,  $C$  tem duas arestas disjuntas.

Após demonstrar que o corte  $C$  tem duas arestas disjuntas, vamos demonstrar que  $\neg(i) \rightarrow \neg(iv)$ . Suponha por absurdo que o grafo  $G$  contenha um emparelhamento perfeito  $M$  contendo  $e_1$  e  $e_2$ . Seja  $k$  o número de vértices em  $X^-$ , note que  $|X^+| = k + 1$ . O emparelhamento perfeito  $M$  emparelha dois vértices de  $X^+$  com dois vértices de  $Y$ . Portanto no máximo  $|X^+| - 2 = k - 1$  vértices de  $X^+$  emparelham com vértices de  $X^-$ . Portanto algum vértice de  $X^-$  não é coberto por  $M$ , uma contradição.  $\square$

### 4.1.3 4-Somas

Depois da decomposição em cortes justos, obteremos um conjunto de presilhas. Mas estas presilhas não são presilhas simples. Para obtermos presilhas simples devemos remover os laços e as arestas paralelas.

Uma vez que obtivermos as presilhas simples, devemos reduzir todas as presilhas simples redutíveis usando a decomposição 4-soma descrita abaixo.

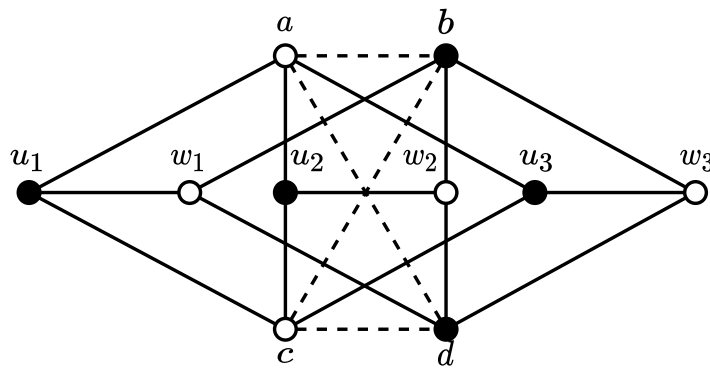
Utilizaremos a notação presente na dissertação de Miranda [2006]. Sejam  $G_1, G_2, \dots, G_n$ , ( $n \geq 2$ )  $n$  grafos e  $Q$  um quadrilátero tal que:

(i) para  $i = 1, 2, \dots, n$ , o grafo  $G_i$  tem seis ou mais vértices.

(ii) para quaisquer dois índices  $i, j$  tal que  $1 \leq i < j \leq n$ ,  $G_i \cap G_j = Q$

Para cada subconjunto  $R$  (possivelmente vazio) de  $E(Q)$ , o grafo  $G := (\bigcup_{i=1}^n G_i) - R$  é a  $(Q, R)$ -soma das  $n$  parcelas  $G_1, G_2, \dots, G_n$ . Se  $Q$  e  $R$  estão sub-entendidos, diz-se simplesmente 4-soma, ao invés de  $(Q, R)$ -soma. Se  $R$  é vazio, dizemos que a 4-soma é *completa*: nesse caso,  $G$  é a união das  $n$  parcelas. Se  $R = E(Q)$  então dizemos que a 4-soma é *fin*: nesse caso, nenhuma aresta de  $Q$  está em  $G$ . O grafo  $H_{10}$ , mostrado na Figura 4.9 é a 4-soma fina de três  $K_{3,3}$ 's.

Note que uma presilha pode ser 4-soma de 2, 3 ou mais parcelas. Nos restringiremos a seguir, na definição de redução 4-soma e de decomposição 4-soma, aos casos em que a 4-soma se aplica sobre 3 ou mais parcelas, ou seja, nos restringiremos à redução 4-soma em presilhas redutíveis (explicaremos isso abaixo quando estivermos discutindo o Teorema 4.1.12). O processo de substituir uma presilha  $G$  pelas suas  $n \geq 3$  parcelas é chamado *redução 4-soma*. E o processo de fazer iteradamente a redução 4-soma em cada presilha de um conjunto de presilhas até que não haja mais presilhas redutíveis é chamado de *decomposição 4-soma*, e o conjunto de presilhas irredutíveis geradas a partir da decomposição 4-soma é o conjunto resultado desta decomposição.



**Figura 4.9.** O grafo  $H_{10}$ , a 4-soma fina de três  $K_{3,3}$ 's. (o quadrilátero  $Q$  é mostrado em linhas pontilhadas) (Miranda [2006]).

#### 4.1.3.1 Presilhas e 4-somas

No grafo  $H_{10}$  mostrado na Figura 4.9, nenhum emparelhamento perfeito de  $H_{10}$  contém tanto  $u_1w_1$  quanto  $u_3w_3$ . Portanto,  $H_{10} - u_1 - u_3 - w_1 - w_3$  não tem emparelhamento perfeito. Sendo assim,  $H_{10}$  não é presilha, de acordo com o Teorema 4.1.7.

O Teorema 4.1.8 foi descoberto por McCuaig [2004][Lema 19 e Lema 20] e descreve algumas propriedades importantes sobre presilhas.

**Teorema 4.1.8.** *Seja  $G$  um grafo bipartido coberto por emparelhamentos que é uma 4-soma de  $n \geq 2$  grafos bipartidos. Então,*

- (i) *se  $G$  é uma presilha, então cada parcela é uma presilha;*
- (ii) *se  $G$  não é uma presilha, mas cada parcela é uma presilha, então  $n \leq 3$  e a soma não é completa. Se, além disso,  $n = 3$ , então a soma é fina e  $G = H_{10}$ .*

O Teorema 4.1.9 nos mostra outra propriedade envolvendo presilhas e 4-somas. Na dissertação de Miranda [2006][Teorema 3.2.1], onde nos baseamos, há uma prova para este teorema.

**Teorema 4.1.9.** *Seja  $G$  uma  $(Q, R)$ -soma de  $n \geq 2$  grafos bipartidos. Se a 4-soma é completa ou se  $n \geq 3$  então  $G$  é uma presilha pfaffiana, se e somente se, cada parcela é uma presilha pfaffiana.*

**Corolário 4.1.10.** *Se  $G$  é uma presilha redutível, então  $G$  é pfaffiana, se e somente se, cada parcela obtida em uma redução 4-soma de  $G$  é pfaffiana.*

*Demonstração.* Como  $G$  é redutível, então  $G$  tem três ou mais parcelas. Deste modo,  $G$  é pfaffiano, se e somente se, cada uma das  $n \geq 3$  parcelas de  $G$  é pfaffiana, de acordo com o Teorema 4.1.9.  $\square$

**Corolário 4.1.11.** *Uma presilha  $G$  é pfaffiana, se e somente se, as presilhas obtidas em uma decomposição 4-soma forem pfaffianas.*

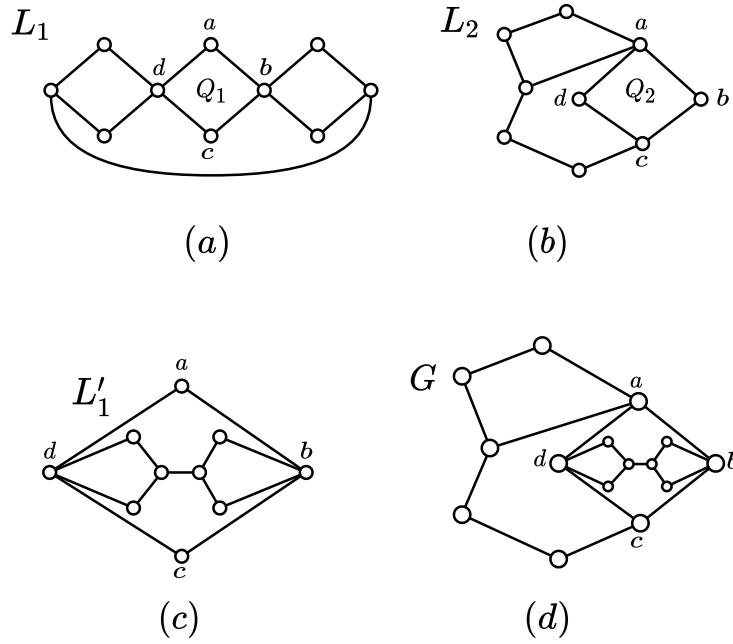
*Demonstração.* Consequência da aplicação indutiva do Corolário 4.1.10.  $\square$

O Teorema 4.1.12, até onde os autores sabem, apareceu pela primeira vez explicado brevemente no artigo de McCuaig [2004][página 7].

**Teorema 4.1.12.** *Seja  $G$  uma presilha irredutível, tal que  $G$  seja uma 4-soma de  $G_1$  e  $G_2$ . Se  $G_1$  e  $G_2$  são planares, então  $G$  é planar.*

*Demonstração.* Sabemos que se  $G_1$  e  $G_2$  são parcelas de uma 4-soma de  $G$ , então existe um quadrilátero  $Q := G_1 \cap G_2$ . Sejam  $Q_1$  o quadrilátero  $Q$  em  $G_1$  e  $Q_2$  o quadrilátero  $Q$  em  $G_2$ . Como  $G_1$  e  $G_2$  são planares, podemos desenhar tanto  $Q_1$  quanto  $Q_2$  como faces em desenho planar  $L_1$  e  $L_2$  de  $G_1$  e  $G_2$ , respectivamente (item (a) e (b) da Figura 4.10). Podemos manipular o desenho planar  $L_1$  de  $G_1$  obtendo um desenho planar  $L'_1$  de tal forma que  $Q_1$  se torne a face externa de  $L'_1$  para que todo o grafo  $G_1 \setminus Q_1$  esteja dentro de  $Q_1$  (item (c) da Figura 4.10), com  $Q_1$  fazendo o contorno do  $G_1$  (como se estivéssemos virando  $G_1$  do avesso através da face  $Q_1$ ) mantendo a planalidade de  $G_1$ . Finalizamos com a substituição da face  $Q_2$  de  $L_2$  por  $L'_1$ , colocando cada vértice de  $Q_1$  no seu respectivo vértice em  $Q_2$  obtendo um grafo isomorfo à  $G$  (item (d) da Figura 4.10). Repare que este grafo isomorfo a  $G$  é planar.  $\square$

Usamos a decomposição 4-soma para reduzir o problema de se decidir se uma presilha  $G$  é pfaffiana para o problema de decidir se as presilhas resultantes da decomposição 4-soma de  $G$  (que também são presilhas) são pfaffianas. De acordo com o Teorema 4.1.12, se uma presilha possui apenas duas parcelas planares então a presilha é planar.



**Figura 4.10.** (a) O desenho planar  $L_1$  da parcela  $G_1$  com o quadrilátero  $Q_1$ , (b) O desenho planar  $L_2$  da parcela  $G_2$  com o quadrilátero  $Q_2$ , (c) É o desenho planar  $L'_1$  e mostra o quadrilátero  $Q_1$  como a face externa, (d) O grafo  $G$  depois da substituição do quadrilátero  $Q_2$  de  $L_2$  por  $L'_1$ .

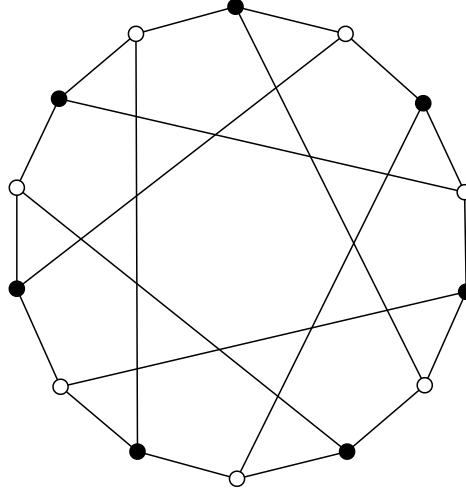
Kasteleyn provou que grafos planares são Pfaffianos (Teorema 3.0.4). Sendo assim, no processo de decomposição em 4-soma de  $G$ , podemos nos restringir a somente reduzir em 4-somas os grafos que sejam 4-soma de 3 ou mais parcelas (presilhas redutíveis). Dessa forma, nem ao menos chamamos de redução 4-soma a operação de substituir um grafo que é 4-soma de 2 parcelas por suas parcelas. Em resumo, o motivo está baseado nos Teoremas 4.1.8, 4.1.13, 3.0.4 e 4.1.12, que mostram que uma reversão de 4-soma de somente duas parcelas não contribuiria para decidir se um grafo é pfaffiano, por isso ignoramos o item (ii) do Teorema 4.1.8.

#### 4.1.4 Presilhas Pfaffianas e grafo de Heawood

Após remover todas as arestas inadmissíveis, decompor em cortes justos (com a remoção de arestas múltiplas) e fazer a decomposição 4-soma de um grafo  $G$  bipartido, obteremos um conjunto de presilhas simples irreduzíveis. Note que, no caso em que  $G$  não tem emparelhamento perfeito, o grafo  $G$  é pfaffiano por vacuidade (todos os seus, nenhum, emparelhamentos perfeitos têm o mesmo sinal). Para decidir se  $G$  é pfaffiano, basta verificarmos se suas presilhas simples irreduzíveis são pfaffianas.

De acordo com o Teorema 3.0.4 as presilhas simples planares são pfaffianas. De acordo com o Teorema 4.1.13 abaixo, o grafo de Heawood é a única presilha simples

Pfaffiana irreduzível não planar (a Figura 4.11 nos mostra o grafo de Heawood). Portanto, uma presilha simples irreduzível não planar será pfaffiana, se e somente se, ela for isomorfa ao grafo de Heawood. Este resultado foi obtido por McCuaig [2004] e separadamente por Robertson et al. [1999]. A dissertação de Miranda [2006] possui uma prova alternativa para este teorema.



**Figura 4.11.** O grafo de Heawood.

**Teorema 4.1.13.** *Seja  $G$  uma presilha simples pfaffiana irreduzível e não planar. Então, a presilha  $G$  é isomorfa ao grafo de Heawood, a menos de arestas múltiplas.*

Portanto, para decidir se um grafo bipartido  $G$  é pfaffiano, devemos fazer os seguintes passos:

- remover as arestas inadmissíveis de  $G$ . Este processo pode resultar em uma ou várias componentes conexas  $\{H_1, H_2, \dots, H_p\}$  de  $G$ ;
- decompor em cortes justos. Para cada componente conexa  $H$  resultante da remoção das arestas inadmissíveis, fazemos sucessivas decomposições em cortes justos até que reste apenas presilhas (que são livres de cortes justos não triviais). Removemos as arestas múltiplas durante este processo. No final teremos um conjunto  $\{I_1, I_2, \dots, I_q\}$  de presilhas;
- decompor em 4-soma. Para cada uma das presilhas  $I$  resultantes da decomposição em cortes justos, fazemos sucessivas decomposições 4-soma das presilhas redutíveis até que restem apenas presilhas irreduzíveis. Após este passo, temos um conjunto  $\{J_1, J_2, \dots, J_r\}$  de presilhas irreduzíveis;



- checar se os grafos obtidos são planares ou isomorfos ao grafo de Heawood. Para cada presilha irreduzível  $J$  obtida no passo anterior, verificamos se  $J$  é um grafo planar, ou é isomorfo ao grafo de Heawood. Se todas as presilhas irreduzíveis forem planares ou isomorfas ao grafo de Heawood, então o grafo  $G$  é pfaffiano.

## 4.2 Descrição do Algoritmo MRST

O Algoritmo MRST que será desenvolvido neste projeto tem como entrada um grafo  $G = (V, E)$  bipartido, a saída é a informação sobre o grafo ser ou não pfaffiano. Sejam  $n = |V|$  e  $m = |E|$ . Os passos do algoritmo são sucessivas reduções em grafos menores de forma a facilitar a análise da propriedade de  $G$  ser ou não pfaffiano. Inicialmente precisamos de um algoritmo para encontrar emparelhamentos máximos em grafos bipartidos, que será utilizado como sub-rotina de algoritmos utilizados pelas reduções.

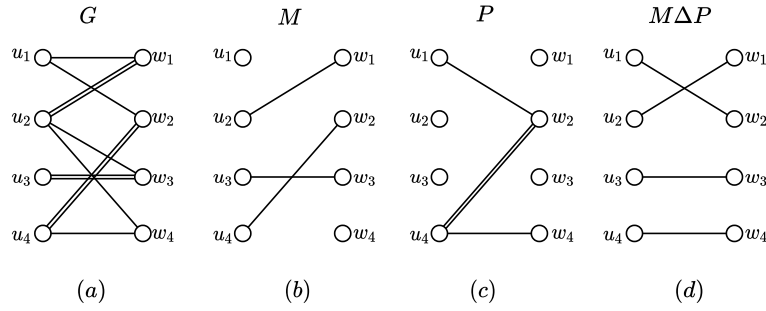
### 4.2.1 Algoritmo para encontrar emparelhamento máximo em grafo bipartido

O Teorema de Berge (Teorema 3.0.3) nos sugere uma abordagem para encontrar emparelhamentos máximos em um grafo. Podemos começar com qualquer emparelhamento  $M$  (inclusive um emparelhamento vazio) e procuramos por caminhos  $M$ -aumentantes através de buscas em largura (BFS) em caminhos  $M$ -alternados iniciados por vértices que não estejam cobertos por  $M$  até encontrar outro vértice que não esteja coberto por  $M$ . Como no exemplo da Figura 4.12 o vértice  $u_1$  do grafo  $G$  não está coberto por  $M$ , uma busca em largura iniciando em  $u_1$  seguindo por caminhos  $M$ -alternados irá encontrar o caminho  $P$  mostrado no item  $c$  da Figura 4.12, finalizando a busca ao encontrar o vértice  $w_4$  (que não é coberto por  $M$ ).

O processo de percorrer os caminhos  $M$ -alternados pode ser simplificado no caso de grafos bipartidos para uma simples busca em um grafo orientado. A construção deste grafo orientado é descrito abaixo.

Seja  $G[U, W]$  um grafo bipartido com bipartições  $U$  e  $W$ . Seja  $M$  qualquer emparelhamento de  $G$ . O grafo  $D(G, U, W, M)$  é definido como um grafo orientado  $D = (V, A)$  com a seguinte construção:

- o conjunto dos vértices de  $D$  é o conjunto de vértices de  $G$ , ou seja, O conjunto  $V(D) := V(G)$ ;
- todas as arestas de  $G$  que estão no emparelhamento  $M$  são orientadas de  $W$  para  $U$  em  $D$ ;



**Figura 4.12.** Representação de uma iteração do algoritmo baseado no teorema de Berge para encontrar emparelhamentos máximos. (a) O grafo  $G$  com as arestas que estão no emparelhamento  $M$  sendo representadas pelas arestas duplas. (b) Um emparelhamento  $M$  de  $G$ . (c) Um caminho  $M$ -aumentante  $P$ . (d) O novo emparelhamento  $M \Delta P$ .

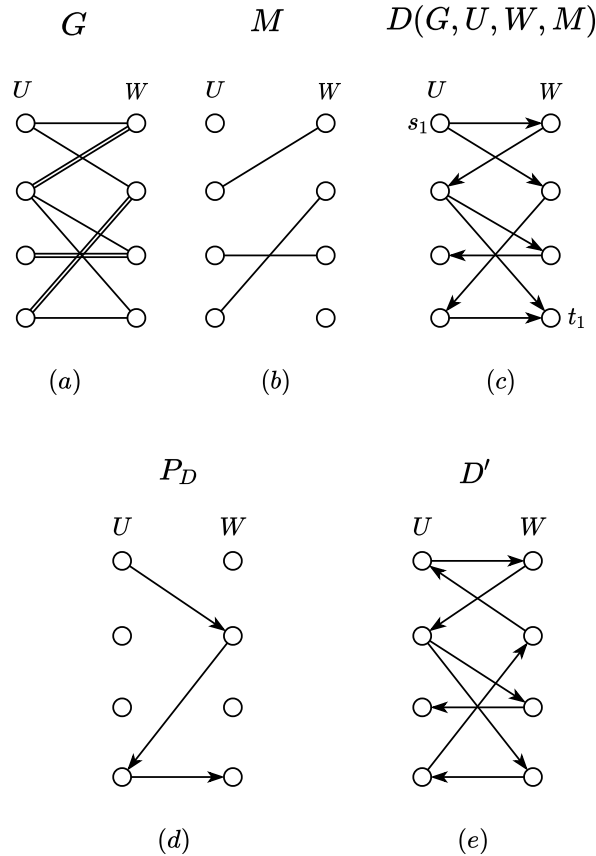
- todas as arestas de  $G$  que não estão no emparelhamento  $M$  são orientadas de  $U$  para  $W$  em  $D$ ;

Usando o grafo  $G$  (item (a)) e o emparelhamento  $M$  (item (b)) da Figura 4.13 como exemplo, obteremos o grafo orientado  $D(G, U, W, M)$  do item (c). O emparelhamento  $M$  é representado pelas arestas com orientação da partição  $W$  para a partição  $U$ .

Sejam  $S$  o conjunto de todos os vértices de  $U$  que não são cobertos por  $M$ , e  $T$  o conjunto de todos os vértices de  $W$  que não são cobertos por  $M$ . Com a construção do grafo  $D(G, U, W, M)$ , podemos seguir um caminho  $M$ -alternado apenas fazendo uma simples busca em largura de um vértice de  $S$  até um vértice de  $T$ . Se fizermos uma busca em largura em  $D(G, U, W, M)$  da Figura 4.13 de um vértice de  $S$  para um vértice de  $T$ , encontraremos um caminho  $P_D$ , mostrado no item (d) da Figura 4.13, que é um caminho  $M$ -aumentante. Repare que o caminho  $P_D$  sem as orientações das arestas é o mesmo caminho encontrado no item (c) da Figura 4.12.

Com o caminho  $P_D$  podemos aumentar o emparelhamento atual de  $D$  fazendo a inversão das arestas de  $D$  por onde passa o caminho  $P_D$ . Assim, obtemos o grafo  $D'$  no item (e) da Figura 4.13, cujo emparelhamento é máximo. Note que o emparelhamento de  $D'$  (as arestas com orientação da partição  $W$  para a partição  $U$ ) é o mesmo emparelhamento encontrado no item (d) da Figura 4.12.

Aplicando este algoritmo de transformar um grafo  $G$  bipartido e um emparelhamento  $M$  em um grafo  $D(G, U, W, M)$  e ir aumentando repetidas vezes o emparelhamento de  $D$  até que não haja um caminho de um vértice de  $S$  para um vértice de  $T$ , podemos encontrar uma emparelhamento máximo em um grafo bipartido. Se o emparelhamento máximo de um grafo bipartido cobre todos os seus vértices, então este emparelhamento também é perfeito. Como dito anteriormente, as arestas de  $D$  com



**Figura 4.13.** (a) Um grafo  $G$  com as arestas do emparelhamento  $M$  sendo representadas pelas arestas duplas, (b) O emparelhamento  $M$  e (c) Representam o grafo  $D(G, U, W, M)$  e os conjuntos  $S = \{s_1\}$  e  $T = \{t_1\}$ .

orientação da partição  $W$  para a partição  $U$  são as arestas do emparelhamento de  $D$  e consequentemente de  $G$ .

O Algoritmo de Berge possui complexidade (tempo de execução) de  $O(nm)$ , por isso ele não é a melhor forma de encontrarmos emparelhamentos máximos em um grafo bipartido. Existem algoritmos para encontrar emparelhamento máximo com menor complexidade. Com a finalidade de desenvolver um algoritmo eficiente para o reconhecimento de grafos bipartidos pfaffianos com o melhor tempo de execução possível, inicialmente, tenderíamos a usar como sub-rotina um algoritmo para encontrar emparelhamentos máximos com a menor complexidade possível. O algoritmo de Hopcroft Karp e Karzanov com a complexidade de  $O(\sqrt{nm})$  seria o algoritmo escolhido para calcular o emparelhamento máximo em grafos bipartidos e poderia ser usado como sub-rotina para o algoritmo MRST (McCuaig, Robertson, Seymour e Thomas). No entanto, esta não foi nossa escolha final.

### 4.2.1.1 Algoritmo de Hopcroft Karp e Karzanov

Este algoritmo é mais eficiente por um fator relevante de  $\sqrt{n}$ . No entanto, ele não será utilizado neste projeto, pois o algoritmo de Berge será suficiente para atingirmos a complexidade assintótica mínima para o algoritmo completo. Isto se deve ao fato de que fases posteriores têm seu melhor tempo pelo menos  $\Theta(m^2)$ , dominando assim a diferença de tempo entre  $\Theta(nm)$  e  $\Theta(\sqrt{nm})$ .

Para encontrar um emparelhamento máximo em um grafo bipartido, podemos usar o eficiente algoritmo de Hopcroft Karp e Karzanov que encontra tal emparelhamento em tempo  $O(\sqrt{nm})$ .

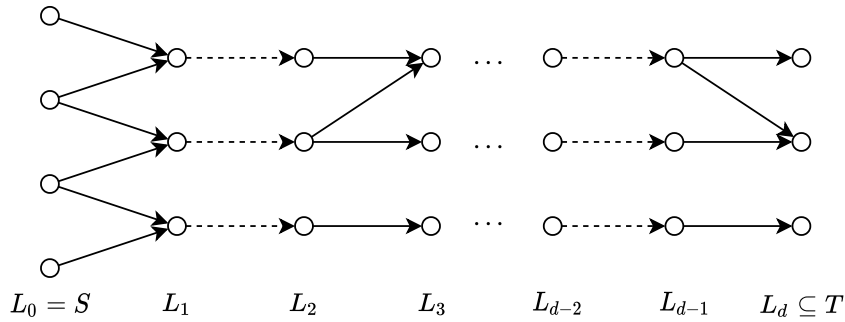
O funcionamento Algoritmo de Hopcroft Karp e Karzanov é muito parecido com o Algoritmo de Berge, a diferença é que o Algoritmo de Berge encontra um caminho  $M$ -aumentante por iteração, enquanto que o Algoritmo de Hopcroft Karp e Karzanov encontra vários caminhos  $M$ -aumentante por iteração.

O Algoritmo de Hopcroft Karp e Karzanov é descrito no artigo Hopcroft & Karp [1973][página 229 a 231]. Contudo, explicaremos uma variante deste algoritmo abaixo para simplificar o entendimento e o desenvolvimento deste algoritmo.

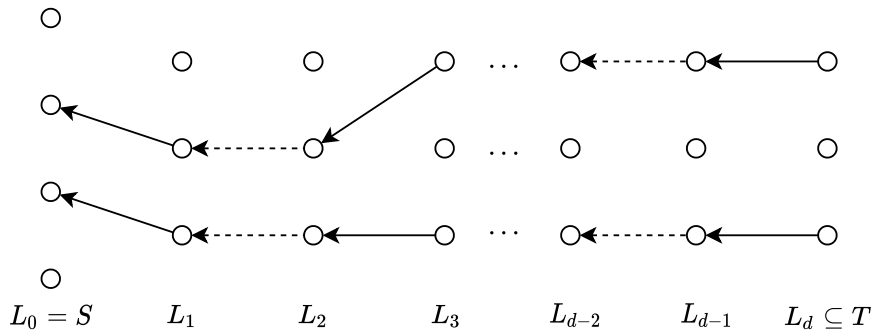
Seja  $G = (V, E)$  um grafo bipartido com bipartição  $\{U, W\}$ . Seja  $M$  um emparelhamento (que pode ser vazio) de  $G$ . Criaremos um grafo orientado  $D(G, U, W, M)$  que nos traz a propriedade de que caminhos aumentantes relativos a  $M$  se tornam caminhos orientados. Sejam  $S$  o conjunto dos vértices de  $U$  que não estão coberto por  $M$  e  $T$  o conjunto dos vértices de  $W$  que não estejam cobertos por  $M$ . Com uma busca em largura partindo de todos os vértices de  $S$  até encontrarmos um vértice de  $T$ , podemos encontrar os menores caminhos  $A$  de  $S$  para  $T$  e todos os vértices que estão a uma distância  $i$  de  $S$ , chamamos de  $L_i$  o conjunto de todos os vértices de  $D$  que estão a uma distância  $i$  de  $S$  (o conjunto  $L_0 = S$ ), a Figura 4.14 nos mostra essa busca.

Depois de termos os caminhos mais curtos do conjunto  $S$  para o conjunto  $T$  representados pelo grafo  $A$ , descobrimos a distância  $d$  dos vértices de  $S$  para os vértices de  $T$  em  $D(G, U, W, M)$ . Em seguida, executaremos uma busca a partir dos vértices de  $L_d$  no grafo inverso de  $D(G, U, W, M)$ , nos restringindo àquelas arestas que vão de um vértice de  $L_k$  para um vértice de  $L_{k-1}$ , para qualquer  $k$ . Esta busca iniciará em  $L_d$  e potencialmente encontrará um vértice de  $S$ . Caso uma tal busca encontre um vértice de  $S$ , teremos um caminho  $M$ -aumentante. O modo em que o algoritmo Hopcroft Karp Karzanov consegue eficientemente obter vários tais caminhos é através da execução desta busca a partir de  $L_d$  por repetidas buscas em profundidade, de tal forma que as subsequentes buscas não utilizem os vértices utilizados em buscas anteriores. A Figura 4.15 mostra dois caminhos  $\bar{P}$  encontrados no grafo inverso de  $D(G, U, W, M)$ . Obtemos os caminhos vértices-disjuntos  $P$  invertendo as orientações dos arcos de  $\bar{P}$ .

A



**Figura 4.14.** Representação do grafo  $A$  obtido a partir de uma busca em largura dos vértices de  $S$  até encontrar um vértice de  $T$  (Hopcroft & Karp [1973]).



**Figura 4.15.** .

Os próximos passos são a inversão de todos os arcos de  $D$  por onde passam os caminhos  $P$  gerando um novo grafo  $D'$  e a atualização dos conjuntos  $S$  e  $T$  para o novo grafo  $D'$ , de modo a preparar o novo grafo  $D'$  para uma nova iteração. Assim terminamos a execução de uma iteração do Algoritmo de Hopcroft Karp e Karzanov.

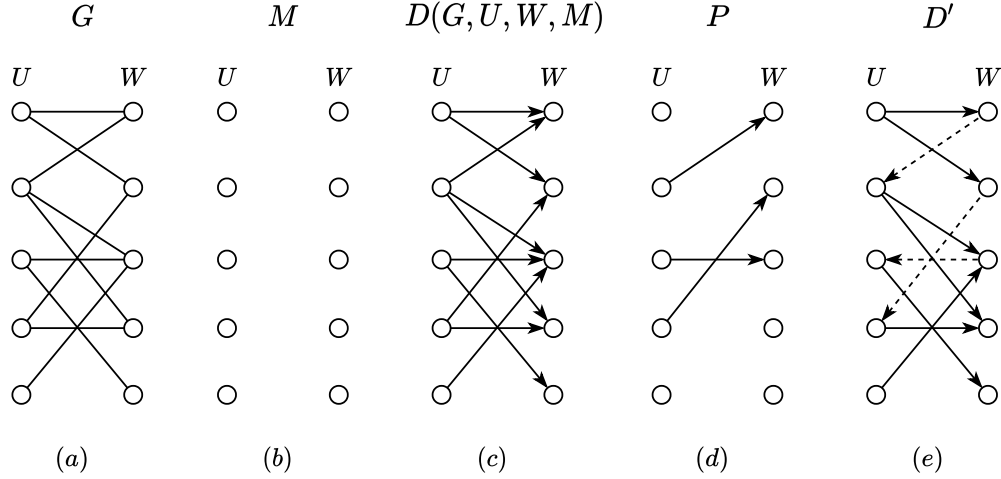
Para a execução completa do Algoritmo de Hopcroft Karp e Karzanov, basta executarmos os passos anteriores e tomar o grafo  $D'$  da iteração anterior como sendo o grafo  $D$ . O algoritmo é finalizado quando não há mais caminhos de  $S$  para  $T$ .

A busca em largura de  $S$  para  $T$  nos garante que o grafo  $A$  gerado dessa busca tenha os menores caminhos de  $S$  para  $T$ . A busca em profundidade nos permite encontrar um conjunto maximal de caminhos vértices-disjuntos  $P$  do grafo  $A$  de modo que os caminhos em  $P$  são os mais curtos dos caminhos de  $S$  para  $T$ , mais precisamente todos os caminhos de  $P$  têm o mesmo tamanho (o tamanho de um caminho é medido em relação a quantidade de arcos ou arestas). Assim, o Algoritmo de Hopcroft Karp e Karzanov encontra os menores caminhos  $M$ -aumentantes por iteração.

O Algoritmo de Hopcroft Karp e Karzanov apresenta um tempo de execução melhor

que o algoritmo de Berge, pois, ao invés de pegarmos um caminho  $M$ -aumentante por vez, conseguimos pegar vários caminhos  $M$ -aumentantes de uma só vez.

A Figura 4.16 mostra a execução da primeira iteração deste algoritmo para o grafo do item (a) da Figura 4.16 com um emparelhamento vazio. A Figura 4.17 mostra a segunda iteração (a iteração final) para o grafo  $D'$  da Figura 4.16. No final teremos o grafo  $D''$  como a saída do Algoritmo de Hopcroft Karp e Karzanov. .

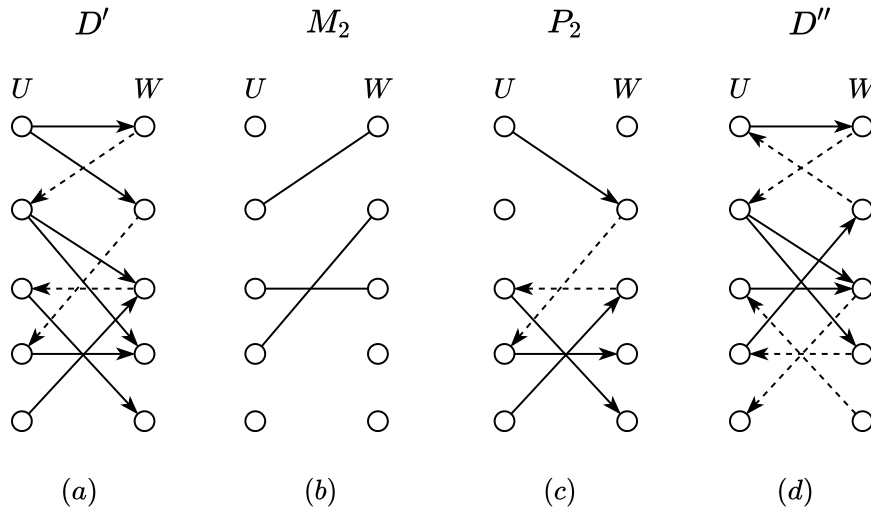


**Figura 4.16.** (a) um grafo bipartido  $G$  com bipartições  $U$  e  $W$ . (b) Um emparelhamento  $M$  vazio de  $G$ . (c) o grafo orientado  $D(G, U, W, M)$ . (d) um conjunto  $P$  de caminhos mais curtos maximais e vértice disjunto de  $S = U$  para  $T = W$ . (e) o grafo  $D'$  gerado pela inversão dos arcos por onde passa os caminhos  $P$ . As arestas de  $D'$  correspondentes ao novo emparelhamento encontrado está sendo representado pelos arcos seccionados.

### 4.2.2 Obtendo grafo(s) coberto(s) por emparelhamentos

Seja  $G$  um grafo bipartido. Para encontrar as arestas inadmissíveis de  $G$ , precisamos de um algoritmo que encontre um emparelhamento máximo em um grafo bipartido, por isso, iremos utilizar o algoritmo de Kuhn Munkres. Devemos primeiramente verificar se  $G$  possui algum emparelhamento perfeito, caso não possua, o grafo  $G$  é pfaffiano por vacuidade, visto que todos seus emparelhamentos perfeitos (ou seja, nenhum) têm o mesmo sinal. Se um grafo possui um emparelhamento perfeito, o seu emparelhamento máximo é também um emparelhamento perfeito.

**Teorema 4.2.1.** *Seja  $G$  um grafo. Seja  $e$  uma aresta de  $G$  com extremos  $u, v$ . Existe um emparelhamento perfeito em  $G - u - v$ , se e somente se, existe um emparelhamento perfeito de  $G$  contendo  $e$ .*



**Figura 4.17.** (a) é o grafo  $D'$  do item (e) da Figura 4.16. (b) É o emparelhamento  $M$  obtido pelos arcos de  $D'$  com orientação de  $W$  para  $U$ . (c) Um conjunto  $P_2$  de caminhos (que possui apenas um caminho) mais curtos maximais e vértice disjunto de  $S$  para  $T$  em  $D'$ . (d) o grafo  $D''$  obtido através da inversão dos arcos por onde passa os caminhos  $P_2$ , e assim encontrando um novo emparelhamento maior que o anterior. As arestas seccionadas representam as arestas dos emparelhamentos.

*Demonstração.* Seja  $M$  um emparelhamento perfeito de  $G - u - v$ . Podemos então criar um emparelhamento perfeito  $M' = M + e$  de  $G$ , de forma que  $e$  emparelha os vértices  $u$  e  $v$ .  $\square$

**Teorema 4.2.2.** *Dados um grafo  $G$ , uma aresta  $e$  de  $G$  com extremos  $\{u, v\}$  e um emparelhamento perfeito  $M$  de  $G$ . Podemos verificar se  $e$  pertence a algum emparelhamento perfeito de  $G$  em tempo  $\Theta(m)$ .*

*Demonstração.* De forma simples, se a aresta  $e$  pertence a  $M$ , então verificamos em tempo  $\Theta(m)$  que  $e$  pertence a algum emparelhamento perfeito. Se a aresta  $e$  não pertence a  $M$  então devemos buscar por um caminho  $M'$ -aumentante em  $G' := G - V(e)$ , onde  $M'$  é a restrição de  $M$  a  $G'$ . Note que, com a operação  $G - V(e)$ , retiramos duas arestas do emparelhamento  $M$ . Se tal caminho  $M'$ -aumentante existe, então existe um emparelhamento perfeito de  $G'$  e portanto existe um emparelhamento perfeito em  $G$  contendo  $e$  (Teorema 4.2.1). Essa operação de buscar um caminho  $M$ -aumentante em  $G'$  é o custo de uma busca em largura, e o tempo para essa operação é de  $\Theta(m)$   $\square$

**Teorema 4.2.3.** *Dado um grafo  $G$ . Podemos decidir se  $G$  é coberto por emparelhamentos em tempo  $O(m^2)$ .*

*Demonstração.* Para sabermos se  $G$  é coberto por emparelhamentos, precisamos saber se cada uma das arestas de  $G$  está em pelo menos um emparelhamento perfeito de  $G$ .

Primeiro temos que encontrar um emparelhamento perfeito  $M$  de  $G$  com o Algoritmo de Kuhn Munkres que levará tempo de  $O(nm)$ . Para cada aresta  $e$  de  $G$ , decidimos se  $e$  está em algum emparelhamento perfeito de  $G$  em tempo  $O(m)$ , como é mostrado no Teorema 4.2.2. Então o tempo total é de  $O(nm) + O(m) \cdot m$ , que simplificando, temos  $O(m^2)$ . Note que uma troca do Algoritmo de Kuhn Munkres pelo Algoritmo Hopcroft Karp Karzanov não diminuiria a complexidade deste passo.  $\square$

**Corolário 4.2.4.** *A remoção das arestas inadmissíveis de um grafo bipartido leva tempo  $O(m^2)$*

*Demonstração.* Podemos adaptar o algoritmo descrito pelo Teorema 4.2.3 para remover todas as arestas que não pertençam a nenhum emparelhamento perfeito. O resultado deste procedimento é um conjunto de grafos cobertos por emparelhamentos.  $\square$

### 4.2.3 Decompondo em cortes justos

Seja  $G$  um grafo bipartido com bipartição  $\{U, W\}$ , e coberto por emparelhamentos. Um corte justo  $C$  de  $G$  é definido pela sua praia  $X$  de onde temos uma parte majoritária de vértices  $X^+$  e uma parte minoritária  $X^-$  (Corolário 4.1.6), a outra praia do corte  $C$  será chamada de  $Y = V(G) \setminus X$ , e conseqüentemente a sua parte majoritária e minoritária são os conjuntos  $Y^+$  e  $Y^-$  respectivamente.

O Algoritmo de decomposição em cortes justos se baseia em encontrar uma das praias do corte.

De acordo com o Teorema 4.2.1, para verificarmos se uma aresta de  $G$  pertence a um emparelhamento perfeito, podemos remover seus extremos e verificar se o grafo resultante possui emparelhamento perfeito. Podemos estender o Teorema 4.2.1 para verificar se duas arestas estão em um emparelhamento perfeito. Para verificarmos se duas arestas  $e_1 = \{u_1, w_1\}$  e  $e_2 = \{u_2, w_2\}$  estão em um emparelhamento perfeito de  $G$ , basta verificarmos se  $G - u_1 - w_1 - u_2 - w_2$  possui emparelhamento perfeito.

De acordo com o Teorema 4.1.7, um grafo  $G$  é uma presilha, se e somente se, todo emparelhamento de tamanho 2 de  $G$  é subconjunto de um emparelhamento perfeito de  $G$ . Sendo assim, podemos testar a propriedade de um grafo ser ou não uma presilha testando se todos os pares de arestas não adjacentes estão em um emparelhamento perfeito. Além disso, como veremos abaixo, caso um par de arestas não adjacentes não esteja em um emparelhamento perfeito é possível a partir destas arestas encontrar uma praia de um corte justo não trivial do grafo.

**Teorema 4.2.5.** *Dado um grafo  $G[U, W]$  bipartido e coberto por emparelhamentos, e um emparelhamento perfeito  $M$  de  $G$  é possível encontrar um corte justo  $C$  de  $G$  ou decidir que  $G$  não possui um corte justo em tempo  $\Theta(m^3)$ .*



*Demonstração.* Para cada aresta  $e_1$ , obtemos um emparelhamento perfeito  $M_1$  de  $G$  contendo  $e_1$ , em tempo  $\Theta(m)$  através de uma busca por caminho aumentante em  $G - V(e_1)$  e  $M$  restrito ao grafo  $G - V(e_1)$ . Tal emparelhamento perfeito  $M_1$  sempre existe, porque  $G$  é coberto por emparelhamentos. Para cada par de arestas não adjacentes  $\{e_1, e_2\}$ , faremos uma busca por caminhos aumentantes em  $G' := G - V(e_1) - V(e_2)$  e  $M'$ , definido como igual ao emparelhamento  $M_1$  restrito ao grafo  $G'$ . Como foi dito anteriormente, podemos estender o Teorema 4.2.1 para verificar se duas arestas estão em um emparelhamento perfeito. Sendo assim, o grafo  $G$  contém um emparelhamento perfeito contendo  $\{e_1, e_2\}$ , se e somente se, esta busca por caminho aumentante tiver sucesso, que é quando  $G'$  tem emparelhamento perfeito. Caso  $G'$  tenha emparelhamento perfeito contendo qualquer par não adjacente de arestas  $\{e_1, e_2\}$ , então  $G$  é livre de cortes justos pelo Teorema 4.1.7. Então, podemos supor que  $G'$  não tem emparelhamento perfeito. Neste caso  $M'$  é um emparelhamento máximo de  $G'$ , e apenas dois vértices  $a$  e  $b$  de  $G'$  não estarão cobertos por  $M'$ . Construiremos a seguir um conjunto de vértices  $X \subset V(G)$  que é praia de um corte justo de  $G$  contendo  $e_1$  e  $e_2$ .

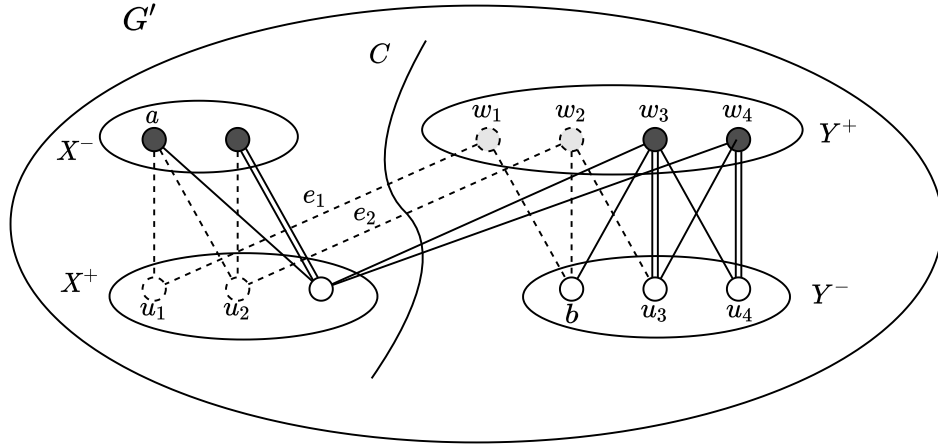
Seja  $R$  o conjunto de vértices de  $G'$  alcançados por uma busca  $M'$ -alternada a partir do vértice  $b$  (não coberto por  $M'$ ). Suponhamos que  $b$  seja um vértice da parte  $U$  da bipartição. Seja  $e_1 = \{u_1, w_1\}$  e  $e_2 = \{u_2, w_2\}$  com  $u_1, u_2 \in U$  e  $w_1, w_2 \in W$ . Veremos que o conjunto  $X = R \cup \{w_1, w_2\}$  é praia de um corte justo  $C$ . Note que  $b$  pertence a  $R$ .

O conjunto  $R$  tem um vértice  $b$  não coberto por  $M'$ , e os demais vértices cobertos por  $M'$ . A parte majoritária  $R^+ = R \cap U$  de  $R$  é a parte contendo  $b$ . Por construção, todas as arestas de  $\partial(R)$  tem extremo em  $R^- = R \cap W$ . O conjunto  $X = R \cup \{w_1, w_2\}$  tem sua parte majoritária  $X^+ = X \cap W$ , no lado oposto da bipartição quando comparado com  $R$ . Todas as arestas de  $\partial(X)$  tem um extremo em  $X^+$ . Portanto, pelo Teorema 4.1.6,  $X$  é praia do corte justo  $C = \partial(X)$ . O corte  $C$  não é trivial pois  $X$  contém  $b, w_1, w_2$  e  $V(G) - X$  contém  $a, u_1, u_2$ .

A Figura 4.18 exemplifica bem esse algoritmo. Podemos ver que uma busca  $M'$ -alternada partindo de  $a$  ou  $b$  em  $G'$  nunca irá alcançar a outra praia do corte justo  $C$ . O conjunto  $R$  de vértices alcançáveis por uma busca  $M'$ -alternada partindo de  $b$  da Figura 4.18 é formado por  $\{b, w_3, u_3, w_4, u_4\}$ , e uma praia de  $C$  é formada por  $X = R \cup \{w_1, w_2\}$ .

Para cada par de arestas  $\{e_1, e_2\}$  a busca por caminho aumentante sobre  $M'$  leva tempo  $\Theta(m)$ . Caso haja uma construção de praia de corte justo, isto pode ser feito no mesmo tempo. Há  $\Theta(m^2)$  pares  $\{e_1, e_2\}$ . Portanto, o tempo total do algoritmo é  $\Theta(m^3)$ .

□



**Figura 4.18.** Representação do grafo  $G'$  com um corte justo  $C$  com praias  $X = X^+ \cup X^-$  e  $Y = Y^+ \cup Y^-$ . O emparelhamento máximo  $M'$  é representada pelas arestas duplas. Os vértices  $u_1, w_1, u_2, w_2$  e todas as arestas incidentes a estes vértices não fazem parte do grafo  $G'$

**Teorema 4.2.6.** *Dados  $G$  um grafo e  $X \subset V(G)$  um conjunto de vértices, o grafo  $H := G\{X \rightarrow x\}$  pode ser gerado em tempo  $\Theta(m + n)$ .*

*Demonstração.* Criamos um grafo  $H$  da seguinte forma. Seja  $V(H) := (V(G) \setminus X) \cup \{x\}$ . Definir  $V(H)$  leva tempo  $O(n)$ . Para cada aresta  $e = \{a, b\}$  de  $E(G)$  faça uma dentre as 3 seguintes alternativas. Caso  $a, b \in V(G) - X$ , adicione  $\{a, b\}$  a  $E(H)$ . Caso  $a \in V(G) - X$  e  $b \in X$ , adicione  $\{a, x\}$  a  $E(H)$ . Caso  $a, b \in X$ , somente descarte a aresta.  $\square$

**Proposição 4.2.7.** *Dados um grafo bipartido e coberto por emparelhamentos  $G$ , e uma praia  $X$  de um corte justo  $C = \partial(X)$  de  $G$ , as  $C$ -contrações de  $G$  podem ser encontradas em tempo  $\Theta(m + n)$*

*Demonstração.* Podemos criar as duas  $C$ -contrações de  $G$  como sendo os grafos  $H_1 := G\{X \rightarrow x\}$  e  $H_2 := G\{\bar{X} \rightarrow \bar{x}\}$  em tempo  $2 \cdot \Theta(m + n)$  através do algoritmo descrito pelo Teorema 4.2.6. O tempo deste algoritmo é de  $\Theta(m + n)$ .  $\square$

**Corolário 4.2.8.** *Dado um grafo bipartido e coberto por emparelhamentos  $G$ . Podemos gerar as  $C$ -contrações de  $G$  se  $G$  possuir um corte justo  $C$  ou decidir que  $G$  não tem um corte justo em tempo  $\Theta(m^3)$ .*

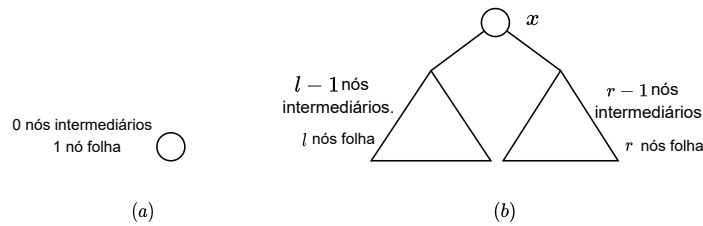
*Demonstração.* Podemos encontrar uma praia  $X \subset V(G)$  de um corte justo  $C = \partial(X)$  de um grafo  $G$  ou decidir que  $G$  não possui um corte justo em tempo  $\Theta(m^3)$  de acordo com o Teorema 4.2.5. Se  $G$  possuir um corte justo, podemos gerar as  $C$ -contrações  $H_1 := G\{X \rightarrow x\}$  e  $H_2 := G\{\bar{X} \rightarrow \bar{x}\}$  como mostra o Teorema 4.2.7 em tempo  $\Theta(m + n)$ . Então, o tempo total é  $\Theta(m^3)$ .  $\square$

**Proposição 4.2.9.** *Uma árvore binária cujos nós internos tenham precisamente 2 filhos com  $k$  nós folha possui  $k - 1$  nós intermediários.*

*Demonstração por indução.* Caso base: para uma árvore binária com 1 nó folha, temos 0 nós intermediários. Assim como é mostrado no item (a) da Figura 4.19.

Hipótese de indução: temos que para uma árvore binária cujos nós internos tenham precisamente 2 filhos com  $l$  nós folha, temos  $l - 1$  nós intermediários.

Passo indutivo: Seja  $T$  uma árvore binária qualquer cujos nós internos tenham precisamente 2 filhos com raiz em um nó  $x$  e cujas sub-árvores à esquerda e à direita tenham  $l$  e  $r$  nós folhas respectivamente. Sendo assim, pela hipótese de indução, a sub-árvore à esquerda e a sub-árvore à direita possuem  $l - 1$  e  $r - 1$  nós intermediários respectivamente, assim como é mostrado no item (b) da Figura 4.19. A árvore  $T$  tem  $r + l$  nós folhas e  $(r - 1) + (l - 1) + 1 = (r + l) - 1$  nós intermediários. Portanto a quantidade de nós intermediários é uma unidade a menos que a quantidade de nós folha.



**Figura 4.19.** (a) Árvore binária do caso base. (b) Árvore binária do passo indutivo.

□

**Proposição 4.2.10.** *Uma decomposição em cortes justos de um grafo  $G$  com  $n$  vértices utiliza no máximo  $n/2 - 2$  cortes justos.*

*Demonstração.* Em cada processo de gerar as duas  $C$ -contrações de  $G$ , obtemos dois grafos resultantes dessa  $C$ -contração e aumentamos o número total de vértices em duas unidades. Podemos fazer uma analogia da decomposição em cortes justos com uma árvore binária, onde cada nó é um grafo(ou o processo de gerar as duas  $C$ -contrações) e seus nós filhos são as suas duas  $C$ -contrações.

Em uma Decomposição em cortes justos geramos  $k$  presilhas que são os nós folha dessa árvore. De acordo com o Teorema 4.2.9, sabemos que em uma árvore binária cujos nós internos tenham sempre 2 filhos com  $k$  folhas, temos  $k - 1$  nós intermediários, onde esses nós intermediários representam operações de se gerar as  $C$ -contrações. Cada contração em corte justo aumenta de 2 o número total de vértices dos grafos resultantes.

Logo com  $n$  vértices de  $G$  geramos  $n + 2(k - 1)$  vértices nos nós folha. Como a menor presilha gerada em uma  $C$ -contração é o  $C_4$ , o número total de vértices nas presilhas geradas é de no mínimo  $4k$ . Portanto  $4k \leq n + 2(k - 1)$ , assim temos que  $k \leq n/2 - 1$ . Com  $n/2 - 1$  folhas, precisamos executar  $n/2 - 2$  cortes justos. Ao total em um grafo com  $n$  vértices, executaremos no máximo  $n/2 - 2$  operações de se gerar as  $C$ -contrações, ou seja, executaremos  $O(n)$  operações.  $\square$

**Teorema 4.2.11.** *Dado um grafo  $G$  bipartido e coberto por emparelhamentos. A decomposição em cortes justos de  $G$  pode ser realizada em tempo  $O(nm^3)$ .*

*Demonstração.* Para fazermos a decomposição em cortes justos devemos fazer os seguintes passos. Procuramos por um corte justo em  $G$ , caso  $G$  possua um corte justo  $C$ , geramos as  $C$ -contrações de  $G$  e repetimos este mesmo procedimento para ambas as  $C$ -contrações de  $G$ . Caso  $G$  não possua um corte justo, então  $G$  é uma presilha. As presilhas encontradas no algoritmo acima são os resultados desta decomposição. Note que estes passos são uma descrição de um algoritmo recursivo que executamos em  $G$  e recursivamente para as  $C$ -contrações de  $G$ . Se  $G$  inicialmente for uma presilha, o grafo  $G$  é o resultado (trivial) desta decomposição. No término deste algoritmo, todos os grafos resultantes são as presilhas do grafo original.

De acordo com o Teorema 4.2.5, a procura por um corte justo  $C$  em  $G$  ou a decisão que  $G$  é ou não uma presilha (não possui corte justo) leva tempo  $\Theta(m^3)$ . Cada uma das contrações em cortes justos aplicadas gasta no máximo  $O(m^3)$  para ser executada (Note que para grafos menores gerados durante a decomposição, o tempo pode ser assintoticamente menor do que  $O(m^3)$ , mas ainda sempre limitado por  $O(m^3)$ ). Teremos no máximo  $O(n)$  aplicações das contrações nos cortes justos em  $G$  de acordo com a Proposição 4.2.10. Um limite superior do tempo total deste algoritmo é de  $O(nm^3)$ .  $\square$

#### 4.2.4 Aplicando a Decomposição 4-soma

Uma vez que possuímos um conjunto de presilhas provenientes da decomposição em cortes justos, devemos reduzir estas presilhas a presilhas irreduzíveis. Dada uma presilha redutível, para que possamos fazer a sua redução, precisamos encontrar uma quádrupla de vértices que o reduz. Chamamos de *blocos* as componentes 2-conexas de um grafo.

O Teorema 4.2.13 nos mostra como encontrar os vértices de corte de um grafo com conectividade igual a 1 em tempo  $\Theta(m)$ , ou seja, podemos encontrar os vértices de corte para o grafo  $G$  que é 1-conexo e não é 2-conexo. A demonstração que este algoritmo funciona está no livro Schrijver [2003][Teorema 15.7a].

**Teorema 4.2.12.** *Sejam  $G$  um grafo,  $X$  e  $Y \subset G$  dois blocos distintos maximais de  $G$  que compartilham vértice. Então o número de vértices compartilhados é 1, ou seja  $A = V(X) \cap V(Y)$  e  $|A| = 1$ . Seja  $A = \{u\}$ , temos que o vértice  $u$  é um vértice de corte.*

*Demonstração.* Suponha por absurdo que existam dois ou mais vértices na intersecção de dois blocos maximais  $X$  e  $Y$ . A remoção de um desses vértices não desconecta os blocos, pois os blocos são 2-conexos e a remoção de um vértice não desconecta o bloco, e como ainda há vértices na intersecção desses blocos, os blocos ainda se mantêm conexos, portanto  $X \cup Y$  é 2-conexo e portanto um bloco, portanto isso é um absurdo pois  $X$  e  $Y$  eram blocos maximais.

Suponha por absurdo que a remoção de  $u$  não desconecta os blocos  $X$  e  $Y$ , logo existe um caminho  $P$  entre os blocos  $X$  e  $Y$  que não passa por  $A$ , Seja  $P'$  o ciclo formado pelo caminho  $P$  e os blocos  $X$  e  $Y$  com o vértice  $u$ . Todos os blocos tocados pelo ciclo  $P'$  (que tem vértice em comum com  $P'$ ) estarão conectados depois da remoção de um vértice qualquer. Então todos os blocos que são tocados pelo caminho  $P'$  e o caminho  $P'$  são um bloco, portanto isso é uma contradição ao fato de  $X$  e  $Y$  serem blocos maximais.  $\square$

**Teorema 4.2.13.** *Dado um grafo  $G$ , é possível em tempo  $\Theta(m)$  encontrar todos os vértices de corte de  $G$ , ou decidir que tais vértices não existem.*

A seguir explicaremos os passos do algoritmo que encontra os blocos de  $G = (V, E)$  em tempo  $\Theta(m)$ .

*Descrição.* Escolha qualquer um vértice  $s \in V(G)$ . Aplique o algoritmo de busca em profundidade começando em  $s$ . Durante esta busca, criaremos um grafo orientado  $D = (V, A)$ , em função da busca. Com essa busca em profundidade, podemos definir uma árvore  $T = (V, E_T)$ , onde  $E_T$  é o conjunto de arcos por onde a busca em profundidade foi realizada. As arestas de  $E_T$  serão adicionadas ao conjunto de arcos  $A$  de  $D$ . As outras arestas de  $G$  que não foram usadas para gerar a árvore  $T$  conectam dois vértices  $u$  e  $v$  de forma que  $T$  tem um caminho  $P$  de  $u$  para  $v$ . Para cada tal aresta  $(u, v)$ , adicione ao conjunto de arcos  $A$  de  $D$  o arco  $(v, u')$ , onde  $u'$  é o segundo vértice de  $P$ . Os arcos de  $E_T$  e esses novos arcos formam um grafo orientado  $D = (V, A)$ . Adaptando a busca em profundidade, podemos encontrar o conjunto  $A$  em tempo linear. Para isso, durante a busca em profundidade, armazenamos a informação para cada vértice  $v$  de qual é o sucessor de  $v$  no caminho de  $v$  para o vértice atualmente sendo tratado pela busca. Se durante a execução da busca em um vértice  $v$  encontrarmos uma aresta  $\{u, v\}$  com  $u$  sendo um vértice já alcançado na busca (e portanto em  $Q$ ), podemos

encontrar o sucessor  $u'$  de  $u$  em  $Q$  e construir um arco  $(v, u')$ . Durante a busca, também armazenamos o antecessor  $ant(x)$  de cada vértice  $x$  na árvore de busca. A Figura 4.20 retrata o grafo  $D$  gerado por este procedimento.

Uma vez que nenhum arco de  $D$  entra em  $s$ , o conjunto  $\{s\}$  é uma componente fortemente conexa. Seja  $CFC(D) = \{J : J \text{ é componente fortemente conexa de } D\}$ . Para cada componente conexa  $J$  de  $CFC(D)$ , com  $J \neq \{s\}$ , buscamos pelo único vértice  $z_J$  tal que  $z_J = ant(x)$ , com  $x$  em  $V(J)$  e  $z_J$  fora de  $V(J)$ . Definimos o conjunto  $J' = J \cup \{z_J\}$ .

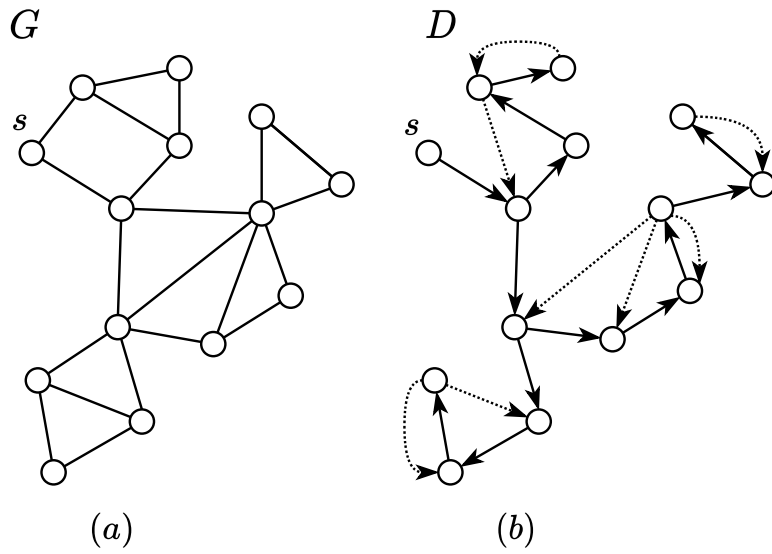
O conjunto  $B(G) = \{J' : J \in CFC(G) - \{s\}\}$  é o conjunto de blocos de  $G$ . Com isso, todos os vértices de  $G$  pertencem a algum bloco. Cada componente fortemente conexa  $J$  corresponde a um bloco maximal  $J^*$  no grafo não orientado subjacente de  $D$  devido a sua construção. No processo de construirmos cada bloco  $J'$ , adicionamos o vértice  $z_J$  correspondente que já pertence a um bloco em  $J^*$ , portanto o vértice  $z_J$  está na intercessão de dois blocos distintos. De acordo com o Teorema 4.2.12, cada vértice  $z_J$  é um vértice de corte de  $G$ . A Figura 4.21 nos mostra as componentes fortemente conexas  $J$  de  $D$  e os conjuntos  $J'$ .

Cada conjunto  $J'$  é um bloco de  $G$ . Se  $G$  tem conectividade igual a 1, então teremos mais de um conjunto  $J'$  e consequentemente teremos um ou vários vértices de corte. Se  $G$  não tem conectividade igual a 1, então  $G$  terá apenas um conjunto  $J'$  (com  $J' = D$ ), o que implica em dizer que  $D$  não possui vértices de corte.  $\square$

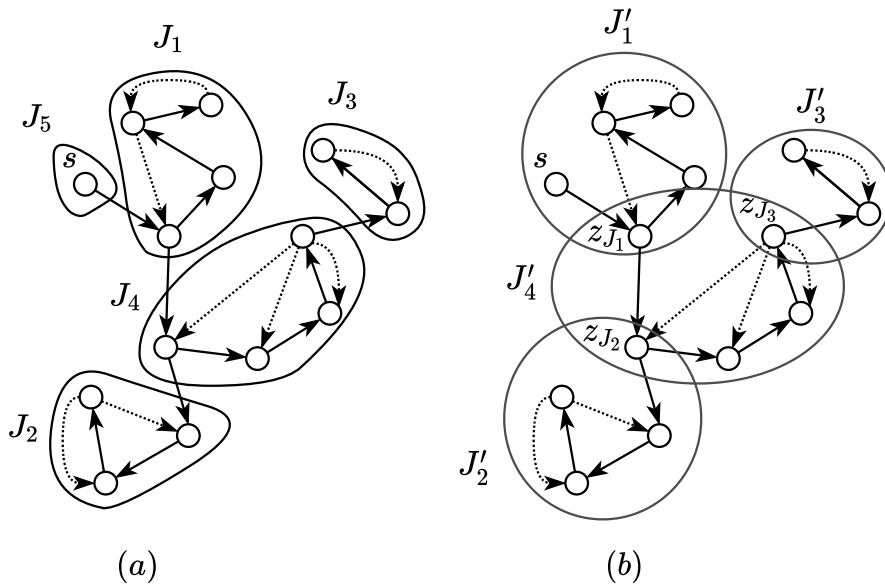
Usando a descrição do algoritmo de encontrar os vértices de corte descrito acima, podemos descrever o algoritmo do Teorema 4.2.14 para a redução 4-soma, que encontra uma quádrupla de vértices de  $G$  que o reduz.

**Teorema 4.2.14.** *Dado uma presilha  $G[U, W]$  com bipartição  $\{U, W\}$ , é possível decidir que  $G$  é irreduzível ou, caso contrário, encontrar todas as quádruplas de vértices que reduzem  $G$  em tempo  $O(n^3m)$ .*

*Demonstração.* Para cada par de vértices  $u_1, u_2$  da parte  $U$  da bipartição, e para cada vértice  $w_1$  da parte  $W$  da bipartição, encontramos os vértices de corte de  $G' := G - u_1 - u_2 - w_1$ . Para cada vértice de corte  $x$  de  $G'$  encontrado, verificamos se  $x \in W$ . Caso  $x$  seja um vértice de corte de  $G'$  que esteja na partição  $W$  de  $G$  e que  $G' - x$  resulte em mais de duas componentes conexas, então os vértices  $u_1, u_2, w_1, x$  formam uma quádrupla de vértices que reduz  $G$ . De acordo com o Teorema 4.2.13 e com a descrição do seu algoritmo, o tempo gasto para encontrar todos os vértices de corte é de  $\Theta(m)$ . Sendo assim, o tempo total para encontrar todas as quádruplas de vértices que reduzem  $G$  é de  $\Theta(n^3m)$ .  $\square$



**Figura 4.20.** Aplicação de busca em profundidade adaptada em  $G$  para encontrar o grafo  $D$ . As arestas da árvore  $T$  é representada pelas arestas contínuas e as arestas de  $A \setminus T$  são representadas pelas arestas pontilhadas.



**Figura 4.21.** o item (a) nos mostra as componentes fortemente conexas de  $D$  que são representadas pelos conjuntos  $J_i$ . O item (b) nos mostra os conjuntos  $J'_i$  e os vértices de corte  $z_{J_1}, z_{J_2}$  e  $z_{J_3}$  que estão nas interseção dos conjuntos  $J'_i$  tomados de dois em dois.

Com as quádruplas de vértices de  $G$  que reduzem  $G$  que podemos obter pelo Teorema 4.2.14, podemos reduzir  $G$  e também todas as suas parcelas e as parcelas das parcelas, e assim sucessivamente.

**Teorema 4.2.15.** Dadas uma presilha  $G$  e uma quádrupla de vértices  $Q$  de  $G$  que

reduzem  $G$ , as parcelas de  $G$  podem ser geradas em tempo  $\Theta(m + n)$

*Demonstração.* Seja  $\{u_1, u_2, w_1, w_2\}$  a quádrupla de vértices  $Q$ . Sejam  $H_1, H_2, \dots, H_k$  as componentes conexas de  $G - u_1 - u_2 - w_1 - w_2$ . Criaremos parcelas de  $G$  a partir das componentes conexas de  $G$ . Sejam  $H'_1, H'_2, \dots, H'_k$  grafos onde  $V(H'_i) := V(H_i) \cup \{u_1, u_2, w_1, w_2\}$  e  $E(H'_i) := E(G[V(H'_i)]) \cup \{\{u_1, w_1\}, \{w_1, u_2\}, \{u_2, w_2\}, \{w_2, u_1\}\}$ , para  $i = 1, 2, \dots, k$ . O tempo de se construir as parcelas de  $G$  é de  $\Theta(m + n)$ , pois precisamos tratar cada vértice e aresta apenas uma vez. O tempo para encontrar as componentes conexas é de  $\Theta(n)$ . Logo o tempo total é de  $\Theta(m + n)$   $\square$

No artigo do Robertson et al. [1999] nos itens (8.6 e 8.8) e (8.9) encontramos demonstrações para o Teorema 4.2.16 e o Teorema 4.2.17. Assim, sabemos que, dada uma presilha pfaffiana  $G$ , cada quádrupla de vértices do conjunto de todas as quádruplas de vértices que reduzem  $G$ , que podemos encontrar usando algoritmo descrito pelo Teorema 4.2.14, reduzirão alguma presilha do grafo  $G$  no processo da decomposição 4-soma, e também que há no máximo  $n - 5$  quádruplas de vértices que reduzem  $G$ .

**Teorema 4.2.16.** *Seja  $G$  uma presilha que possui uma orientação pfaffiana que é uma  $(Q, R)$ -soma das presilhas  $G_1, G_2$  e  $G_3$ . Seja  $X \subseteq V(G)$  com  $X \neq V(Q)$ . Então  $X$  é uma quádrupla de vértices que reduz  $G$ , se e somente se,  $X$  é uma quádrupla de vértices que reduz  $G_1, G_2$  ou  $G_3$ .*

**Teorema 4.2.17.** *Seja  $G$  uma presilha pfaffiana com  $n \geq 5$  vértices. Então  $G$  tem no máximo  $n - 5$  quádruplas de vértices que o reduzem.*

**Teorema 4.2.18.** *Dados uma presilha  $G$  e o conjunto  $S$  de quádruplas que reduzem  $G$ , então a decomposição 4-soma pode ser realizada em tempo  $O(nm)$ .*

*Demonstração.* De acordo com o Teorema 4.2.17, o número total de quádruplas de vértices em  $S$  é no máximo  $n - 5$ , ou seja, o número de vezes que a operação de redução em 4-soma é aplicado é  $\Theta(n)$ . De acordo com o Teorema 4.2.15 podemos gerar as parcelas em tempo  $\Theta(m + n)$ . Assim, o tempo total é de  $O(nm)$ , pois geramos as parcelas de  $G$  e as parcelas das parcelas de  $G$  sucessivamente.  $\square$

**Corolário 4.2.19.** *Dado uma presilha  $G$ , a decomposição 4-soma pode ser realizada em tempo  $O(n^3m)$ .*

*Demonstração.* De acordo com o Teorema 4.2.13 Podemos encontrar todas quádruplas de vértices que reduzem  $G$  em tempo  $O(n^3m)$ . Logo após encontrar todas as quádruplas, podemos realizar a decomposição 4-soma em tempo  $O(mn)$ . Portanto o tempo total é de  $O(n^3m)$ .  $\square$



### 4.2.5 Verificando a Propriedade Pfaffiana

Finalmente, para sabermos se  $G$  é pfaffiano, precisamos apenas saber se todas as presilhas irreduzíveis obtidas da decomposição 4-soma são pfaffianas. Pelo teorema de Kasteleyn (Teorema 3.0.4) sabemos que as presilhas planares são pfaffianas, nos restando saber apenas se as presilhas não planares são o grafo de Heawood, que é a única presilha irreduzível não planar pfaffiana, de acordo com o Teorema 4.1.13.

Para verificar se uma presilha irreduzível é planar podemos usar a biblioteca Boost que possui uma implementação do algoritmo de Boyer & Myrvold [2004]. Este algoritmo recebe um grafo como entrada e verifica se este grafo é ou não planar. A complexidade deste algoritmo é de  $O(n)$ .

Para as presilhas irreduzíveis não planares, falta checar se elas são isomorfas ao grafo de Heawood. Como o grafo de Heawood tem número de vértices fixo, o tempo para verificar se um grafo é isomorfo ao grafo de Heawood é constante.

### 4.2.6 Tempo de Execução do Algoritmo MRST

Vimos que a remoção das arestas inadmissíveis de um grafo bipartido  $G$  e a verificação se  $G$  é coberto por emparelhamentos leva tempo  $O(m^2)$ , de acordo com o Corolário 4.2.4 e o Teorema 4.2.3. A decomposição em cortes justos leva tempo de  $O(nm^3)$  de acordo com o Teorema 4.2.11. A decomposição 4-soma leva tempo de  $O(n^3m)$ , de acordo com o Teorema 4.2.19. E a verificação da propriedade pfaffiana leva tempo linear. Portanto o tempo total do Algoritmo MRST é a soma do tempo de todas as etapas, e portanto, o tempo total é de  $O(nm^3)$ , visto que  $nm^3 \geq n^3m$ , pois  $m \geq n$  para grafos cobertos por emparelhamentos.

## 4.3 Testes e Tecnologias

### 4.3.1 Testes

Após o desenvolvimento do Algoritmo MRST foram feitos os testes para cada etapa do algoritmo e também para o algoritmo MRST como um todo. As entradas para os testes foram criadas manualmente para a certificação de que cada etapa foi desenvolvida corretamente. Durante a implementação do algoritmo MRST, foram feitos testes para a certificação de que cada etapa do Algoritmo MRST tem o tempo condizente com a sua complexidade. Para este tipo de teste, foram analisados vários tempos de execuções das etapas do Algoritmo MRST em relação aos tamanhos das entradas, assim pudemos analisar todos os comportamentos assintóticos. Esse tipo de análise evita que o Algoritmo MRST apresente um tempo de execução maior do que o planejado causadas por

erros em detalhes de implementação, linguagem de programação, estrutura de dados utilizada ou do método usado.

Para testar se um grafo é bipartido, foram fornecidos grafos bipartidos e não bipartidos como entradas. Grafos que são bipartidos com emparelhamento perfeito e grafos bipartidos sem emparelhamentos perfeitos foram usados como entrada para a fase do algoritmo que testa se um grafo possui emparelhamento perfeito. Na etapa da remoção das arestas inadmissíveis, foram fornecidos grafos bipartidos que possuem arestas inadmissíveis e como resposta desta etapa temos um grafo bipartido sem arestas inadmissíveis. Para a Decomposição em cortes justos, fizemos o processo inverso da Decomposição em cortes justos partindo de presilhas para gerar os grafos com cortes justos como entrada, e também foram testados os grafos livres de cortes justos não triviais (presilhas). O resultado foi o conjunto de presilhas usadas para gerar as entradas. Para testar a Decomposição 4-soma, foram gerados os grafos de entrada usando o processo inverso da Decomposição 4-soma, esses grafos de entrada foram gerados a partir de presilhas tomadas como parcelas da 4-soma. O obtido foi são as presilhas usadas inicialmente para gerar as entradas. Finalmente foi feito o teste que verifica se as presilhas irreduzíveis são pfaffianas. Para esse teste foram fornecidos grafos bipartidos planares e não planares e grafos isomorfos ao grafo de Heawood. Todos os grafos planares e os grafos isomorfos ao grafo de Heawood devem ter resposta sim para grafos bipartidos pfaffianos, enquanto que os grafos que não são planares e também não são isomorfos ao grafo de Heawood devem ter resposta não.

Podemos fazer o processo inverso do Algoritmo MRST partindo de presilhas pfaffianas para gerar as entradas para o algoritmo MRST, ou seja, podemos gerar grafos bipartidos pfaffianos.

### 4.3.2 Tecnologias

Para implementar este algoritmo, usamos a linguagem C++. Representamos os grafos usando listas de adjacência, onde os vértices adjacentes a  $v_i$  são os vértices descritos em uma lista na posição  $i$  da lista de adjacência. As bipartições de um grafo foram armazenadas em um vetor, onde o vértice  $v_i$  pertence a parte 0 da bipartição  $\{0, 1\}$  se esse vetor na posição  $i$  for 0, e pertence a parte 1 da bipartição  $\{0, 1\}$  caso contrário. Os emparelhamentos são representados por um vetor, um vértice  $v_i$  está emparelhado com um vértice  $v_j$  se o valor desse vetor na posição  $i$  for  $v_j$ , caso o valor desse vetor na posição  $i$  seja  $-1$  então o vértice  $v_j$  não está emparelhado a outro vértice. Na implementação da parte do algoritmo que trata da verificação da planaridade de um grafo, utilizamos uma implementação pronta do algoritmo de Boyer & Myrvold [2004] disponível na biblioteca Boost que verifica a planaridade de um grafo.

## 4.4 Atividades

As seguintes atividades foram executadas neste projeto.

1. Implementação do algoritmo de Berge para encontrar emparelhamentos máximos.
2. Implementação do algoritmo para a remoção das arestas inadmissíveis.
3. Implementação da Decomposição em cortes justos.
4. Implementação da Decomposição 4-soma.
5. Integração com a biblioteca Boost para o uso da função que determina se um grafo é planar.
6. Implementação do Verificador da propriedade pfaffiana.
7. Testar o algoritmo MRST.

# Capítulo 5

## Resultados Obtidos

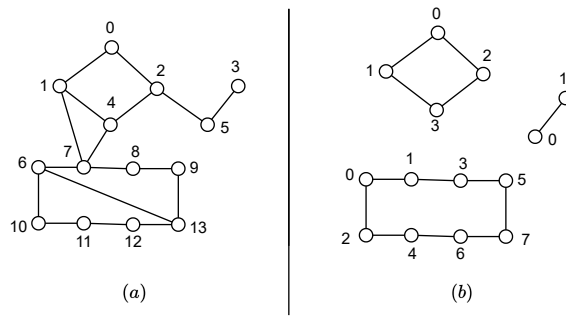
O algoritmo MRST desenvolvido neste projeto está disponível para download através do link <https://github.com/AlvaroEFMota/mrst> juntamente com algumas orientações de como executar este algoritmo. Alguns testes foram implementados e estão disponíveis no projeto.

Cada uma das etapas do algoritmo MRST foi testado com diferentes tipos de grafos a fim de atestar seu funcionamento. Após o teste de cada uma das etapas, o algoritmo MRST foi testado em sua totalidade. Os testes foram divididos em duas partes, a primeira parte são os testes criados manualmente a fim de verificar o comportamento de cada etapa, e a outra parte dos testes foram gerados aleatoriamente a fim de verificarmos o tempo de execução da implementação do algoritmo MRST e se os tempos encontrados correspondem ao comportamento esperado.

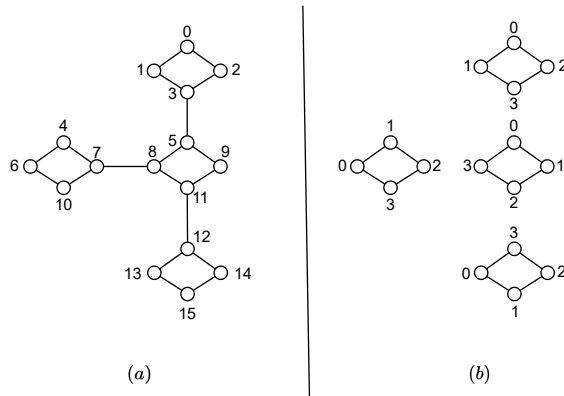
### 5.1 Remoção das arestas inadmissíveis

Na etapa da remoção das arestas inadmissíveis, temos como entrada um grafo que possui emparelhamento perfeito e as saídas são um ou mais grafos cobertos por emparelhamentos. Para a remoção das arestas inadmissíveis mostramos dois testes realizados presentes nas Figuras 5.1 com um grafo com 14 vértices e 17 arestas e 5.2 com um grafo com 16 vértices e 19 arestas. Em ambos os testes temos como saída 3 componentes cobertas por emparelhamentos.

Podemos ver através dos testes realizados na Figura 5.1 e na Figura 5.2 que a implementação da função que remove as arestas inadmissíveis passa neste teste.



**Figura 5.1.** O item (a) corresponde a entrada da função que remove arestas inadmissíveis e o item (b) são os resultados que são as componentes obtidas resultante da remoção das arestas inadmissíveis  $\{1, 7\}$ ,  $\{4, 7\}$ ,  $\{6, 13\}$  e  $\{2, 5\}$  do grafo do item (a).



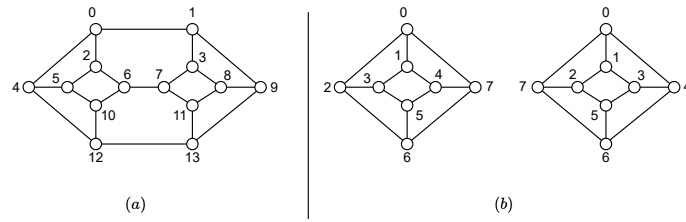
**Figura 5.2.** O item (a) corresponde a entrada da função que remove arestas inadmissíveis e o item (b) são os resultados que são as componentes obtidas resultante da remoção das arestas inadmissíveis  $\{7, 8\}$ ,  $\{3, 5\}$  e  $\{11, 12\}$  do grafo do item (a).

## 5.2 Decomposição em cortes justos

Para a decomposição em cortes justos, a entrada é um grafo bipartido  $G$  coberto por emparelhamentos, se  $G$  tem corte justo, então é aplicado a decomposição em cortes justos até que todos os grafos resultantes sejam presilhas. Para este teste, usamos o grafo da Figura 5.3 com 10 vértices e 21 arestas e também um teste maior com 26 vértices e 39 arestas, como é mostrado na Figura 5.4.

A decomposição em cortes justos da Figura 5.3 tem um grafo como entrada e produz duas presilhas. Neste teste, foi encontrado apenas um corte justo descrito pelas arestas  $\{0, 1\}$ ,  $\{6, 7\}$  e  $\{12, 13\}$ .

Para o teste da Figura 5.4, o algoritmo encontrou um corte justo representado pelo conjunto de arestas  $\{\{2, 6\}, \{7, 10\}, \{5, 8\}\}$  no grafo do item (a), com isso, a decomposição em cortes justos produziu duas componentes, a componente (c) que é uma



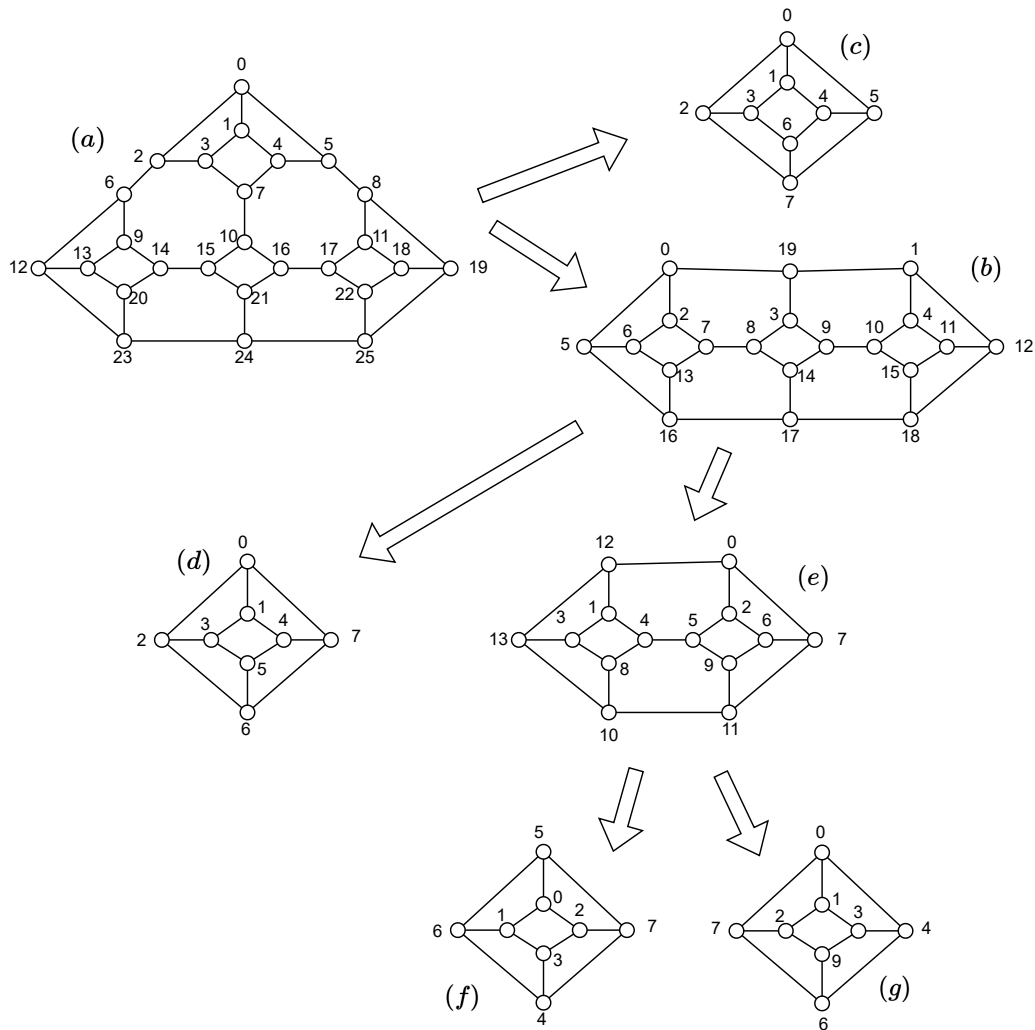
**Figura 5.3.** Aplicação da decomposição em cortes justos. (a) grafo usado como entrada que possui um corte justo descrito pelo conjunto de arestas  $\{\{0, 1\}, \{6, 7\}, \{12, 13\}\}$ . (b) resultado da decomposição em cortes justos do item (a).

presilha, e a componente do item (b) que possui cortes justos. Em sequência, foi encontrado na componente do item (b) o corte justo representado pelo conjunto de arestas  $\{\{0, 19\}, \{7, 8\}, \{16, 17\}\}$ . Após a aplicação da decomposição em cortes justos na componente do item (b), tivemos como resposta a componente do item (d) que é uma presilha e na componente do item (e) que possui mais um corte justo. Finalmente, a componente (e) foi decomposta em corte justos gerando as duas componentes (f) e (g) através do corte justo representado pelo conjunto  $\{\{0, 12\}, \{4, 5\}, \{10, 12\}\}$ . Assim, a decomposição em cortes justos do grafo do item (a) produziu como saída as presilhas (c), (d), (f) e (g).

### 5.3 Decomposição 4-soma

Para a decomposição 4-soma, temos como entrada uma presilha redutível e como saída temos todas as presilhas irredutíveis em que o grafo de entrada pode ser decomposto. O teste realizado na Figura 5.5 conta com um grafo com 26 vértices e 39 arestas.

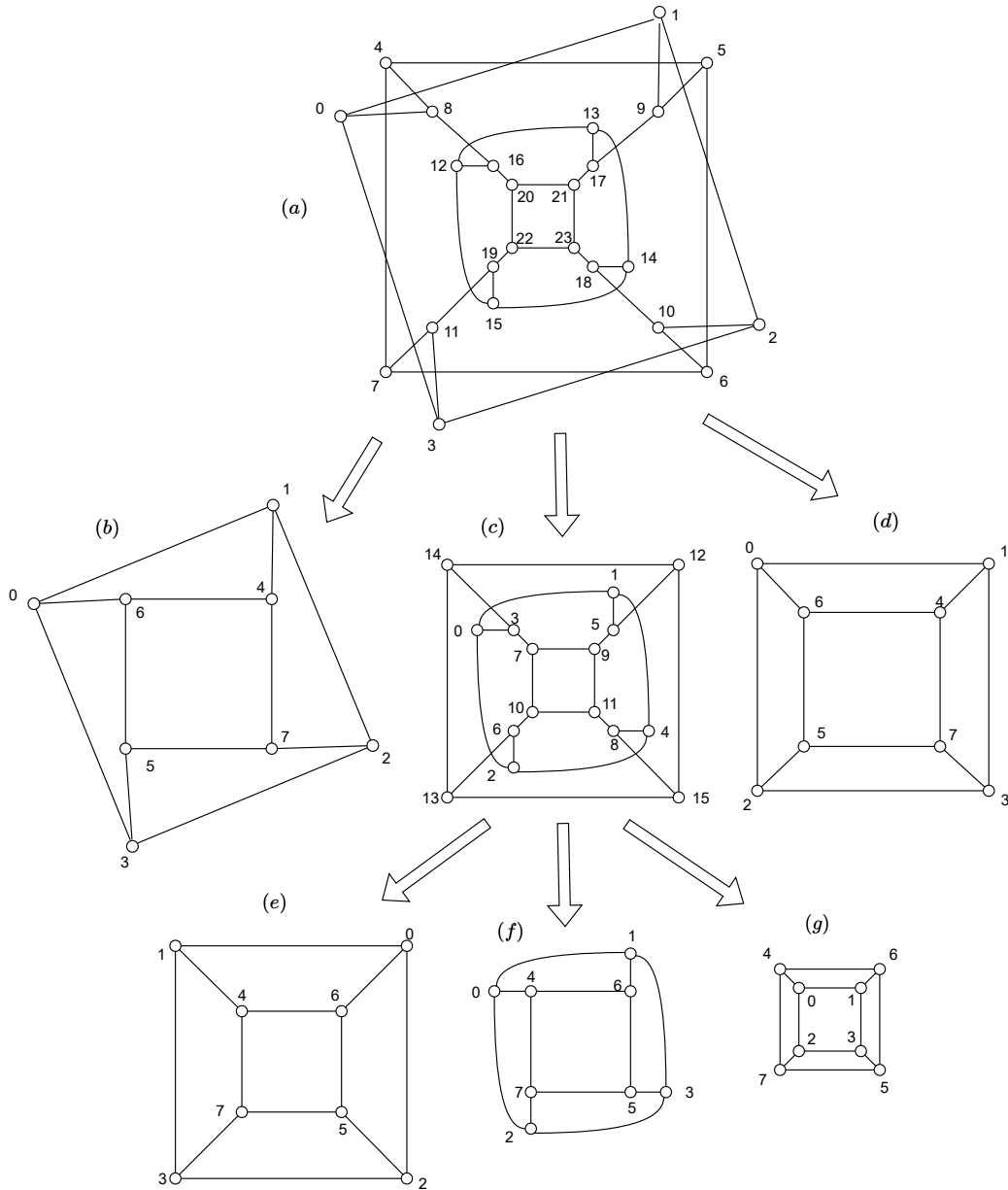
Na Figura 5.5 o grafo de entrada, que é uma presilha redutível, é representada pelo item (a). Inicialmente é encontrado um quadrilátero representado pelos vértices  $\{8, 9, 10, 11\}$  no grafo do item (a) que o reduz. Essa redução produz as presilhas dos itens (b), (c) e (d), sendo que, dessas presilhas geradas, apenas a presilha do item (c) é redutível. Logo em seguida, o algoritmo realiza a decomposição do grafo do item (c) através do quadrilátero formado pelos vértices  $\{3, 5, 8, 6\}$  que o reduz, produzindo as presilhas irredutíveis dos itens (e), (f) e (g). Logo os itens (b), (d), (e), (f) e (g) são as presilhas irredutíveis obtidas como resposta da decomposição 4-soma do grafo do item (a).



**Figura 5.4.** Aplicação da decomposição em cortes justos. O item (a) é o grafo usado como entrada. Os grafos dos itens (c), (d), (f) e (g) são as presilhas obtidas como resultados.

## 5.4 Verificando presilhas irredutíveis

Nesta etapa verificamos se uma dada presilha irredutível é ou não pfaffiana. Para uma presilha irredutível ser pfaffiana, basta que ela seja planar ou isomorfa ao grafo de Heawood. Para a verificação da planaridade de uma presilha, foi usado o algoritmo de Boyer e Myrvold disponível na biblioteca Boost ([https://www.boost.org/doc/libs/1\\_36\\_0/libs/graph/example/simple\\_planarity\\_test.cpp](https://www.boost.org/doc/libs/1_36_0/libs/graph/example/simple_planarity_test.cpp)). Para verificarmos se uma presilha irredutível é isomorfa ao grafo de Heawood, foi implementado um algoritmo que testa usando força bruta em tempo exponencial, mas como o grafo de Heawood tem um número constante de vértices e arestas, é 3 regular e bipartido, isso faz com que o tempo dessa verificação seja bem curto.

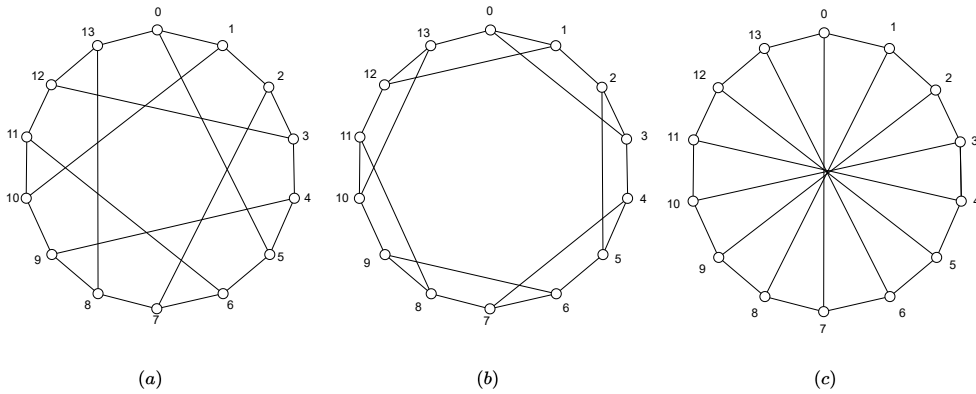


**Figura 5.5.** Aplicação da decomposição 4-soma. O item (a) representa a presilha redutível usada como entrada para o algoritmo. Os grafos (b), (d), (e), (f) e (g) são as presilhas irredutíveis obtidas desta decomposição 4-soma.

Para testarmos a função que verifica se um grafo é isomorfo ao grafo de Heawood, usamos apenas grafos com 14 vértices, caso contrário, seria trivial verificarmos se um grafo com número de vértices diferente de 14 é isomorfo ao grafo de Heawood.

Nos resultados obtidos para os grafos de entrada descritos na Figura 5.6 só o grafo do item (a) foi verificado como isomorfo ao grafo de Heawood.





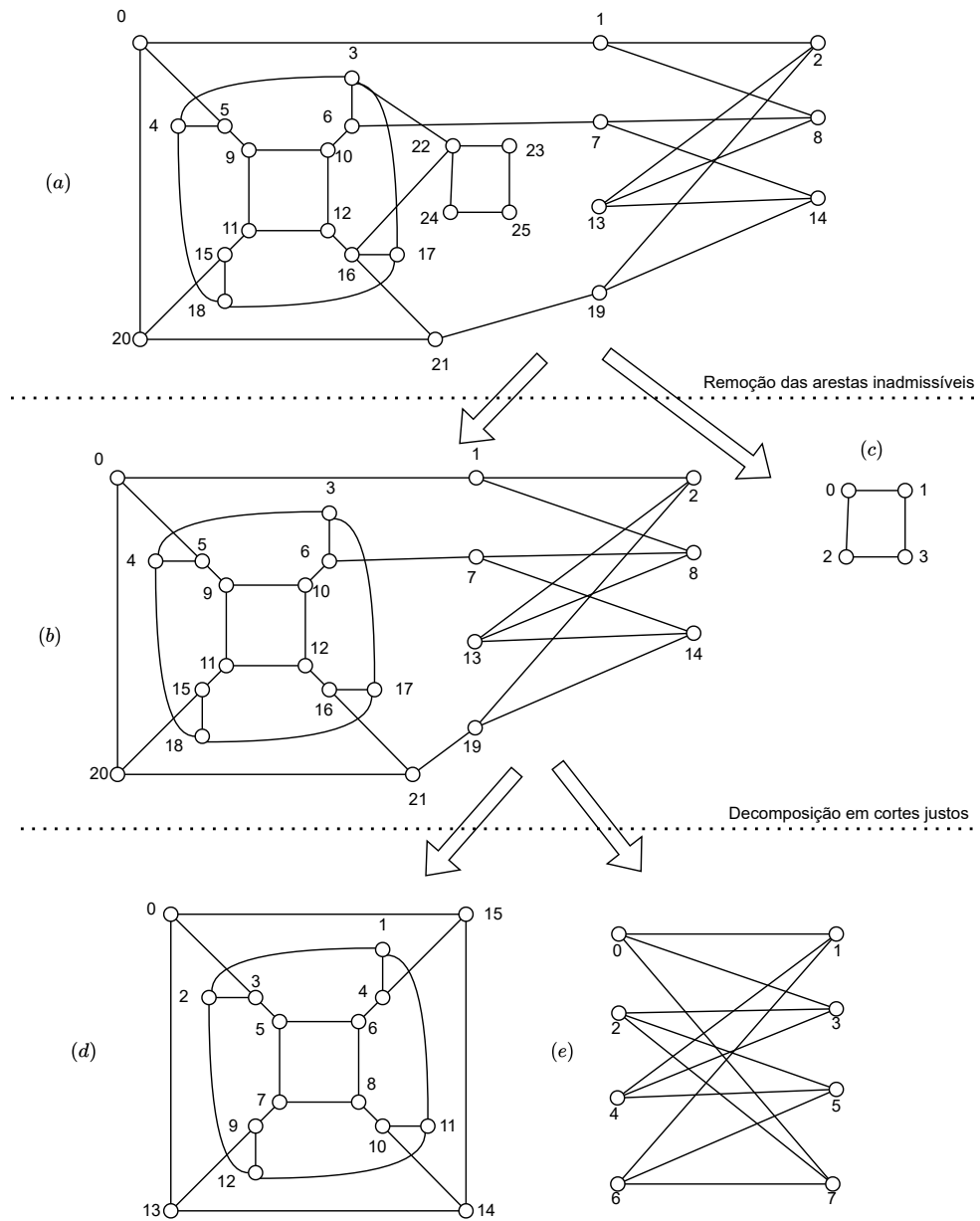
**Figura 5.6.** (a) Um grafo isomorfo ao grafo de Heawood. (b) Um grafo com 14 vértices e 21 arestas, 3 regular, bipartido e não isomorfo ao grafo de Heawood. (c) Um outro grafo com 14 vértices e 21 arestas, 3 regular, bipartido e não isomorfo ao grafo de Heawood.

## 5.5 Verificação da propriedade pfaffiana

Para a realização deste teste foram criados manualmente os grafos de entrada. O grafo mostrado no item (a) da Figura 5.7 foi dado como entrada para a função que verifica se um grafo é pfaffiano. A primeira etapa que o algoritmo realizou foi remover as arestas inadmissíveis, produzindo as componentes cobertas por emparelhamentos (b) e (c). Em seguida foi realizada a decomposição em cortes justos do grafo do item (b), produzindo como resultado as presilhas dos itens (d) e (c). Como a componente (c) não possui cortes justos, a decomposição em cortes justos não foi aplicado. O grafo do item (e) é uma presilha irreduzível, já o grafo do item (d) é uma presilha redutível. Na Figura 5.8, que é continuação da Figura 5.7, podemos ver que o algoritmo aplicou a decomposição 4-soma na presilha redutível do item (d), gerando as presilhas irreduzíveis (f), (g) e (h). Como o grafo da item (e) da Figura 5.8 é uma presilha irreduzível, não foi aplicado a decomposição 4-soma. Como não há mais reduções a serem aplicadas, o algoritmo verifica se todas as presilhas irreduzíveis obtidas descritas na Figura 5.9 são pfaffianas. Como neste caso todas as presilhas da Figura 5.9 são pfaffianas, o grafo de entrada também é pfaffiano.

## 5.6 Análise da complexidade

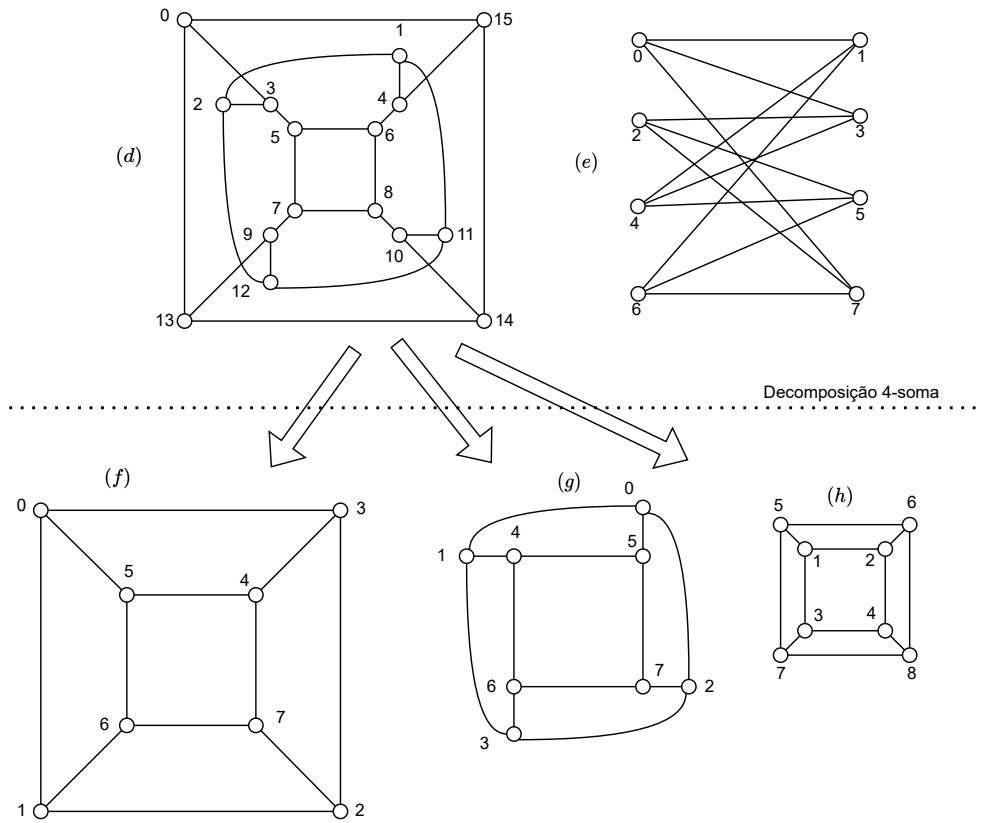
Para entendermos o comportamento assintótico do tempo de execução do algoritmo MRST implementado, chamamos a função que verifica se um grafo é pfaffiano para vários grafos aleatórios com densidades constantes. As Figuras 5.10, 5.11, 5.12, 5.13, 5.14 e 5.15 nos mostram algumas amostras dos grafos encontrados na geração dos grafos



**Figura 5.7.** Aplicação da verificação da propriedade pfaffiana do grafo descrito no item (a), onde podemos observar as etapas de remoção das arestas inadmissíveis e a decomposição em cortes justos.

aleatórios. Esses grafos aleatórios foram gerados da seguinte forma: fixa-se o número de vértices e para cada número de vértice encontramos o número de arestas (por exemplo usando a densidade  $m = 2n$ ). Para cada grafo  $G$  com  $n$  vértices e  $m$  arestas a ser gerado, sorteamos pares de vértices e os adicionamos no grafo  $G$  caso esses pares de vértices não foram adicionados anteriormente até atingirmos um número  $m$  de arestas adicionadas em  $G$ .

As Figuras 5.10, 5.11, 5.12, 5.13, 5.14 e 5.15 nos mostram o tempo de execução da

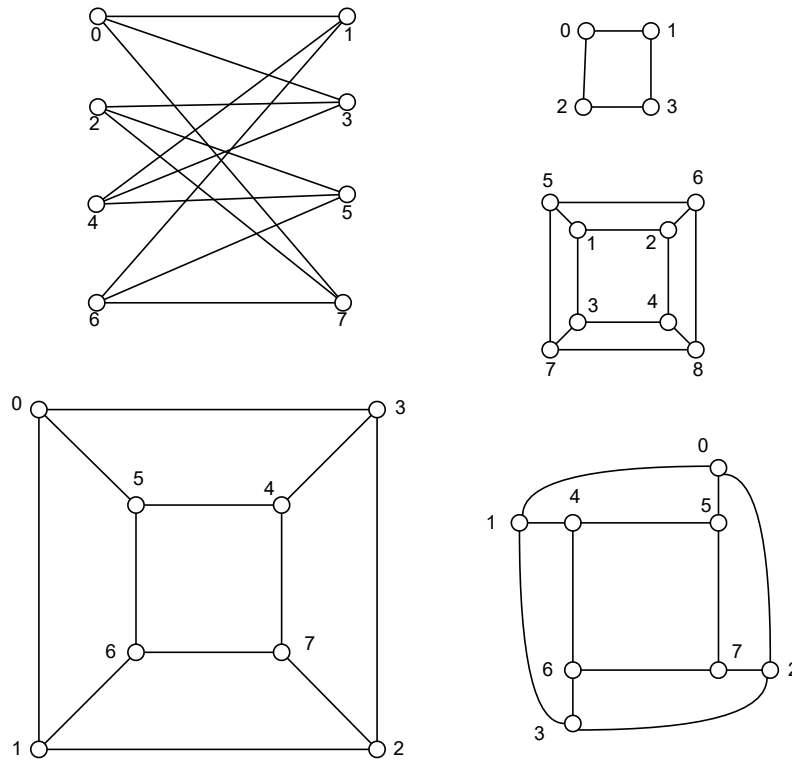


**Figura 5.8.** Continuação da verificação da propriedade pfaffiana da Figura 5.7. Podemos observar a decomposição 4-soma realizada pelo algoritmo na presilha redutível do item (d) gerando as presilhas irreduzíveis (f), (g) e (h).

implementação do algoritmo MRST representados por gráficos com densidades constantes de  $m = 1.5n$ ,  $m = 2n$ ,  $m = 4n$ ,  $m = 10n$ ,  $m = n^{1.5}$  e  $m = 0.1n^2$ . O eixo  $n$  representa o número de vértices e o eixo  $t$  o tempo em segundos. A constante  $c$  da função do tempo  $c \cdot nm^3$  encontrada é  $5.5 \cdot 10^{-26}$ . Apenas a última figura teve a sua constante recalculada para  $10^{-23}$  devido a quantidade de testes realizados. O motivo da quantidade de testes ser pequena para os grafos com densidade maior é devido à demora na geração dos grafos aleatórios.

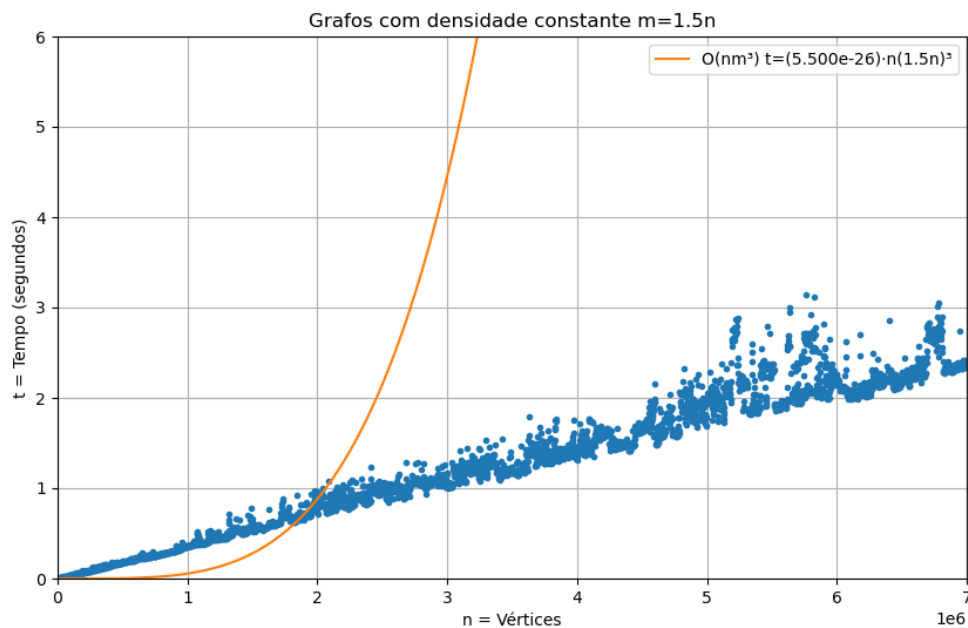
## 5.7 Conclusão

Com a realização dos testes, foi observado que de fato a complexidade  $O(nm^3)$  é um limite superior, mostrando que a implementação tem um comportamento assintótico dentro do esperado. Os resultados mostraram que o comportamento do tempo de execução foi muito satisfatório para o nível de reduções e decomposições existentes. Cabe ressaltar que não necessariamente todos os grafos dos testes passaram por todas

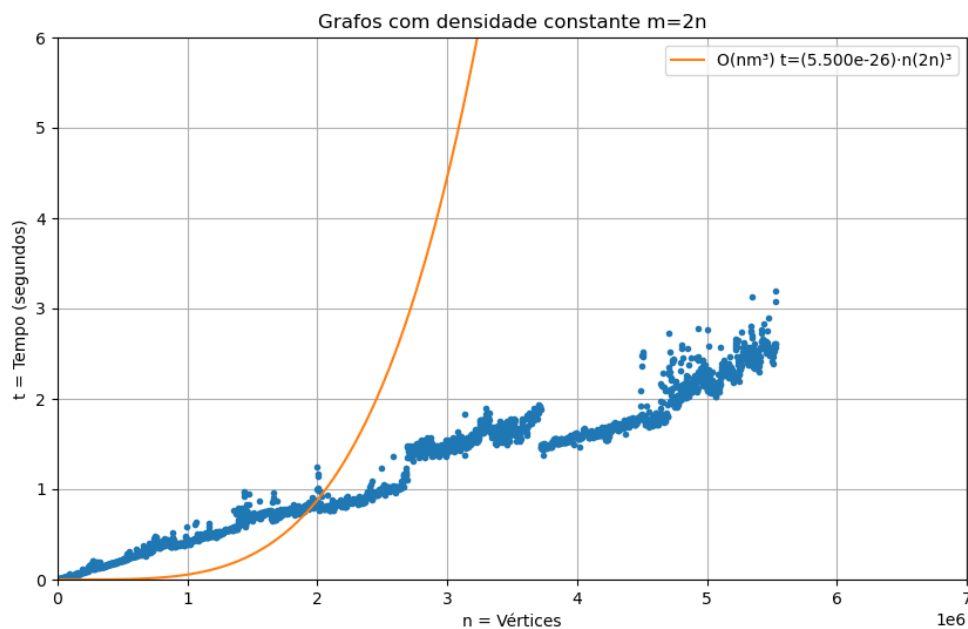


**Figura 5.9.** presilhas irreduzíveis obtidas através da aplicação do algoritmo que verifica a propriedade pfaffiana. Como todas essas presilhas irreduzíveis são pfaffianas, logo o grafo de entrada representada pelo item (a) da Figura 5.7 é pfaffiana.

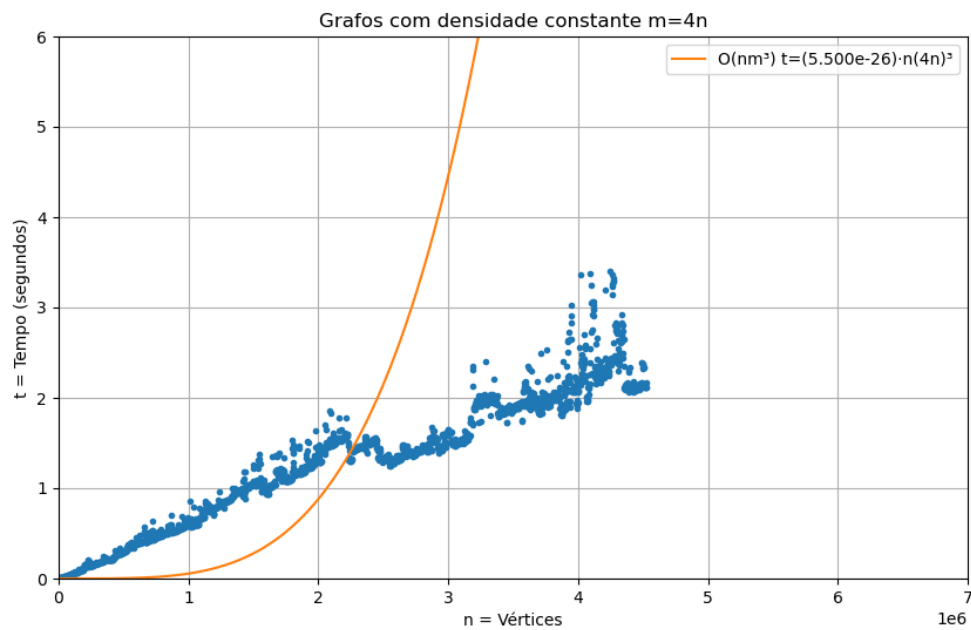
as reduções e decomposições para a verificação da propriedade pfaffiana, uma vez que a procura dos grafos pfaffianos é feita aleatoriamente.



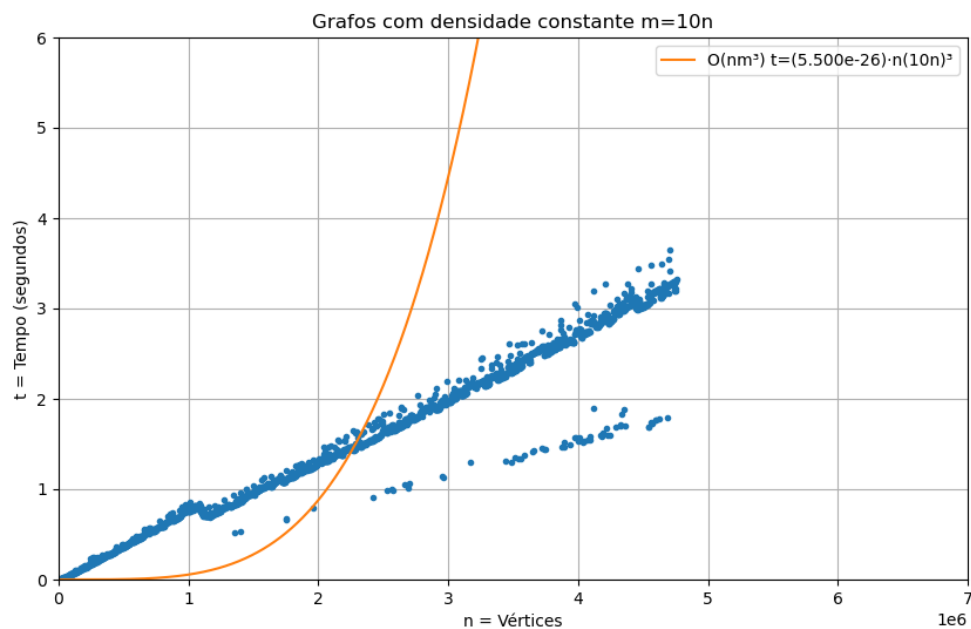
**Figura 5.10.** Gráfico dos tempos de execuções para grafos obtidos com densidade constante  $m = 1.5n$  em função do número de vértices.



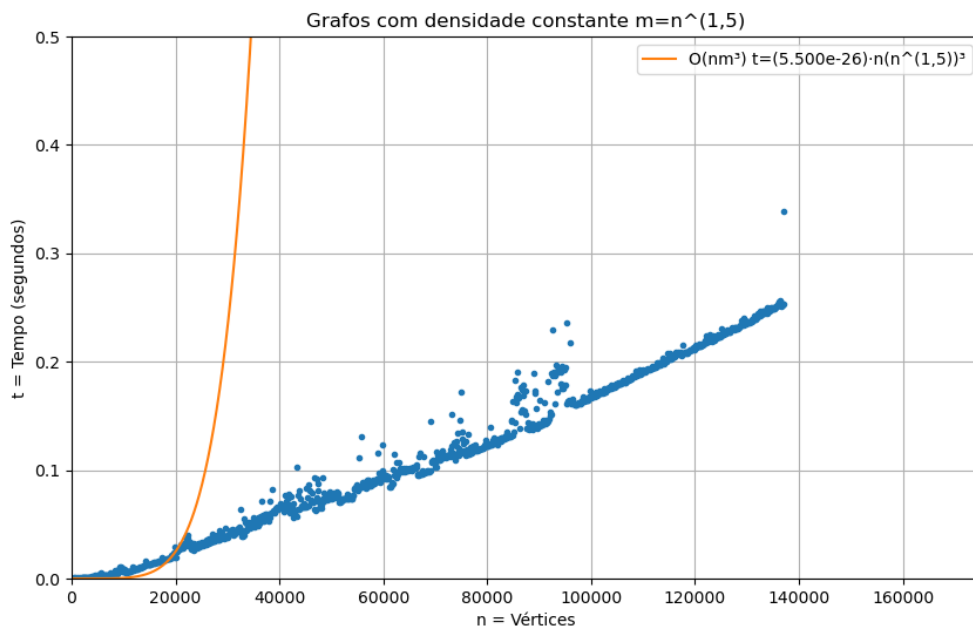
**Figura 5.11.** Gráfico dos tempos de execuções para grafos obtidos com densidade constante  $m = 2n$  em função do número de vértices.



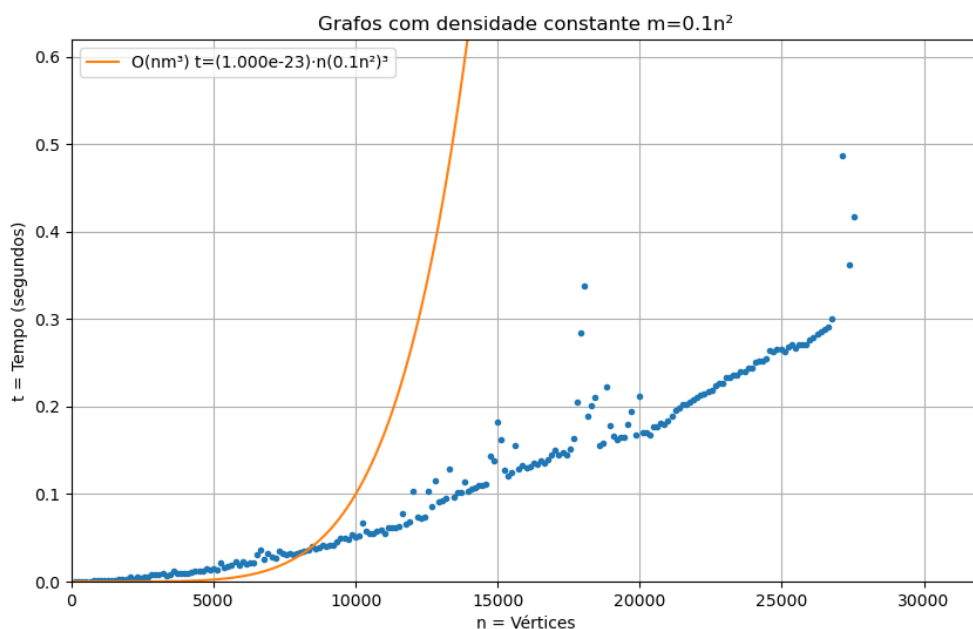
**Figura 5.12.** Gráfico dos tempos de execuções para grafos obtidos com densidade constante  $m = 4n$  em função do número de vértices.



**Figura 5.13.** Gráfico dos tempos de execuções para grafos obtidos com densidade constante  $m = 10n$  em função do número de vértices.



**Figura 5.14.** Gráfico dos tempos de execuções para grafos obtidos com densidade constante  $m = n^{1,5}$  em função do número de vértices.



**Figura 5.15.** Gráfico dos tempos de execuções para grafos obtidos com densidade constante  $m = 0.1n^2$  em função do número de vértices.

# Referências Bibliográficas

- Berge, C. (1957). Two theorems in graph theory. *Proceedings of the National Academy of Sciences of the United States of America*, 43:842–844.
- Bondy, A. & Murty, U. (2008). Graph theory (graduate texts in mathematics 244).
- Boyer, J. & Myrvold, W. (2004). On the cutting edge: Simplified  $O(n)$  planarity by edge addition. *Journal of Graph Algorithms and Applications*, 8(3):241–273.
- de Carvalho, M. H.; Lucchesi, C. L. & Murty, U. S. (2005). On the number of dissimilar pfaffian orientations of graphs. *RAIRO - Theoretical Informatics and Applications*, 39(1):93–113.
- Fisher, M. E. (1961). Statistical mechanics of dimers on a plane lattice. *Physical Review*, 124(6):1664–1672.
- Frobenius, G. (1917). *Über zerlegbare determinanten*. Sitzungsber. Königl. Preuss. Akad. Wiss., XVIII (1917), 274–277.
- Hall, P. (1935). On representatives of subsets. *J. London Math. Soc.*, 10:26–30.
- Hopcroft, J. E. & Karp, R. M. (1973). An  $n^{5/2}$  algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, 2(4):225–231.
- Kasteleyn, P. (1967). Graph theory and crystal physics. *Graph theory and theoretical physics*, pp. 43–110.
- Kasteleyn, P. W. (1961). The statistics of dimers on a lattice: I. the number of dimer arrangements on a quadratic lattice. *Physica*, 27(12):1209–1225.
- Kasteleyn, P. W. (1963). Dimer statistics and phase transitions. *Journal of Mathematical Physics*, 4(2):287–293.
- Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2:83–97.



- Kuratowski, C. (1930). Sur le problème des courbes gauches en topologie. *Fundamenta Mathematicae*, 15(1):271–283.
- Little, C. H. C. (1974). *An extension of Kasteleyn's method of enumerating the 1-factors of planar graphs*, pp. 63–72. Springer Berlin Heidelberg.
- Little, C. H. C. (1975). A characterization of convertible  $(0, 1)$ -matrices. *Journal of Combinatorial Theory, Series B*, 18(3):187–208.
- Little, C. H. C. & Rendl, F. (1991). Operations preserving the pfaffian property of a graph. *Journal of the Australian Mathematical Society. Series A. Pure Mathematics and Statistics*, 50(2):248–257.
- Lovász, L. & Plummer, M. D. (1986). *Matching theory*, volume Number 29 in Annals of Discrete Mathematics. Elsevier Science.
- Lovász, L. (1987). Matching structure and the matching lattice. *Journal of Combinatorial Theory, Series B*, 43:187–222.
- McCuaig, W. (2004). Pólya's permanent problem. *The electronic journal of combinatorics*, pp. R79–R79.
- Miranda, A. A. A. (2006). *Orientações Pfaffianas e o furtivo grafo de Heawood*.
- Miranda, A. A. A. (2009). *Grafos pfaffianos e problemas relacionados*. PhD thesis, IC-Unicamp.
- Morrison, R. T. & Boyd, R. N. (1973). Organic chemistry. *Allyn and Bacon, Toronto*, third edition.
- Robertson, N.; Seymour, P. D. & Thomas, R. (1999). Permanents, pfaffian orientations, and even directed circuits. *Annals of Mathematics*, 150(3):929–975.
- Samuelson, P. A. (1971). Foundations of economic analysis. *Atheneum, New York*.
- Schrijver, A. (2003). *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media.
- Temperley, H. N. & Fisher, M. E. (1961). Dimer problem in statistical mechanics-an exact result. *Philosophical Magazine*, 6(8):1061–1063.
- Tutte, W. T. (1947). The factorization of linear graphs. *Journal of the London Mathematical Society*, (22):107–111.
- Tutte, W. T. (1998). *Graph theory as I have known it*, volume 11. Clarendon Press.