



**COPPE/UFRJ**

CARACTERIZAÇÕES E RECONHECIMENTO DE GRAFOS BIPARTIDOS  
CORDAIS

Cristiane Barbosa da Cruz

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Márcia Rosana Cerioli


Rio de Janeiro  
Outubro de 2009

CARACTERIZAÇÕES E RECONHECIMENTO DE GRAFOS BIPARTIDOS  
CORDAIS

Cristiane Barbosa da Cruz


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

Examinada por:



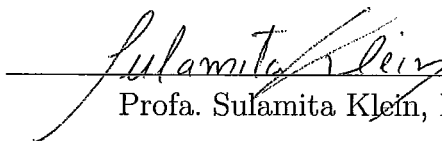
---

Profa. Márcia Rosana Cerioli, D.Sc.



---

Profa. Cristina Gomes Fernandes, Ph.D.



---

Profa. Sulamita Klein, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

OUTUBRO DE 2009

Cruz, Cristiane Barbosa da

Caracterizações e Reconhecimento de Grafos Bipartidos Cordais/Cristiane Barbosa da Cruz. – Rio de Janeiro: UFRJ/COPPE, 2009.

X, 70 p.: il.; 29,7cm.

Orientador: Márcia Rosana Cerioli

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2009.

Referências Bibliográficas: p. 68 – 70.

1. Caracterização de classes de grafos. 2. Grafo bipartido cordal. 3. Problema de reconhecimento. I. Cerioli, Márcia Rosana. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

*A Fabiano Oliveira, meus esposo  
e a Melissa, nossa filha.*

# Agradecimentos

A Deus, por todas as oportunidades, derrotas e conquistas que me concedeu.

Ao meu amigo e amado esposo, pelo companheirismo, dedicação e empenho que permitiram que este trabalho fosse desenvolvido.

Aos meus pais, pela minha formação e apoio.

Ao CEPTEL que, nas pessoas de Maria Elvira Piñeiro Maceira e Vitor Silva Duarte, incentivou minha vida acadêmica.

Às Professoras Cristina e Sulamita, por participarem da banca de defesa desta dissertação, agregando valor inestimável a este trabalho.

Em especial, à minha orientadora, Professora Márcia Cerioli, pelo incentivo desde a graduação, pela excelente orientação e por servir para mim de exemplo a se seguir. Muito obrigada por não desistir dessa orientação mesmo diante de tantas intercorrências. Os meus sinceros agradecimentos.

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

## CARACTERIZAÇÕES E RECONHECIMENTO DE GRAFOS BIPARTIDOS CORDAIS

Cristiane Barbosa da Cruz

Outubro/2009

Orientador: Márcia Rosana Cerioli

Programa: Engenharia de Sistemas e Computação

O estudo dos grafos bipartidos cordais foi motivado originalmente por suas aplicações em problemas de diversos contextos, como naqueles relacionados a matrizes não-simétricas, eliminação Gaussiana de matrizes, programação inteira e análise de matrizes. Tais estudos foram apresentados por diversos autores, tanto do ponto de vista teórico quanto prático. Os grafos bipartidos cordais são semelhantes aos grafos cordais em muitos aspectos. Os grafos cordais formam uma importante classe de grafos havendo inúmeros trabalhos a seu respeito.

Como os grafos cordais, os grafos bipartidos cordais são ricos em estrutura e muitos resultados originam-se de seu estudo. Sua importância está para a classe dos grafos bipartidos assim como a importância dos grafos cordais está para a classe dos grafos em geral.

O principal objetivo deste trabalho é realizar um estudo compreensivo da classe dos grafos bipartidos cordais, consolidando as caracterizações e algoritmos de reconhecimento mais eficientes existentes para esta classe em um único trabalho. Para tanto, é necessário conhecer a resolução de problemas relacionados a este problema, como a obtenção de ordenações lexicográficas duplas de matrizes, bem como suas soluções. As provas originais foram em sua maior parte reescritas de modo a obter uma padronização de notação em todo este trabalho.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

## CHARACTERIZATIONS AND RECOGNITION OF BIPARTITE CHORDAL GRAPHS

Cristiane Barbosa da Cruz

October/2009

Advisor: Márcia Rosana Cerioli

Department: Systems Engineering and Computer Science

The study of the bipartite chordal graphs was originally motivated by their applications in problems from different contexts, such as in those related to non-symmetric matrices, Gaussian matrices elimination, integer programming, and matrix analysis. Such studies were presented by several authors, both from the theoretical and application viewpoints. Bipartite chordal graphs are similar to chordal graphs in many aspects. Chordal graphs form an important class of graphs to which many papers were devoted.

As chordal graphs, bipartite chordal graphs are rich in structure and many results are derived from their study. Its importance is to the class of bipartite graphs as the class of chordal graphs is to the class of general graphs.

The main goal of this work is to carry out a comprehensive study of the class of bipartite graphs, consolidating the known characterizations and most efficient recognition algorithms for this class of graphs in a unique work. In order to do that, it is necessary to introduce related problems, as the double lexicographical ordering of matrices, and their solution. The original proofs in its major part have been rewritten in order to be present in a uniform notation throughout this work.

# Sumário

Lista de Figuras	x
<b>1 Introdução</b>	<b>1</b>
1.1 Definições básicas . . . . .	2
1.2 Alguns exemplos . . . . .	7
<b>2 Problemas Relacionados à Caracterização e Reconhecimento dos Grafos Bipartidos Cordais</b>	<b>9</b>
2.1 Ordenação lexicográfica dupla . . . . .	9
2.2 Algoritmos de ordenação lexicográfica dupla . . . . .	15
2.2.1 O algoritmo de Lubiw, Paige e Tarjan . . . . .	15
2.2.2 O algoritmo de Spinrad . . . . .	20
2.3 Matrizes livres de gama . . . . .	29
2.4 Grafos bipartidos de eliminação perfeita . . . . .	30
<b>3 Propriedades, Caracterizações e Aplicações dos Grafos Bipartidos Cordais</b>	<b>35</b>
3.1 Propriedades gerais dos bipartidos cordais . . . . .	36
3.2 Caracterizações do grafos bipartidos cordais . . . . .	43
3.3 Aplicação prática . . . . .	49
3.3.1 Problema da eliminação Gaussiana perfeita . . . . .	49
3.3.2 Eliminação Gaussiana perfeita e os grafos bipartidos cordais . . . . .	51
<b>4 Reconhecimento dos Grafos Bipartidos Cordais</b>	<b>54</b>
4.1 Reconhecimento de grafos bipartidos cordais . . . . .	54
4.2 Reconhecimento de matrizes livres de $\Gamma$ . . . . .	58



5	Comparação entre os Grafos Cordais e Grafos Bipartidos Cordais	62
6	Conclusão	66
	Referências Bibliograficas	68

# Lista de Figuras

1.1	O grafo de Petersen é um exemplo de grafo 3-regular. . . . .	4
1.2	$\hat{C}_6 = 3K_2$ e $\hat{C}_8 = C_8$ . . . . .	5
1.3	Hipercubos. . . . .	7
2.1	Triangularização desejada de 1's e refinamento do particionamento. . .	22
2.2	Procedimento de Refinamento . . . . .	24
3.1	Exemplo de grafo bipartido cordal separável. . . . .	37
5.1	Subgrafo $B_c$ de $B(G)$ . . . . .	64
5.2	Exemplo de um grafo cordal $G$ cujo grafo $B(G)$ não é bipartido cordal. .	65

# Capítulo 1

## Introdução

Um grafo é *bipartido cordal* se é bipartido e cada um de seus ciclos de tamanho maior ou igual a 6 tem uma corda. Como um grafo bipartido só possui ciclos induzidos de tamanho par, um grafo bipartido cordal só pode possuir ciclos induzidos de tamanho 4.

Introduzidos em 1978 por Golubic e Goss [13], o estudo dos grafos bipartidos cordais foi motivado originalmente por sua aplicação em matrizes não-simétricas. As aplicações em eliminação Gaussiana de matrizes esparsas foram descritas por Bakonyi e Bono em 1997 [3], em programação inteira por Hoffman, Kolen e Sakarovitch em 1985 [14], e em análise de matrizes por Johnson e Whitney em 1991 [16] e por Johnson e Miller em 1997 [15]. Uma descrição sucinta destas aplicações encontra-se na Seção 3.3.

Note que em geral os grafos bipartidos cordais não são cordais. Por exemplo, o  $C_4$  é bipartido cordal mas não é cordal. Naturalmente, também existem grafos cordais que não são bipartidos cordais como o  $C_3$ , por exemplo. Portanto, um grafo bipartido cordal não é necessariamente um grafo bipartido e cordal, como poderia ser imaginado baseando-se no nome dado à classe.

O nome bipartido cordal vem da idéia de que estes são os grafos bipartidos que possuem como característica principal aquela análoga a dos grafos cordais: um ciclo induzido possui o menor tamanho possível.

Os grafos bipartidos cordais são semelhantes aos grafos cordais em muitos outros aspectos. Para mencionar um exemplo por ora, considere as seguintes caracterizações:

- (i) Um grafo  $G$  é cordal se e somente se todo separador de vértices minimal de  $G$  é uma clique; e
- (ii) Um grafo  $G$  é bipartido cordal se e somente se todo separador de arestas minimal (definido mais a frente) de  $G$  é uma biclique.

De tais semelhanças, originam-se resultados análogos para as classes. No entanto, nem todo resultado de uma classe migra naturalmente para a outra. Às vezes, as demonstrações possuem raciocínios distintos. Em outros casos, os resultados simplesmente não se aplicam para a outra classe. Como os grafos cordais, os grafos bipartidos cordais são ricos em estrutura e muitos resultados interessantes originam-se de seu estudo. Sua importância para a classe dos grafos bipartidos é análoga a importância dos grafos cordais para a classe de todos os grafos.

O objetivo deste trabalho é fazer um estudo compreensivo da classe dos grafos bipartidos cordais, consolidando as caracterizações e algoritmos de reconhecimento mais eficientes existentes para esta classe de grafos. Como veremos adiante no trabalho, para tanto é necessário conhecer a resolução de problemas relacionados, como a obtenção de ordenações lexicográficas duplas de matrizes.

O restante do trabalho está assim organizado. A Seção 1.1 fornece as definições básicas da teoria dos grafos que serão utilizadas ao longo do trabalho. O Capítulo 2 apresenta problemas relacionados aos grafos bipartidos cordais. O Capítulo 3 apresenta propriedades, caracterizações e aplicações dos grafos bipartidos cordais. O Capítulo 4 descreve os algoritmos mais eficientes conhecidos para o problema de reconhecimento dos grafos bipartido cordais. O Capítulo 5 consiste de um estudo comparativo entre os grafos bipartidos cordais e os grafos cordais, relacionando resultados similares entre as classes e evidenciando suas diferenças. Finalmente, o Capítulo 6 traz um resumo dos temas abordados neste trabalho.

## 1.1 Definições básicas

A seguir apresentamos os conceitos e notações da teoria de grafos que serão utilizados no decorrer do trabalho. Procuramos seguir as convenções usuais [4].

**Definição 1.1.** Um *grafo*  $G$  é um par ordenado  $(V(G), E(G))$ , onde  $V(G)$  é um conjunto finito não-vazio de elementos denominados *vértices* e  $E(G)$  é um conjunto

finito de elementos denominados *arestas*. Cada aresta é um par não ordenado de vértices distintos.

Note que estamos desconsiderando a possibilidade de um grafo possuir *laços* (uma aresta formada por um par de vértices idênticos) e *multiarestas* (arestas distintas formadas por pares de vértices idênticos).

A menos que seja dito de forma diferente no contexto, denotaremos por  $n = |V(G)|$  o número de vértices de um dado grafo  $G$  e por  $m = |E(G)|$  seu número de arestas.

**Definição 1.2.** Se  $e = (u, v)$  é uma aresta de um grafo, denotaremos  $e$  por simplicidade como  $uv$ .

**Definição 1.3.** Se  $e = uv$  é uma aresta de um grafo, então dizemos que  $e$  é *incidente* a  $u$  e a  $v$  ou ainda que  $e$  *liga* os vértices  $u$  e  $v$ .

**Definição 1.4.** Dois vértices  $u$  e  $v$  são ditos *adjacentes* quando existir uma aresta que os liga.

**Definição 1.5.** Duas arestas distintas  $ab$  e  $cd$  são ditas *adjacentes* quando  $\{a, b\} \cap \{c, d\} \neq \emptyset$ .

**Definição 1.6.** A *vizinhança* de  $v$  em um grafo  $G$  é o conjunto  $N_G(v) = \{u \mid uv \in E(G)\}$ . Podemos denotar a vizinhança de  $v$  simplesmente por  $N(v)$  quando o grafo  $G$  for claro no contexto.

**Definição 1.7.** A *vizinhança fechada* de  $v$  em um grafo  $G$  é definida por  $N_G[v] = N_G(v) \cup \{v\}$ . De maneira similar, denotamos a vizinhança fechada de  $v$  simplesmente por  $N[v]$  quando o grafo  $G$  for claro no contexto.

**Definição 1.8.** O *grau*  $d_G(v)$  de um vértice no grafo  $G$  é o número de arestas de  $G$  incidentes a  $v$ . Podemos denotar o grau de  $v$  simplesmente por  $d(v)$  quando o grafo  $G$  for claro no contexto.

**Definição 1.9.** Um vértice  $v$  é dito *isolado* num grafo  $G$  se  $d(v) = 0$ .

**Definição 1.10.** Um grafo é dito *k-regular* (ou simplesmente *regular*) quando todos os seus vértices têm o mesmo grau  $k$ .

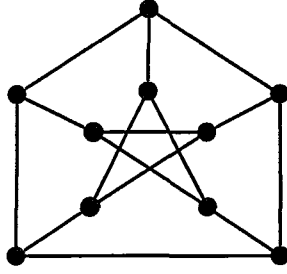


Figura 1.1: O grafo de Petersen é um exemplo de grafo 3-regular.

**Exemplo 1.11.** A Figura 1.1 ilustra um exemplo de grafo 3-regular.

**Definição 1.12.** Um *caminho*  $C$  em um grafo  $G$  é uma sequência de vértices distintos  $C = v_1, \dots, v_n$  onde  $v_i v_{i+1} \in E(G)$  para  $i = 1, \dots, n - 1$ .

O tamanho do caminho  $C = v_1, \dots, v_n$ , denotado por  $|C|$ , é  $n - 1$ .

**Definição 1.13.** Um *ciclo* em um grafo  $G$  é uma sequência de vértices  $(v_1, \dots, v_n)$ , onde  $n \geq 3$ ,  $(v_1, \dots, v_n)$  é um caminho e  $v_1 v_n \in E(G)$ .

**Definição 1.14.** Um grafo  $G$  é dito *bipartido* quando  $V(G)$  pode ser particionado em dois conjuntos  $X$  e  $Y$  tais que toda aresta de  $G$  liga um vértice na parte  $X$  e outro na parte  $Y$ . Frequentemente, utilizaremos a notação  $G = (X, Y, E)$  para nos referir ao grafo bipartido com partição  $X$  e  $Y$  e  $E(G) = E$ .

**Observação 1.15.** Seja  $G = (X, Y, E)$  um grafo bipartido e  $v_1, \dots, v_n$  um caminho em  $G$ . Se  $v_1, v_n \in X$ , então  $n$  é ímpar. Se  $v_1 \in X$  e  $v_n \in Y$ , então  $n$  é par.

**Definição 1.16.** Um *grafo completo*  $K_n$  é um grafo com  $n$  vértices tal que existe aresta ligando quaisquer dois de seus vértices.

**Definição 1.17.** Um conjunto de vértices  $V \subseteq V(G)$  *induz* o grafo  $G[V] = (V, E)$  tal que para todo  $u, v \in V$ ,  $uv \in E$  se e somente se  $uv \in E(G)$ .

Diz-se que  $G[V]$  é o subgrafo induzido de  $G$  pelos vértices de  $V$ .

**Definição 1.18.** Um conjunto de arestas  $E \subseteq E(G)$  *induz* o grafo  $G[E] = (V, E)$  tal que para  $V = \{v \mid \exists e = uv \in E\}$ .

Diz-se que  $G[E]$  é o subgrafo induzido de  $G$  pelas aresta de  $E$ .

**Definição 1.19.** Um grafo  $H$  é um *subgrafo induzido* de  $G$ , denotado por  $H \subset G$ , se existe  $S$  tal que  $H = G[S]$  com  $S \subset V(G)$  ou  $S \subset E(G)$ .

**Definição 1.20.** Uma *clique* de um grafo é um subconjunto de vértices deste grafo que induz um grafo completo.

**Definição 1.21.** Dois grafos  $G$  e  $H$  são *isomorfos* se existe uma função bijetiva  $f : V(G) \rightarrow V(H)$  que preserve as adjacências, ou seja,  $uv \in E(G)$  se e somente se  $f(u)f(v) \in E(H)$ .

**Definição 1.22.** Seja  $G$  um grafo. O *complemento*  $\overline{G}$  de  $G$  é o grafo tal que  $V(\overline{G}) = V(G)$  e  $E(\overline{G}) = \{uv \mid u, v \in V(G), u \neq v \text{ e } uv \notin E(G)\}$ .

**Definição 1.23.** Seja  $G = (X, Y, E)$  um grafo bipartido. O *complemento bipartido*  $\hat{G}$  de  $G$  é o grafo bipartido  $(X, Y, \hat{E})$  tal que  $\hat{E} = \{xy \mid x \in X, y \in Y \text{ e } xy \notin E\}$ . A Figura 1.2 mostra o complemento bipartido do  $C_6$  e do  $C_8$ .

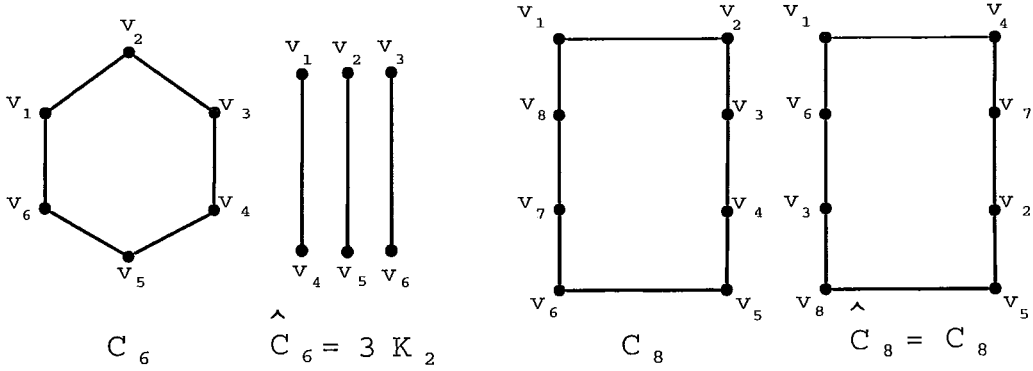


Figura 1.2:  $\hat{C}_6 = 3K_2$  e  $\hat{C}_8 = C_8$ .

**Definição 1.24.** Sejam  $G$  e  $H$  grafos. Dizemos que  $G$  é *livre* de  $H$  se  $G$  não contém um subgrafo induzido isomorfo a  $H$ .

**Definição 1.25.** Um grafo bipartido  $G = (X, Y, E)$  é *completo* quando possuir uma aresta ligando cada par de vértices  $x \in X, y \in Y$ . Denotamos  $G$  por  $K_{n,m}$ , onde  $n = |X|$  e  $m = |Y|$ .

**Definição 1.26.** Uma *biclique* de um grafo bipartido é um conjunto de vértices deste grafo que induz um grafo bipartido completo.

**Exemplo 1.27.** Os grafos bipartidos completos  $K_{m,n}$  são bipartidos cordais.

**Definição 1.28.** Um grafo  $G$  é dito *conexo* se existir um caminho entre qualquer par de vértices pertencentes a  $V(G)$ , e *desconexo* caso contrário.

Uma *componente conexa* de um grafo  $G$  é um subgrafo induzido conexo maximal.

Sempre que não for mencionado o contrário, consideraremos  $G$  um grafo conexo.

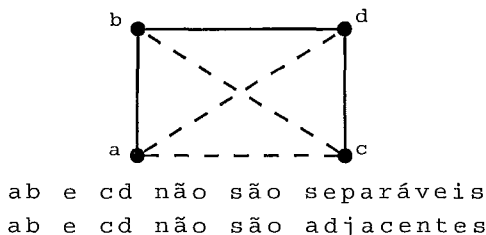
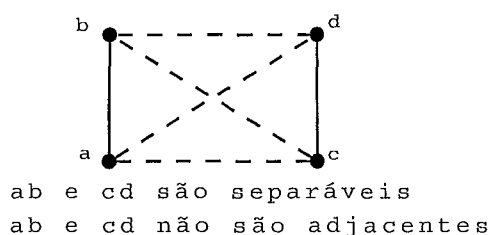
**Definição 1.29.** Se  $S$  e  $S'$  são conjuntos tais que  $S' \subseteq S$ , então  $S'$  é *maximal* em relação a uma propriedade  $P$  se  $S'$  satisfizer  $P$  e não existir  $S''$ ,  $S' \subset S'' \subseteq S$ , tal que  $S''$  satisfaça  $P$ .

**Definição 1.30.** Sejam  $S$  e  $S'$  conjuntos tais que  $S' \subseteq S$ . O conjunto  $S'$  é *minimal* em relação a uma propriedade  $P$  se  $S'$  satisfizer  $P$  e não existir  $S'' \subset S'$  tal que  $S''$  satisfaça  $P$ .

**Definição 1.31.** Seja  $G$  um grafo. Um conjunto  $S \subseteq V(G)$  é um *separador de vértices* em relação aos vértices não adjacentes  $a$  e  $b$ , se  $a$  e  $b$  estão em componentes conexas distintas de  $G - S$ .

**Definição 1.32.** Duas arestas  $ab$  e  $cd$  de  $G$  são ditas *separáveis* quando  $c, d \notin N[a] \cup N[b]$ .

Portanto, note que arestas separáveis não são adjacentes. O inverso não é verdadeiro: arestas não-adjacentes podem não ser separáveis.



**Definição 1.33.** Seja  $G$  um grafo. Um conjunto  $S \subseteq V(G)$  é um *separador de arestas* em relação às arestas separáveis  $ab$  e  $cd$ , se  $ab$  e  $cd$  estão em componentes conexas distintas de  $G - S$ .

**Observação 1.34.** Seja  $S$  um separador de arestas minimal em relação às arestas separáveis  $ab$  e  $cd$ . Então cada vértice de  $S$  é adjacente a algum vértice de cada uma das componentes conexas contendo  $ab$  e  $cd$ . Caso contrário  $S$  não seria minimal.



**Definição 1.35.** Um vértice  $v$  de um grafo  $G$  é *simplicial* se  $N[v]$  for uma clique em  $G$ .

**Definição 1.36.** Uma aresta  $vy$  de um grafo bipartido  $G$  é *bi-simplicial* se  $N[v] \cup N[y]$  é uma biclique.

**Observação 1.37.** A propriedade de ser uma aresta bi-simplicial é hereditária, isto é, se  $e \in E(G)$  é uma aresta bi-simplicial de  $G$  e  $H$  é um subgrafo induzido de  $G$  tal que  $e \in E(H)$ , então  $e$  também é bi-simplicial em  $H$ .

## 1.2 Alguns exemplos

**Definição 1.38.** Um *hipercubo* de dimensão  $k$ ,  $k \geq 1$ ,  $Q_k$  é um grafo tal que:

- O conjunto de vértices  $V(Q_k)$  é formado por todas as  $k$ -uplas binárias;
- Existe uma aresta entre dois vértices de  $Q_k$  se as  $k$ -uplas correspondentes aos vértices diferem em exatamente uma de suas coordenadas.

A Figura 1.3 ilustra os hipercubos  $Q_1$ ,  $Q_2$  e  $Q_3$ .

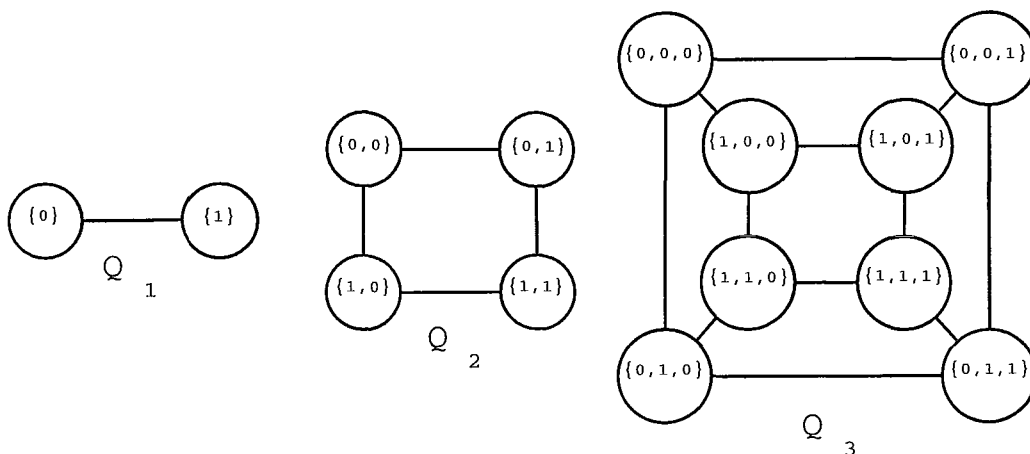


Figura 1.3: Hipercubos.

**Observação 1.39.** Um  $Q_k$  é um grafo  $k$ -regular com  $2^k$  vértices e  $k \times 2^{k-1}$  arestas.

*Demonstração.* Claramente  $Q_k$  é um grafo  $k$ -regular pois, dado um vértice  $v \in V(Q_k)$ , existem exatamente outros  $k$  vértices que diferem em apenas uma de suas

coordenadas sendo portanto são adjacentes a  $v$ . Logo, para cada  $v \in V(Q_k)$ ,  $|N(v)| = k$ .

Note que,  $|V(Q_k)|$  é igual ao número de  $k$ -uplas distintas constituídas por 0's e 1's. Para a formação das  $k$ -uplas constituídas por 0's e 1's existem duas possibilidades para cada uma das  $k$  coordenadas, então  $|V(Q_k)| = 2 \times \dots \times 2 = 2^k$ .

Quanto ao número de arestas  $|E(Q_k)|$ , se somássemos o total de vizinhos de cada vértice de  $Q_k$  então contabilizaríamos duas vezes cada aresta, uma para cada vértice que a define. Portanto,  $|E(Q_k)| = |V(Q_k)| \times (|N(v)| \div 2) = |V(Q_k)| \times (k \div 2) = 2^k \times (k \div 2) = k \times 2^{k-1}$ .  $\square$

**Observação 1.40.** Os hipercubos são bipartidos.

*Demonstração.* De fato, podemos construir as partes da bipartição da seguinte forma: os vértices cuja soma das coordenadas é par constituem uma das partes e os de soma ímpar, a outra parte. Observe que os vértices cuja soma das coordenadas é par não são adjacentes entre si. O mesmo ocorre entre os de soma ímpar.  $\square$

**Observação 1.41.** Apesar dos hipercubos serem bipartidos,  $Q_1$  e  $Q_2$  são os únicos hipercubos que são também bipartidos cordais.

*Demonstração.* Claramente por inspeção verifica-se que não existem ciclos induzidos de tamanho maior do que 4 nem em  $Q_1$ , nem em  $Q_2$ , sendo ambos portanto bipartidos cordais. Por outro lado, também por inspeção verifica-se que há um ciclo induzido de tamanho 6 em  $Q_3$  ( $(\{0, 0, 0\}, \{0, 0, 1\}, \{0, 1, 1\}, \{1, 1, 1\}, \{1, 1, 0\}, \{1, 0, 0\})$ , Figura 1.3). Além disso, note que  $Q_3 \subset Q_n$ , para cada  $n \geq 4$ .  $\square$

# Capítulo 2

## Problemas Relacionados à Caracterização e Reconhecimento dos Grafos Bipartidos Cordais

Neste capítulo fornecemos uma visão geral de três conceitos relacionados aos grafos bipartidos cordais: ordenação lexicográfica dupla, matrizes livres de gama e grafos bipartidos de eliminação perfeita. Estes conceitos, além de definirem problemas em si, são utilizados na literatura como auxílio na resolução de diversos problemas. Eles se relacionam de maneira particular com os grafos bipartidos cordais servindo de ferramenta para a obtenção de caracterizações e algoritmos de reconhecimento. Dado a importância destes conceitos neste trabalho, dedicamos este capítulo a mostrar sucintamente os resultados relevantes no nosso contexto.

Na Seção 2.1, introduzimos o problema da ordenação lexicográfica dupla de uma matriz real. Na Seção 2.3, apresentamos as matrizes livres de gama. Na Seção 2.4, apresentamos os grafos bipartidos de eliminação perfeita e sua relação com os grafos bipartidos cordais.

### 2.1 Ordenação lexicográfica dupla

Uma *ordenação lexicográfica dupla* de uma matriz real é uma ordenação das linhas e colunas desta matriz de tal forma que os vetores formados pelas linhas lidas da direita para a esquerda e pelas colunas lidas de baixo para cima se tornem

lexicograficamente ordenados.

O interesse em tal ordenação se deve ao fato de que o conceito de ordenação lexicográfica dupla é utilizado para provar e unificar resultados em várias classes de matrizes e grafos: matrizes totalmente balanceadas [2], grafos bipartidos cordais [14] e grafos fortemente cordais [18]. Em nosso caso, o interesse é ainda mais abrangente uma vez que os algoritmos de ordenação lexicográfica dupla compõem de forma essencial os algoritmos mais eficientes de reconhecimento dos grafos bipartidos cordais.

Mais especificamente, Spinrad utilizou a ordenação lexicográfica dupla de matrizes binárias como subrotina em algoritmos eficientes para reconhecer grafos de trapézio (cf. [19]) e arco-circulares (cf. [9]). Nessas aplicações, a ordenação lexicográfica dupla foi usada no complemento do grafo original.

**Definição 2.1.** Se  $M$  é uma matriz real  $m \times n$ , denote por  $M(i, *)$  o *vetor-linha*  $(M(i, 1), \dots, M(i, n))$  e por  $M(*, i)$  o *vetor-coluna*  $(M(1, i), \dots, M(m, i))$ .

**Definição 2.2.** Dados os vetores  $u = (u_1, \dots, u_n)$  e  $v = (v_1, \dots, v_n)$ , diremos que  $u$  é *menor* que  $v$  (resp.  $v$  é *maior* que  $u$ ), denotando por  $u < v$  (resp.  $v > u$ ) se, para algum  $1 \leq k \leq n$ ,  $u_k < v_k$  e  $u_i = v_i$  para todo  $k < i \leq n$ . De maneira alternativa,  $u < v$  se o reverso do vetor  $u$  é lexicograficamente menor do que o reverso do vetor  $v$ .

**Definição 2.3.** Uma *ordenação lexicográfica dupla* de  $M$  é uma matriz  $M'$  obtida a partir de  $M$  por uma sequência de permutações de linhas ou colunas de tal forma que  $M'(i, *) < M'(i + 1, *)$  para cada  $1 \leq i < m$  e  $M'(*, i) < M'(*, i + 1)$  para cada  $1 \leq i < n$ .

**Exemplo 2.4.** Seja  $M$  a matriz definida abaixo e  $M'$  sua ordenação lexicográfica dupla.

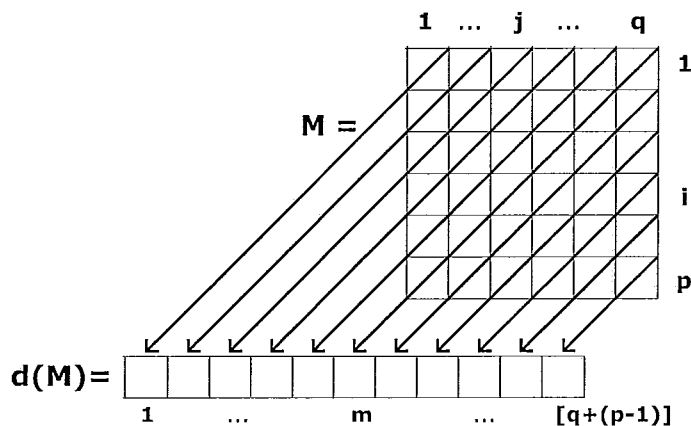
$$M = \begin{pmatrix} 2 & 3 & 4 \\ 1 & 4 & 3 \\ 4 & 8 & 3 \\ 2 & 3 & 3 \end{pmatrix} \quad M' = \begin{pmatrix} 3 & 2 & 3 \\ 4 & 2 & 3 \\ 3 & 1 & 4 \\ 3 & 4 & 8 \end{pmatrix}$$

Claramente, existe uma permutação das linhas de uma matriz  $M$  que tornam seus vetores-linha ordenados lexicograficamente. De maneira análoga, o mesmo vale

para a obtenção de uma ordenação dos vetores-coluna de  $M$ . Note que, ao se ordenar as linhas de  $M$ , a alteração da ordem de suas colunas pode comprometer a ordenação de linhas já feita. Isto leva à pergunta natural sobre a existência de uma ordenação lexicográfica dupla, respondida a seguir.

**Teorema 2.5** (Lubiw [18]). *Toda matriz real possui uma ordenação lexicográfica dupla.*

*Demonstração.* Seja  $M$  uma matriz real  $m \times n$ . Seja  $d_M$  o vetor associado a  $M$  de tamanho  $m + n - 1$  definido da seguinte maneira: para cada  $1 \leq k \leq m + n - 1$ ,  $d_M(k) = \sum_{i+j=k+1} M(i, j)$ . Isto é, cada componente de  $d_M$  é igual a soma dos valores de uma diagonal de  $M$ , como pode ser visto na figura a seguir:



Por exemplo, considere a matriz  $M'$  obtida de uma ordenação lexicográfica dupla da matriz  $M$  conforme o Exemplo 2.4. O vetor diagonal de  $M'$  é  $d = (3, 6, 8, 7, 8, 8)$ .

Suponha, com o propósito de encontrar um absurdo, que não exista uma ordenação lexicográfica dupla de  $M$ . Seja  $\mathcal{M}$  o conjunto das matrizes que podem ser obtidas a partir de  $M$  por uma sequência de permutações de linhas ou colunas. Naturalmente,  $\mathcal{M}$  é um conjunto finito. Seja  $M' \in \mathcal{M}$  a matriz que possua o maior vetor  $d_{M'}$  associado. Como  $M'$  não é uma ordenação lexicográfica dupla, então existem colunas ou linhas de  $M'$  fora de ordem.

Suponha que existam colunas em tal condição, e sejam  $1 \leq d < e \leq n$  tais que  $M'(*, d) > M'(*, e)$ . Seja  $M''$  a matriz obtida por  $M'$  trocando-se de posição as

colunas  $d$  e  $e$ . Seja  $k$  a maior linha de  $M'$  tal que  $M'(k, d) > M'(k, e)$ . Note que a maior componente onde há diferença de valores no vetor  $d(M'')$  em relação ao vetor  $d(M')$  é  $e + k - 1$ . Mais precisamente,  $d_{M''}(e + k - 1) = d_{M'}(e + k - 1) - M'(k, e) + M'(k, d)$ . Como  $M'(k, d) > M'(k, e)$ , então  $d_{M''} > d_{M'}$ , contrariando a escolha de  $M'$ . Uma contradição análoga pode ser derivada no caso de existirem linhas de  $M'$  fora de ordem.  $\square$

Para algumas classes de matrizes, as ordenações lexicográficas duplas podem aparecer sob uma forma especial. Por exemplo, mostraremos que uma permutação especial de linhas e colunas de uma matriz simétrica é uma ordenação lexicográfica dupla desta matriz.

**Definição 2.6.** Uma matriz real  $M$ ,  $n \times n$ , é dita *simétrica* quando  $M(i, j) = M(j, i)$ , para todo  $1 \leq i \leq j \leq n$ .

**Definição 2.7.** Uma *ordenação simétrica* de uma matriz simétrica  $M$  é uma matriz  $M'$  obtida a partir de  $M$  por uma sequência de permutações de linhas onde, a cada operação de permutação entre linhas  $i$  e  $j$ , permutam-se também as colunas  $i$  e  $j$  de  $M$ .

Note que toda ordenação simétrica de uma matriz  $M$  também é uma matriz simétrica. Uma ordenação lexicográfica dupla de uma matriz simétrica  $M$ , por outro lado, não é necessariamente uma matriz simétrica, como mostra o Exemplo 2.8.

**Exemplo 2.8.** Seja  $M$  a matriz simétrica definida abaixo e  $M'$  a matriz obtida de  $M$  pela permutação da segunda com a terceira coluna. Note que  $M'$  é uma matriz não-simétrica e é uma ordenação lexicográfica dupla de  $M$ .

$$M = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 5 \end{pmatrix}, M' = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 6 & 4 \\ 3 & 5 & 6 \end{pmatrix}$$

**Definição 2.9.** Uma matriz  $n \times n$  simétrica  $M$  possui *diagonal dominante* se  $M(i, i) \geq M(j, i) = M(i, j)$ , para todo  $1 \leq i, j \leq n$ .

**Teorema 2.10** (Lubiw [18]). *Toda matriz simétrica com diagonal dominante possui uma ordenação simétrica que é uma ordenação lexicográfica dupla.*

*Demonstração.* Seja  $M$  uma matriz simétrica real  $n \times n$  com diagonal dominante. Suponha, por absurdo, que não seja verdade o resultado. Seja  $d_M$  o vetor associado a  $M$  de tamanho  $2n - 1$  definido da seguinte maneira: para cada  $1 \leq k \leq 2n - 1$ ,  $d_M(k) = \sum_{i+j=k+1} M(i, j)$ . Logo, cada componente de  $d_M$  representa a soma dos valores de uma diagonal de  $M$ .

Suponha que exista  $1 < e \leq n$  consecutivos tal que  $M(e - 1, *) > M(e, *)$  (caso não existam, então  $M$  é uma ordenação lexicográfica dupla). Seja  $M'$  a matriz obtida por  $M$  trocando-se de posição suas linhas  $e - 1$  e  $e$ , em seguida, trocando-se de posição suas colunas  $e - 1$  e  $e$ .

Seja  $k$  a maior coluna de  $M$  tal que  $M(e - 1, k) > M(e, k)$ . Naturalmente, como  $M(e - 1, e) \leq M(e, e)$  pois  $M$  possui diagonal dominante, então  $k \neq e$ . Dependendo dos valores de  $k$  e  $e$ , a maior componente  $m$  onde há diferença de valores no vetor  $d(M')$  em relação ao vetor  $d(M)$  é dada pelos seguintes casos:

- $m = e + k - 1$ , quando  $k \neq e - 1$ : Neste caso, a componente é alterada para  $d_{M'}(m) = d_M(m) - M(e, k) + M(e - 1, k) - M(k, e) + M(k, e - 1) = d_M(m) - 2M(e, k) + 2M(e - 1, k)$ . Como  $M(e - 1, k) > M(e, k)$ , então  $d_{M'} > d_M$ .
- $m = 2e - 1$ , quando  $k = e - 1$ : Neste caso, a componente é alterada para  $d_{M'}(m) = d_M(m) - M(e, e) + M(e - 1, e - 1)$ . Como  $M(e - 1, e - 1) = M(e - 1, k) > M(e, k) = M(k, e) = M(e - 1, e) = M(e, e)$ , então  $d_{M'} > d_M$ .

Note que  $M'$  continua sendo matriz simétrica com diagonal dominante. Fazendo-se  $M'$  ser  $M$ , podemos aplicar sucessivamente este raciocínio para obter uma sequência infinita de ordenações simétricas com diagonal dominante, dado que por suposição nenhuma delas será uma ordenação lexicográfica dupla e, portanto, podemos sempre escolher um par de linhas desordenadas para gerar uma próxima ordenação simétrica. Como  $d_{M'} > d_M$ , então nenhuma ordenação nesta sequência se repete. Isto é uma contradição com o fato de haver um número finito de ordenações simétricas de uma matriz.  $\square$

De agora em diante apresentamos os algoritmos mais eficientes de reconhecimento de ordenações lexicográficas duplas. Hoffman, Kolen, e Sakarovitch [14] desenvolveram o primeiro algoritmo eficiente para encontrar uma ordenação lexicográfica dupla. Os mais eficientes são devidos a Lubiw [18], para ordenações lexicográficas

duplas de matrizes reais, e de Paige e Tarjan [21], para ordenações lexicográficas duplas de matrizes binárias.

Antes de apresentar os algoritmos propriamente ditos, tratamos de descrever algumas propriedades e definições adicionais utilizadas por eles.

**Definição 2.11.** Uma *partição ordenada* de um conjunto finito  $E$  é uma tupla ordenada  $(E_1, \dots, E_k)$  onde  $E_1, \dots, E_k$  são as partes de uma partição de  $E$ .

**Definição 2.12.** Uma ordem linear  $e_1 < \dots < e_{|E|}$  dos elementos de  $E$  é *consistente* com a partição ordenada  $(E_1, \dots, E_k)$  de  $E$  se  $e_i \in E_p$  e  $e_j \in E_q$  com  $p < q$  implicar em  $i < j$ .

**Definição 2.13.** Dada uma matriz  $M$  com partição ordenada  $(L_1, \dots, L_r)$  de seu conjunto de linhas  $\{1, \dots, m\}$  e partição ordenada  $(C_1, \dots, C_k)$  de seu conjunto de colunas  $\{1, \dots, n\}$ , para todo  $1 \leq i \leq k$ ,  $1 \leq j \leq l$ ,  $(L_i, C_j)$  é um *bloco* de  $M$ .

**Definição 2.14.** Um bloco  $(L, C)$  é um *bloco constante* se o valor  $M(l, c)$  é igual para todo  $l \in L, c \in C$ .

**Definição 2.15.** O *máximo* de um bloco  $B = (L, C)$ , denotado por  $\max(B)$ , é dado por  $\max\{M(l, c) \mid l \in L, c \in C\}$ .

**Definição 2.16.** Uma *linha separadora* de um bloco  $B = (L, C)$  é um  $l \in L$  tal que  $\{c \in C \mid M(l, c) = \max(B)\}$  é um subconjunto próprio não-vazio de  $C$ . Analogamente, uma *coluna separadora* de  $B$  é um  $c \in C$  tal que  $\{l \in L \mid M(l, c) = \max(B)\}$  é um subconjunto próprio não-nulo de  $L$ .

**Exemplo 2.17.** Considerando a matriz  $M$  abaixo como um único bloco, a primeira linha é a única linha separadora. Quanto as colunas, todas são colunas separadoras.

$$M = \begin{pmatrix} 7 & 1 & 7 \\ 6 & 6 & 6 \\ 7 & 7 & 7 \end{pmatrix}$$

A seguir apresentamos uma propriedade importante sobre linhas e colunas separadoras em blocos.

**Lema 2.18.** *Todo bloco não-constante possui uma linha ou uma coluna separadora.*



*Demonstração.* Por absurdo, suponha que não existe nenhuma linha ou coluna separadora num bloco  $B = (L, C)$  não-constante. Seja  $i \in L$ ,  $j \in C$  tais que  $M(i, j) = \max(B)$ . Como  $i$  não é uma linha separadora de  $B$  e  $j \in \{c \in C \mid M(i, c) = \max(B)\} \neq \emptyset$ , então  $M(i, c) = \max(B)$  para todo  $c \in C$ .

Para todo  $c \in C$ , como  $c$  não é uma coluna separadora de  $B$  e  $i \in \{l \in L \mid M(l, c) = \max(B)\} \neq \emptyset$ , então  $M(l, c) = \max(B)$  para todo  $l \in L$ . Portanto,  $B$  é um bloco constante, contrariando a hipótese inicial.  $\square$

## 2.2 Algoritmos de ordenação lexicográfica dupla

O primeiro algoritmo para encontrar uma ordenação lexicográfica dupla de uma matriz foi apresentado por Hoffman, Kolen e Sakarovitch [14] e chamado por eles de ordenação léxica. Apesar deste algoritmo ter sido construído para encontrar uma ordenação de uma classe especial de matrizes, as matrizes totalmente balanceadas, ele originou a noção atual de ordenação lexicográfica dupla. O tempo gasto para sua execução é  $O(i^2j)$ , onde  $i$  é o número de linhas de  $M$  e  $j$  é o número de colunas de  $M$ .

Lubiw [18] e Paige e Tarjan [21] desenvolveram outros algoritmos para ordenação lexicográfica dupla com complexidade de tempo respectivamente  $O(m \log^2(i + j) + i + j)$  e  $O(m \log(i + j) + i + j)$ , onde  $m$  é o número de entradas não-nulas de  $M$ .

Spinrad [23] apresentou um algoritmo que encontra uma ordenação lexicográfica dupla de matrizes binárias densas em tempo  $O(ij)$ . Observe que o algoritmo de Lubiw e o de Paige e Tarjan na verdade resolvem um problema mais geral do que o resolvido por Spinrad [23], uma vez que no caso deles as entradas de  $M$  são valores arbitrários e portanto necessitam de estruturas de dados mais complexas para conseguir atingir o tempo de execução estabelecido.

### 2.2.1 O algoritmo de Lubiw, Paige e Tarjan

O algoritmo apresentado a seguir é baseado nos artigos de Lubiw [18] e de Paige e Tarjan [21]. A idéia central do algoritmo resume-se da seguinte maneira: o invariante do algoritmo é a manutenção de uma partição ordenada  $\mathcal{L}$  do conjunto de linhas  $\{1, \dots, m\}$  e outra  $\mathcal{C}$  do conjunto de colunas  $\{1, \dots, n\}$  da matriz de entrada, além

de uma lista  $\mathcal{K}$  dos blocos definidos por  $\mathcal{L}$  e  $\mathcal{C}$  que sabidamente são constantes. A cada iteração, ou alguma parte de  $\mathcal{L}$  ou  $\mathcal{C}$  é substituída por duas outras, ou um novo bloco é adicionado em  $\mathcal{K}$ . O algoritmo termina quando  $\mathcal{K}$  possuir todos os blocos definidos por  $\mathcal{L}$  e  $\mathcal{C}$ , condição sob a qual o algoritmo poderá então gerar uma ou mais ordenações lexicográficas duplas de  $M$  (no caso de várias, todas serão equivalentes, isto é, todas resultam numa mesma ordenação lexicográfica dupla). Nem todas as ordenações lexicográficas duplas são geradas pelo algoritmo.

### Descrição do Algoritmo de Lubiw – Paige e Tarjan

**Entrada:** Matriz racional  $M$   $m \times n$

**Saída:** Uma ordenação lexicográfica dupla de  $M$

(i)  $\mathcal{L} \leftarrow (\{1, \dots, m\})$ ;  $\mathcal{C} \leftarrow (\{1, \dots, n\})$ ;  $\mathcal{K} \leftarrow \emptyset$

Seja  $\mathcal{B}(\mathcal{L}, \mathcal{C})$  o conjunto dos blocos definidos por  $\mathcal{L}$  e  $\mathcal{C}$

(ii) Enquanto existe bloco não-constante faça

(a) Seja  $B = (L_i, C_j)$  um bloco não-constante tal que não exista  $(L_r, C_s) \in \mathcal{B}(\mathcal{L}, \mathcal{C}) \setminus \mathcal{K}$  de modo que  $i \leq r$  e  $j \leq s$ . Em outras palavras,  $B$  é o bloco posterior em relação às partes de  $\mathcal{L}$  e posterior em relação às partes de  $\mathcal{C}$ .

(b) Se  $B$  é um bloco constante, adicione-o a  $\mathcal{K}$ .

Senão se  $B$  possui uma linha separadora  $l \in L_i$ , então substitua  $C_j$  em  $\mathcal{C}$  pelo par de partes  $\{c \in C_j \mid M(l, c) \neq \max(B)\}$  e  $\{c \in C_j \mid M(l, c) = \max(B)\}$ , nesta ordem. Caso contrário, pelo Lema 2.18,  $B$  possui uma coluna separadora  $c \in C_j$  e  $L_i$  deve ser substituído em  $\mathcal{L}$  analogamente pelo par de partes  $\{l \in L_i \mid M(l, c) \neq \max(B)\}$  e  $\{l \in L_i \mid M(l, c) = \max(B)\}$ , nesta ordem.

(iii) Qualquer ordenação linear de  $\{1, \dots, m\}$  consistente com  $\mathcal{L}$  e qualquer ordenação linear de  $\{1, \dots, n\}$  consistente com  $\mathcal{C}$  definem uma permutação de linhas e colunas respectivamente que resulta numa ordenação lexicográfica dupla.

Vejam agora um exemplo de execução do algoritmo da Lubiw, Paige e Tarjan. Considere a matriz real  $M$  abaixo.

$$M = \begin{pmatrix} 2 & 3 & 4 \\ 1 & 4 & 3 \\ 4 & 8 & 3 \\ 2 & 3 & 3 \end{pmatrix}$$

O algoritmo é inicializado com a partição das colunas contendo uma única parte que é o conjunto das linhas de  $M$  assim como a partição das colunas contendo uma única parte que é o conjunto das colunas de  $M$ , ou seja,  $\mathcal{L} = (\{1, 2, 3, 4\})$  e  $\mathcal{C} = (\{1, 2, 3\})$ . O algoritmo escolhe o único bloco  $B = (\{1, 2, 3, 4\}, \{1, 2, 3\})$  para ser refinado, onde  $\max(B) = 8$  e a linha separadora é a 3. Assim a partição de colunas foi refinada para  $\mathcal{C} = (\{1, 3\}, \{2\})$  e a de linhas não foi alterada. Na iteração seguinte, o algoritmo escolhe o bloco  $B = (\{1, 2, 3, 4\}, \{2\})$  para ser refinado, para o qual  $\max(B) = 8$  e a coluna separadora é a 2. Assim a partição de linhas foi refinada para  $\mathcal{L} = (\{1, 2, 4\}, \{3\})$  e a de colunas não foi alterada. O algoritmo segue escolhendo o bloco  $B = (\{1, 2, 4\}, \{2\})$  para ser refinado (o bloco  $(\{3\}, \{1, 3\})$  também era uma opção), para o qual  $\max(B) = 4$  e a coluna separadora é a 2. Assim a partição de linhas foi refinada para  $\mathcal{L} = (\{1, 4\}, \{2\}, \{3\})$  e a de colunas não foi alterada. Na iteração posterior, o bloco escolhido para ser refinado é  $B = (\{3\}, \{1, 3\})$ , para o qual  $\max(B) = 4$  e a linha separadora é a 3. Assim a partição de colunas foi refinada para  $\mathcal{C} = (\{3\}, \{1\}, \{2\})$  e a de linhas não foi alterada. O algoritmo continua escolhendo o bloco  $B = (\{1, 4\}, \{3\})$  para ser refinado.  $\max(B) = 4$  e a coluna separadora é a 3. Assim a partição de linhas foi refinada para  $\mathcal{L} = (\{4\}, \{1\}, \{2\}, \{3\})$  e a de colunas não foi alterada. O algoritmo termina pois todos os blocos são constantes. As partições terminam da seguinte forma  $\mathcal{L} = (\{4\}, \{1\}, \{2\}, \{3\})$  e  $\mathcal{C} = (\{3\}, \{1\}, \{2\})$ , as quais definem a seguinte permutação de linhas e colunas de  $M$ , constituindo uma ordenação lexicográfica dupla desta matriz.

$$M' = \begin{pmatrix} 3 & 2 & 3 \\ 4 & 2 & 3 \\ 3 & 1 & 4 \\ 3 & 4 & 8 \end{pmatrix}$$

### Correção do algoritmo

**Teorema 2.19** (Correção do Algoritmo). *Se  $M$  é uma matriz racional, o algoritmo descrito acima tendo  $M$  como entrada devolve uma ordenação lexicográfica dupla de  $M$ .*

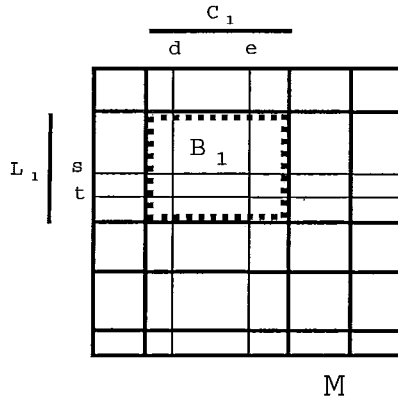
*Demonstração.* Seja  $M$  uma matriz  $m \times n$  e  $(\{1, \dots, m\}, \prec_L), (\{1, \dots, n\}, \prec_C)$  as ordens lineares devolvidas pelo algoritmo, especificando uma permutação de linhas e colunas para gerar uma ordenação lexicográfica dupla  $M'$ . Queremos mostrar que  $M'$  é de fato uma ordenação lexicográfica de  $M$ . Sejam  $d \prec_C e$  e um par de índices de colunas que nesta ordem tenham ficado na ordenação final. Mostraremos que a coluna  $d$  é menor ou igual a coluna  $e$  em  $M'$ . Desse resultado, juntamente com o análogo para as linhas, segue a prova.

Suponha o contrário com o propósito de encontrar um absurdo. Logo, existe  $1 \leq t \leq m$  tal que  $M(t, d) > M(t, e)$  e para todo  $t \prec_L t', M(t', d) = M(t', e)$ .

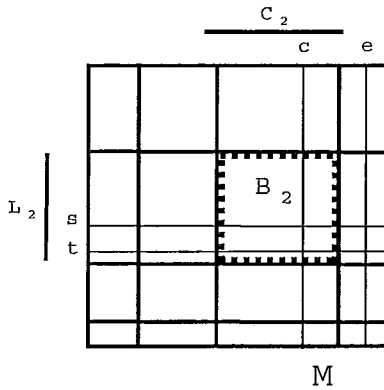
Como os valores das colunas  $d$  e  $e$  de  $M'$  para ao menos a linha  $t$  não são iguais, tais colunas não podem ter terminado numa mesma parte da partição de colunas. Logo, em algum passo  $P_1$  do algoritmo, as colunas  $d$  e  $e$  foram colocadas em partes diferentes da partição de colunas. Seja  $B_1 = (L_1, C_1)$  o bloco que foi dividido no passo  $P_1$  pela linha separadora escolhida  $s \in L_1$ . Note que  $s \prec_L t$ , pois caso contrário se  $t \prec_L s$ , então  $M(s, d) = M(s, e)$  pela definição de  $t$ , o que contradiz o fato de  $s$  ser uma linha separadora em que as colunas  $d$  e  $e$  passem a estar em partes diferentes. Além disso,  $t$  não pode pertencer a algum bloco  $L_i$  posterior a  $L_1$ , pois pela escolha de  $B$ , todos os blocos  $(L_i, C_1)$  são constantes, mas  $M(t, d) > M(t, e)$ . Como partes distintas de uma partição de linhas ou colunas não trocam de ordem durante o algoritmo, então  $t \in L_1$ . A próxima figura mostra uma visão geral deste passo  $P_1$ .

Como  $t \in L_1$ , então  $M(t, d) \leq \max(B_1)$ . Observe que, como  $d \prec_C e$ , então  $M(s, d) < \max(B_1) = M(s, e)$ . Logo,  $M(t, d) \leq M(s, e)$ . Além disso, como  $M(t, d) > M(t, e)$ , segue que  $M(t, e) < M(s, e)$ .

Note que as linhas  $s, t$  pertencem a um mesmo bloco no passo  $P_1$  mas a blocos diferentes na ordenação final, dado que discordam de um valor ao menos pela coluna  $e$ . Seja  $P_2$  o passo, posterior a  $P_1$ , no qual as linhas  $s$  e  $t$  são colocadas em diferentes partes da partição de linhas. Seja  $B_2 = (L_2, C_2)$  o bloco que será dividido pela coluna separadora escolhida  $c \in C_2$  no passo  $P_2$ . Como  $s \prec_L t$ ,  $M(s, c) < \max(B_2) =$



Passo 1 ( $P_1$ )



Passo 2 ( $P_2$ )

$M(t, c)$ .

Vejamus onde estaria a coluna  $c$  posicionada no passo  $P_1$ . Note que  $c$  não poderia pertencer a nenhuma parte  $C_j$  na partição de colunas posterior a  $C_1$ , pois neste caso  $M(s, c)$  seria igual a  $M(t, c)$  pela escolha de  $B_1$ . Além disso, ao final de  $P_1$ ,  $c$  não pode ter ficado na mesma parte em que ficou a coluna  $e$ , pois neste caso  $M(s, c) = \max(B_1) \geq M(t, c)$  contrariando  $M(s, c) < M(t, c)$ . Portanto, ao final do passo  $P_1$ , a coluna  $c$  ficou numa parte da partição de colunas anterior àquela que contém  $e$ . Como a posição entre as partes é um invariante, então em  $P_2$  a coluna  $e$  está numa parte  $C_j$  posterior à  $C_2$ . Logo, pela escolha de  $B_2$ , o bloco  $(L_2, C_j)$  é constante. Em particular,  $M(s, e) = M(t, e)$ , contrariando a conclusão anterior de que  $M(t, e) < M(s, e)$ .  $\square$

### Complexidade do algoritmo

Em seu trabalho, Lubiw [18] propôs uma estrutura de dados e uma imple-

mentação cuja complexidade de tempo é de  $O((e + m + n) \log^2(e + m + n) + m + n)$ , sendo  $M$  uma matriz de entrada  $m \times n$  e  $e$  o número de elementos de  $M$  diferentes do menor elemento.

Paige e Tarjan, baseados no algoritmo acima, observaram que com uma implementação melhorada a complexidade de tempo é de  $O((e \log(m + n)) + m + n)$ . No caso particular de matrizes binárias representando as matrizes bipartidas de grafos bipartidos, temos que  $e$  é igual ao número de arestas do grafo e  $m + n$  é igual ao número de seus vértices. Embora hajam muitas semelhanças nas duas implementações, como por exemplo, a matriz de entrada que é dada por um conjunto de triplas (das entradas não-nulas da matriz), a implementação de Paige e Tarjan difere em dois pontos em relação à de Lubiw, os quais descrevemos a seguir.

A primeira diferença é na seleção do bloco a ser refinado no passo (ii)(a). Na implementação de Lubiw, é feita uma ordem total dos blocos de baixo para cima dentro de cada conjunto de colunas que define um bloco e da direita para esquerda dentro de cada conjunto de linhas que define um bloco. Paige e Tarjan propuseram uma ordem parcial destes blocos, de modo que um bloco não-constante  $(L, C)$  precede um bloco não-constante  $(L', C')$  quando  $L = L'$  e  $C$  está após  $C'$  na partição ordenada de colunas ou  $C = C'$  e  $L$  está após  $L'$  na partição ordenada de linhas. Esta relaxação permite melhorar a complexidade de tempo do algoritmo.

A segunda diferença é quando há tanto uma linha separadora quanto uma coluna separadora no bloco sendo refinado no passo (ii)(b). Pela implementação de Lubiw, escolhe-se arbitrariamente uma linha ou coluna separadora. Na implementação de Paige e Tarjan contudo uma coluna separadora só é escolhida quando não há linha separadora do bloco.

Embora haja essas diferenças na implementação, a redução da complexidade de tempo na abordagem de Paige e Tarjan se deve também a uma análise mais precisa do tempo gasto por cada passo da implementação.

## 2.2.2 O algoritmo de Spinrad

O algoritmo a seguir foi apresentado por Spinrad e encontra uma ordenação lexicográfica dupla de matrizes binárias densas em tempo  $O(ij)$ , melhorando o tempo de execução para determinar quando uma matriz binária densa é totalmente ba-

lanceada, quando um grafo é fortemente cordal e para o nosso interesse principal, quando um grafo é bipartido cordal. Informativamente, o autor deste algoritmo também usou ordenação lexicográfica dupla como uma sub-rotina em algoritmos eficientes para reconhecer grafos de trapézios (cf. [19]) e grafos arco-circulares (cf. [9]). O algoritmo de Spinrad produz uma ordenação lexicográfica dupla de uma matriz binária  $M$  com o seguinte processo.

Mantém-se uma partição  $\mathcal{L}$  do conjunto de linhas e  $\mathcal{C}$  do conjunto de colunas e define-se a matriz  $M'$  como uma permutação das linhas  $\pi_{\mathcal{L}}$  e colunas  $\pi_{\mathcal{C}}$  de  $M$ , tal que a reordenação de colunas deve ser consistente com  $\mathcal{C}$  e a reordenação de linhas consistente com  $\mathcal{L}$ . Em outras palavras, se  $\pi_{\mathcal{L}}$  é uma permutação do conjunto de linhas e  $\pi_{\mathcal{C}}$  uma permutação do conjunto de colunas consistentes respectivamente com  $\mathcal{L}$  e  $\mathcal{C}$ , então (i) se  $L_1, L_2 \in \mathcal{L}$ ,  $L_1 < L_2$ , então  $\pi_{\mathcal{L}}(l_1) < \pi_{\mathcal{L}}(l_2)$  para todo  $l_1 \in L_1, l_2 \in L_2$ ; e (ii) se  $C_1, C_2 \in \mathcal{C}$ ,  $C_1 < C_2$ , então  $\pi_{\mathcal{C}}(c_1) < \pi_{\mathcal{C}}(c_2)$  para todo  $c_1 \in C_1, c_2 \in C_2$ .

Iterativamente, respeita-se o seguinte invariante: (i) se  $L_1, L_2 \in \mathcal{L}$ ,  $L_1 < L_2$ , então para todo  $l_1 \in L_1, l_2 \in L_2$ ,  $M'(\pi_{\mathcal{L}}(l_1), *) < M'(\pi_{\mathcal{L}}(l_2), *)$ ; (ii) se  $C_1, C_2 \in \mathcal{C}$ ,  $C_1 < C_2$ , então para todo  $c_1 \in C_1, c_2 \in C_2$ ,  $M'(*, \pi_{\mathcal{C}}(c_1)) < M'(*, \pi_{\mathcal{C}}(c_2))$ . O processo iterativo termina quando é possível garantir as seguintes condições: (i) se  $L \in \mathcal{L}$ , então para todo  $l_1, l_2 \in L$ ,  $M'(\pi_{\mathcal{L}}(l_1), *) = M'(\pi_{\mathcal{L}}(l_2), *)$ ; (ii) se  $C \in \mathcal{C}$ , então para todo  $c_1, c_2 \in C$ ,  $M'(*, \pi_{\mathcal{C}}(c_1)) = M'(*, \pi_{\mathcal{C}}(c_2))$ . Tal condição de escape da iteração, reforçada pelas condições garantidas por seu invariante, mostram que  $\pi_{\mathcal{L}}$  e  $\pi_{\mathcal{C}}$  constituem uma ordenação lexicográfica dupla de  $M$ .

Inicialmente, o invariante pode ser trivialmente atingido fazendo-se  $\mathcal{L}$  ser o conjunto com todo o conjunto de linhas e  $\mathcal{C}$  ser o conjunto com todo o conjunto de colunas. A estratégia geral da iteração, então, é descrita da seguinte forma. Dados  $L \in \mathcal{L}$  e  $C \in \mathcal{C}$ , de modo que todo bloco posterior a  $(L, C)$  em  $\mathcal{L}$  ou em  $\mathcal{C}$  já tenham sido visitados (o único bloco inicial é considerado não-visitado), realiza-se o procedimento de refinamento, que constitui de uma tentativa de particionar os conjuntos  $L$  e  $C$ . Tal processo de refinamento toma como entrada portanto um bloco que pode gerar diversos blocos. Alguns de tais blocos serão marcados como visitados, enquanto outros como não-visitados.

O objetivo do refinamento é produzir na submatriz definida pelo bloco  $(L, C)$

$$\begin{pmatrix} \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & 0 & 0 & 1 \\ \dots & \dots & \dots & \dots & 0 & 0 & 1 \\ \dots & \dots & \dots & 0 & 1 & 1 & 1 \\ \dots & \dots & \dots & 0 & 1 & 1 & 1 \\ \dots & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & 0 & 0 & 1 \\ \dots & \dots & \dots & \dots & 0 & 0 & 1 \\ \dots & \dots & \dots & 0 & 1 & 1 & 1 \\ \dots & \dots & \dots & 0 & 1 & 1 & 1 \\ \dots & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Figura 2.1: Triangularização desejada de 1's e refinamento do particionamento.

uma configuração triangular de 1's conforme esquematizado na Figura 2.1 (à esquerda). Considerando-se isto possível, então podemos refinar  $L$  e  $C$  mantendo-se o invariante conforme representado pela Figura 2.1 (à direita). Neste ponto, como podemos notar no exemplo, produz-se diversos blocos constantes, os quais são então marcados como visitados. O processo continua refinando-se um outro bloco não-visitado para o qual todo bloco posterior a ele em  $\mathcal{L}$  ou em  $\mathcal{C}$  já tenham sido visitados.

Note que, supondo que cada chamada deste procedimento de refinamento dos blocos não-visitados retornarem partições tais que permutações consistentes com tais partições são ordenações lexicográficas duplas das respectivas submatrizes, então de fato a partição da matriz inteira é tal que permutações consistentes com ela serão ordenações lexicográficas duplas da matriz.

A descrição do algoritmo, que utiliza a rotina Refinar como sub-rotina, é formalizada a seguir por seus pseudo-códigos.

### Descrição da Rotina Particionar

**Entrada:** Matriz binária  $M$   $m \times n$

**Saída:** Uma ordenação lexicográfica dupla de  $M$

- (i)  $\mathcal{L} \leftarrow (\{1, \dots, m\}); \mathcal{C} \leftarrow (\{1, \dots, n\})$  // conjunto de linhas e colunas
- (ii) Marcar o único bloco criado como não-visitado
- (iii) Enquanto houver um bloco não-visitado faça
  - (a) Seja  $L$  a última parte de  $\mathcal{L}$  que possua um bloco não-visitado;



- (b) Seja  $C$  a parte de  $\mathcal{C}$  mais à direita tal que  $(L, C)$  seja não-visitado;
  - (c) Se  $C$  é constante então marque  $(L, C)$  como visitado, senão substituir  $L$  e  $C$  pelos refinamentos devolvidos por  $\text{Refinar}(L, C)$
- (iv) Devolva uma permutação de linhas e colunas consistente com  $\mathcal{L}$  e  $\mathcal{C}$

### Descrição da Rotina Refinar

**Entrada:**  $L \in \mathcal{L}$ ,  $C \in \mathcal{C}$  tal que  $(L, C)$  não é um bloco constante

**Saída:** Refinamento de  $L$  e  $C$

- (i)  $\mathcal{L}' \leftarrow \emptyset$ ;  $\mathcal{C}' \leftarrow (C)$
- (ii) Para cada  $l \in L$  faça
  - (a) Seja  $C'$  a última parte de  $\mathcal{C}'$
  - (b) Enquanto ( $C'$  é definido) e  $(M(l, c) = 1$  para todo  $c \in C'$ ) faça
    - i.  $C' \leftarrow$  parte anterior a  $C'$  de  $\mathcal{C}'$
  - (c) Se ( $C'$  é definido) e  $(M(l, c) = 1$  para algum  $c \in C'$ ) então
    - i.  $C'_0 \leftarrow \{c \in C' \mid M(l, c) = 0\}$ ;  $C'_1 \leftarrow \{c \in C' \mid M(l, c) = 1\}$
    - ii. Substituir  $C'$  em  $\mathcal{C}'$  por  $C'_0, C'_1$  (nesta ordem)
- (iii) Para cada  $l \in L$  faça
  - (a)  $F(l) \leftarrow$  a parte  $C'$  mais à direita de  $\mathcal{C}'$  tal que  $(M(l, c) = 0$  para todo  $c \in C'$ ). Deixe  $F(l)$  indefinida se  $M(l, c) = 1$  para todo  $c \in C'$ .
- (iv) Para cada  $C'$  em  $\mathcal{C}'$  decrescentemente e  $C' = \emptyset$  faça
  - (a)  $L' \leftarrow \{l \in L \mid F(l) = C'\}$
  - (b) Se  $L' \neq \emptyset$  então
    - i. Acrescente  $L'$  como última parte de  $\mathcal{L}'$
- (v) Devolva  $\mathcal{L}'$ ,  $\mathcal{C}'$

A Figura 2.2 mostra graficamente o resultado do refinamento de uma matriz hipotética realizado pelo algoritmo, evidenciando a estrutura de 1's triangularizados obtida em sua conclusão. À esquerda da figura, é representada por linhas verticais

a partição de colunas gerada após a finalização do processo iterativo do passo (ii). À direita da figura, o refinamento das linhas após a finalização do processo iterativo do passo (iv) é representado por linhas horizontais.

$$\left( \begin{array}{c|c|c|c|c|c|c|c|c|c|c} \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & 0 & 0 & 1 & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 & 0 & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \end{array} \right) \leftarrow \left( \begin{array}{c|c|c|c|c|c|c|c|c|c|c} \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{array} \right)$$

Figura 2.2: Procedimento de Refinamento

Vejam agora um exemplo de execução do procedimento Refinar. Considere a matriz  $M$  abaixo.

$$M = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

O algoritmo é inicializado com a partição das colunas contendo uma única parte  $(\{1, 2, 3, 4, 5, 6\})$  com o conjunto das colunas de  $M$ . O algoritmo percorre cada linha tentando fazer uma partição das colunas. A linha 1 divide a única parte da partição das colunas em duas partes que correspondem aos vizinhos e não-vizinhos de 1, resultando em  $\mathcal{C} = (\{1, 3, 6\}, \{2, 4, 5\})$ . A linha 2 divide a parte  $\{2, 4, 5\}$  em  $\{2\}, \{4, 5\}$ , portanto  $\mathcal{C} = (\{1, 3, 6\}, \{2\}, \{4, 5\})$ . A linha 3 não divide a parte  $\{4, 5\}$  pois são ambos vizinhos de 3, o mesmo ocorre com a parte  $\{2\}$ . Já a parte  $\{1, 3, 6\}$  possui vizinhos e não-vizinhos e será portanto dividida em  $\{1, 6\}, \{3\}$ . A nova partição será  $\mathcal{C} = (\{1, 6\}, \{3\}, \{2\}, \{4, 5\})$ . A linha 4 não dividirá nenhuma parte pois  $\{4, 5\}$  são ambos não-vizinhos de 4 (condição do passo (ii)(c)) e portanto a partição continuará  $\mathcal{C} = (\{1, 6\}, \{3\}, \{2\}, \{4, 5\})$ . A linha 5 é vizinha de 4 mas não de 5, e portanto  $\mathcal{C} = (\{1, 6\}, \{3\}, \{2\}, \{5\}, \{4\})$ . Como 4 é vizinho da linha 6, a próxima parte a ser analisada é a parte  $\{5\}$  que também é vizinha da linha 6. Mas 2 não é vizinho da linha 6. Assim a partição de colunas termina igual a  $\mathcal{C} = (\{1, 6\}, \{3\}, \{2\}, \{5\}, \{4\})$ .

Para a partição das linhas, colocamos as linhas que possuem o primeiro 0 mais à direita em partes mais anteriores, sendo que linhas que possuem a mesma quantidade de 1's antes do aparecimento do primeiro 0 ficam na mesma parte. Assim, no exemplo acima a linha 1 fica na parte  $\{3\}$ , a linha 2 fica na parte  $\{2\}$ , a linha 3 fica na parte  $\{1, 6\}$ , a linha 4 fica na parte  $\{4\}$ , a linha 5 fica na parte  $\{5\}$  e a linha 6 fica na parte  $\{2\}$ . A partição das linhas será portanto  $\mathcal{L} = (\{4\}, \{5\}, \{2, 6\}, \{1\}, \{3\})$ .

Portanto, se o algoritmo tiver como entrada a matriz  $M$  apresentada anteriormente, a primeira chamada à sub-rotina Refinar gera a partição de colunas  $\mathcal{C} = (\{1, 6\}, \{3\}, \{2\}, \{5\}, \{4\})$  e a partição de linhas  $\mathcal{L} = (\{4\}, \{5\}, \{2, 6\}, \{1\}, \{3\})$ . Como as partes  $\{3\}$  e  $\{1\}$  da partição de linhas já refinaram todas as partes da

partição de colunas, então o algoritmo continuará com o refinamento do bloco  $(\{2, 6\}, \{3\})$ , ou seja, será invocado  $\text{Refinar}(\{2, 6\}, \{3\})$ , que refinará a parte em duas partes obtendo uma nova partição de linhas  $\mathcal{L} = \{\{4\}, \{5\}, \{6\}, \{2\}, \{1\}, \{3\}\}$ . O próximo a ser refinado,  $(\{2\}, \{1, 6\})$ , faz com que a nova partição de colunas seja  $\mathcal{C} = (\{6\}, \{1\}, \{3\}, \{2\}, \{5\}, \{4\})$ . Nesse momento todas as partes contém exatamente um elemento, logo não haverá mais refinamentos.

### Correção do algoritmo

**Lema 2.20.** *Sejam  $l_1$  e  $l_2$  linhas de  $M$ . Considerando a ordenação final das linhas e colunas gerada pela rotina  $\text{Particionar}(M)$ , seja  $c$  a última coluna onde essas duas linhas diferem. Então  $l_1$  e  $l_2$  são colocadas em partes diferentes da partição de linhas após a chamada  $\text{Refinar}(X, Y)$  tal que  $X$  contém  $l_1$  e  $l_2$  e  $Y$  contém  $c$ .*

*Demonstração.* Seja  $X$  uma parte da partição de linhas que contém as duas linhas  $l_1$  e  $l_2$ . Seja  $C$  a parte que contém  $c$  na partição de colunas.

Como  $l_1$  e  $l_2$  diferem em pelo menos uma coluna  $c$  e ao final da execução da rotina  $\text{Particionar}(M)$  todos os pares de partes serão examinados, então em alguma chamada à rotina  $\text{Refinar}(X, Y^*)$ ,  $l_1$  e  $l_2$  serão colocadas em partes diferentes. Vamos mostrar que isso ocorre exatamente quando  $Y^* = C$ .

Suponha que  $Y^*$  seja uma parte que sucede da parte  $C$  na partição de colunas. Como  $c$  é a última coluna onde essas duas linhas diferem na ordenação final das linhas e colunas, então todas as colunas que aparecem em  $Y^*$  aparecerão depois de  $c$  na ordenação final e portanto  $l_1$  e  $l_2$  concordam em todas as colunas de  $Y^*$ . Logo,  $l_1$  e  $l_2$  ficariam na mesma parte ao final da execução rotina  $\text{Refinar}(X, Y^*)$ .

Se por outro lado,  $Y^*$  for uma parte que está antes da parte  $C$  na partição de colunas, então, como a rotina  $\text{Particionar}$  sempre escolhe a última parte não examinada então a chamada  $\text{Refinar}(X, C)$  será executada antes da  $\text{Refinar}(X, Y^*)$ . Lembre que  $l_1$  e  $l_2$  discordam na coluna  $c \in C$  portanto após essa chamada  $l_1$  e  $l_2$  não ficaram ambos em  $X$ , contradizendo o assumido.  $\square$

A prova do Lema a seguir é análoga a anterior.

**Lema 2.21.** *Sejam  $c_1$  e  $c_2$  colunas de  $M$ . Considerando a ordenação final das linhas e colunas gerada pela rotina  $\text{Particionar}(M)$ , seja  $l$  a última linha onde essas duas*

colunas diferem. Então  $c_1$  e  $c_2$  são colocadas em partes diferentes da partição de colunas após a chamada  $\text{Refinar}(X, Y)$  tal que  $X$  contém  $l$  e  $Y$  contém  $c_1$  e  $c_2$ .

*Demonstração.* Seja  $Y$  uma parte da partição de colunas que contém as duas colunas  $c_1$  e  $c_2$ . Seja  $L$  a parte que contém  $l$  na partição de linhas.

Como  $c_1$  e  $c_2$  diferem em pelo menos uma linha  $l$  e ao final da execução da rotina  $\text{Particionar}(M)$  todos os pares de partes serão examinados, então em alguma chamada à rotina  $\text{Refinar}(X^*, Y)$ ,  $c_1$  e  $c_2$  serão colocadas em partes diferentes. Vamos mostrar que isso ocorre exatamente quando  $X^* = L$ .

Suponha que  $X^*$  seja uma parte que sucede da parte  $L$  na partição de linhas. Para cada linha  $l_i \in X$ , ou  $l_i$  é adjacente a  $c_1$  e  $c_2$ , ou  $l_i$  não é adjacente nem a  $c_1$  e nem a  $c_2$ . Portanto, em ambos os casos quando  $Y$  for separado por  $l_i$   $c_1$  e  $c_2$  ficarão na mesma parte da partição de colunas.

Se por outro lado,  $X^*$  for uma parte que está antes da parte  $L$  na partição de linhas. Como a rotina  $\text{Refinar}(X^*, Y)$  não será executada antes da chamada  $\text{Refinar}(L, Y)$  e  $c_1$  e  $c_2$  discordam na linha  $l \in L$  portanto após essa chamada  $c_1$  e  $c_2$  serão separados.  $\square$

**Lema 2.22.** *Considere qualquer par de linhas  $l_1$  e  $l_2$  tal que  $l_1$  vem antes de  $l_2$  na ordenação final. Seja  $c$  a última coluna da ordenação final na qual as duas linhas diferem. Então  $c$  é um vizinho de  $l_2$ .*

*Demonstração.* Seja  $\text{Refinar}(X, Y)$  a última chamada tal que  $X$  contenha o par de linhas  $l_1$  e  $l_2$  e  $Y$  contenha  $c$ . Neste passo,  $l_1$  e  $l_2$  serão colocadas em partes distintas conforme o Lema 2.20. Seja  $Y_1, Y_2, \dots, Y_c, \dots$ , onde  $c \in Y_c$ , o refinamento de  $Y$  ocasionado pela execução deste passo do algoritmo.

Lembrando que o refinamento de  $Y$  refina  $X$ , onde cada linha  $l$  de  $X$  é colocada na lista correspondente ao último bloco de  $Y$  que contém um não-vizinho de  $l$ , então nenhuma das linhas  $\{l_1, l_2\}$  pode ser colocada na lista de partes que sucedem  $Y_c$ . Isso porque todas as partes que sucedem  $Y_c$  também ocorrem depois de  $c$  na ordenação final e  $c$  é a última coluna da ordenação final na qual as duas linhas diferem, portanto  $l_1$  e  $l_2$  concordam em todas as colunas que ocorrem em partes que sucedem  $Y_c$ . Como  $l_1$  e  $l_2$  são colocadas em listas de partes distintas e discordam na coluna  $c$ , então exatamente uma das linhas  $\{l_1, l_2\}$  é colocada na lista correspondentes a parte

predecessora de  $Y_c$ . Como as linhas colocadas em listas de partes que ocorrem antes ocorrem depois na ordenação final, então  $l_2$  deve ser a linha que é colocada na parte predecessora de  $Y_c$ . Como tal parte é a última de  $Y$  que contém não-vizinhos de  $l_2$ , por definição, então  $l_2$  tem um 1 na coluna  $c$ .  $\square$

**Lema 2.23.** *Considere qualquer par de colunas  $c_1$  e  $c_2$  tal que  $c_1$  vem antes de  $c_2$  na ordenação final. Seja  $l$  a última linha da ordenação final na qual as duas colunas diferem. Então  $l$  é um vizinho de  $c_2$ .*

*Demonstração.* Seja  $\text{Refinar}(X, Y)$  a última chamada tal que  $X$  contenha  $l$  e  $Y$  contenha o par de colunas  $c_1$  e  $c_2$ . Neste passo,  $c_1$  e  $c_2$  serão colocadas em partes distintas conforme Lema 2.21. Seja  $Y_1, Y_2, \dots, Y_{c_1}, \dots, Y_{c_2}, \dots$ , onde  $c_1 \in Y_{c_1}$  e  $c_1 \in Y_{c_2}$ , o refinamento de  $Y$  ocasionado pela execução deste passo do algoritmo.

Observe que existe uma linha  $l_i \in X$  que separa a parte  $W$  que contém  $c_1$  e  $c_2$  em vizinhos e não-vizinhos de  $l_i$ . Como  $l_i$  separa  $W$  e o algoritmo sempre refina a última parte que contém pelo menos um não-vizinho, então  $l_i$  é adjacente a todas as colunas que ocorrem depois de  $W$  na subpartição de  $Y$ .

Como os vizinhos de  $l_i$  são colocados depois dos não-vizinhos de  $l_i$  e  $l_i$  é adjacente a  $c_2 \in Y_{c_2}$ , então  $l_i$  é adjacente a todas as colunas que ocorrem na parte  $Y_{c_2}$  e sucessoras na subpartição de  $Y$ . Portanto,  $l_i$  será colocado na lista de uma parte que precede  $Y_{c_2}$  na subpartição de  $Y$ .

Como  $l$  é a última linha na qual  $c_1$  e  $c_2$  diferem, então  $l$  aparece depois de  $l_i$  na ordenação final. Como as linhas colocadas em lista de partes que ocorrem antes, aparecem depois na ordenação final, então  $l$  é colocada na lista de uma parte que ocorre antes de  $Y_{c_2}$ . Isso significa que tal parte é a última parte que contém um não-vizinho de  $l$ , portanto  $l$  é vizinho de todas as colunas em  $Y_{c_2}$ , em particular  $l$  é vizinho de  $c_2$ .  $\square$

A combinação dos Lemas 2.22 e 2.23 nos remete ao seguinte Teorema.

**Teorema 2.24** (Spinrad [23]). *A saída da chamada à rotina  $\text{Particionar}(M)$  é uma ordenação lexicográfica dupla da Matriz  $M$ .*

## 2.3 Matrizes livres de gama

Seja  $M$  uma matriz binária. Uma estrutura  $\Gamma$  em  $M$  consiste de um par de linhas  $l_1 < l_2$  e um par de colunas  $c_1 < c_2$  tais que  $M(l_1, c_1) = M(l_1, c_2) = M(l_2, c_1) = 1$  e  $M(l_2, c_2) = 0$ . Uma matriz é dita ser *livre de  $\Gamma$*  quando não possuir nenhuma estrutura  $\Gamma$ . Uma *ordenação livre de  $\Gamma$*  de uma matriz é uma permutação de suas linhas e colunas tal que a matriz resultante seja livre de  $\Gamma$ .

$$M = \begin{pmatrix} \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \mathbf{1} & \dots & \dots & \dots & \mathbf{1} & \dots \\ \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \mathbf{1} & \dots & \dots & \dots & \mathbf{0} & \dots \\ \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

Veremos adiante que as matrizes livres de  $\Gamma$  serão fundamentais no reconhecimento dos grafos bipartidos cordais. Um dos resultados principais a ser utilizado no reconhecimento é o apresentado a seguir, que relaciona matrizes livres de  $\Gamma$ , ordenações lexicográficas duplas e grafos bipartidos cordais.

**Teorema 2.25** (Anstee e Farber [2], Lubiw [18], Hoffman, Kolen e Sakarovitch [14]).  
*Se um grafo  $G$  é bipartido cordal então toda ordenação lexicográfica dupla de sua matriz bipartida é livre de  $\Gamma$ .*

*Demonstração.* Seja  $M$  uma ordenação lexicográfica dupla da matriz bipartida de  $G$ . Suponha, com o propósito de encontrar um absurdo, que  $M$  possua um  $\Gamma$ . Desta maneira, sejam  $l_1 < l_2$  linhas e  $c_1 < c_2$  colunas formando um  $\Gamma$  em  $M$ , conforme ilustrado a seguir.

	...	$c_1$	...	$c_2$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$l_1$	...	1	...	1	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$l_2$	...	1	...	0	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Como  $M$  é resultado de uma ordenação lexicográfica dupla e  $l_2 > l_1$ , então  $M(l_2, *) > M(l_1, *)$ . Seja  $c_3$  a última coluna em que as linhas  $l_1$  e  $l_2$  diferem. Logo,  $M(l_1, c_3) = 0$  e  $M(l_2, c_3) = 1$ . Como  $M(l_1, c_2) = 1$  e  $M(l_2, c_2) = 0$ , então  $c_3 > c_2$ .

Analogamente para as linhas, concluímos que existe  $l_3 > l_2$  representando a última linha em que as colunas  $c_1$  e  $c_2$  diferem, tal que  $M(l_3, c_1) = 0$  e  $M(l_3, c_2) = 1$ . A próxima figura resume a configuração deduzida.

	...	$c_1$	...	$c_2$	...	$c_3$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$l_1$	...	1	...	1	...	0	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$l_2$	...	1	...	0	...	1	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$l_3$	...	0	...	1	...	?	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Note que  $M(l_3, c_3) = 0$ , pois caso contrário se  $M(l_3, c_3) = 1$ , existiria um ciclo induzido  $(l_3, c_2, l_1, c_1, l_2, c_3)$  em  $G$  de tamanho 6, contrariando o fato do grafo ser bipartido cordal.

Raciocinando de maneira similar, como  $M(l_3, *) > M(l_2, *)$ ,  $M(*, c_3) > M(*, c_2)$  por estarmos tratando de uma ordenação lexicográfica dupla, podemos definir  $l_4 > l_3$  (resp.  $c_4 > c_3$ ) representando a última linha (resp. coluna) em que as colunas  $c_2$  e  $c_3$  (resp.  $l_2$  e  $l_3$ ) diferem, e seremos forçados a concluir que  $M(l_4, c_4) = 0$  evitando-se a formação de um  $C_8$  induzido. Este processo se aplica indefinidamente para as últimas duas linhas e duas colunas definidas, contrariando o fato de  $G$  ser um grafo finito.  $\square$

## 2.4 Grafos bipartidos de eliminação perfeita

Para esta seção, por conveniência, denotaremos por  $G - \{e_1, \dots, e_k\}$  como sendo  $G[\{v \in V(G) \mid v \text{ não é um extremo de } e_i \text{ para algum } 1 \leq i \leq k\}]$ .

Um grafo bipartido  $G = (X, Y, E)$  é chamado *grafo bipartido de eliminação perfeita* se existir uma sequência de arestas duas a duas não-adjacentes  $e_1, \dots, e_n$  tal que:

- (i)  $e_i$  é uma aresta bi-simplicial em  $G - \{e_1, \dots, e_{i-1}\}$  para todo  $i$ ;



(ii)  $E(G - \{e_1, \dots, e_n\}) = \emptyset$ .

Chamamos  $\sigma = [e_1, \dots, e_n]$  de um *esquema* se as duas condições anteriores são satisfeitas e de um *esquema parcial* se somente a primeira condição é satisfeita.

**Exemplo 2.26.** Os grafos bipartidos completos  $K_{m,n}$  são bipartidos de eliminação perfeita, pois todas as arestas são bi-simpliciais, a união das vizinhanças dos extremos de qualquer aresta corresponde ao conjunto de vértices do próprio grafo, que é um bipartido completo, e a retirada de uma aresta bi-simplicial do grafo resulta num novo bipartido completo, isomorfo ao  $K_{m-1,n-1}$ . Em particular, um esquema é dado por  $[x_1y_1, \dots, x_ky_k]$ , onde  $k = \min\{m, n\}$ .

O próximo teorema introduz a propriedade fundamental de um esquema de eliminação perfeita, conduzindo inclusive a um algoritmo de reconhecimento.

**Teorema 2.27** (Golumbic e Goss [13]). *Se  $e$  é uma aresta bi-simplicial em um grafo bipartido de eliminação perfeita  $G = (X, Y, E)$ , então  $G - \{e\}$  é um grafo bipartido de eliminação perfeita.*

*Demonstração.* Mostraremos que se  $[e_1, \dots, e_n]$  é um esquema de  $G$ , então existe um outro esquema começando por  $e = xy$ . Digamos que  $e_i = x_iy_i$ , para cada  $1 \leq i \leq n$ .

**Caso I:** Se  $e = e_i$ , para algum  $1 \leq i \leq n$ , então  $[e, e_1, \dots, e_{i-1}, e_{i+1}, \dots, e_n]$  é um esquema, pois as arestas  $e_1, \dots, e_{i-1}$  continuam bi-simpliciais pela hereditariedade desta propriedade.

**Caso II:** Se  $x = x_i$  e  $y = y_j$ , para algum  $1 \leq i, j \leq n$ , com  $i \neq j$ , então por uma troca de rótulos entre  $X$  e  $Y$ , considere sem perda de generalidade que  $i < j$ . Mostraremos que  $e' = x_jy_i \in E$  e  $[e, e_1, \dots, e_{i-1}, e_{i+1}, \dots, e_{j-1}, e', e_{j+1}, \dots, e_n]$  é um esquema de  $G$ . Para as arestas  $e_1, \dots, e_{i-1}$ , a bi-simplicialidade continua a ocorrer por ser esta uma propriedade hereditária. Além disso, para as arestas  $e_{j+1}, \dots, e_n$  a bi-simplicialidade continua a ocorrer pois  $G - \{e, e_1, \dots, e_{i-1}, e_{i+1}, \dots, e_{j-1}, e'\} = G - \{e_1, \dots, e_{j-1}\}$ . Mostraremos o restante em dois passos:

(i) aresta  $e_h$ ,  $i < h < j$ , é bi-simplicial em  $G - \{e, e_1, \dots, e_{i-1}, e_{i+1}, \dots, e_{h-1}\}$ :

Seja  $E' = E(G - \{e, e_1, \dots, e_{i-1}, e_{i+1}, \dots, e_{h-1}\})$ . Suponha que  $e_h$  não é bi-simplicial em  $G[E']$ . Logo, existem  $w \in N_{G[E']}(x_h)$  e  $v \in N_{G[E']}(y_h)$  tais que  $wv \notin E'$ . Como  $e_h$  é uma aresta bi-simplicial no esquema original, temos então existe uma única possibilidade:

(a) somente  $w$ , entre  $w$  e  $v$ , é um vértice pertencente a uma aresta entre  $e_1, \dots, e_{h-1}$ , e portanto não pertence a nenhuma aresta de  $E(G - \{e_1, \dots, e_{h-1}\})$ . Neste caso, como no caso anterior, concluímos que  $w = y_i$ . Seja  $E'' = E - \{e_1, \dots, e_{h-1}\}$ . Como  $e_k$  é uma aresta bi-simplicial em  $G[E'']$ , ou  $w \notin V(G[E''])$  ou  $v \notin V(G[E''])$  (ou ambos). Como  $w, v \in V(G[E'])$  e  $V(G[E']) - V(G[E'']) = \{y_i\}$ , temos que  $w = y_i$ . Sem perda de generalidade, digamos que  $w = y_i$ . Portanto a aresta  $x_h y_i \in E$ . Como  $x_i y_i$  é bi-simplicial em  $G - \{e_1, \dots, e_{i-1}\}$ , então  $x_h y_j \in E$ . Como  $x_h y_h$  é bi-simplicial em  $G - \{e_1, \dots, e_{h-1}\}$ , então  $v y_j \in E$ . Como  $x_i y_j$  é bi-simplicial em  $G$ , então  $v y_i \in E$  e portanto  $v y_i \in E'$ , o que contradiz  $v y_i = v w \notin E'$ .

O caso no qual somente  $v$ , entre  $w$  e  $v$ , é um vértice pertencente a uma aresta entre  $e_1, \dots, e_{h-1}$  é análogo.

(ii)  $e'$  é bi-simplicial em  $G - \{e, e_1, \dots, e_{i-1}, e_{i+1}, \dots, e_{j-1}\}$ :

Seja  $E' = E(G - \{e, e_1, \dots, e_{i-1}, e_{i+1}, \dots, e_{j-1}\})$ . Como  $e = x_i y_j$  é bi-simplicial em  $G$  e  $x_i y_i, x_j y_j \in E$ , então  $e' = x_j y_i \in E$ . Se  $e'$  não é bi-simplicial em  $G[E']$ , então existem  $w \in N_{G[E']}(x_j)$  e  $v \in N_{G[E']}(y_i)$  tais que  $wv \notin E'$ . Como  $x_i y_i$  é bi-simplicial em  $G - \{e_1, \dots, e_{i-1}\}$  então  $v y_j \in E$ . Como  $x_j y_j$  é bi-simplicial em  $G - \{e_1, \dots, e_{j-1}\}$  então  $v w \in E$ , uma contradição.

**Caso III:** Se um dos extremos de  $e = xy$  não é ponta de nenhuma das arestas que participam do esquema original  $[e_1, \dots, e_n]$ , sem perda de generalidade, considere  $x = x_i$  para algum  $1 \leq i \leq n$  e  $y \neq y_j$  para cada  $1 \leq j \leq n$ . Mostraremos que  $[e, e_1, \dots, e_{i-1}, e_{i+1}, \dots, e_n]$  é um esquema.

Para as arestas  $e_1, \dots, e_{i-1}$  a bi-simplicialidade continua a ocorrer pela hereditariedade da propriedade. Seja  $i < h \leq n$  e  $E' = E(G - \{e, e_1, \dots, e_{i-1}, e_{i+1}, \dots, e_{h-1}\})$ . Mostremos que  $e_h$  é bi-simplicial em  $G[E']$ .

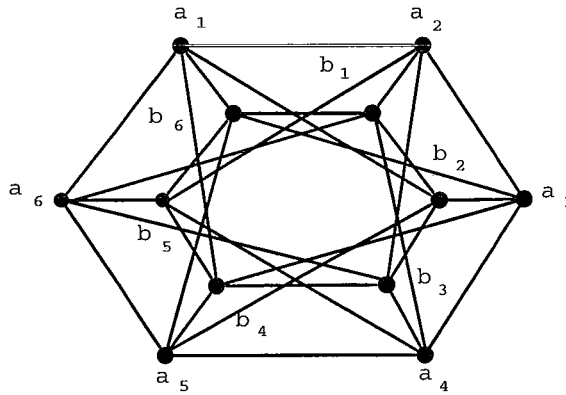
Suponha o contrário e portanto que existem  $w \in N_{G[E']}(x_h)$  e  $v \in N_{G[E']}(y_h)$  tais que  $wv \notin E'$ . O único vértice em  $Y$  que deixou de ser removido foi  $y_i$  ( $e_i$  não participa mais do esquema), logo  $w = y_i$ . Portanto,  $x_h y_i \in E$ . Como  $x_i y_i$  é bi-simplicial em  $G - \{e_1, \dots, e_{i-1}\}$ , então  $x_h y_j \in E$ . Como  $x_h y_h$  é bi-simplicial em  $G - \{e_1, \dots, e_{h-1}\}$ , então  $v y_j \in E$ . Como  $x_i y_j$  é bi-simplicial em  $G$ , então  $v y_i \in E$ ,

contrariando  $vy_i = vw \notin E$ . □

Nem todo grafo bipartido é de eliminação perfeita, como ilustrado a seguir.

**Teorema 2.28** (Golumbic [12] – cap. 12, exerc. 3). *Seja  $G$  um grafo constituído pelos ciclos induzidos disjuntos  $C_a = (a_1, \dots, a_n)$  e  $C_b = (b_1, \dots, b_n)$ ,  $n$  par, de modo que  $a_i b_j$  é uma aresta se e somente se  $i + j$  é ímpar. Então,  $G$  é um grafo bipartido que não é de eliminação perfeita para cada  $n \geq 6$ .*

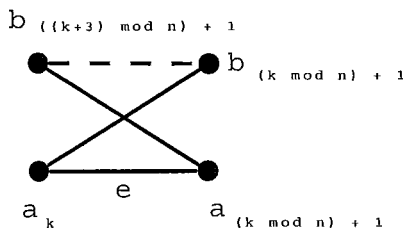
*Demonstração.* Um exemplo para  $n = 6$  pode ser visto na figura a seguir.

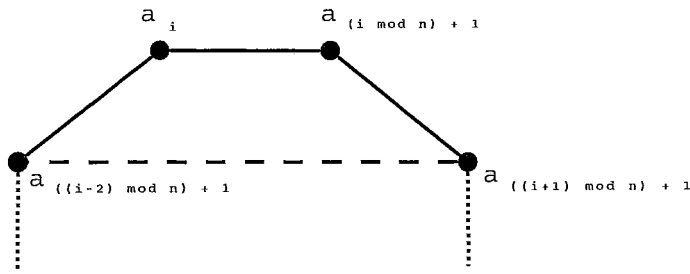


É fácil verificar que  $\{a_i, b_i \mid i \text{ é ímpar}\}$  e  $\{a_i, b_i \mid i \text{ é par}\}$  formam uma bipartição dos vértices de  $G$ . Supondo que  $G$  seja bipartido de eliminação perfeita, então existe aresta  $e$  bi-simplicial em  $G$ . Uma das seguintes possibilidades ocorre:

- (i)  $e \in C_a$  (análogo se  $e \in C_b$ );

Neste caso,  $e = a_k a_{(k \bmod n) + 1}$ , para algum  $1 \leq k \leq n$ . Note que existem as arestas  $a_k b_{(k \bmod n) + 1}$  e  $a_{(k \bmod n) + 1} b_{((k+3) \bmod n) + 1}$ , pois em ambos os casos as somas dos índices dos vértices é ímpar. Além disso,  $b_{((k+3) \bmod n) + 1} b_{(k \bmod n) + 1} \notin E(G)$ , contrariando a hipótese de que  $e$  é bi-simplicial.

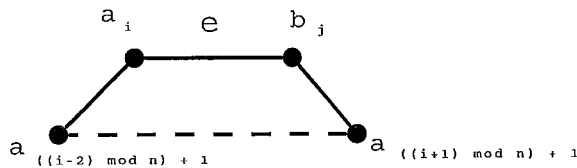




(ii)  $e = a_i b_j$  com  $i + j$  ímpar:

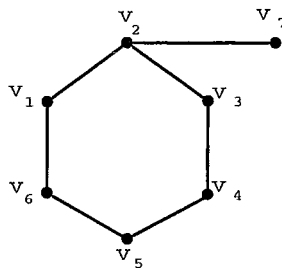
Observe que numa ordem circular os índices são representados com a seguir:

Assim,  $a_{((i-2) \bmod n+1)} a_i \in E$  mas  $a_{((i-2) \bmod n+1)} a_{((i+1) \bmod n+1)} \notin E(G)$ . Além disso,  $a_{((i+1) \bmod n+1)} b_j \in E(G)$  pois  $i + j + 2$  é ímpar uma vez que  $i + j$  também o é. Portanto, novamente a hipótese de que  $e$  é bi-simplicial foi contradita.



□

A classe dos grafos bipartidos de eliminação perfeita inclui propriamente a classe dos grafos bipartidos cordais 3.15. O grafo  $C_6$  com uma aresta pendurada em um de seus vértices (Figura 2.4) é grafo bipartido de eliminação perfeita mas não é grafo bipartido cordal. O esquema de eliminação para ele, com os rótulos abaixo, é  $[v_2 v_7, v_1 v_6, v_3 v_4]$ .



No Capítulo 3 veremos mais resultados relacionando a classe dos grafos bipartidos de eliminação perfeita com a classe dos grafos bipartidos cordais.

# Capítulo 3

## Propriedades, Caracterizações e Aplicações dos Grafos Bipartidos Cordais

Neste capítulo compilamos as propriedades e caracterizações dos grafos bipartidos cordais encontradas na literatura. Como veremos adiante, tais caracterizações utilizam abordagens bem distintas, entre elas: propriedades da vizinhança, esquema de eliminação, separador minimal, subgrafos proibidos, matrizes livres de gama, entre outras. Neste sentido, percebe-se que os grafos bipartidos cordais são ricos em estrutura, podendo ser vistos por diferentes pontos de vista, relacionando-se a diversos problemas. Além disso, neste capítulo mostramos exemplos de aplicações onde os grafos bipartidos cordais são empregados.

Na Seção 3.1, discutimos algumas propriedades gerais dos grafos bipartidos cordais. Na Seção 3.2, apresentamos um resumo sobre as principais caracterizações dos grafos bipartidos cordais, em particular aquelas que serão importantes para as caracterizações e algoritmos de reconhecimento que se encontram no capítulo seguinte. Finalmente, na Seção 3.3, apresentamos uma aplicação na álgebra linear onde os grafos bipartidos cordais são usados.

### 3.1 Propriedades gerais dos bipartidos cordais

Os resultados a seguir apresentam propriedades importantes do grafos bipartidos cordais. Tais propriedades serão utilizadas direta ou indiretamente nas caracterizações e algoritmos de reconhecimento descritos nas seções seguintes. Em primeiro lugar, uma propriedade básica, mas importante, dos grafos bipartidos cordais:

**Lema 3.1.** *A classe dos grafos bipartidos cordais é hereditária.*

*Demonstração.* Com o propósito de encontrar um absurdo, suponha que o grafo  $G$  seja bipartido cordal, mas  $G - v$  não o seja para certo  $v \in V(G)$ , isto é,  $G - v$  não seja bipartido ou tenha um ciclo induzido par de comprimento maior que 4. Como a classe dos grafos bipartidos é hereditária, existe um ciclo induzido  $C_k$  em  $G - v$  para algum  $k = 3$  ou  $k \geq 5$ . Como  $v \notin V(C_k)$ , então  $C_k$  também é subgrafo induzido de  $G$ , uma contradição.  $\square$

A próxima propriedade de grafos bipartidos cordais é baseada no conceito de separação de arestas.

**Definição 3.2.** Um grafo bipartido  $G$  é dito *separável* se existirem um par  $ab, cd \in E(G)$  com  $a, b, c, d$  distintos e  $S \subseteq V(G)$  tais que as arestas  $ab$  e  $cd$  fiquem em componentes conexas distintas de  $G - S$ .

Como consequência direta, podemos enunciar o seguinte resultado.

**Lema 3.3.** *Um grafo bipartido  $G = (X, Y, E)$  é separável se e somente se existir um par  $x_1y_1, x_2y_2 \in E$  com  $x_1, x_2 \in X$  e  $y_1, y_2 \in Y$  tais que  $x_1y_2 \notin E$  e  $x_2y_1 \notin E$ .*

*Demonstração.* Se  $G$  é separável, então existem  $x_1y_1, x_2y_2 \in E$  com  $x_1, x_2 \in X$  tal que a remoção de um certo conjunto  $S \subset V(G)$  faz com que estas arestas fiquem em componentes conexas distintas. Logo,  $x_1y_2 \notin E$  e  $x_2y_1 \notin E$ . Por outro lado, se existir um par  $x_1y_1, x_2y_2 \in E$  com  $x_1, x_2 \in X$ ,  $y_1, y_2 \in Y$  tal que  $x_1y_2 \notin E$  e  $x_2y_1 \notin E$ , definimos  $S = V(G) - \{x_1, x_2, y_1, y_2\}$  e claramente  $x_1y_1$  e  $x_2y_2$  ficam em componentes conexas distintas de  $G - S$ .  $\square$

O conceito de separação tem papel importante na classe dos grafos bipartidos cordais. De fato, todo grafo bipartido que não é separável é um grafo bipartido cordal, como mostra o seguinte resultado.

**Lema 3.4** (Golumbic [12]). *Todo grafo bipartido não-separável é bipartido cordal.*

*Demonstração.* Seja  $G = (X, Y, E)$  um grafo bipartido não-separável com ciclo induzido  $C_n = (x_1, y_1, x_2, y_2, \dots, x_k, y_k)$ ,  $k \geq 3$ . Como as arestas  $x_1y_1$  e  $x_3y_2$  não são separáveis, então pelo menos umas das arestas  $x_1y_2$ ,  $x_3y_1$  pertence a  $E(G)$ . Ou seja,  $C_k$  tem uma corda, contrariando o fato do ciclo ser induzido. Logo,  $G$  é um grafo bipartido cordal.  $\square$

**Observação 3.5.** Existem grafos bipartidos cordais separáveis, como pode ser visto na Figura 3.1.

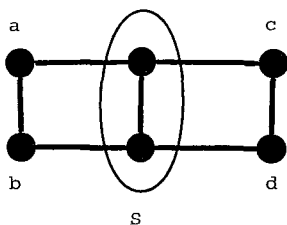


Figura 3.1: Exemplo de grafo bipartido cordal separável.

Em resumo: a classe dos grafos bipartidos não-separáveis está propriamente contida naquela dos grafos bipartidos cordais. Além disso, podemos citar mais uma propriedade de um grafo bipartido não-separável.

**Teorema 3.6** (Golumbic [12]). *Se um grafo bipartido é não-separável, então cada vértice  $v$  não-isolado deste grafo possui uma aresta bi-simplicial incidente a  $v$ .*

*Demonstração.* Seja  $G = (X, Y, E)$  um grafo bipartido não-separável. Suponha que exista um vértice não-isolado  $y_0 \in V(G)$  que não possui nenhuma aresta bi-simplicial incidente a ele. Sem perda de generalidade considere  $y_0 \in Y$ . Seja  $x_0 \in N(y_0)$ . Mostraremos, por indução em  $k$ , que existem  $X_k = \{x_0, x_1, \dots, x_k\}$  e  $Y_k = \{y_0, y_1, \dots, y_k\}$  para todo  $k \geq 0$  tais que:

(i)  $X_0 \subset X_1 \subset \dots \subset X_k$ ;

(ii)  $x_iy_j$  se e somente se  $0 \leq i, j \leq k$  e  $(j = 0 \text{ ou } i < j)$ .

Logo, como  $X_k$  existe para todo  $k \geq 0$ , pela propriedade (i) concluímos que  $G$  é infinito, o que contraria o fato de estarmos trabalhando com grafos finitos, seguindo o resultado. Tratemos de provar a indução.

Trivialmente a proposição vale para  $X_0 = \{x_0\}$  e  $Y_0 = \{y_0\}$ . Suponha que as condições sejam verdadeiras para algum  $k \geq 0$ . Formemos  $X_{k+1}$  e  $Y_{k+1}$  como se segue. Note que  $x_k$  é adjacente somente a  $y_0$  em  $Y_k$  pela propriedade (ii). Como  $x_k y_0$  não é bi-simplicial, sejam  $x_{k+1} \in X$  e  $y_{k+1} \in Y$ , distintos de  $x_k$  e  $y_0$ , tais que  $x_{k+1} y_0, x_k y_{k+1} \in E$  e  $x_{k+1} y_{k+1} \notin E$ . Como  $x_k$  é adjacente em  $Y_k$  somente a  $y_0$ , então  $y_{k+1} \notin Y_k$ . Por outro lado, para cada  $0 \leq i < k$ , como  $x_k y_{i+1} \notin E$  e além disso as arestas  $x_i y_{i+1}$  e  $x_k y_{k+1}$  não são separáveis, então  $x_i y_{k+1} \in E$ . Como  $x_{k+1} y_{k+1} \notin E$ , então  $x_{k+1} \notin X_k$ . Finalmente, para cada  $0 < i \leq k$ ,  $x_{k+1} y_i \notin E$ , pois caso contrário, como  $x_k y_i \notin E, x_{k+1} y_{k+1} \notin E$ , as arestas  $x_{k+1} y_i$  e  $x_k y_{k+1}$  seriam separáveis. É trivial verificar que as condições (i) e (ii) valem para  $X_{k+1}$  e  $Y_{k+1}$ .  $\square$

Tratemos agora dos grafos bipartidos separáveis. O próximo lema evidencia como são os separadores de arestas de grafos bipartidos cordais.

**Lema 3.7.** *Se  $G = (X, Y, E)$  é um grafo bipartido cordal separável e  $S$  é um separador de arestas minimal, então  $S$  é uma biclique.*

*Demonstração.* Se  $S \subset X$  ou  $S \subset Y$  o resultado é trivialmente verdade. Suponha, com o propósito de encontrar um absurdo, que existam  $s_x \in S \cap X$  e  $s_y \in S \cap Y$  tais que  $s_x s_y \notin E$ .

Sejam  $x_1 y_1$  e  $x_2 y_2$  as arestas separadas por  $S$  e  $G_1, G_2$  as componentes conexas de  $G - S$ . Como  $S$  é minimal, seja  $x'_1 \in V(G_1)$  (resp.  $y'_1 \in V(G_1)$ ) tal que  $x'_1 s_y \in E$  (resp.  $y'_1 s_x \in E$ ) e  $x'_1$  (resp.  $y'_1$ ) pertença a um caminho entre um vértice da aresta  $x_1 y_1$  e um vértice da aresta  $x_2 y_2$ . Defina  $x'_2, y'_2 \in V(G_2)$  de maneira análoga. Seja  $x'_1 = u_1^1, \dots, u_k^1 = y'_1$  (resp.  $x'_2 = u_1^2, \dots, u_l^2 = y'_2$ ) um caminho mínimo entre  $x'_1$  e  $y'_1$  (resp.  $x'_2$  e  $y'_2$ ) tal que  $u_i^1 \in V(G_1)$  para cada  $1 \leq i \leq k$  (resp.  $u_i^2 \in V(G_2)$  para cada  $1 \leq i \leq l$ ). Como  $u_i^1 u_j^2 \notin E$  para todo  $1 \leq i \leq k, 1 \leq j \leq l$ , então  $(x'_1 = u_1^1, \dots, u_k^1 = y'_1, s_x, y'_2 = u_l^2, \dots, u_1^2 = x'_2, s_y, x'_1)$  é um ciclo induzido de tamanho maior ou igual a 6, um absurdo.  $\square$

Os grafos bipartidos cordais não-separáveis possuem, pelo Teorema 3.6, arestas bi-simpliciais incidentes a todo vértice não-isolado. O próximo teorema mostra como as arestas bi-simpliciais se relacionam com os grafos bipartidos cordais separáveis.

**Teorema 3.8** (Golumbic [12, 11]). *Seja  $G = (X, Y, E)$  um grafo bipartido cordal. Se  $G$  é separável, então  $G$  tem duas arestas bi-simpliciais separáveis.*



*Demonstração.* Trivialmente, os menores grafos separáveis são aqueles que deixam, após a remoção de um separador de arestas minimal  $S$ , apenas duas arestas. Pelo Lema 3.7,  $S$  é uma biclique de  $G$ . Logo, tais arestas são simpliciais de  $G$ . Suponha que o resultado seja verdadeiro para todo grafo com menos vértices do que  $G$ . Como  $G$  é separável, seja  $S$  um separador minimal de um par de arestas, digamos  $x_a y_a$  e  $x_b y_b$ . Novamente pelo Lema 3.7 sabemos que  $S$  é uma biclique. Sejam  $G_A$  e  $G_B$  as componentes conexas de  $G[X \cup Y - S]$ . Mostraremos que  $G_A \cup G[S]$  (resp.  $G_B \cup G[S]$ ) possui uma aresta  $xy$  bi-simplicial tal que  $x, y \in V(G_A)$  (resp.  $x, y \in V(G_B)$ ), seguindo o resultado. Tratemos de provar esta afirmação.

**Caso I:**  $G_A \cup G[S]$  é separável.

Suponha com o propósito de encontrar um absurdo que  $G_A \cup G[S]$  não seja separável. Por hipótese de indução,  $G_A \cup G[S]$  possui arestas bi-simpliciais separáveis  $x_1 y_1$  e  $x_2 y_2$ . Se  $x_1, y_1 \in V(G_A)$  ou  $x_1, y_1 \in V(G_B)$ , não há nada a provar. Caso contrário,  $x_1, y_1$  (resp.  $x_2, y_2$ ) não pertencem ambos a  $V(G_A)$ . Por outro lado, como  $S$  é uma biclique,  $x_1, y_2$  (resp.  $x_2, y_1$ ) não pertencem ambos a  $S$ . Portanto, resta mostrar que a configuração  $x_1, x_2 \in S$ ,  $y_1, y_2 \in V(G_A)$  (e analogamente  $x_1, x_2 \in V(G_B)$ ,  $y_1, y_2 \in S$ ) também não é possível, seguindo a afirmação que tínhamos por objetivo demonstrar.

Suponha que  $x_1, x_2 \in S$  e  $y_1, y_2 \in V(G_A)$ . Unindo-se um caminho mínimo  $y_1, a_1, \dots, a_r, y_2$  em  $G_A$  com um caminho mínimo  $x_2, b_1, \dots, b_s, x_1$  tal que  $b_i \in V(G_B)$ ,  $1 \leq i \leq s$ , obtém-se um ciclo de tamanho maior ou igual a 6 que portanto deve ter uma corda. No entanto,  $a_i b_j \notin E(G)$  para todo  $1 \leq i \leq r$ ,  $1 \leq j \leq s$ . Portanto, existe ou a corda  $x_1 y_2$ , ou a corda  $x_2 y_1$ , ambas contradizendo a separabilidade de  $x_1 y_1$  e  $x_2 y_2$ .

**Caso II:**  $G_A \cup G[S]$  é não-separável.

Seja  $x_1 y_2 \in E(G_A)$ . Pelo Teorema 3.6, existem vértices  $y_1$  e  $x_2$  tais que  $x_1 y_1 \in E(G_A \cup G[S])$  e  $x_2 y_2 \in E(G_A \cup G[S])$  são arestas bi-simpliciais em  $G_A \cup G[S]$ . Se  $x_2$  ou  $y_1$  pertencem a  $V(G_A)$ , então segue a afirmação que pretendíamos mostrar. Logo, considere  $x_2, y_1 \in S$ . Como  $S$  é uma biclique de  $G$ , então  $x_2 y_1 \in E(G_A \cup G[S])$ . Além disso,  $N_{G_A \cup G[S]}(x_1) \supseteq N_{G_A \cup G[S]}(x_2)$ , pois  $x_2 y_2$  é bi-simplicial em  $G_A \cup G[S]$  e  $x_1 y_2$  é uma aresta. Do mesmo modo  $N_{G_A \cup G[S]}(x_2) \supseteq N_{G_A \cup G[S]}(x_1)$ , pois  $x_1 y_1$  é bi-simplicial em  $G_A \cup G[S]$  e  $x_2 y_1$  é uma aresta. Portanto,  $N_{G_A \cup G[S]}(x_1) = N_{G_A \cup G[S]}(x_2)$ . Logo,

$N_{G_A \cup G[S]}(x_1) \cup N_{G_A \cup G[S]}(y_2) = N_{G_A \cup G[S]}(x_2) \cup N_{G_A \cup G[S]}(y_2)$ . Como  $N_{G_A \cup G[S]}(x_2) \cup N_{G_A \cup G[S]}(y_2)$  é uma biclique em  $G_A \cup G[S]$ , então  $x_1 y_2$  é uma aresta bi-simplicial em  $G_A \cup G[S]$ .  $\square$

O resultado a seguir é uma propriedade dos separadores de arestas minimais dos grafos bipartidos cordais que generaliza o resultado deixado como exercício 12 do capítulo 4 de [12]:

**Teorema 3.9** (Bakonyi e Bono [3]). *Seja  $G = (X, Y, E)$  um grafo bipartido cordal separável sendo  $S$  um separador minimal das arestas  $x_a y_a$  e  $x_b y_b$ . Existe  $x'_a \in X \cup Y - S$  na componente conexa de  $G - S$  que contém  $x_a y_a$  tal que  $S \cap Y \subset N(x'_a)$ .*

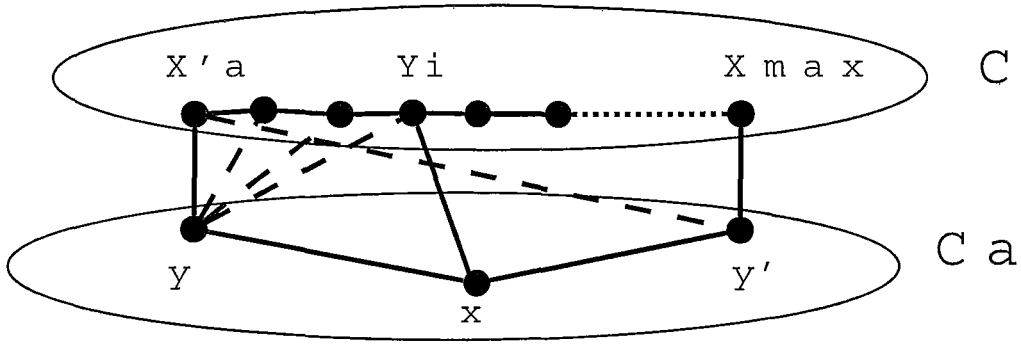
*Demonstração.* Seja  $G_A = (X_A, Y_A, E_A)$  a componente conexa de  $G[X \cup Y - S]$  que contém  $x_a y_a$  e  $G_B = (X_B, Y_B, E_B)$  a que contém  $x_b y_b$ . Claro que  $X_A \neq \emptyset$ . Seja  $S_Y = S \cap Y$ . Considere  $x_{MAX} \in X_A$ , tal que  $|N(x_{MAX}) \cap S_Y| \geq |N(x) \cap S_Y|$  para todo  $x \in X_A$ .

Se, para todo  $x \in X_A$ ,  $N(x) \cap S_Y \subseteq N(x_{MAX}) \cap S_Y$ , então não pode existir  $v \in S_Y$  tal que  $v \notin N(x_{MAX})$  pois tal  $v$  não seria adjacente a nenhum vértice de  $X_A$  contrariando o fato de  $S$  ser minimal. Portanto,  $N(x_{MAX}) \cap S_Y = S_Y$ , ou seja,  $x_{MAX}$  é o  $x'_a$  que procurávamos.

Caso contrário, seja  $x'_a \in X_A$  um vértice a menor distância de  $x_{MAX}$  tal que  $N(x'_a) \cap S_Y \not\subseteq N(x_{MAX}) \cap S_Y$  e seja  $C = x'_a, y_1, \dots, y_k, x_{MAX}$  esse caminho. Observe que todo vértice  $x_i \in C$  é tal que  $N(x_i) \cap S_Y \subseteq N(x_{MAX}) \cap S_Y$ . Tome  $y \in N(x_{MAX}) \cap S_Y - N(x'_a)$  e  $y' \in N(x'_a) \cap S_Y - N(x_{MAX})$ . É claro que existe um caminho induzido  $C_b = y, x_{b_1}, y_{b_1}, \dots, y'$  de tamanho pelo menos 2, onde para todo  $i$ ,  $x_{b_i} \in X_B$  e  $y_{b_i} \in Y_B$ . Mostraremos que existe um caminho induzido  $C_a = y, x_{a_1}, y_{a_1}, \dots, y'$  de tamanho pelo menos 4, onde para todo  $i$ ,  $x_{a_i} \in X_A$  e  $y_{a_i} \in Y_A$ . Portanto, teríamos um ciclo induzido de tamanho maior ou igual a 6 em  $G$  formado pela junção de  $C_b$  e  $C_a$ , contrariando a hipótese de se tratar de um grafo bipartido cordal. Note que não existe aresta entre vértices de  $C_b$  e  $C_a$  em  $G[X \cup Y - S]$  uma vez que estão em componentes conexas diferentes e além disso não existe corda entre vértices do mesmo caminho pois ambos são induzidos.

De fato, existe um caminho induzido  $C_a$ . Se  $|C_a| \geq 4$ , então o resultado segue. Suponha, portanto que,  $|C_a| = 2$ , ou seja,  $C_a = y, x, y'$ . Como  $x'_a y \notin E$ , então

$x \neq x'_a$ . Além disso,  $x \neq x_i, \forall x_i \in C$  pois nenhum  $x_i \in C$  é adjacente a  $y'$  para satisfazer  $N(x_i) \cap S_Y \subseteq N(x_{MAX}) \cap S_Y$ .



Seja  $y_i \in C$  tal que  $y_i x \in E$  com  $i$  mínimo. Assim considere  $C'$  tal que  $C = x'_a, C', y_i, \dots, x_{MAX}$ . Agora, observe o seguinte ciclo  $C^* = (y', x'_a, C', y_i, x)$ . Note que não existem cordas entre os vértices de  $x'_a, C', y_i$  pois fazem parte de um caminho induzido. Além disso, nenhum  $x_i \in C'$  é adjacente a  $y'$  pois satisfazem  $N(x_i) \cap S_Y \subseteq N(x_{MAX}) \cap S_Y$ . Para finalizar as possibilidades de corda em  $C^*$ , nenhum  $y_l \in C'$  é adjacente a  $x$  pois  $y_i$  foi escolhido com menor índice. Portanto,  $C^*$  é induzido contrariando o fato de  $G$  ser bipartido cordal.  $\square$

O teorema a seguir mostra que é possível adicionar uma aresta a qualquer vértice de um grafo bipartido não-separável de forma que o grafo resultante continue sendo um grafo bipartido não-separável.

**Proposição 3.10** (Bakonyi e Bono [3]). *Dado um grafo bipartido não-separável  $G = (X, Y, E)$  e  $x \in X$  tal que  $N(x) \neq Y$ , então existe  $y \in Y$  tal que  $G' = (X, Y, E \cup \{xy\})$  também é não-separável. Afirmação similar vale para  $y \in Y$  tal que  $N(y) \neq X$ .*

**Teorema 3.11** (Bakonyi e Bono [3]). *Seja  $G = (X, Y, E)$  um grafo bipartido cordal e  $x \in X$  tal que  $N(x) \neq Y$ . Então existe  $y \in Y$  tal que  $G' = (X, Y, E \cup \{xy\})$  também é bipartido cordal. Afirmação similar vale para  $y \in Y$  tal que  $N(y) \neq X$ .*

*Demonstração.* Seja  $m = |E|$ . Se  $m \leq 4$ , o resultado é claramente válido, uma vez que a adição de uma aresta não pode formar um ciclo induzido  $C_n, n \geq 6$ . Suponha que isto seja verdade para todo grafo com menos do que  $m$  arestas para um  $m > 4$ .

Se  $G$  é não-separável, então o grafo  $G'$  da Proposição 3.10 também é não-separável. Como todo bipartido não-separável é bipartido cordal pelo Lema 3.4, então  $G'$  é bipartido cordal. Suponha, portanto, que  $G$  é um grafo separável.

Seja  $x \in X$ , com  $N(x) \neq Y$ . Pelo Teorema 3.8, existem arestas bi-simpliciais separáveis  $x_a y_a$  e  $x_b y_b$ . Como  $G$  é bipartido cordal e  $x_a y_a$  é bi-simplicial, então pela Proposição 3.18,  $\hat{G} = (X, Y, E - \{x_a y_a\})$  é bipartido cordal.

Sem perda de generalidade, por uma re-rotulação entre as arestas  $x_a y_a$  e  $x_b y_b$ , assumamos que  $x \neq x_a$ . Por hipótese de indução, considerando  $\hat{G}$  e  $x$ , existe  $y$  tal que  $\hat{G} = (X, Y, E - \{x_a y_a\} \cup \{xy\})$  é bipartido cordal. Suponha, por absurdo, que  $G' = (X, Y, E \cup \{xy\})$  não seja bipartido cordal. Logo, existe um ciclo induzido  $C_n$  em  $G'$ ,  $n \geq 6$ . Como  $\hat{G}$  é bipartido cordal mas  $G'$  não o é, então  $x_a y_a \in C_n$ . Logo, podemos representar  $V(C_n)$  numa sequência de leitura do ciclo por  $(\dots, y', x_a, y_a, x', \dots)$ . Vejamos as possibilidades de existência de tal vértice  $y$ :

**Caso I:**  $y \neq y_a$ .

Como  $x_a y_a$  é bi-simplicial em  $G$ ,  $x \neq x_a$  e  $y \neq y_a$ , a adição da aresta  $xy$  em  $G$  não altera o fato de que  $N_{G'}(x_a) \cup N_{G'}(y_a)$  é uma biclique. Assim,  $x_a y_a$  é bi-simplicial em  $G'$ . Como  $x' \in N_{G'}(y_a)$  e  $y' \in N_{G'}(x_a)$ , então  $x' y' \in E(G')$ , contrariando o fato de  $C_n$  não ter cordas.

**Caso II:**  $y = y_a$ .

Note que  $x y_a \in C_n$ , pois caso contrário  $C_n$  seria induzido também em  $G' - \{x y_a\} = G$ , contradizendo  $G$  ser bipartido cordal. Logo,  $x' = x$ . Observe que, ao contrário do caso anterior, não podemos concluir que  $x_a y_a$  é bi-simplicial em  $G'$ , acarretando a existência da corda  $x y'$  e de uma contradição. Vejamos, portanto, como prosseguir neste caso.

Seja  $G_a = G[X \cup Y - \{y_a\}]$ . Aplicando indução em  $G_a$ , podemos obter o grafo bipartido cordal  $G''_a$  a partir de  $G_a$  adicionando-se a aresta  $xz$ ,  $z \in Y - \{y_a\}$ . Seja  $G'_a = (X, Y, E \cup \{xz\})$ .

Se  $G'_a$  não é bipartido cordal, existindo um ciclo induzido  $C'_n$  em  $G'_a$ ,  $n \geq 6$ , então como  $G$  é bipartido cordal mas  $G'_a$  não o é, temos que  $xz \in C'_n$ . Além disso, como  $G''_a$  é bipartido cordal mas  $G'_a$  não o é, temos que  $y_a \in C'_n$ . Como  $x y_a \in E$ , podemos representar  $C'_n$  portanto pela sequência  $\dots, y_a, x, z, \dots$ . Como  $x'_i, x'_{i+1} \in N_G(y_a)$ ,  $y' \in N_G(x_a)$  e  $x_a y_a$  é bi-simplicial em  $G$ , então  $x'_i y', x'_{i+1} y' \in E$ . Portanto, o

ciclo  $C_n'''$  representado pela sequência  $(x, z, x'_1, y'_1, \dots, x'_i, y'_i, x'_{i+1}, y'_{i+1}, \dots, x'_{n/2}, y'_{n/2})$  é um ciclo de  $G_a''$ . Como  $G_a''$  é bipartido cordal, então existe a corda  $xy'$  em  $C_n'''$ , contrariando assim o fato de  $C_n$  não ter cordas. Portanto um dos grafos,  $G'$  ou  $G_a''$ , é bipartido cordal, em ambos os casos seguindo o resultado.  $\square$

## 3.2 Caracterizações do grafos bipartidos cordais

Enquanto na seção anterior apresentamos uma coleção de propriedades gerais dos grafos bipartidos cordais, neste seção descreveremos as caracterizações dos grafos bipartidos cordais existentes na literatura.

**Teorema 3.12** (Golumbic e Goss [13]). *Um grafo bipartido é bipartido cordal se e somente se todo separador de arestas minimal for uma biclique.*

*Demonstração.* Se  $G$  é um bipartido cordal separável, e  $S$  é um separador de arestas minimal, pela Lema 3.7,  $S$  é uma biclique de  $G$ .

Por outro lado, considere  $G$  um grafo bipartido tal que todo separador de arestas minimal é uma biclique. Mostraremos que  $G$  é bipartido cordal. Suponha, por absurdo, que  $G$  tenha um ciclo induzido  $C = (v_1, v_2, \dots, v_k)$  de tamanho  $k \geq 6$ . Seja  $S = N(v_2) \cup N(v_3) \cup N(v_5) \cup N(v_6) - \{v_2, v_3, v_5, v_6\}$  o conjunto de todos os vértices adjacentes a pelo menos um dos vértices  $v_2, v_3, v_5$  e  $v_6$  sem incluir nenhum destes quatro vértices. Note que  $S$  separa a aresta  $v_2v_3$  de  $v_5v_6$ . Seja  $S' \subseteq S$  um separador minimal das arestas  $v_2v_3$  e  $v_5v_6$ . Sabemos, por hipótese, que  $S'$  é uma biclique. Além disso,  $v_4 \in S'$  e  $v_7 \in S'$  (ou  $v_1 \in S'$ , quando  $k = 6$ ). Portanto, como  $v_4$  e  $v_7$  (ou  $v_1$ ) estão em partes diferentes devido a paridade oposta dos índices destes vértices, existe uma corda em  $C$ , um absurdo pois  $C$  não tem cordas. Logo,  $G$  é bipartido cordal.  $\square$

O próximo lema mostra uma propriedade importante sobre complementos bipartidos de grafos, que será especialmente útil para os grafos bipartidos cordais a seguir.

**Lema 3.13.** *Seja  $G$  um grafo bipartido. Se  $G$  é livre de  $C_6$ , então  $\hat{G}$  é livre de  $C_n$ ,  $n \geq 10$ .*

*Demonstração.* Mostraremos que se  $\hat{G}$  contém um subgrafo induzido isomorfo a um  $C_n$ ,  $n \geq 10$ , então  $G$  contém um subgrafo induzido isomorfo a  $C_6$ .

Suponha, portanto, que  $\hat{G}$  contenha um subgrafo induzido isomorfo a  $C_n = (v_1, v_2, \dots, v_n)$ , com  $n \geq 10$ ,  $n$  par (pois  $\hat{G}$  é bipartido). Sabemos que  $N_{\hat{G}}(v_i) \supseteq \{v_{(i \bmod n) + 1}, v_{((i-2) \bmod n) + 1}\}$  para todo  $i = 1, \dots, n$ .

Observe que existem as seguintes arestas em  $G$ :  $v_1v_4, v_4v_7, v_7v_2, v_2v_5, v_5v_8, v_8v_1$ , formando um  $C_6$  induzido pois nenhuma corda é possível neste ciclo. Lembre-se que  $v_iv_j \notin E$  e  $v_iv_j \notin \hat{E}$  sempre que  $i$  e  $j$  tem a mesma paridade.  $\square$

O próximo teorema caracteriza os grafos bipartidos cordais auto-complementares. Esta caracterização conduz a um algoritmo eficiente de reconhecimento.

**Teorema 3.14** (Golumbic e Goss [13]). *Seja  $G$  um grafo bipartido. Então  $G$  e  $\hat{G}$  são ambos bipartidos cordais se e somente se  $G$  não contiver nenhum subgrafo induzido isomorfo ao  $C_6$ ,  $3K_2$ , ou  $C_8$ .*

*Demonstração.* Suponha que  $G$  e  $\hat{G}$  são grafos bipartidos cordais. Claro que  $G$  não tem subgrafo induzido isomorfo ao  $C_6$  nem  $C_8$ . Suponha, por absurdo, que  $G$  tenha um subgrafo induzido  $H$  isomorfo ao  $3K_2$ . Veja que  $\hat{G}[V(H)]$  será isomorfo ao  $C_6$ , um absurdo pois  $\hat{G}$  é bipartido cordal.

Por outro lado, suponha que  $G$  não contenha nenhum subgrafo induzido isomorfo ao  $C_6$ ,  $3K_2$ , ou  $C_8$ . Primeiramente, mostraremos que  $G$  é bipartido cordal. Para isso precisamos mostrar apenas que  $G$  não contém nenhum subgrafo induzido isomorfo ao  $C_n$ ,  $n \geq 10$  (pois  $G$  não tem nem  $C_6$ , nem  $C_8$  por hipótese). Suponha, por absurdo, que  $\hat{G}$  tenha um subgrafo induzido  $H$  isomorfo ao  $C_6$ . Veja que  $G[V(H)]$  será isomorfo ao  $3K_2$ , contrariando a hipótese. Logo, pelo Lema 3.13,  $G$  não contém nenhum subgrafo induzido isomorfo ao  $C_n$ ,  $n \geq 10$  e, portanto, é bipartido cordal. Mostraremos na sequência que  $\hat{G}$  é bipartido cordal.

Pelo Lema 3.13 e pelo fato de  $G$  não possuir nenhum subgrafo induzido isomorfo ao  $C_6$ , sabemos que  $\hat{G}$  não contém nenhum subgrafo induzido isomorfo a  $C_n$ ,  $n \geq 10$ . Resta mostrar que  $\hat{G}$  não contém nenhum subgrafo induzido isomorfo nem ao  $C_6$ , nem ao  $C_8$ . Suponha que  $H$  seja um subgrafo induzido em  $\hat{G}$  isomorfo ao  $C_6$ . Então,  $G[V(H)]$  é isomorfo ao  $3K_2$ , o que é um absurdo. Suponha agora, que  $H$

seja um subgrafo induzido em  $\hat{G}$  isomorfo ao  $C_8$ . Então,  $G[V(H)]$  é isomorfo ao  $C_8$ , novamente conduzindo a uma contradição. Portanto,  $\hat{G}$  é bipartido cordal.  $\square$

**Teorema 3.15** (Golumbic e Goss [13]). *Todo grafo bipartido cordal é um grafo bipartido de eliminação perfeita.*

*Demonstração.* Como a propriedade de ser bipartido cordal é hereditária, é suficiente mostrar que todo grafo bipartido cordal tem uma aresta bi-simplicial. De fato isto ocorre ou pelo Teorema 3.6, ou pelo Teorema 3.8, caso o grafo seja não-separável ou separável, respectivamente.  $\square$

**Corolário 3.16** (Golumbic e Goss [13]). *Se um grafo bipartido  $G$  não contém  $2K_2$  induzido, então  $G$  e  $\hat{G}$  são ambos bipartidos cordais e bipartidos de eliminação perfeita.*

*Demonstração.* Consequência imediata do Teorema 3.14 e do Teorema 3.15.  $\square$

**Corolário 3.17** (Golumbic e Goss [13]). *Um grafo é bipartido cordal se e somente se todo subgrafo induzido é bipartido de eliminação perfeita.*

*Demonstração.* Se  $G$  não é bipartido cordal então possui um ciclo induzido  $C_n$ ,  $n \geq 6$ . Mas tal  $C_n$  não é bipartido de eliminação perfeita pois  $C_n$  não tem nenhuma aresta bi-simplicial. Logo  $G$  tem um subgrafo induzido que não é bipartido de eliminação perfeita.

Por outro lado se  $G$  é bipartido cordal, então todo subgrafo induzido  $H$  de  $G$  é bipartido cordal e, pelo Teorema 3.15, é bipartido de eliminação perfeita.  $\square$

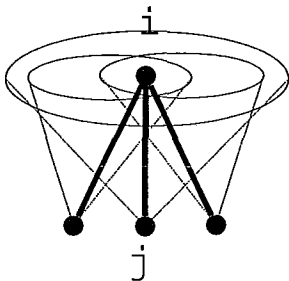
**Proposição 3.18** (Bakonyi e Bono [3]). *Um grafo bipartido  $G = (X, Y, E)$  com aresta bi-simplicial  $xy \in E$  é bipartido cordal se e somente se  $G' = (X, Y, E - \{xy\})$  é bipartido cordal.*

*Demonstração.* Sendo  $G$  um grafo bipartido cordal, suponha que exista em  $G'$  um ciclo induzido  $C_n$ ,  $n \geq 6$ . Como  $G$  é livre de  $C_m$  com  $m \geq 6$ , então  $n = 6$  e  $C_6 = (x, y_1, x_1, y, x_2, y_2)$ . Logo,  $y_1, y_2 \in N(x)$ ,  $x_1, x_2 \in N(y)$ . Como  $xy$  é bi-simplicial em  $G$ ,  $x_1y_2$  e  $x_2y_1$  são cordas de  $C_n$  em  $G'$ , resultando em um absurdo.

Por outro lado, suponha que  $G'$  é um grafo bipartido cordal e que  $G' \cup \{xy\}$  contenha um ciclo induzido  $C_n = (x, y, x_1, y_1, \dots, x_{[n/2]-1}, y_{[n/2]-1})$ ,  $n \geq 6$ ,  $n$  par. Pela Observação 1.37,  $xy$  é uma aresta bi-simplicial de  $G[\{x, y, x_1, y_1, \dots, x_{[n/2]-1}, y_{[n/2]-1}\}]$ .

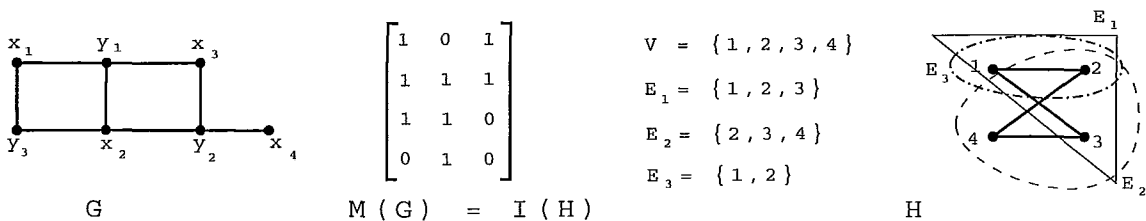
Novamente, como  $x_1 \in N(y)$ ,  $y_{[n/2]-1} \in N(x)$  e  $xy$  é uma aresta bi-simplicial de  $G[\{x, y, x_1, y_1, \dots, x_{[n/2]-1}, y_{[n/2]-1}\}]$ , então  $x_1 y_{[n/2]-1} \in E$ , contrariando o fato de  $C_n$  não ter cordas.  $\square$

**Teorema 3.19** (Dragan e Voloshin [8]). *Um grafo bipartido  $G$  é bipartido cordal se e somente se todo subgrafo induzido  $H$  de  $G$  contém vértices adjacentes  $i$  e  $j$  tais que  $N_H(j)$  contém a vizinhança restrita a  $H$  de todos os vizinhos de  $i$ .*



**Definição 3.20.** Um hipergrafo  $\mathcal{H}$  é *totalmente balanceado*  $\iff \forall$  ciclo de  $\mathcal{H}$  existe uma aresta contendo pelo menos três vértices deste ciclo.

Um exemplo pode ser visto a seguir.



**Teorema 3.21** (Brandstädt, Le e Spinrad [5], Lehel [17]). *Um grafo bipartido  $G$  é bipartido cordal se e somente se o hipergrafo definido por  $M(G)$  é totalmente balanceado.*

**Definição 3.22.** Uma *ordem perfeita* de um grafo é uma ordenação de seus vértices tal que a coloração gulosa seguindo tal ordem de qualquer um de seus subgrafos induzidos é ótima.

Um grafo *perfeitamente ordenável* é um grafo que possui uma ordem perfeita.

**Teorema 3.23** (Chvátal [6]). *Um grafo bipartido  $G$  é bipartido cordal se e somente se  $\overline{G}$  é perfeitamente ordenável.*

**Lema 3.24.** *A propriedade de uma matriz bipartida  $M$  de  $G$  ser livre de  $\Gamma$  é hereditária.*



*Demonstração.* Seja  $M$  a matriz bipartida do grafo bipartido  $G$  e suponha que  $M$  seja livre de  $\Gamma$ .

Seja  $v$  um vértice de  $G$  tal que a matriz bipartida  $M'$  de  $G' = G - v$  não seja livre de  $\Gamma$ . A matriz  $M'$  é obtida de  $M$  pela remoção da linha ou da coluna que corresponde a  $v$ . Mas então um  $\Gamma$  em  $M'$  está também presente em  $M$ , uma contradição.  $\square$

**Definição 3.25.** Um vértice  $v \in V(G)$  é dito ser *centro de  $P_5$*  se existir em  $G$  um caminho induzido de tamanho 5 no qual  $v$  é o vértice central deste caminho.

**Lema 3.26.** *Se a matriz bipartida de  $G$  é livre de  $\Gamma$ , então existe um vértice  $v \in V(G)$  que não é centro de  $P_5$ .*

*Demonstração.* Seja  $M$  a matriz bipartida de  $G$  e suponha que ela é livre de  $\Gamma$ . Suponha, por absurdo, que todo vértice é centro de  $P_5$  e seja  $v \in V(G)$  centro do  $P_5 = (x, y, v, w, z)$ . Sem perda de generalidade, por uma transposição de  $M$ , considere que  $v$  corresponda a uma linha de  $M$ . Observe que se  $v$  for anterior a  $x, z$  na ordenação de linhas de  $M$ , então, para toda permutação válida entre  $y, w, x, z$ , existe um  $\Gamma$ , como pode ser analisado na Tabela 3.1. Logo,  $v$  não pode estar na primeira linha. Como  $v$  é um vértice geral, nenhum vértice pode representar a primeira linha, o que obviamente é um absurdo.

	...	$y$	...	$w$	...
:	:	:	:	:	:
$v$	...	1	...	1	...
:	:	:	:	:	:
$x$	...	1	...	0	...
:	:	:	:	:	:
$z$	...	0	...	1	...
:	:	:	:	:	:

Tabela 3.1:  $M$  uma matriz bipartida de  $G$ .

$\square$

**Teorema 3.27** (Spinrad [24]). *Se  $G$  é grafo bipartido cordal, então existe  $v \in V(G)$  tal que  $v$  não é centro de  $P_5$ .*

*Demonstração.* Seja  $G$  um grafo bipartido cordal. Pelo Teorema 2.5 e pelo Teorema 2.25, existe uma ordenação livre de  $\Gamma$  da matriz bipartida  $M$  de  $G$ . Logo, pelo Lema 3.26, existe  $v \in V(G)$  tal que  $v$  não é centro de  $P_5$ .  $\square$

**Teorema 3.28** (Exercício 9.3 do Livro do Spinrad [24]). *Seja  $G$  um grafo bipartido. O grafo  $G$  é bipartido cordal se e somente se  $G$  tem um esquema de eliminação baseado em sempre remover vértices que não são centro de  $P_5$ .*

*Demonstração.* Seja  $G$  um grafo bipartido e  $v \in V(G)$  um vértice que não é no centro de  $P_5$ . Então:

- Se  $G$  é bipartido cordal, então pelo Lema 3.1,  $G - v$  é bipartido cordal.
- Se  $G$  não é bipartido cordal, então como  $v$  não é no centro de  $P_5$ ,  $v$  não está em nenhum  $C_n$ ,  $n \geq 6$ . Logo,  $G - v$  não é bipartido cordal.

Portanto,  $G$  é bipartido cordal se e somente se  $G - v$  é bipartido cordal. **(I)**

Seja  $G'$  o grafo resultante de eliminações sucessivas de vértices que não são centro de  $P_5$  até onde seja possível. Se  $G' = \emptyset$ , ou seja, existe esquema de eliminação perfeita, então, por **(I)** e dado que o grafo nulo é bipartido cordal,  $G$  é bipartido cordal. Se  $G' \neq \emptyset$ , então pela contra-positiva do Teorema 3.27,  $G'$  não é bipartido cordal. Logo, por **(I)**,  $G$  não é bipartido cordal.

Conclui-se que existe tal esquema de eliminação se e somente se  $G$  é bipartido cordal.  $\square$

**Lema 3.29.** *Se um grafo bipartido  $G$  tem uma ordenação livre de  $\Gamma$  de sua matriz bipartida, então  $G$  é bipartido cordal.*

*Demonstração.* Seja  $G$  um grafo bipartido com uma ordenação livre de  $\Gamma$  de sua matriz bipartida  $M$ . Pelo Lema 3.26, existe  $v_1$  que não é centro de  $P_5$ .

Pelo Lema 3.24,  $G - v_1$  também possui uma ordenação livre de  $\Gamma$  de sua matriz bipartida. Aplicando-se sucessivamente este procedimento, existe esquema de eliminação perfeita  $[v_1, \dots, v_k]$  baseado na remoção de vértices que não são centro de  $P_5$ . Portanto, pelo Teorema 3.28,  $G$  é um grafo bipartido cordal.  $\square$

**Teorema 3.30** (Spinrad [24]). *Um grafo  $G$  é bipartido cordal se e somente se existir uma ordenação livre de  $\Gamma$  de sua matriz bipartida.*

*Demonstração.* Seja  $G$  um grafo bipartido cordal. Pelo Teorema 2.5 e pelo Teorema 2.25, existe uma ordenação livre de  $\Gamma$  da sua matriz bipartida.

Por outro lado, se  $G$  tem uma ordenação livre de  $\Gamma$  de sua matriz bipartida, então, pelo Lema 3.29,  $G$  é bipartido cordal.  $\square$

### 3.3 Aplicação prática

Na seção anterior, mostramos que a classe dos grafos bipartidos cordais é rica em estrutura. Diversos resultados interessantes e não-triviais, fruto do trabalho de diversos pesquisadores, foram apresentados. Nesta seção, apresentaremos uma aplicação dos grafos bipartidos cordais em eliminação Gaussiana de matrizes (bem conhecida na álgebra linear sob o nome “pivoteamento”), demonstrando que além da riqueza teórica, esta classe possui aplicação direta em outros problemas. A descrição a seguir está baseada naquela de Golumbic [12] e Bakonyi e Bono [3].

**Definição 3.31.** Uma matriz quadrada é dita *singular* quando não admite uma inversa.

Uma matriz é *singular* se e somente se seu determinante é nulo.

#### 3.3.1 Problema da eliminação Gaussiana perfeita

Seja  $M$  uma matriz real não-singular  $n \times n$ . É bem conhecido em álgebra linear que pode-se reduzir  $M$  à matriz identidade  $I$  pelo processo repetido de substituir uma linha por uma combinação linear de outras linhas. Tal processo é chamado de *eliminação Gaussiana* ou simplesmente de *pivoteamento*. Abaixo apresentamos um exemplo de pivoteamento de uma matriz. Os pivôs foram destacados em cada etapa por uma moldura retangular.

$$\begin{pmatrix} \boxed{4} & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \xrightarrow{1} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \boxed{3} & -1 & -1 \\ 0 & -1 & 3 & -1 \\ 0 & -1 & -1 & 3 \end{pmatrix} \xrightarrow{2} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \boxed{8} & -4 \\ 0 & 0 & -4 & 8 \end{pmatrix} \xrightarrow{3} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \boxed{12} \end{pmatrix} \xrightarrow{4} I$$

Combinações lineares efetuadas:

$$\xrightarrow{1}: M(2, *) = 4 \times M(2, *) - M(1, *)$$

$$M(3, *) = 4 \times M(3, *) - M(1, *)$$

$$M(4, *) = 4 \times M(4, *) - M(1, *)$$

$$M(1, *) = 1/4 \times M(1, *) - 1/4 \times M(2, *) - 1/4 \times M(3, *) - 1/4 \times M(4, *)$$

$$\begin{aligned} \xrightarrow{2}: M(3, *) &= 3 \times M(3, *) + M(2, *) \\ M(4, *) &= 3 \times M(4, *) + M(2, *) \\ M(2, *) &= 1/3 \times M(2, *) + 1/12 \times M(3, *) + 1/12 \times M(4, *) \\ \xrightarrow{3}: M(4, *) &= 2 \times M(4, *) + M(3, *) \\ M(3, *) &= 1/8 \times M(3, *) + 1/24 \times M(4, *) \\ \xrightarrow{4}: M(4, *) &= 1/12 \times M(4, *) \end{aligned}$$

Quando a matriz é muito grande e esparsa, é bem sabido que a maneira mais eficiente de se representar a matriz, para armazenamento e processamento em memória, se faz guardando somente suas entradas não-nulas. Para que um processo de pivoteamento seja eficiente, ele deve ser feito se possível de uma maneira que entradas nulas da matriz não se tornem entradas não-nulas durante o pivoteamento, pois neste caso o armazenamento eficiente desta matriz potencialmente estaria comprometido quando então o processo de pivoteamento fracassaria. Note que o pivoteamento exemplificado anteriormente não cumpre este requisito.

O problema da *eliminação Gaussiana perfeita* é encontrar, se possível, uma sequência de pivôs tal que o processo de pivoteamento, seguindo tal ordem de pivôs, nunca torne não-nula uma entrada nula da matriz.

A matriz acima admite uma eliminação Gaussiana perfeita, conforme demonstrado abaixo.

$$\begin{pmatrix} 4 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & \boxed{1} \end{pmatrix} \xrightarrow{1} \begin{pmatrix} 3 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & \boxed{1} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \xrightarrow{2} \begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & \boxed{1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \xrightarrow{3} \mathbf{I}$$

Combinações lineares efetuadas:

$$\begin{aligned} \xrightarrow{1}: M(1, *) &= M(1, *) - M(4, *) \\ M(4, *) &= M(4, *) - M(1, *) + M(2, *) + M(3, *) \end{aligned}$$

$$\begin{aligned} \xrightarrow{2}: M(1, *) &= M(1, *) - M(3, *) \\ M(3, *) &= M(3, *) - M(1, *) + M(2, *) \end{aligned}$$

$$\xrightarrow{3}: M(1, *) = M(1, *) - M(2, *)$$

Note que este pivoteamento utilizou menos combinações lineares. Isto é apenas uma coincidência. O objetivo procurado, e alcançado no exemplo acima, é não introduzir entradas não-nulas onde originalmente são entradas nulas.

### 3.3.2 Eliminação Gaussiana perfeita e os grafos bipartidos cordais

Em primeiro lugar, investiguemos a causa para que uma entrada não-nula se torne nula, durante uma operação de pivoteamento.

Suponha que, em determinado momento, um pivô  $M(i, j)$  seja escolhido. As linhas que deverão ser substituídas são as linhas de índice  $r \neq i$ ,  $1 \leq r \leq n$  tais que  $M(r, j) \neq 0$ . Uma entrada em  $M(r, *)$  nula se torna não-nula na coluna  $s$  precisamente quando  $M(r, s) = 0$  e  $M(i, s) \neq 0$ . Em outras palavras, para evitar tornar não-nula uma entrada nula, o pivô escolhido  $M(i, j) \neq 0$  deve atender o seguinte requisito: para todo  $r \neq i$ ,  $1 \leq r \leq n$ , tal que  $M(r, j) \neq 0$ ,  $s \neq j$ ,  $1 \leq s \leq n$  tal que  $M(i, s) \neq 0$ ,  $M(r, s)$  deve ser não-nulo.

Modelemos este problema sob o ponto de vista de teoria de grafos.

O grafo bipartido  $G(M) = (X, Y, E)$  é o *grafo bipartido de uma matriz não-nula quadrada*  $M$   $n \times n$  se  $X = \{x_1, \dots, x_n\}$ ,  $Y = \{y_1, \dots, y_n\}$  e  $x_i y_j \in E$  se e somente se  $M(i, j) \neq 0$ .

É fácil ver que o requisito estabelecido para não se criar entradas não-nulas onde havia entradas nulas, adaptada para a nova estrutura  $G(M)$ , é assim enunciado: um pivô  $x_i y_j \in E$  evita o efeito de tornar não-nulas entradas nulas se e somente se para todo  $x_r \in X$  tal que  $x_r y_j \in E$ ,  $y_s \in Y$  tal que  $x_i y_s \in E$ , ocorra de  $x_r y_s \in E$ . Equivalentemente, um pivô  $x_i y_j \in E$  evita o efeito de tornar não-nulas entradas nulas se e somente se  $x_i y_j \in E$  é uma *aresta bi-simplicial* de  $G(M)$ .

Logo, a escolha de pivôs em uma eliminação Gaussiana perfeita de uma matriz  $M$  é um esquema de eliminação perfeita de  $G(M)$ . De fato, a matriz admite uma eliminação Gaussiana perfeita se e somente se  $G(M)$  for um grafo bipartido de eliminação perfeita [12]. Conforme a Seção 2.4, a classe dos grafos bipartidos de eliminação perfeita inclui propriamente aquela dos grafos bipartidos cordais.

Para se encontrar uma eliminação Gaussiana perfeita de uma matriz  $M$ , portanto, a estratégia é seguir um esquema de eliminação perfeita de  $G(M)$ . No entanto, existe um inconveniente. Nem todo esquema de eliminação perfeita pode ser seguido até o fim, devido a uma dificuldade técnica descrita a seguir.

Um *zero aritmético* no processo de eliminação Gaussiana é uma entrada nula criada como resultado de uma coincidência aritmética (operações aritméticas sendo

efetuadas durante o pivoteamento). Zeros na matriz original são chamados de *originais*. Um problema crucial no pivoteamento de uma matriz  $M$  seguindo um esquema de eliminação perfeita de  $G(M)$  ocorre se um zero aritmético aparecer durante o processo numa posição que, segundo o esquema de eliminação perfeita, será o próximo pivô a ser considerado. Neste caso, tal esquema não poderá ser utilizado. Encontrar um esquema de eliminação perfeita que não produza zeros aritméticos pode ser uma tarefa não-trivial.

Um grafo bipartido  $G = (X, Y, E)$  é chamado de grafo bipartido de *redução completa de arestas* se as arestas em  $E$  puderem ser ordenadas na sequência  $(x_1y_1, \dots, x_my_m)$ , onde  $x_iy_i \neq x_jy_j$  para  $i \neq j$  e cada  $x_ky_k$  é bi-simplicial em  $G[\{x_ky_k, \dots, x_my_m\}]$ . Note que a classe dos grafos bipartidos de redução completa de arestas não é equivalente à classe dos grafos bipartidos de eliminação perfeita, pois num esquema de eliminação perfeita não estão presentes necessariamente todas as arestas do grafo.

O teorema a seguir relaciona os grafos bipartidos cordais aos grafos de redução completa de arestas.

**Teorema 3.32** (Bakonyi e Bono [3]). *Um grafo bipartido é bipartido cordal se e somente se ele é um grafo de redução completa de arestas. Além disso, se um grafo bipartido  $G = (X, Y, E)$  é um grafo de redução completa de arestas e  $xy$  é uma aresta bi-simplicial em  $G$ , então  $G' = (X, Y, E - \{xy\})$  também é um grafo de redução completa de arestas.*

*Demonstração.* Consequência direta da Proposição 3.18. □

Considere agora uma matriz  $M$  tal que seu grafo bipartido associado  $G(M)$  seja bipartido cordal. Considere novamente um esquema de eliminação perfeita  $[x_1y_1, \dots, x_ky_k]$  de  $G[M]$ . Suponha que, para algum  $1 \leq i \leq k$ , ocorra um zero aritmético justamente na próxima posição de pivoteamento  $x_iy_i$ .

Como ser um grafo bipartido cordal é uma propriedade hereditária, então  $G[\{x_iy_i, \dots, x_ky_k\}]$  é um grafo bipartido cordal. Como  $x_iy_i$  é uma aresta bi-simplicial de  $G[\{x_iy_i, \dots, x_ky_k\}]$ , de acordo com o Teorema 3.32,  $G[\{x_{i+1}y_{i+1}, \dots, x_ky_k\}]$  é um grafo bipartido cordal. Portanto, podemos encontrar um outro esquema de eliminação perfeita para  $G[\{x_{i+1}y_{i+1}, \dots, x_ky_k\}]$ , e prosseguir o pivoteamento com

a nova primeira aresta bi-simplicial deste novo esquema. Logo, se  $M$  é uma matriz com grafo bipartido cordal associado, então dado um esquema de eliminação perfeita, é possível utilizá-lo para gerar uma eliminação Gaussiana perfeita pois as coincidências aritméticas podem ser contornadas.

# Capítulo 4

## Reconhecimento dos Grafos

### Bipartidos Cordais

No capítulo anterior apresentamos uma coleção de propriedades e caracterizações relacionadas aos grafos bipartidos cordais. Este capítulo destina-se a descrever os algoritmos de reconhecimento de grafos bipartidos cordais. Como veremos adiante, o reconhecimento dos grafos bipartidos cordais é fortemente baseado nas ordenações lexicográficas duplas de matrizes e reconhecimento de matrizes livres de  $\Gamma$ , vistos no capítulo anterior.

Nas Seções 4.1 e 4.2, mostramos de forma mais detalhada a abordagem mais eficiente conhecida para decidir se um grafo é bipartido cordal.

#### 4.1 Reconhecimento de grafos bipartidos cordais

Nesta seção trataremos de apresentar os algoritmos de reconhecimento dos grafos bipartidos cordais.

Como um grafo é bipartido cordal se é bipartido (tempo linear de reconhecimento, conforme Observação 4.1) e se não possui ciclos induzidos de tamanhos maiores ou iguais a 5, decorre que a complexidade de tempo do problema de reconhecimento dos grafos bipartidos cordais é, no pior caso, a mesma de se reconhecer ciclos induzidos de tamanhos maiores ou iguais a 5. O problema de decidir se um grafo possui ciclos induzidos de tamanho ao menos  $k$ , para algum valor fixo  $k \geq 4$ , possui complexidade de tempo  $O(n^k)$  (Observação 4.2). Spinrad [22] forneceu uma solução



melhorada levando tempo  $O(n^{k-3}M)$ , onde  $M = O(n^{2.376})$  é o tempo requerido para multiplicar duas matrizes  $n \times n$ . Recentemente, Nikolopoulos e Palios [20] forneceram um algoritmo de tempo  $O(n^4)$  para reconhecer ciclos induzidos de tamanho ao menos 5. Portanto, utilizando-se tais resultados, podemos estabelecer que a complexidade de tempo do problema de reconhecimento dos grafos bipartidos cordais é  $O(n^4)$ . Levando-se em conta que tais grafos são bipartidos, é possível complexidades melhores conforme veremos nesta seção. É interessante observar que o problema de determinar em particular se um grafo contém um ciclo induzido de tamanho no mínimo 4 pode ser resolvido em tempo  $O(n + m)$  (problema bem conhecido de se reconhecer um grafo cordal).

**Observação 4.1.** É possível decidir se um grafo  $G$  é bipartido em tempo  $O(n + m)$ .

*Demonstração.* Seja  $G$  um grafo. Sejam  $A$  e  $B$  dois conjuntos de vértices, inicialmente vazios. Percorre-se a lista de adjacências fazendo-se o seguinte processo. O conjunto  $A$  é inicializado com um vértice arbitrário  $v \in V(G)$ . Em seguida, o outro conjunto recebe  $N(v)$  e  $v$  é marcado como visitado. O algoritmo prossegue visitando cada  $w \in N(v)$  ainda não visitado, sempre colocando  $N(w)$  no outro conjunto em relação a qual  $w$  está. Claramente  $G$  é bipartido se e somente se o algoritmo produzir uma bipartição de  $V(G)$ . Outra forma de enunciar este processo é dizer que podemos fazer uma busca em largura a partir de um vértice arbitrário  $v \in V(G)$  verificando se as arestas de retorno saem e entram em níveis da árvore de busca cuja diferença seja ímpar.  $\square$

**Observação 4.2.** É possível decidir se um grafo  $G$  possui um ciclo induzido de tamanho maior ou igual a  $k$  em tempo  $O(n^k)$ .

*Demonstração.* Seja  $P$  um caminho induzido de tamanho  $k - 3$ , ou seja, com  $k - 2$  vértices. Sejam  $x, y$  os vértices extremos de  $P$ . Seja  $G'$  o grafo construído a partir de  $G$  removendo-se os seguintes vértices: (i) vértices internos ao caminho  $P$  e os vizinhos de tais vértices excetuando-se  $x$  e  $y$ ; e (ii) vértices que são vizinhos a ambos os vértices  $x$  e  $y$ .

Se  $P$  estiver em um ciclo induzido de tamanho maior ou igual a  $k$  em  $G$ , então claramente haverá um caminho entre  $x$  e  $y$  em  $G'$ . Por outro lado, se existe um caminho entre  $x$  e  $y$  em  $G'$ , então, pela remoção dos vértices do tipo (ii) e pelo fato

de que  $xy \notin E(G')$ , existe um caminho induzido entre  $x$  e  $y$  em  $G'$  de tamanho maior ou igual 3. Logo,  $P$  faz parte de um ciclo induzido de tamanho maior ou igual a  $k$  em  $G$ .

Portanto,  $P$  está em um ciclo induzido de tamanho maior ou igual a  $k$  em  $G$  se e somente se existir um caminho entre  $x$  e  $y$  em  $G'$ . A construção de  $G'$  e a verificação de um caminho entre  $x$  e  $y$  em  $G'$  pode ser feita em tempo  $O(n + m)$ . Como existem no máximo  $O(n^{k-2})$  caminhos de tamanho  $k - 3$  e leva-se tempo  $O(n + m)$  para verificar se cada caminho está em um ciclo induzido de tamanho maior ou igual a  $k$ , então o tempo total para a verificação da existência de um tal ciclo é de  $O(n^{k-1} + mn^{k-2}) = O(n^k)$ .  $\square$

Embora o reconhecimento dos grafos bipartidos cordais seja portanto inerentemente polinomial, é possível realizar esta tarefa com uma complexidade de tempo bem melhor. Apresentamos os algoritmos com a menor complexidade conhecida para este problema. Estes algoritmos estão baseados na seguinte idéia.

Pelo Teorema 3.30, um grafo  $G$  é bipartido cordal se e somente se existir uma ordenação livre de  $\Gamma$  da sua matriz bipartida. Por outro lado, pelo Teorema 2.25, se um grafo  $G$  é bipartido cordal então toda ordenação lexicográfica dupla de sua matriz bipartida é livre de  $\Gamma$ . Combinando estes resultados, podemos reconhecer se um grafo  $G$  é bipartido cordal fazendo uma ordenação lexicográfica dupla de sua matriz bipartida e  $G$  será bipartido cordal se e somente se a ordenação lexicográfica dupla obtida for livre de  $\Gamma$ .

Portanto, existem duas fases distintas no reconhecimento de grafos bipartidos cordais utilizando-se esta abordagem: produzir uma ordenação lexicográfica dupla de uma matriz bipartida  $M$  de um grafo  $G$  e decidir se  $M$  é livre de  $\Gamma$ . Na Seção 2.1, descrevemos os algoritmos mais eficientes conhecidos para se produzir uma ordenação lexicográfica dupla de uma matriz. Neste capítulo, portanto, descreveremos como decidir se uma matriz  $M$  é livre de  $\Gamma$ .

Note que a tarefa de decidir se uma matriz  $M$  é livre de  $\Gamma$  é trivialmente polinomial. De fato, considere o seguinte algoritmo.

### Reconhecimento de Matrizes Livres de $\Gamma$

**Entrada:** matriz  $M$   $I \times J$

**Saída:** ‘Sim’ se  $M$  é livre de  $\Gamma$ , ‘Não’ caso contrário.

(i) Para cada  $1 \leq i \leq I$ ,  $1 \leq j \leq J$  tais que  $M(i, j) = 1$  faça

(a) Para cada  $j < w \leq J$  tal que  $M(i, w) = 1$  faça

i. Para cada  $i < z \leq I$  tal que  $M(z, j) = 1$  faça

Se  $M(z, w) = 0$  então pare e devolva ‘Não’

(ii) Devolva ‘Sim’.

O laço (i) é iterado  $IJ$  vezes. Para cada valor 1 encontrado, o laço (a) é iterado  $O(J)$  e o laço i. é iterado  $O(I)$  vezes. Logo, a complexidade total de tempo é de  $O(I^2J^2)$ . No problema em questão, reconhecimento de grafos bipartidos cordais,  $M$  é a matriz bipartida do grafo  $G$  o qual se deseja saber se é bipartido cordal ou não. Portanto  $I = O(n)$  e  $J = O(n)$ . Logo, apenas a fase de reconhecer se a matriz resultante de uma ordenação lexicográfica dupla é livre de  $\Gamma$  levaria tempo  $O(n^4)$ , não apresentando nenhuma melhora ao tempo máximo já esperado conforme mencionado na introdução desta seção. Mostraremos que é possível testar se uma matriz bipartida de um grafo  $G$  é livre de  $\Gamma$  em tempo linear  $O(m + n)$ .

Observe que tal matriz tem tamanho  $O(mn)$  mas esse algoritmo evitará a construção explícita dessa matriz usando um interessante truque algorítmico.

Desta forma, a complexidade de tempo para se reconhecer se um grafo é bipartido cordal é dada pela complexidade de tempo dos diversos métodos de se obter uma ordenação lexicográfica dupla da matriz bipartida deste grafo, sendo os mais eficientes apresentados na Seção 2.1.

O algoritmo a seguir resume o procedimento de reconhecimento dos grafos bipartidos cordais.

### Algoritmo de reconhecimento dos grafos bipartidos cordais

**Entrada:** grafo  $G$  (lista de adjacências)

**Saída:** ‘Sim’ se  $G$  é bipartido cordal, ‘Não’ caso contrário.

(i) Se  $G$  não é bipartido, devolva ‘Não’.

- (ii) Constrói-se a representação adequada da matriz bipartida  $M$  de  $G$ .
- (iii) Obtém-se uma ordenação lexicográfica dupla  $M'$  de  $M$  (utilizando um algoritmo da Seção 2.1).
- (iv) Se  $M'$  for livre de  $\Gamma$  (utilizando o algoritmo da Seção 4.2), devolva ‘Sim’; caso contrário, devolva ‘Não’.

**Consumo de tempo:**  $O(n^2)$  usando o algoritmo de Spinrad e a ‘primeira’ parte da próxima seção.

## 4.2 Reconhecimento de matrizes livres de $\Gamma$

Nesta seção, apresentaremos como reconhecer a ausência de  $\Gamma$  em uma matriz bipartida de um grafo em tempo ótimo  $O(m + n)$ . Para tanto, será necessário fazer com que seja possível testar a vizinhança entre um par de vértices em tempo  $O(1)$ , mesmo que a entrada seja a lista de adjacências do grafo (caso contrário, se tivermos como entrada a matriz de adjacência deste grafo, o tempo seria  $O(n^2)$ ). Isto é possível assumindo-se um modelo de computação no qual *a simples alocação de espaço para uma matriz ou vetor, sem sua inicialização, leva tempo constante*. Tendo esta premissa por base, de forma engenhosa, pode-se conseguir tal feito conforme descrito a seguir.

**Lema 4.3** (Aho, Hopcroft e Ullman [1] – exerc. 2.12). *Seja  $G$  um grafo descrito sob forma de sua lista de adjacências. Assumindo-se um modelo de computação no qual a alocação de espaço para uma matriz ou vetor, sem sua inicialização, leva tempo constante, podemos construir em tempo  $O(m + n)$  uma estrutura de dados que permite testar a adjacência entre qualquer par de vértices em tempo  $O(1)$ .*

*Demonstração.* Seja  $\mathcal{L}$  a lista de adjacências de  $G$  dada como entrada. A estrutura de dados que será construída, que permitirá fazer tal teste de adjacência em tempo constante, é constituída dos seguintes elementos:

- (i) matriz  $M$   $n \times n$
- (ii) matriz  $M_{pont}$   $n \times n$

(iii) pilha  $P_{verif}$  (implementada por um vetor de  $m$  posições)

Todas estas estruturas são alocadas (mas não inicializadas) em tempo constante, de acordo com a premissa sendo assumida sobre o modelo de computação. A idéia é que  $M$  seja a matriz de adjacência de  $G$ . Gastando-se somente tempo  $O(n + m)$ , nem todas as posições de  $M$  serão inicializadas. De fato, inicializaremos somente as posições da matriz que devem ter o valor 1. As demais ficarão com o valor arbitrário após o processo de alocação (lixo de memória). As demais estruturas  $M_{pont}$  e  $P_{verif}$  serão utilizadas para distinguir quando uma entrada da matriz  $M$  é um valor inicializado pelo algoritmo (necessariamente valor 1) de quando ela representa um lixo (necessariamente interpretado como valor 0).

A inicialização destas estruturas ocorrerá da seguinte forma. Para cada vértice  $v$ , percorre-se cada vizinho  $w \in N(v)$  em  $\mathcal{L}$ , inicializando as estruturas da seguinte forma:

- (i)  $M(v, w) \leftarrow 1$
- (ii) empilhar em  $P_{verif}$  o endereço de memória de  $M_{pont}(v, w)$
- (iii)  $M_{pont}(v, w) \leftarrow$  endereço de memória da última posição do vetor  $P_{verif}$  sendo utilizada (topo da pilha)

Podemos inferir para qualquer par de vértices  $v, w$  se  $vw \in E(G)$  da seguinte forma:  $vw \in E(G)$  se e somente se todas as condições abaixo são satisfeitas:

- (i)  $M(v, w) = 1$
- (ii) endereço armazenado em  $M_{pont}(v, w)$  pertence ao vetor  $P_{verif}$
- (iii) o endereço armazenado em  $P_{verif}$  na posição apontada pelo endereço armazenado em  $M_{pont}(v, w)$  é igual ao endereço de  $M_{pont}(v, w)$

Em outras palavras, a estrutura  $P_{verif}$  serve de certificação de que a posição  $M_{pont}(v, w)$ , para todo  $v, w \in V(G)$ , foi inicializada.  $\square$

Apenas esta estrutura eficiente para testar adjacências como ferramenta ainda não é suficiente para se testar otimamente se uma matriz bipartida de um grafo possui  $\Gamma$ . Precisamos também observar que nem todas as possibilidade de formação de  $\Gamma$  necessitam de fato serem verificadas, fato que segue do próximo resultado.

**Lema 4.4** (Lubiw [18]). *Se uma matriz binária  $M$  possuir um  $\Gamma$ , então  $M$  possui um  $\Gamma$  formado por  $M(l_1, c_1) = M(l_1, c_2) = M(l_2, c_1) = 1$ ,  $M(l_2, c_2) = 0$  tal que  $M(l_1, c) = 0$  para cada  $c_1 < c < c_2$  e  $M(l, c_1) = 0$  para cada  $l_1 < l < l_2$ .*

*Demonstração.* Sejam  $l_1 < l_2$  e  $c_1 < c_2$  tais que  $M(l_1, c_1) = M(l_1, c_2) = M(l_2, c_1) = 1$  e  $M(l_2, c_2) = 0$ . Se  $M(l_1, c) = 0$  para cada  $c_1 < c < c_2$  e  $M(l, c_1) = 0$  para cada  $l_1 < l < l_2$ , segue o resultado. Logo, suponha que exista (i)  $c_1 < c_3 < c_2$  tal que  $M(l_1, c_3) = 1$ ; ou (ii)  $l_1 < l_3 < l_2$  tal que  $M(l_3, c_1) = 1$ .

...	...	$c_1$	...	...	$c_3$	...	...	$c_2$	...
...	...	:	:	:	:	:	:	:	:
$l_1$	...	<b>1</b>	0	...	<b>1</b>	...	0	<b>1</b>	...
...	...	0	:	:	:	:	:	:	:
...	...	:	:	:	:	:	:	:	:
...	...	0	:	:	:	:	:	:	:
$l_2$	...	<b>1</b>	...	...	?	...	...	<b>0</b>	...
...	...	:	:	:	:	:	:	:	:

Se o caso (i) ocorre, considere o valor  $M(l_2, c_3)$ . Se  $M(l_2, c_3) = 1$ , então existe um outro  $\Gamma$  ( $\Gamma'$ ) formado por  $M(l_1, c_3) = M(l_1, c_2) = M(l_2, c_3) = 1$ ,  $M(l_2, c_2) = 0$ . Se  $M(l_2, c_3) = 0$ , então existe um outro  $\Gamma$  ( $\Gamma'$ ) formado por  $M(l_1, c_1) = M(l_1, c_3) = M(l_2, c_1) = 1$ ,  $M(l_2, c_3) = 0$ . Em ambos os casos,  $\Gamma'$  está propriamente “contido” em  $\Gamma$ . De maneira análoga, encontramos um  $\Gamma'$  propriamente “contido” em  $\Gamma$  se o caso (ii) ocorre. Tomando-se  $\Gamma'$  por  $\Gamma$ , podemos repetir este processo, que deve ser finito pelo fato de  $M$  ser uma matriz de dimensões finitas. O último  $\Gamma'$  encontrado claramente atende as exigências do resultado. □

Como consequência do Lema 4.4, para testar se uma matriz binária  $M$  possui um  $\Gamma$  é suficiente testar se cada entrada 1 de  $M$  forma um  $\Gamma$  com o primeiro 1 à direita e o primeiro 1 abaixo. O próximo algoritmo apresenta esta estratégia de maneira detalhada.

### Reconhecimento de matrizes bipartidas livres de $\Gamma$ em tempo linear

**Entrada:** lista de adjacências  $\mathcal{L}$  tal que, para todo  $v \in V(G)$ , a lista  $\mathcal{L}(v)$  está ordenada conforme as linhas e colunas da matriz  $M$ . Além disso, para todo  $v \in V(G)$ , cada entrada  $w \in \mathcal{L}(v)$  é duplamente ligada com a respectiva entrada  $v \in \mathcal{L}(w)$ .

**Saída:** Sim/Não

- (i) Construir a matriz de adjacência  $M$  conforme o Lema 4.3
- (ii) Para cada  $i \in V(G)$  faça
  - (a) Para cada  $j \in \mathcal{L}(i)$  faça
    - i. Seja  $z \in \mathcal{L}(i)$  após a entrada  $j \in \mathcal{L}(i)$
    - ii. Seja  $i' \in \mathcal{L}(j)$  a entrada respectiva à entrada  $j \in \mathcal{L}(i)$
    - iii. Seja  $w \in \mathcal{L}(j)$  após a entrada  $i' \in \mathcal{L}(j)$
    - iv. Se  $M(z, w) = 0$  então pare e devolva 'Não'
- (iii) Devolva 'Sim'.

Seja  $n$  o número de vértices de  $G$  e  $m$  o número de arestas de  $G$ .

O passo (i) conforme visto pode ser implementado em tempo  $O(n + m)$ . Além disso, os laços (ii) e (a) percorrem toda a lista de adjacências  $\mathcal{L}$ , para cada entrada, executando as operações i. a iv. de tempo constante, portanto requerendo também tempo  $O(n + m)$ . Logo, o tempo total de execução deste algoritmo é  $O(n + m)$ .

# Capítulo 5

## Comparação entre os Grafos Cordais e Grafos Bipartidos Cordais

O objetivo deste capítulo é mostrar os casos nos quais a similaridade de estrutura entre os grafos cordais e os bipartidos cordais geram um paralelismo entre os resultados das duas classes, bem como os casos onde essa relação não ocorre.

Vimos no Teorema 3.12 que um grafo bipartido é bipartido cordal se e somente se todo separador de arestas minimal for uma biclique [Golumbic e Goss [13]]. Para os grafos cordais temos o seguinte resultado:

**Teorema 5.1** (cf. Dirac [7]). *Um grafo  $G = (V, E)$  é cordal se e somente se todo separador de vértices minimal é uma clique.*

Ainda em relação aos separadores, vimos no Teorema 3.9 que, dado um grafo bipartido cordal separável  $G = (X, Y, E)$  e sendo  $S$  um separador minimal de suas arestas  $x_a y_a$  e  $x_b y_b$ , então existe um vértice  $x'_a$  na componente conexa de  $G[X \cup Y - S]$  que contém  $x_a y_a$  tal que  $S \cap Y \subset N(x'_a)$ . Veja que existe um resultado semelhante para os grafos cordais.

**Teorema 5.2** (Golumbic [12] – exerc. 12, cap. 4). *Para todo separador de vértices minimal  $S$  de um grafo cordal  $G = (V, E)$ , existe um vértice  $v$  em cada componente conexa de  $G[V - S]$  tal que  $S \subseteq N(v)$ .*



De acordo com o Teorema 3.8, sendo  $G = (X, Y, E)$  um grafo bipartido cordal separável, então  $G$  tem no mínimo duas arestas bi-simpliciais separáveis. Como todo grafo bipartido não-separável é um grafo bipartido cordal (Lema 3.4), e por consequência um grafo bipartido de eliminação perfeita (Teorema 3.15), então todo grafo bipartido cordal tem no mínimo uma aresta bi-simplicial, ou duas, caso seja separável. Um resultado semelhante para grafos cordais encontra-se a seguir.

**Teorema 5.3** (cf. Dirac [7]). *Todo grafo cordal  $G$  tem um vértice simplicial. Se  $G$  não é um grafo completo, então existem pelo menos dois vértices simpliciais não-adjacentes.*

Nem todos os paralelismos com os grafos cordais ocorrem diretamente com a classe dos grafos bipartidos cordais. Na verdade, alguns ocorrem com a classe dos grafos bipartidos de eliminação perfeita que é uma superclasse da classe dos grafos bipartidos cordais. Por exemplo, pela própria definição dos grafos bipartidos de eliminação perfeita, um grafo bipartido  $G = (X, Y, E)$  é bipartido de eliminação perfeita se e somente se  $G$  tem um esquema de eliminação perfeita de arestas bi-simpliciais. Para os grafos cordais temos o seguinte resultado:

**Teorema 5.4** (cf. Fulkerson e Gross [10]). *Um grafo  $G$  é cordal se e somente se  $G$  tem um esquema de eliminação perfeita de vértices simpliciais.*

Seja  $G$  um grafo tal que  $V(G) = \{v_1, v_2, \dots, v_n\}$ . O grafo bipartido  $B(G) = (X, Y, E)$  é tal que  $X = \{x_1, x_2, \dots, x_n\}$ ,  $Y = \{y_1, y_2, \dots, y_n\}$ , e  $x_i y_j \in E$  se e somente se  $i = j$  ou  $v_i v_j \in E(G)$ . Para  $S \subset V(G)$ , seja  $B(S) = \{x_i \mid v_i \in S\} \cup \{y_i \mid v_i \in S\}$ . O próximo resultado mostra uma relação entre os grafos cordais e os grafos bipartidos cordais, e o seguinte entre seus separadores minimais.

**Teorema 5.5** (Golumbic [12] – exerc. 2, cap. 12). *Se  $B(G)$  é um grafo bipartido cordal, então  $G$  é um grafo cordal.*

*Demonstração.* Suponha que  $B(G)$  é bipartido cordal e que  $G$  possua um ciclo induzido pelos vértices  $(v_1, v_2, \dots, v_m)$ ,  $m \geq 4$ . Note que em  $B(G)$  temos o grafo  $B_c$  definido pela Figura 5.1 como um subgrafo induzido.

Como  $B(G)$  é um grafo bipartido cordal, pelo Teorema 3.15,  $B(G)$  é também um grafo bipartido de eliminação perfeita. Seja, portanto,  $\sigma = [e_1, \dots, e_k]$  um esquema

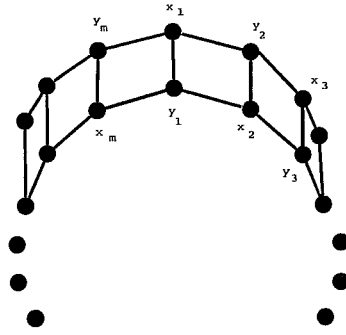
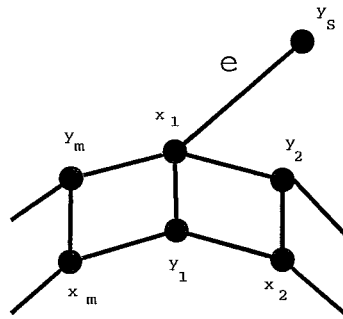
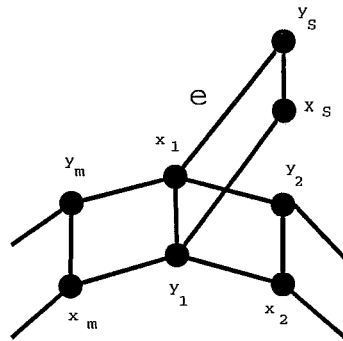


Figura 5.1: Subgrafo  $B_c$  de  $B(G)$ .

de  $B(G)$ . Seja  $e$  a primeira aresta bi-simplicial na sequência de arestas definida por  $\sigma$  que envolve um vértice de  $B_c$ . Claramente  $e$  possui exatamente um vértice de  $B_c$ , pois nenhuma aresta de  $B_c$  é bi-simplicial. Sem perda de generalidade,  $e = x_1y_s$ ,  $s \notin \{1, \dots, m\}$ , e a situação é conforme mostrado a seguir.



Por construção de  $B(G)$ , existe  $x_s$  tal que  $x_sy_s \in E(B(G))$ . Como  $x_1y_s \in E(B(G))$ , então  $x_sy_1 \in E(B(G))$ . A situação revisada é conforme mostrado na figura a seguir.



Pela bi-simplicialidade de  $e$ , concluímos que  $x_sy_2 \in E(B(G))$  e que, portanto,  $x_2y_s \in E(B(G))$ . Novamente pelo fato de  $e$  ser uma aresta bi-simplicial,  $x_2y_m \in$

$E(B(G))$ , o que contradiz  $B_c$  ser um subgrafo induzido . □

**Observação 5.6.** Se  $G$  é um grafo cordal,  $B(G)$  não é necessariamente um grafo bipartido cordal. A figura a seguir exemplifica tal caso.

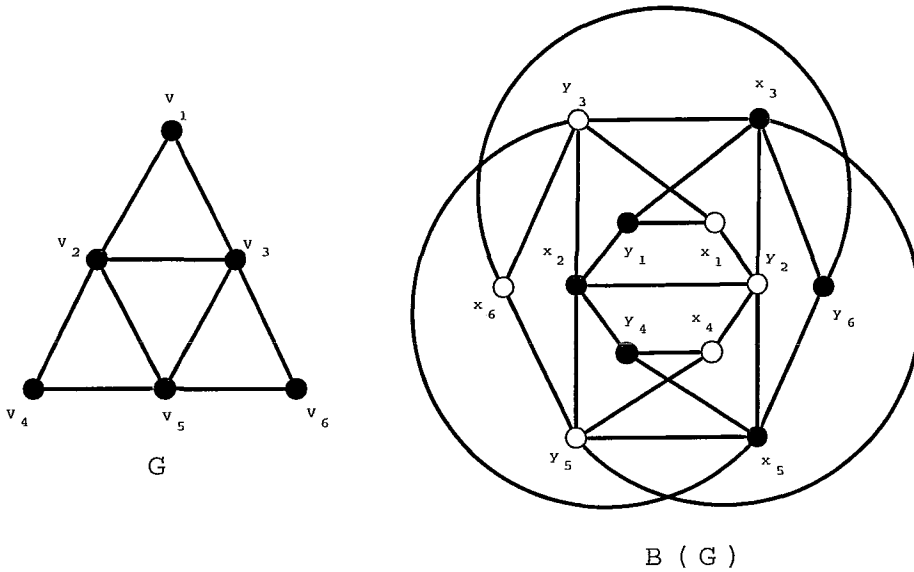


Figura 5.2: Exemplo de um grafo cordal  $G$  cujo grafo  $B(G)$  não é bipartido cordal.

Note que existe um  $C_6 = (x_1, y_2, x_4, y_5, x_6, y_3)$  induzido em  $B(G)$ . Além disso, existe um separador de arestas minimal  $S = \{x_2, y_2, y_6, y_5, y_3\}$  que não forma uma biclique em  $B(G)$ , o que seria necessário pelo Lema 3.7 caso  $B(G)$  fosse bipartido cordal. De fato,  $S$  possui as arestas  $x_2y_2, x_2y_3, x_2y_5$  mas não possui a aresta  $x_2y_6$ .

A exemplo do que também ocorre para os grafos cordais e, portanto, reforçando o paralelismo que ocorre entre as classes, os algoritmos de reconhecimento dos grafos bipartidos cordais são divididos em duas fases. No caso dos grafos bipartidos cordais, a primeira delas é construtiva produzindo uma ordenação lexicográfica dupla da matriz bipartida do grafo. A fase de verificação checa se a ordenação produzida é de fato livre de  $\Gamma$  (conforme Capítulo 4).

# Capítulo 6

## Conclusão

Um grafo  $G$  é *bipartido cordal* se  $G$  é bipartido e todo ciclo de tamanho maior ou igual a 6 tem uma corda. Em outras palavras, dado que um grafo bipartido só possui ciclos induzidos de tamanho par, um grafo é bipartido cordal quando todos os seus ciclos induzidos têm tamanho 4.

Os grafos bipartidos cordais foram inicialmente introduzidos em 1978 por Golumbic e Goss [13]. Seu estudo foi motivado originalmente por sua aplicação em matrizes não-simétricas. As aplicações em eliminação Gaussiana de matrizes esparsas são descritas por Bakonyi e Bono em 1997 [3], e em programação inteira por Hoffman, Kolen e Sakarovitch em 1985 [14] em análise de matrizes por Johnson e Whitney em 1991 [16] e por Johnson e Miller em 1997 [15]. Exemplos de aplicações foram descritos na Seção 3.3.

Como vimos, o nome bipartido cordal foi motivado pelo fato de serem os grafos bipartidos que possuem uma característica fundamental dos grafos cordais: um grafo bipartido  $G$  é bipartido cordal se todo ciclo induzido de  $G$  possui o menor tamanho possível. Muitos resultados semelhantes sobre os grafos cordais e os grafos bipartidos cordais, assim como diferenças, foram relacionados no Capítulo 5, muito embora tais classes de grafos sejam bem distintas. Como foi notado, os grafos bipartidos cordais não são cordais em geral e vice-versa.

Como os grafos cordais, os grafos bipartidos cordais são ricos em estrutura e muitos resultados interessantes originaram-se do estudo de diversos autores. Podemos dizer que sua importância está para a classe dos grafos bipartidos assim como a importância dos grafos cordais está para a classe dos grafos em geral.

O objetivo deste trabalho foi compilar um estudo abrangente da classe dos grafos bipartidos cordais, consolidando as caracterizações e algoritmos de reconhecimento mais eficientes existentes para esta classe de grafos. Para tanto, foi necessário conhecer a resolução de problemas relacionados interessantes, como a obtenção de ordenações lexicográficas duplas de matrizes.

# Referências Bibliográficas

- [1] AHO, A. V., HOPCROFT, J. E., ULLMAN, J. D. *The Design and Analysis of Computer Algorithms*. Addison–Wesley, 1974.
- [2] ANSTEE, R. P., FARBER, M. Characterizations of totally balanced matrices, *J. Algorithms* v. 5, n. 2, pp. 215–230, 1984.
- [3] BAKONYI, M., BONO, A. Several results on chordal bipartite graphs, *Czechoslovak Math. J.* v. 47, n. 4, pp. 577–583, 1997.
- [4] BONDY, J. A., MURTY, U. S. R. *Graph Theory with Applications*. North-Holland, 1976.
- [5] BRANDSTÄDT, A., LE, V. B., SPINRAD, J. P. *Graph Classes: a Survey*. SIAM, 1999.
- [6] CHVÁTAL, V. Which claw-free graphs are perfectly orderable?, *Discrete Appl. Math.* v. 44, n. 1–3, pp. 39–63, 1993.
- [7] DIRAC, G. A. On rigid circuit graphs, *Abh. Math. Sem. Univ. Hamburg*, pp. 71–76, 1961.
- [8] DRAGAN, F. F., VOLOSHIN, V. I. Incidence graphs of biacyclic hypergraphs, *Discrete Appl. Math.* v. 68, n. 3, pp. 259–266, 1996.
- [9] ESCHEN, E. M., SPINRAD, J. P. An  $O(n^2)$  algorithm for circular-arc graph recognition. In: *SODA'93: Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 128–137, Philadelphia, PA, USA, SIAM, 1993.
- [10] FULKERSON, D. R., GROSS, O. A. Incidence matrices and interval graphs, *Pacific J. Math.* v. 15, n. 3, pp. 835–855, 1965.

- [11] GOLUBIC, M. C. A generalization of Dirac's theorem on triangulated graphs, *Ann. New York Acad. Sci.*, pp. 242–246, 1979.
- [12] GOLUBIC, M. C. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, 1980.
- [13] GOLUBIC, M. C., GOSS, C. F. Perfect elimination and chordal bipartite graphs, *J. Graph Theory* v. 2, n. 2, pp. 155–163, 1978.
- [14] HOFFMAN, A. J., KOLEN, A. W. J., SAKAROVITCH, M. Totally balanced and greedy matrices, *SIAM J. Alg. Disc. Meth.*, pp. 721–730, 1985.
- [15] JOHNSON, C. R., MILLER, J. Rank decomposition under combinatorial constraints, *Linear Algebra Appl.*, pp. 97–104, 1997.
- [16] JOHNSON, C. R., WHITNEY, G. T. Minimum rank completions, *Linear and Multilinear Algebra*, pp. 271–273, 1991.
- [17] LEHEL, J. A characterization of totally balanced hypergraphs, *Discr. Math.* v. 57, n. 1–2, pp. 59–65, 1985.
- [18] LUBIW, A. Doubly lexical orderings of matrices, *SIAM J. Comput.* v. 16, n. 5, pp. 854–879, 1987.
- [19] MA, T.-H., SPINRAD, J. An  $O(n^2)$  time algorithm for the 2-chain cover problem and related problems. In: *SODA'91: Proceedings of the Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 363–372, Philadelphia, PA, USA, SIAM, 1991.
- [20] NIKOLOPOULOS, S. D., PALIOS, L. Hole and antihole detection in graphs. In: *SODA'04: Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 850–859, Philadelphia, PA, USA, SIAM, 2004.
- [21] PAIGE, R., TARJAN, R. E. Three partition refinement algorithms, *SIAM J. Comput.* v. 16, n. 6, pp. 973–989, 1987.
- [22] SPINRAD, J. P. Finding large holes, *Inf. Process. Lett.* v. 39, n. 4, pp. 227–229, 1991.

- [23] SPINRAD, J. P. Doubly lexical ordering of dense 0-1 matrices, *Inf. Process. Lett.*, pp. 229–235, 1993.
- [24] SPINRAD, J. P. *Efficient Graph Representations*. vol. 19. Fields Institute Monographs. AMS, 2003.