

**GRAFOS TRIANGULARIZADOS:**

**CARACTERIZAÇÕES, RESTRIÇÕES E EXTENSÕES**

Mónica Villanueva Ilufi

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA 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.

Aprovada por:



Prof. Jayme Szwarcfiter, Ph. D.  
(PRESIDENTE)



Prof. Nelson Maculan, D. Sc.



Prof. Lillian Markenzon, D. Sc.



Prof. Celso Carneiro, Dr. Ing.

RIO DE JANEIRO, RJ - BRASIL

MAIO DE 1989

VILLANUEVA, MONICA

Grafos Triangularizados: Caracterizações,  
Restrições e Extensões [Rio de Janeiro]  
1989

X, 139 p. 29,7 cm (COPPE/UFRJ, M. Sc.,  
Engenharia de Sistemas e Computação, 1989)  
Tese - Universidade Federal do Rio de  
Janeiro, COPPE

1. Grafos I. COPPE/UFRJ II. Título (série)

of Mass

## AGRADECIMENTOS

A COPPE e ao povo brasileiro pela possibilidade de realizar a pós-graduação e conhecer uma experiência diferente de vida.

A Jayme pela motivação, confiança e orientação geral.

A Nelson por seu apoio constante e amigo.

A Celso e Lilian, participantes da banca examinadora.

Aos colegas pela amizade, companheirismo e experiências compartilhadas.

A Célia, Celina, Cláudia e Cristina por traduzir meu "portunhol".

Aos colegas do Departamento de Matemática e Estatística, Universidad de la Frontera, por substituir-me no trabalho.

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

GRAFOS TRIANGULARIZADOS:  
CARACTERIZAÇÕES, RESTRIÇÕES E EXTENSÕES

Mónica Villanueva Ilufi

Maio de 1989

Orientador: Jayme Luiz Szwarcfiter

Programa: Engenharia de Sistemas e Computação

Um grafo é perfeito se o grafo e cada um de seus subgrafos induzidos satisfazem a propriedade que seu número cromático é igual ao tamanho da maior clique. Os grafos triangularizados foram uns dos primeiros a serem reconhecidos como perfeitos.

A classe dos grafos triangularizados se diferencia da dos grafos perfeitos em vários aspectos. Por exemplo, é possível fazer o reconhecimento de um grafo triangularizado em tempo polinomial e não se sabe se o mesmo pode ser realizado para os grafos perfeitos. Outra diferença importante entre estas classes de grafos diz respeito aos problemas de determinar o número cromático, o tamanho da maior clique, o tamanho do maior conjunto estável e uma cobertura mínima por cliques, para um grafo dado.

Grotschel, Lovász e Schrijver desenvolveram algoritmos polinomiais para estes quatro problemas, quando restringidos à classe dos grafos perfeitos. Esses algoritmos são baseados no método do elipsóide mas não são explorados os aspectos combinatorios dos problemas. No caso dos grafos

triangularizados, diferentes autores têm desenvolvido algoritmos polinomiais para estes quatro problemas, baseados nas características particulares da classe.

Neste trabalho, são sumarizadas diferentes caracterizações para os grafos triangularizados. São expostos os algoritmos sequenciais e paralelos com tempo polinomial, encontrados na literatura, para seu reconhecimento, para sua representação como grafo interseção e para os quatro problemas clássicos de otimização em grafos.

São mostradas várias subclasses dos grafos triangularizados e algumas classes de grafos que os contêm.

Descrevem-se aplicações deles à solução de sistemas lineares, à minimização de uma função duas vezes diferenciável e a um problema de "scheduling".

Finalmente, mostram-se várias classes de grafos interseção de caminhos de uma árvore, apresentando um enfoque unificado destas classes, algumas das quais são subclasse dos grafos perfeitos.

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

TRIANGULATED GRAPHS:  
CHARACTERIZATIONS, RESTRICTIONS AND EXTENSIONS

Mónica Villanueva Ilufi

May, 1989

Thesis Supervisor: Jayme Luiz Szwarcfiter

Department: Systems Engineering and Computer Science

A graph is perfect if the graph and each of its induced subgraphs have the property that the chromatic number equals the size of maximum clique. Triangulated graphs was one of the first classes of graphs to be recognised as perfect.

The class of triangulated graphs differs from the class of perfect graphs in the fact that a triangulated graph can be recognised in polynomial time and it is unknown the NP-P status of perfection.

Another important difference between the classes is related with the algorithms to solve the problems of chromatic number, clique number, independent set and partition into cliques. Grotschel, Lovász e Schrijver developed polynomial time algorithms for these problems in a perfect graph, using the ellipsoid method. In the case of triangulated graphs, many authors have solved these problems by polynomial time algorithms which exhibit the particular structure of this class.

In this work, we survey some different

characterizations of triangulated graphs. It is given sequential and parallel polynomial time algorithms for recognizing them, for finding an intersection graph representation and to solve the four classical optimization problems.

We give a glimpse of some subclasses of triangulated graphs and some classes of graphs which contain them.

It is shown an application of triangulated graphs to solve linear systems problems, to large sparse numerical optimization and to a problem of scheduling.

Finally, we also survey the study of several classes of intersection graphs arising from families of paths in a tree, some of which are perfects.

## INDICE

I. INTRODUÇÃO .....	1
II. CONCEITOS GERAIS .....	6
II.1. Definições Básicas e Notação .....	6
II.2. Grafos Perfeitamente Ordenáveis .....	16
II.3. Problema da Busca em Grafos .....	21
II.4. Grafos de Comparabilidade .....	24
II.4.1. Grafos Bipartite .....	26
II.4.2. Grafos de Permutação .....	27
II.4.3. Reconhecimento de Grafos de Comparabilidade .....	28
II.5. Grafos de Interseção .....	28
II.5.1. Grafos de Intervalo .....	29
II.5.2. Grafos Circulares .....	30
II.5.3. Grafos Arco-Circulares .....	31
II.5.4. Grafos Triangularizados .....	32
II.5.5. Grafos Interseção de Caminhos .....	32
III. GRAFOS TRIANGULARIZADOS .....	33
III.1. Definições .....	33
III.2. Caracterização por Vértices Simpliciais ....	35
III.3. Grafo Triangularizado é um Grafo Interseção	38
III.4. Grafo Triangularizado é Perfeito .....	44
III.5. Grafo Triangularizado é Perfeitamente Ordenável .....	46
III.6. Reconhecimento de Grafos Triangularizados ..	48
III.6.1. Algoritmo Sequencial de Reconhecimento .....	48
III.6.2. Algoritmos Paralelos de Reconhecimento .....	52
III.7. Problemas em Grafos Triangularizados .....	62
III.7.1. Clique .....	63
III.7.2. Número Cromático .....	63
III.7.3. Conjunto Estável .....	64
III.7.4. Cobertura por Cliques .....	65
III.7.5. Número de Intervalo .....	69
III.7.6. Problemas sem Vantagens .....	70

III.8. Aplicações dos Grafos Triangularizados .....	72
III.8.1. Método de Eliminação de Gauss .....	72
III.8.2. Minimização de uma Função com Matriz Hessiana Esparsa .....	74
III.8.3. Schedulling com Tempo de Processamento Unitário .....	75
III.9. Subclasses dos Grafos Triangularizados .....	76
III.9.1. Grafos Split .....	76
III.9.2. Grafos de Intervalo .....	78
III.9.3. Grafos Threshold .....	80
III.9.4. Grafos Fortemente Triangularizados.	81
III.9.5. Grafos Complemento de Threshold com Tolerâncias .....	84
III.9.6. Grafos Cordais de Interseção de Caminhos .....	85
III.10. Extensões dos Grafos Triangularizados .....	86
III.10.1. Grafos de Gallai .....	86
III.10.2. Grafos de Paridade .....	88
III.10.3. Grafos Meyniel .....	89
III.10.4. Grafos Separáveis por Cliques ...	90
III.10.5. Grafos Fracamente Triangularizados .....	92
IV. GRAFOS INTERSEÇÃO DE CAMINHOS .....	94
IV.1. Definições .....	94
IV.2. Representação pela Árvore de Cliques .....	98
IV.3. Decomposição por Cliques Separadoras .....	100
IV.4. Reconhecimento de Grafos Interseção de Caminhos .....	107
IV.4.1. Grafos UV .....	107
IV.4.2. Grafos RDV e RDE .....	116
IV.4.3. Grafos DV .....	122
IV.4.4. Grafos DE .....	125
IV.4.5. Grafos UE .....	128
V. CONCLUSÕES .....	129
REFERÊNCIAS BIBLIOGRÁFICAS .....	131
APÊNDICE A .....	138
APÊNDICE B .....	139

## Capítulo I: INTRODUÇÃO

A teoria de grafos nasceu em 1736 quando Euler resolveu o problema das sete pontes da cidade de Königsberg. A cidade localiza-se ao lado do rio Pregel, onde sete pontes ligam a cidade com duas ilhas situadas no meio do rio. O problema diz: é possível fazer um percurso tal que partindo de uma das porções de terra, passa-se por todas as pontes, uma e somente uma vez, e retorna-se ao lugar inicial?

Nestes três séculos a *Teoria de Grafos* tem tido grande desenvolvimento, motivado fundamentalmente pelas diversas aplicações tanto na Matemática como em outras áreas, tais como eletricidade, química, genética e, principalmente, ciência da computação.

O estudo dos grafos é feito somente no seu aspecto teórico e também no desenvolvimento de algoritmos eficientes, em termos da sua complexidade computacional, para resolver problemas específicos.

O desenvolvimento da Teoria dos Grafos tem atingido uma amplitude tal, que agora distingue-se várias subáreas dentro dela. Uma destas subáreas corresponde aos denominados *Grafos Perfeitos* introduzidos por *Claude Berge* em 1960. Berge inicialmente denominou grafos  $\alpha$ -perfeitos aqueles onde o tamanho do maior conjunto estável é igual à dimensão da menor cobertura por cliques para todo subgrafo induzido; e aos grafos tais que o tamanho da maior clique é igual ao número cromático, para todo subgrafo induzido, ele chamou de grafos  $\chi$ -perfeitos. Berge foi motivado pelo trabalho de *Claude Shannon*, citado por LOVÁSZ, (1979) que demonstrou que o menor grafo onde o tamanho do maior conjunto estável é estritamente menor que a dimensão da menor cobertura por cliques, é o ciclo de cinco vértices. Berge formulou duas conjecturas, denominadas *Conjectura Fraca* e *Conjectura Forte*, pois a forte implica a fraca. A *Conjectura Fraca* diz que um grafo é  $\alpha$ -perfeito se e somente se é  $\chi$ -perfeito. Esta

denominada *Teorema dos Grafos Perfeitos*; desde esse momento os grafos que satisfazem as condições de  $\alpha$ -perfeitos e  $\chi$ -perfeitos são denominados somente *grafos perfeitos*. Por outro lado, em forma paralela Ray Fulkerson estudava a conjectura de um ponto de vista de Programação Linear, dando origem à *Teoria dos Antiblocking*. Quando Fulkerson soube que Lovász tinha demonstrado a conjectura fraca, ele em poucas horas conseguiu sua demonstração para o problema equivalente que tinha formulado.

A *Conjectura Forte* diz que um grafo é perfeito se e somente se ele e seu complemento não contêm um subgrafo induzido isomorfo a um ciclo sem cordas com um número ímpar de vértices maior que cinco. Até agora, esta conjectura não foi provada. Existem várias classes de grafos que se sabe que são perfeitos, os mais conhecidos são os grafos *triangularizados*, os de *comparabilidade*, os grafos *Meyniel* e os grafos *perfeitamente ordenáveis*.

Também conhecem-se classes de grafos onde a conjectura forte é válida, por exemplo, os grafos *planares* o qual foi provado por TUCKER (1973), os grafos *circulares* apresentado por BUCKINGHAM e GOLUMBIC (1983), os grafos *arco-circulares* dado por TUCKER (1975), PARTHASARATHY e RAVINDRA (1976) provaram para os grafos *livres de  $K_{13}$*  e os grafos *livres de  $K_4$  menos uma aresta*, apresentado em PARTHASARATHY e RAVINDRA (1979).

As linhas atuais de pesquisa tentam, por um lado, caracterizar os grafos perfeitos em termos da sua estrutura. Nesta linha está o trabalho de BURLET e FONLUPT (1984) que provaram que todo grafo Meyniel é obtido por uma operação em alguns grafos primitivos perfeitos, e a operação preserva a perfeição. HSU (1987) fez uma generalização para compor e decompor grafos perfeitos, baseada em ciclos induzidos e seus complementos.

Outra linha de pesquisa busca caracterizar os grafos que não são perfeitos, mas têm subgrafos induzidos que o

são. Os mais importantes são os grafos minimalmente imperfeitos, ou seja, um grafo que não é perfeito mas é suficiente retirar um vértice qualquer para torná-lo perfeito.

Também tenta-se provar a conjectura forte para algumas classes especiais de grafos. E continua-se na procura de condições suficientes para um grafo ser perfeito. CHVÁTAL (1984b) formulou a *Conjectura Semi-Forte*, a qual foi demonstrada por REED (1987) e que diz que os grafos  $P_4$ -isomorfos a grafos perfeitos são perfeitos.

Do ponto de vista algorítmico, para os quatro problemas clássicos de grafos (clique máxima, coloração mínima, conjunto estável máximo e cobertura por cliques) há o algoritmo de GROTSCHEL, LOVÁSZ e SCHRIJVER (1981) para grafos perfeitos, baseado no método do elipsóide. Esse algoritmo termina em tempo *polinomial*, mas não é baseado em aspectos combinatorios dos problemas. Além dele, existem algoritmos *polinomiais* para o reconhecimento e os quatro problemas clássicos, em algumas classes de grafos perfeitos, baseados em características particulares de cada uma. Mas, ainda não há solução para várias outras classes e continua em aberto o reconhecimento de grafos perfeitos. Sabe-se somente que o problema de reconhecer se um grafo é imperfeito pertence à classe *NP*.

GOLUMBIC (1980) é um livro específico sobre *grafos perfeitos*, apresentando muitos dos resultados teóricos e dos algoritmos desenvolvidos até esse momento e disseminados numa grande quantidade de artigos.

Dos trabalhos feitos em português tem-se a tese de NETO (1987), que apresenta um "survey" dos grafos perfeitos e imperfeitos, suas propriedades, algumas classes de grafos perfeitos e algumas classes de grafos onde a conjectura forte é válida. A tese de BARROSO (1987), embora não seja sobre grafos perfeitos, apresenta um estudo dos grafos circulares, que são uma subclasse dos grafos de interseção,

família que possui certa relação com os grafos perfeitos.

Neste trabalho, apresenta-se um "survey" dos *grafos triangularizados*, a primeira classe a ser reconhecida como perfeita, as subclasses dela, classes de grafos que a contêm e alguns dos problemas clássicos em grafos, estudados para esta classe particular. Tenta-se fazer o estudo principalmente sob um enfoque algorítmico.

No capítulo II definem-se os *conceitos gerais* da Teoria de Grafos necessários para os capítulos seguintes e introduz-se a noção de grafos perfeitos, imperfeitos e as conjecturas formuladas por Berge a respeito desses grafos.

No capítulo III são apresentados os *grafos triangularizados* e suas diferentes caracterizações que permitem o reconhecimento deles em tempo *polinomial*. São mostrados os algoritmos sequenciais e paralelos para o reconhecimento, para a construção da sua representação como grafo interseção e para alguns problemas que em um grafo qualquer estão em aberto ou são *NP-completo*, mas ao restringir-se à classe dos grafos triangularizados existem algoritmos *polinomiais* para resolvê-los. Logo mostram-se várias subclasses dos grafos triangularizados. Introduz-se três aplicações destes grafos: a solução de sistemas lineares, a minimização de uma função duas vezes diferenciável e a um problema de "scheduling". Finalmente, apresentam-se algumas classes de grafos perfeitos que contêm os grafos triangularizados.

O capítulo IV é dedicado à classe dos *grafos interseção de caminhos de uma subárvore*, alguns destes grafos são triangularizados dependendo se os caminhos são de vértices ou de arestas, se a árvore é direcionada ou não, e se é enraizada. Tenta-se dar um enfoque unificado para toda esta classe, apresentando sua relação com os grafos triangularizados e com os grafos perfeitos. São também discutidos os algoritmos de reconhecimento.

Em todos os temas e problemas mencionados no trabalho tenta-se dar vastas referências, indicando sempre a complexidade dos algoritmos. Os algoritmos expostos são apresentados em pseudo-linguagem semelhante a um tipo *ALGOL* em português, segundo a descrição dada por SZWARCFITER (1983).

No apêndice A apresenta-se um esquema com as relações de inclusão entre as diferentes classes de grafos introduzidas.

O apêndice B consiste de uma tabela resumo das classes de grafos vistas e os problemas de otimização em grafos mencionados no capítulo III, segundo a idéia da tabela apresentada por JOHNSON (1985). A entrada para cada classe e cada problema é a complexidade do algoritmo para resolver o problema restrito a essa classe de grafos, se existir. Caso contrário, é informado o que se sabe ao respeito do problema, isto é, se é *NP-completo* ou se é ainda desconhecida a sua pertencencia a esta classe.

## Capítulo II: CONCEITOS GERAIS

### II.1. Definições Básicas e Notação

Um *grafo não direcionado*  $G$  consiste de um conjunto finito não-vazio  $V$  e um conjunto  $E$  de pares não-ordenados de elementos distintos de  $V$ .  $G$  denota-se na forma  $G=(V,E)$ , onde  $V$  é o conjunto de vértices e  $E$  o conjunto de arestas. Considera-se  $|V|=n$  e  $|E|=m$ . Fala-se grafo em vez de grafo não direcionado, a não ser quando for necessário.

Cada aresta denota-se na forma do par  $(v,w)$ , diz-se que os vértices  $v$  e  $w$  são os extremos da aresta e eles são *vértices adjacentes*.

Se um vértice é extremo de duas arestas diferentes então elas são *adjacentes*.

Caso interesse a ordem em cada aresta, então  $(v,w)$  é diferente de  $(w,v)$  e o grafo é chamado *grafo direcionado*  $\bar{G}=(V,\bar{E})$ ; diz-se que  $(v,w)$  está orientada de  $v$  a  $w$  e denota-se  $v \longrightarrow w$ .

O grafo  $\bar{G}=(V,\bar{E})$  onde  $\bar{E}=\{(x,y) \in V \times V / x \neq y, (x,y) \notin E\}$  é o *grafo complemento* de  $G$ . Por exemplo, o grafo  $G$  e seu complemento  $\bar{G}$  da figura (II.1).

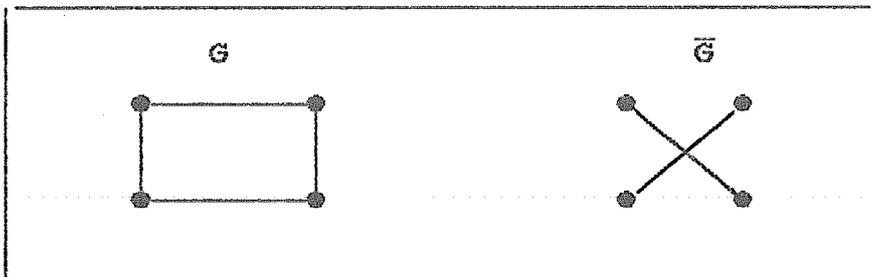


Figura II.1.

Dois grafos,  $G_1=(V_1,E_1)$  e  $G_2=(V_2,E_2)$  são ditos *isomorfos* se  $|V_1|=|V_2|$  e existe uma função unívoca  $f:V_1 \longrightarrow V_2$  tal que  $\forall v,w \in V_1$  tem-se  $(v,w) \in E_1$  se e somente se  $(f(v),f(w)) \in E_2$ .

Um grafo é *completo* se existe uma aresta entre cada par de seus vértices. Denota-se  $K_n$  ao grafo completo com  $n$  vértices, por exemplo na figura (II.2), apresenta-se  $K_n$  para  $n=1, 2, 3$  e  $4$ .

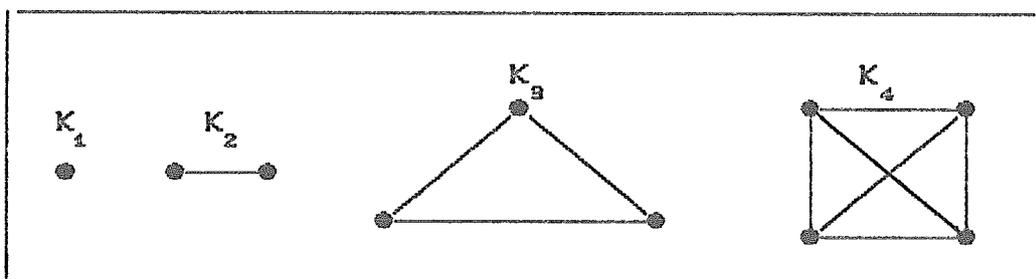


Figura II.2.

Cada vértice de  $K_n$  tem  $(n-1)$  arestas incidentes a ele. Para  $v \in V$  denota-se  $\text{Adj}(v)$  ao conjunto de vértices adjacentes a  $v$ . Chama-se *grau do vértice*  $v$  ao número de vértices adjacentes a ele, e denota-se  $\text{grau}(v)$ . Ou seja  $\text{grau}(v) = |\text{Adj}(v)|$ .

Diz-se que um grafo é *regular de grau*  $r$  se todos os vértices possuem grau  $r$ . Por exemplo,  $K_n$  é regular de grau  $(n-1)$ .

Um vértice com grau 0 é denominado *isolado*.

O grafo  $H=(V', E')$  com  $V' \subseteq V$  e  $E' \subseteq E$ , é um *subgrafo* de  $G$ ; um *subgrafo importante* é, dado  $A \subseteq V$ , o grafo  $G_A=(A, E_A)$ , onde  $E_A = \{(x, y) \in E / x \in A, y \in A\}$  contém todas as arestas de  $G$  cujos dois pontos finais são elementos de  $A$ ,  $G_A$  chama-se *subgrafo induzido por*  $A$ .

Dados  $v_1$  e  $v_k \in V$ , chama-se *caminho* de  $v_1$  a  $v_k$  a uma sequência de vértices  $P_k = v_1, v_2, \dots, v_k$  tal que  $(v_j, v_{j+1}) \in E$ , para  $1 \leq j \leq k-1$ , com  $k > 1$ ; o *comprimento do caminho* é o número de arestas que o compõem. Se todos os vértices do caminho forem distintos então o caminho é *simples* ou *elementar*. Em alguns casos, fala-se de *caminhos de arestas*

que é uma sequência de arestas  $a_1, a_2, \dots, a_{k-1}$  tal que  $a_i = (v_i, v_{i+1})$  com  $1 \leq i \leq k-1$ ,  $k > 1$ . Se não se especificar, o caminho será de vértices.

Um ciclo  $C_k$  é um caminho  $v_1, v_2, \dots, v_k, v_{k+1}$ , onde  $v_{k+1} = v_1$ , com  $k \geq 3$ . Por exemplo na figura (II.3) mostra-se o caminho  $P_3$  e o ciclo  $C_3$ .

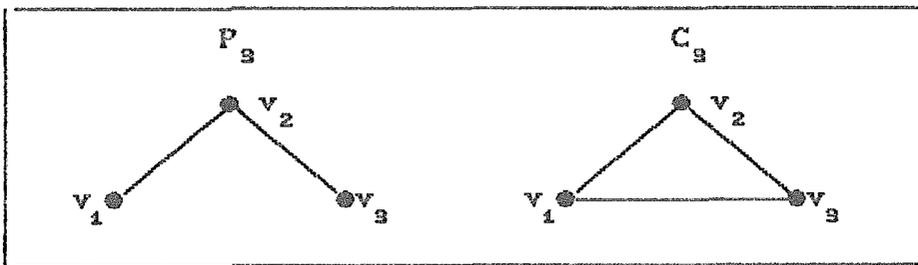


Figura II.3.

Uma corda é uma aresta não pertencente ao ciclo e que une dois vértices dele.

Se um grafo não tem nenhum ciclo, então diz-se que o grafo é acíclico.

Um caminho que contenha cada vértice do grafo exatamente uma vez, é chamado *caminho hamiltoniano*. O ciclo  $v_1, v_2, \dots, v_k, v_1$  é *hamiltoniano* se o caminho  $v_1, \dots, v_k$  o for.

Dado um grafo  $G=(V,E)$  e um vértice  $v \in V$ , denota-se  $G-v$  ao grafo obtido ao remover de  $G$  o vértice  $v$  e todas as arestas que tem num extremo o vértice  $v$ . Dado um vértice  $w \notin V$ , o grafo  $G+w$  é  $G^*=(V^*,E)$  onde  $V^* = V \cup \{w\}$ . Em geral, dado  $G$  e um subconjunto de vértices  $A \subseteq V$ ,  $G-A$  é o grafo obtido ao remover de  $G$  todos os vértices de  $A$  e todas as arestas com pelo menos um vértice em  $A$ . Dado  $G'=(V',E')$ , o grafo  $G+G'$  tem o conjunto de vértices  $V \cup V'$  e as arestas de  $E \cup E'$ .

Um grafo é *conexo* se existe algum caminho entre cada par de seus vértices, caso contrario é *desconexo*. Um grafo sem arestas é chamado *totalmente desconexo*. Se um grafo é desconexo, chamam-se *componentes conexas* a todo subgrafo induzido dele que seja conexo e não esteja contido em outro subgrafo conexo.

Uma *árvore* é um grafo conexo acíclico. Distinguem-se dentro de uma árvore os vértices com grau igual a um, os quais são chamados de *folhas*. Se a árvore é direcionada,  $\vec{T}=(V, \vec{E})$ , chama-se  $\forall v \in V$ : *grau de entrada*(v) =  $|\{(w, v) \in \vec{E}\}|$  corresponde ao número de arestas chegando a v e *grau de saída*(v) =  $|\{(v, w) \in \vec{E}\}|$  é o número de arestas saindo de v.

No caso das folhas, grau de entrada(v) = 1 e grau de saída(v) = 0.

Se existe um e só um vértice r com indegree(r)=0, esse vértice será chamado de *raiz* da árvore e diz-se que a árvore é *enraizada*.

Uma *subárvore* é um subgrafo conexo de uma árvore.

Um *multigrafo* é um grafo com arestas múltiplas entre pares de vértices. Um *grafo linha* L(M) de um multigrafo M é um grafo com um vértice por cada aresta de M e dois vértices de L(M) são adjacentes se as arestas correspondentes são adjacentes em M. Um multigrafo é *bipartite* se seu conjunto de vértices pode ser particionado em dois subconjuntos tais que vértices de um mesmo subconjunto não são adjacentes. Um multigrafo é *livre de triângulos* se não tem três vértices mutuamente adjacentes.

Uma *clique* é um subgrafo completo. Um *conjunto estável* ou *independente* é um subgrafo induzido, totalmente desconexo. Um conjunto estável de G é o grafo complemento de

uma clique de  $\bar{G}$ . O tamanho de uma clique ou de um conjunto estável é igual à cardinalidade do seu conjunto de vértices.

Uma clique é *maximal* se não existe uma outra clique de  $G$  que a contenha propriamente.

Uma clique é a *maior* (máxima) se não existe uma outra clique que possua cardinalidade maior.

Uma *cobertura*, de tamanho  $k$ , por cliques é uma partição dos vértices  $V=K_1+K_2+ \dots ,K_k$ , onde cada  $G_{K_j}$  é uma clique.

Uma *coloração* de  $G$  é uma partição dos vértices  $V=V_1+\dots+V_c$ , tal que cada  $G_{V_i}$  é um conjunto estável. Os elementos de  $V_i$  são todos coloridos da mesma cor e vértices adjacentes possuem cores diferentes.

O *índice cromático* de um grafo é o número de cores necessárias para colorir as arestas do grafo de tal forma que duas arestas adjacentes possuam cores diferentes.

### *Parâmetros de um Grafo*

Num grafo  $G$ , define-se quatro parâmetros:

$\omega(G)$  = tamanho da maior clique de  $G$ .

$\chi(G)$  = número cromático, é o menor número de cores necessários para colorir os vértices de  $G$ , cada um com uma cor, de tal forma que vértices adjacentes possuam cores diferentes. Corresponde à dimensão da menor cobertura por conjuntos estáveis.

$\alpha(G)$  = tamanho do maior conjunto estável.

$\kappa(G)$  = dimensão da menor cobertura por cliques.

Na figura (II.4) apresenta-se  $C_1, C_2, C_3, C_5$  e  $\bar{C}_5$  para os quais:

$$\begin{aligned} \omega(C_1) &= \chi(C_1) = \alpha(C_1) = \kappa(C_1) = 1, \\ \omega(C_2) &= \chi(C_2) = 2, \quad \alpha(C_2) = \kappa(C_2) = 1, \\ \omega(C_3) &= \chi(C_3) = 3, \quad \alpha(C_3) = \kappa(C_3) = 1, \\ \omega(C_5) &= 2, \quad \chi(C_5) = 3, \quad \alpha(C_5) = 2, \quad \kappa(C_5) = 3, \\ \omega(\bar{C}_5) &= 2, \quad \chi(\bar{C}_5) = 3, \quad \alpha(\bar{C}_5) = 2, \quad \kappa(\bar{C}_5) = 3. \end{aligned}$$

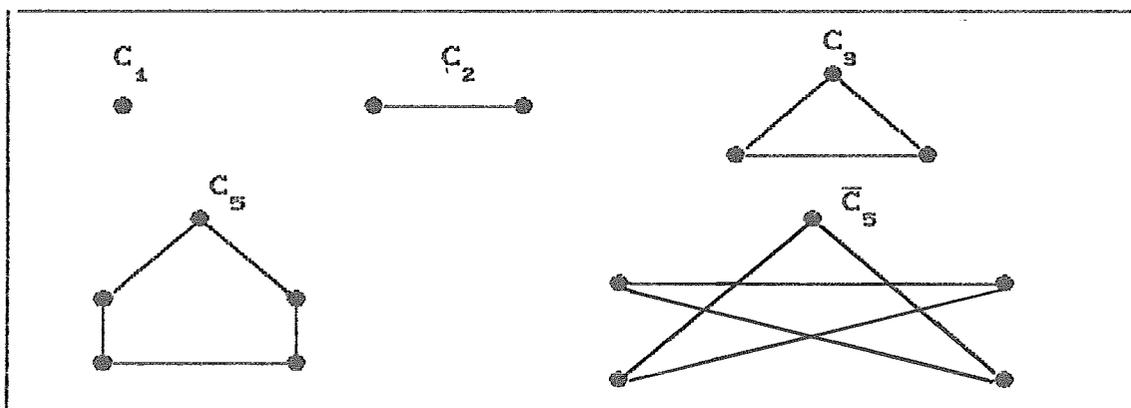


Figura II.4.

**Propriedade II.1:**

$$\omega(G) = \alpha(\bar{G}) \quad (\text{II.1})$$

$$\chi(G) = \kappa(\bar{G}) \quad (\text{II.2})$$

Estas igualdades dão uma dualidade entre o grafo e seu complemento.

**Propriedade II.2:**

$$\omega(G) \leq \chi(G) \quad (\text{II.3})$$

$$\alpha(G) \leq \kappa(G) \quad (\text{II.4})$$

Estas desigualdades vêm dos fatos que: cada vértice pertencente a uma clique tem cor diferente, logo precisa de pelo menos  $\omega(G)$  cores e a cobertura por clique deve ter

pelo menos uma clique por cada vértice do maior conjunto estável.

**Observação II.1:**

A interseção de uma clique e um conjunto estável pode ser no máximo igual a um. Isto vem das definições de clique e conjunto estável.

**Observação II.2:**

Em alguns grafos tem-se  $\omega(G) = \chi(G)$ , mas esta não é uma *propriedade de hereditariedade*, ou seja, se  $G$  satisfaz a igualdade isto não é suficiente para que todo subgrafo induzido de  $G$  também a satisfaça. Os grafos que se comportam desta última forma merecem atenção especial, são os denominados *grafos perfeitos*.

**Definição II.1:**

$G$  é  $\omega$ -perfeito se:

$$\omega(G_A) = \chi(G_A) \quad \forall A \subseteq V. \quad (\text{II.5})$$

$G$  é  $\alpha$ -perfeito se:

$$\alpha(G_A) = \kappa(G_A) \quad \forall A \subseteq V. \quad (\text{II.6})$$

$G$  é perfeito se é  $\omega$ -perfeito e  $\alpha$ -perfeito.

No grafo da figura (II.5) apresenta-se um caminho  $P_n$ ,  $n=4$ , um ciclo par  $C_{2k}$  e um ciclo ímpar  $C_{2k+1}$ ,  $k=3$ . Tem-se  $\omega(P_n) = 2$ ,  $\chi(P_n) = 2$ ,  $\alpha(P_n) = n/2$  se  $n$  é par e  $\alpha(P_n) = (n+1)/2$  se  $n$  é ímpar,  $\kappa(P_n) = \alpha(P_n)$  e qualquer subgrafo dele é da mesma forma, logo  $P_n$  é perfeito.

Para os ciclos tem-se  $\omega(C_{2k}) = \chi(C_{2k}) = 2$ ,  $\alpha(C_{2k}) = \kappa(C_{2k}) = k$  e todos os subgrafos são caminhos, logo  $C_{2k}$  é perfeito; para  $C_7$  tem-se  $\omega(C_7) = 2$ ,  $\chi(C_7) = 3$ ,  $\alpha(C_7) = 3$ ,  $\kappa(C_7) = 4$ , logo  $C_7$  não é perfeito.

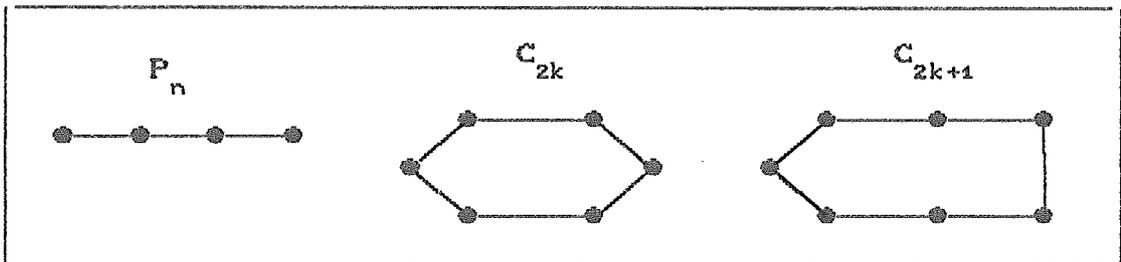


Figura II.5.

**Observação II.3:**

Todo grafo completo  $K_n$  é perfeito  $\chi(K_n) = n$ .

Todo grafo bipartite é perfeito  $\chi(B) = 2$ .

Todo ciclo sem cordas de comprimento ímpar maior do que 3 não é perfeito:

$$\omega(C_{2n+1}) = 2,$$

$$\chi(C_{2n+1}) = 3.$$

**Propriedade II.3:**

$G$  perfeito  $\Leftrightarrow \bar{G}$  perfeito

Esta equivalência vem da dualidade de  $G$  e  $\bar{G}$ .

**Conjectura Fraca dos Grafos Perfeitos**

As igualdades (II.5) e (II.6) são equivalentes.

### *Teorema dos Grafos Perfeitos*

As equações (II.5) e (II.6) são equivalentes à seguinte desigualdade:

$$\omega(G_A) + \alpha(G_A) \geq |A| \quad \forall A \subseteq V \quad (\text{II.7})$$

A conjectura fraca dos grafos perfeitos foi provada por LOVÁSZ (1972) fazendo uso da condição (II.7) do teorema. Com isto a definição de grafo perfeito passou a ser,

#### *Definição II.2:*

$G$  é perfeito se  $\omega(G_A) = \alpha(G_A) \quad \forall A \subseteq V$ .

### *Conjectura Forte dos Grafos Perfeitos*

$G$  é perfeito se e somente se  $G$  não contém um subgrafo induzido isomorfo a  $C_{2k+1}$  ou  $\bar{C}_{2k+1}$  para  $k \geq 2$ .

Esta conjectura ainda está em aberto, conhecem-se algumas classes de grafos onde a conjectura forte foi demonstrada. Muito do trabalho desenvolvido após a formulação da conjectura, tende a se aproximar dela, pesquisando grafos onde ela é válida, grafos que não são perfeitos e as características deles que os fazem imperfeitos, os quais dentre eles os mais importantes são os minimalmente imperfeitos.

#### *Definição II.3:*

Um grafo  $G=(V,E)$  com  $|V|=n$ , é *particionável* se:

$$n = \alpha(G) + \omega(G) \quad (\text{II.8})$$

$$\forall v \in V \quad \omega(G) = \chi(G-v) \quad (\text{II.9})$$

$$e \quad \forall v \in V \quad \alpha(G) = \kappa(G-v) \quad (\text{II.10})$$

**Propriedade II.4:**

$G$  é particionável se e somente se:

- $n = \alpha(G) \omega(G) + 1$ .
- $G$  tem exatamente  $n$  cliques máximas e  $n$  conjuntos estáveis máximos.
- Cada vértice de  $G$  pertence exatamente a  $\omega(G)$  cliques máximas e a  $\alpha(G)$  conjuntos estáveis máximos.
- Cada clique máxima intercepta a todos salvo um conjunto estável máximo.

**Definição II.4:**

Um grafo  $G$  é *minimalmente imperfeito* se ele não é perfeito mas, qualquer subgrafo induzido próprio de  $G$  é perfeito. Isto é,  $\forall v \in V: \omega(G-v) = \chi(G-v)$  e  $\alpha(G-v) = \kappa(G-v)$  mas,  $\omega(G) < \chi(G)$  e  $\alpha(G) < \kappa(G)$ .

**Propriedade II.5:**

Todo grafo minimalmente imperfeito é particionável.

Com estes grafos tem-se uma formulação equivalente para a conjectura forte dos grafos perfeitos: os únicos grafos minimalmente imperfeitos são  $C_{2k+1}$  e  $\bar{C}_{2k+1}$ , para todo  $k \geq 2$ .

## II.2. Grafos Perfeitamente Ordenáveis

Uma das linhas de pesquisa é tentar *caracterizar a classe dos grafos perfeitos*. Uma caracterização deles é através da determinação do seu *número cromático*, mas não se conhece um algoritmo eficiente para calcular este número.

Uma maneira natural de colorir os vértices de um grafo é:

- 1º) Ordenar os vértices sequencialmente  $v_1, v_2, \dots, v_n$ .
- 2º) Segundo esta ordem, atribuir a cada  $v_i$  a menor cor disponível, isto é a menor  $c_i \in \mathbb{Z}^+$  não usada até o momento para colorir algum vizinho  $v_j$  de  $v_i$ ,  $j < i$ .

Isto não é mais que o *Algoritmo Guloso*, mas surgem dois problemas:

- Como obter a ordenação dos vértices ?
- Quando esta coloração é ótima ?

$$\text{Isto é: } \max_i \{ c_i \} = \chi(G).$$

Observando classes conhecidas de grafos perfeitos, Chvátal viu que na demonstração de que eles são perfeitos, sempre usa-se a mesma estratégia: dar uma ordem ao conjunto de vértices, que depende da classe, tal que o algoritmo guloso colore todo subgrafo induzido de maneira ótima, e prova-se que existe uma clique de tamanho igual ao número de cores utilizadas.

### Definição II.5:

Um *poset* é um conjunto parcialmente ordenado  $(X, P)$ , onde  $X$  é não vazio e  $P$  é uma relação binária em  $X$ .  $P$  é reflexiva, transitiva e anti-simétrica,  $P$  é representada pelo símbolo " $<$ ".

Se  $x < y$  e não existe  $z \in X$  tal que  $x < z < y$ , então

diz-se que  $y$  cobre  $x$ .

Se  $\dim(X)$  é finita, então  $P$  fica determinada pela relação de cobertura, com esta relação faz-se a representação de um poset.

O *Diagrama de Hasse* de um poset é um desenho no qual os pontos de  $X$  são posicionados em forma tal que se  $y$  cobre  $x$  então  $y$  está num nível inferior que  $x$  e eles são ligados por uma linha. Por exemplo, no conjunto  $X = \{a, b, c, d, e, f\}$ , se  $a < b$ ,  $a < c$ ,  $c < d$ ,  $b < e$ ,  $d < f$ ,  $e < f$ , então o diagrama de Hasse é o da figura (II.6).

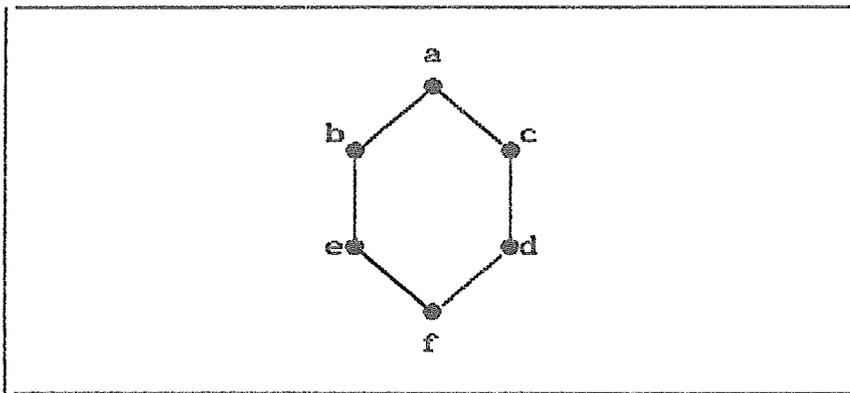


Figura II.6.

Um *conjunto linearmente ordenado* é um poset no qual todo par de elementos distintos é comparável. Também é chamado de cadeia.

Quando tem-se os vértices de um grafo  $G$  ordenados linearmente, é aproveitada esta ordem para colorir os vértices respeitando a condição usual de que dois vértices adjacentes tenham cores diferentes.

Dados os vértices ordenados em uma sequência  $v_1, v_2, \dots, v_n$ , com  $v_i < v_{i+1}$ , associa-se a cada  $v_j$  o menor inteiro positivo  $f(v_j)$  tal que nenhum  $v_i$ ,  $i < j$ , adjacente a  $v_j$  tem  $f(v_i) = f(v_j)$ .

O grafo com uma ordem linear no seu conjunto de vértices chama-se *grafo ordenado*; o maior valor  $f(v_j)$  é chamado *número de Grundy* do grafo ordenado. Tem-se:

$$\chi(G) \leq N^{\circ} \text{ Grundy}(G, <) \quad (\text{II.11})$$

Por exemplo, na figura (II.7) para o grafo  $G_1$  com a ordem linear "b c d e a", tem-se a coloração  $f(b)=1$ ,  $f(c)=2$ ,  $f(d)=3$ ,  $f(e)=2$  e  $f(a)=3$ . Por outro lado,  $\chi(G_1)=3$  e neste caso a coloração é ótima. No grafo  $P_4$  com as restrições  $a < b$  e  $d < c$ , uma ordem linear possível é "a b d c" que dá a coloração  $f(a)=1$ ,  $f(b)=2$ ,  $f(d)=1$  e  $f(c)=3$ , mas  $\chi(P_4)=2$ .

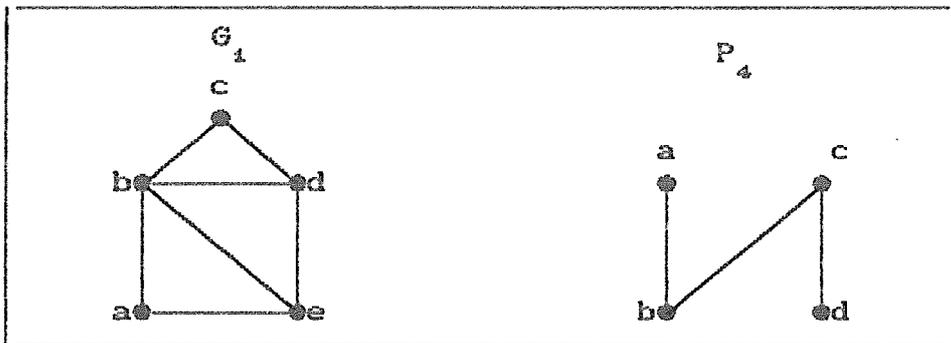


Figura II.7.

Com qualquer ordem linear que respeite as restrições  $a < b$  e  $d < c$ , acontece o mesmo, isto é porque os dois extremos são coloridos antes de acabar de colorir os vértices internos. Este grafo ordenado é chamado de *obstrução*.

Já viu-se que em alguns casos o número de Grundy é igual ao número cromático do grafo, o qual dá a coloração ótima, ou seja, a função  $f$  produz uma boa coloração em  $G$ .

**Definição II.6:**

Uma ordem linear é perfeita se para todo subgrafo induzido  $F$  de  $G$ , a coloração  $f$  definida a partir da ordem, usa somente  $\chi(F)$  cores.

**Observação II.4:**

A ordem linear é perfeita se o algoritmo guloso colore de forma ótima todo subgrafo induzido com a ordem herdada, caracterização dada por REED (1988).

CHVÁTAL (1984a) definiu a ordem perfeita usando a obstrução como subgrafo proibido .

**Definição II.7:**

Uma ordem linear no conjunto de vértices é *admissível* se não cria obstrução.

**Propriedade II.6:**

Uma ordem linear é perfeita se e somente se é admissível.

O grafo  $G$  é *perfeitamente ordenável*, se admite uma ordem perfeita. Da propriedade (II.6) deduz-se que todo grafo perfeitamente ordenável é perfeito. A inclusão é estrita,  $\bar{C}_6$  é perfeito mas não é perfeitamente ordenável.

CHVÁTAL (1984c) define outra classe de grafos perfeitos, os fortemente perfeitos.

*Definição II.8:*

$G$  é fortemente perfeito se todo subgrafo induzido  $H$  de  $G$  contém um conjunto estável que intercepta todas as cliques maximais em  $H$ .

*Propriedade II.7:*

Todo grafo perfeitamente ordenável é fortemente perfeito.

A recíproca é falsa, o grafo da figura (II.8) é fortemente perfeito, mas não é perfeitamente ordenável.

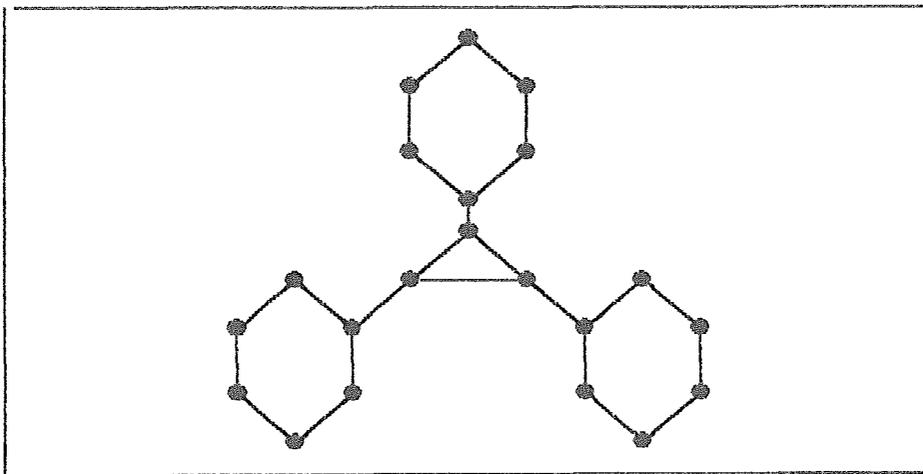


Figura II.8.

Das classes mais conhecidas de grafos perfeitos, tem-se que os grafos de comparabilidade, os triangularizados e os co-triangularizados são perfeitamente ordenáveis.

### II.3. Problema da Busca em Grafos

A busca visa resolver o problema de como explorar um grafo, isto é, dado um grafo, deseja-se obter um processo sistemático de como percorrer os vértices e as arestas do mesmo. O algoritmo geral pode escrever-se na seguinte forma:

#### Algoritmo II.1: BUSCA EM UM GRAFO

- Dado  $G$
- escolher e marcar um vértice inicial
- enquanto existir algum vértice  $v$  marcado e incidente a uma aresta  $(v,w)$  não explorada efetuar
  - escolher o vértice  $v$  e explorar a aresta  $(v,w)$
  - se  $w$  é não marcado então marcar  $w$ .

Neste algoritmo devem ser feitas as escolhas do vértice inicial, do vértice marcado e da aresta incidente. Uma busca em largura tem o critério de escolha do vértice marcado, dado na forma: dentre todos os vértices marcados e incidentes a alguma aresta ainda não explorada, escolher aquele menos recentemente alcançado na busca. Chama-se largura de um vértice ao número de ordem em que ele é marcado.

#### Busca em Largura Lexicográfica

Uma busca em largura lexicográfica, além do critério de escolha do vértice marcado, tem um critério para a seleção da aresta incidente a ser explorada dentro das listas de adjacências. A idéia é, em cada passo, explorar o vértice mais remoto dentro dos vértices ainda não explorados e adjacentes ao vértice atual,  $w_1$  mais remoto que  $w_2$  quer dizer que existe um vértice  $z$  não marcado incidente a  $w_1$  e que não é incidente a  $w_2$ , tal que a largura de  $z$  é menor que a largura de todo vértice não marcado que seja incidente a  $w_2$  mas não a  $w_1$ .

**Definição II.9:**

Sejam  $S_1 = x_1, x_2, \dots, x_p$  e  $S_2 = y_1, y_2, \dots, y_q$  sequências de inteiros.  $S_1$  é *lexicograficamente maior* do que  $S_2$  ( $S_1 \succ_L S_2$ ) se:  $\exists j$ , com  $1 \leq j \leq \min\{p, q\}$  tal que

$$x_j > y_j \text{ e } \forall 1 \leq k < j : x_k = y_k \quad (\text{II.12})$$

$$\text{ou } p > q \text{ e } \forall 1 \leq k \leq q : x_k = y_k \quad (\text{II.13})$$

**Definição II.10:**

Seja  $w \in V$  adjacente aos vértices já marcados  $z_1, \dots, z_k$ , nomeados em ordem crescente de suas larguras.  $R(w)$  é o conjunto dos complementos das larguras dos  $z_i$ :

$R(w) = \{n+1 - \text{largura}(z_i) \mid z_i \text{ é incidente a } w \text{ e está marcado}\}$ .

**Definição II.11:**

Seja  $v \in V$  e  $A^*(v) = \{w \in \text{Adj}(v) \mid w \text{ não marcado}\}$ , sejam  $w_1, w_2 \in A^*(v)$ ,  $w_1$  é mais remoto do que  $w_2$  se:  $\exists z$  marcado com  $(z, w_1) \in E$ , mas  $(z, w_2) \notin E$  tal que  $\forall z'$  marcado com  $(z', w_1) \notin E$  e  $(z', w_2) \in E$ , satisfaz  $\text{largura}(z) < \text{largura}(z')$ .

**Propriedade II.8:**

$w_1$  é mais remoto do que  $w_2$  se e somente se  $R(w_1) \succ_L R(w_2)$ .

**Observação II.5:**

Uma busca em largura é lexicográfica se,  $\forall v \in V$  e  $\forall w_1, w_2 \in A^*(v)$  tem-se: se  $w_1$  é mais remoto do que  $w_2$ , então,  $w_1$  é explorado antes de  $w_2$  na busca. Ou seja, a exploração é feita em ordem lexicográfica decrescente de  $R(w)$ .

Com isto o algoritmo de busca em largura lexicográfica fica:

*Algoritmo II.2: BUSCA EM LARGURA LEXICOGRÁFICA*

- Dado  $G=(V,E)$  conexo,  $|V|=n$  e  $|E|=m$
- desmarcar todos os vértices
- para todo  $v \in V$  definir  $R(v) = \emptyset$
- para  $j=n, n-1, \dots, 1$  efetuar
  - escolher um vértice  $v$  não marcado com  $R(v)$  lexicograficamente máximo
  - marcar  $v$
  - para todo  $w \in \text{Adj}(v)$  e  $w$  não marcado efetuar
    - incluir  $j$  à direita em  $R(w)$
- desmarcar todos os vértices
- definir uma fila  $Q$  vazia
- seja  $s$  o vértice com  $R(s)$  lexicograficamente máximo
- marcar  $s$
- inserir  $s$  em  $Q$
- enquanto  $Q \neq \emptyset$  efetuar
  - seja  $v$  o primeiro elemento de  $Q$
  - enquanto  $\text{Adj}(v) \neq \emptyset$  efetuar
    - seja  $w \in \text{Adj}(v)$  com  $R(w)$  lexicograficamente máximo
    - se  $w$  é não marcado então
      - visitar  $(v,w)$
      - marcar  $w$
      - inserir  $w$  em  $Q$
    - caso contrário
      - se  $w \in Q$  então visitar  $(v,w)$
  - retirar  $w$  de  $\text{Adj}(v)$
- retirar  $v$  de  $Q$
- devolver  $R$

A complexidade deste algoritmo é:  $O(n+m)$ .

## II.4. Grafos de Comparabilidade

### Definição II.12:

Um grafo não direcionado  $G=(V,E)$  é um *grafo de comparabilidade* se existe uma orientação das arestas  $\vec{E}$  tal que o grafo direcionado  $\vec{G}=(V,\vec{E})$  satisfaz:

$$\forall x,y,z \in E: (x,y), (y,z) \in \vec{E} \Rightarrow (x,z) \in \vec{E}$$

$\vec{E}$  é chamada *orientação transitiva de G* e define uma ordem parcial estrita em  $V$  com a interpretação :  $(x,y) \in \vec{E}$  se e somente se  $x < y$ .

Na figura (II.9) tem-se o grafo  $G_2$  e uma possível orientação transitiva de suas arestas.

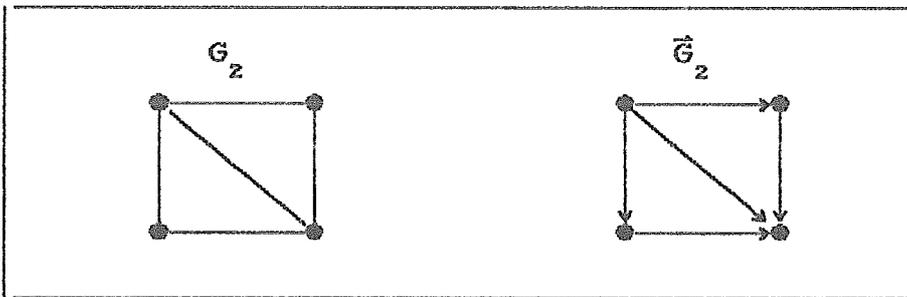


Figura II.9.

Os grafos de comparabilidade também são chamados *orientáveis transitivamente* ou *grafos parcialmente ordenáveis*.

### Observação II.6:

Todos os ciclos ímpares  $C_k$ ,  $k \geq 5$ , não são de comparabilidade,  $C_3$  o é.

Todo ciclo par  $C_k$  é de comparabilidade. Uma orientação

possível é: se  $i$  é ímpar então  $v_i \longrightarrow v_{i+1}$ , se  $i$  é par então  $v_{i+1} \longrightarrow v_i$ , considerando  $v_{k+1} = v_1$ .

Todo grafo completo  $K_n$  é de comparabilidade. Basta numerar os vértices de 1 até  $n$  em forma arbitrária e orientar  $i \longrightarrow j$  se  $i < j$ .

Todo caminho  $P_n = v_1, \dots, v_n$  é de comparabilidade. Orienta-se  $(v_1, v_2)$  e as demais arestas têm uma orientação forçada. Por exemplo,  $P_5$  na figura (II.10).

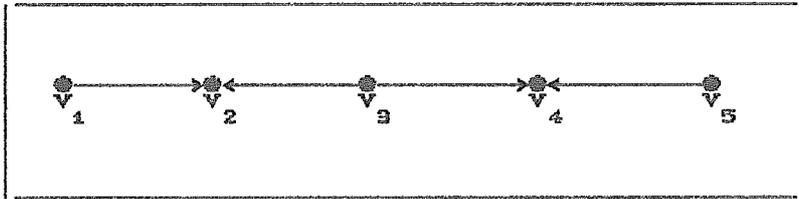


Figura II.10.

**Definição II.13:**

Um grafo  $G$  é de *co-comparabilidade* se seu complemento  $\bar{G}$  é de comparabilidade, por exemplo, o grafo da figura (II.11) é  $\bar{C}_6$  e  $C_6$  é de comparabilidade.

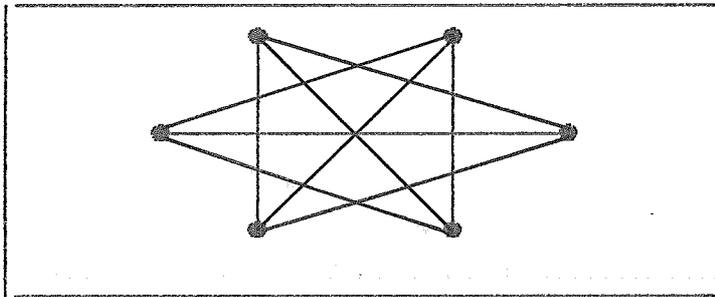


Figura II.11.

**Propriedade II.9:**

Todo grafo de comparabilidade é perfeito.

Todo grafo de co-comparabilidade é perfeito.

**Propriedade II.10:**

Todo grafo de comparabilidade é perfeitamente ordenável.

**II.4.1. Grafos Bipartite**

Um grafo  $G=(V,E)$  é *bipartite* se seu conjunto de vértices pode ser particionado em dois subconjuntos,  $V=V_1+V_2$ , em forma tal que toda aresta une um vértice de  $V_1$  com um vértice de  $V_2$  e não tem nenhuma aresta ligando dois vértices de um mesmo subconjunto.

Por exemplo, no grafo da figura (II.12),  $V_1 = \{a,b,c\}$  e  $V_2 = \{x,y\}$ .

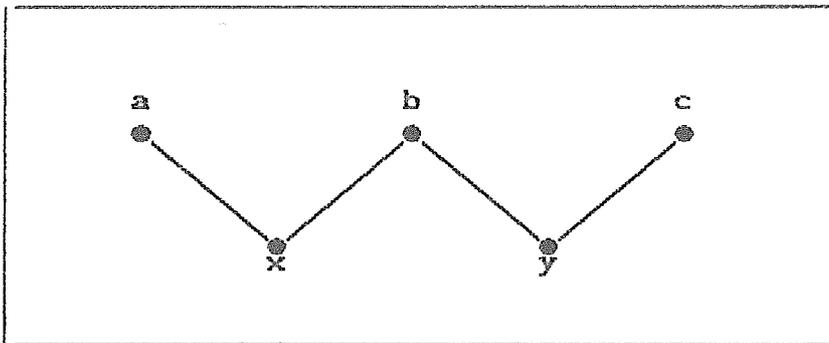


Figura II.12.

Se  $|V_1| = n$  e  $|V_2| = m$ , denota-se  $B_{nm}$  ao grafo bipartite completo, isto é, todo vértice de  $V_1$  ligado com todos os vértices de  $V_2$ . Se  $n=m$ , chama-se simplesmente  $B_n$ .

**Propriedade II.11:**

Todo grafo bipartite é de comparabilidade. Pode-se orientar toda aresta  $(x,y)$  na forma  $x \longrightarrow y$  se  $x \in V_1$  ou  $y \longrightarrow x$  se  $x \in V_2$ .

**Corolário II.1:**

Todo grafo bipartite é perfeito.

**II.4.2. Grafos de Permutação**

Um grafo  $G=(V,E)$  é um grafo de *permutação* se existe uma numeração  $v_1, v_2, \dots, v_n$  dos vértices e uma permutação  $\pi=(\pi_1, \dots, \pi_n)$  da  $n$ -tupla  $(1, 2, \dots, n)$  tal que  $(v_i, v_j) \in E$  se e somente se  $(i-j)(\pi_i^{-1} - \pi_j^{-1}) < 0$ , onde  $\pi_i^{-1}$  é a posição do número  $i$  na sequência  $\pi$ . Isto é,  $(v_i, v_j) \in E$  se os números  $i$  e  $j$  tem uma inversão na ordem em  $\pi$  com respeito à ordem natural  $1, 2, \dots, n$ .

Na figura (II.13) apresenta-se um grafo de permutação.

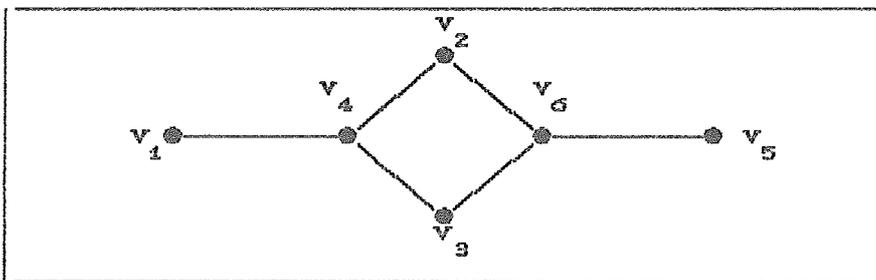


Figura II.13.

**Propriedade II.12:**

$G$  é um grafo de permutação se e somente se  $G$  e  $\bar{G}$  são de comparabilidade.

**Corolário II.2:**

Todo grafo de permutação é perfeito.

### II.4.3. Reconhecimento de Grafos de Comparabilidade

PNUELI et alii (1971) desenvolveram um algoritmo que determina uma orientação transitiva e acíclica de um grafo não direcionado, se existe. Caso contrário, detecta que o grafo não admite orientação transitiva. O algoritmo é:

#### Algoritmo II.3: RECONHECIMENTO DE GRAFOS DE COMPARABILIDADE

- Dado  $G=(V,E)$
- enquanto existir arestas não direcionadas efetuar
  - pegar uma aresta  $(a,b)$  não direcionada e orientá-la  $a \longrightarrow b$
  - pegar uma aresta  $(s,u)$  orientada na forma  $s \longrightarrow u$
  - enquanto existir arestas não orientadas adjacentes a  $(s,u)$  efetuar
    - se  $(u,w) \in E$  e  $(s,w) \notin E$  orientar  $w \longrightarrow u$
    - se  $(s,t) \in E$  e  $(t,u) \notin E$  orientar  $s \longrightarrow t$
    - se deve orientar-se  $w \longrightarrow u$  e  $u \longrightarrow w$  então devolver  $G$  não é de comparabilidade
  - apagar as arestas orientadas
- devolver  $G$  é grafo de comparabilidade e  $\vec{G}$  é uma orientação possível.

A complexidade deste algoritmo é  $O(n^3)$ .

### II.5. Grafos de Interseção

Dada uma família  $\mathcal{F}$  de conjuntos não vazios, constroi-se o grafo interseção  $G=(V,E)$  da família, fazendo uma correspondência um a um entre os conjuntos da família e os vértices do grafo, e uma aresta entre dois vértices existe se os correspondentes conjuntos têm interseção não vazia.

Quando a família de conjuntos tem uma estrutura topológica específica, os grafos de interseção associados recebem um nome particular. Alguns destes grafos de

interseção têm sido estudados do ponto de vista algorítmico, considerando sua importância pelas aplicações que eles têm.

Para algumas famílias, o grafo de interseção associado é perfeito.

### II.5.1. Grafos de Intervalo

Um *grafo de intervalo* é o grafo interseção de um conjunto de intervalos de uma reta. Existe uma correspondência um a um entre os vértices do grafo e os intervalos da reta, há uma aresta entre dois vértices se os intervalos associados a eles, se interceptam.

Sejam  $I_1, I_2, \dots, I_n$  os intervalos, então o grafo está determinado pelo conjunto de vértices  $V = \{v_1, v_2, \dots, v_n\}$  e o conjunto de arestas  $E = \{(v_i, v_j) / I_i \cap I_j \neq \emptyset\}$

Por exemplo, na figura (II.14) apresenta-se um conjunto de cinco intervalos reais e o grafo de intervalo associado.

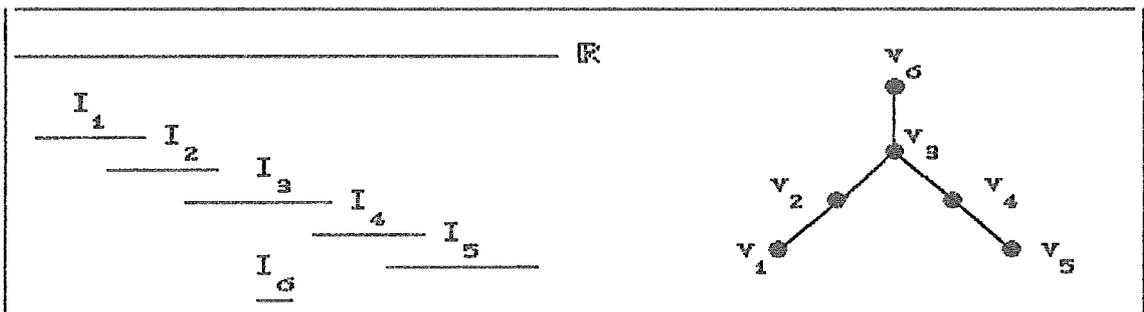


Figura II.14.

Em geral todo caminho  $P_n = \langle v_1, v_2, \dots, v_n \rangle$  é um grafo de intervalo.

#### Propriedade II.13:

Os grafos de intervalo são perfeitos.

Um tratamento mais detalhado destes grafos será realizado na seção III.5.1.

### II.5.2. Grafos Circulares

Um *grafo circular* é o grafo interseção de uma coleção finita de cordas de um círculo, isto é, cada corda do círculo está associada com um vértice, e dois vértices estão ligados se as correspondentes cordas se interceptam. Considera-se, sem perda de generalidade, que duas cordas não compartilham extremidades.

Na figura (II.15) tem-se um conjunto de cordas num círculo e o grafo circular associado a ele.

Em geral  $P_n$ ,  $C_n$  e  $K_n$  são grafos circulares.

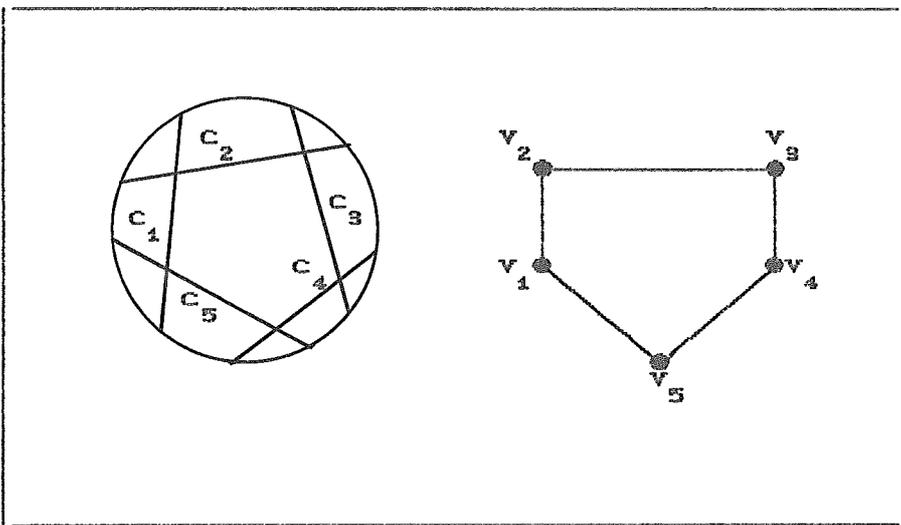


Figura II.15.

#### Observação II.7:

Nem todo grafo circular é perfeito. Como um exemplo tem-se que  $C_5$  não é perfeito e  $K_5$  é perfeito. Um tratamento mais aprofundado deste tipo de grafos encontra-se no

trabalho de BARROSO (1987).

### II.5.3. Grafos Arco-circulares

Um *grafo arco-circular* é modelado como um conjunto de arcos ao redor de um círculo e existe uma aresta entre dois vértices se os arcos correspondentes se interceptam.

Na figura (II.16) mostra-se um conjunto de arcos e o grafo associado a ele.

Em geral  $P_n$ ,  $C_n$  e  $K_n$  são arco-circulares.

**Propriedade II.14:**

Todo grafo de intervalo é arco-circular.

A recíproca é falsa, o grafo da figura (II.16) é arco-circular mas não é de intervalo.

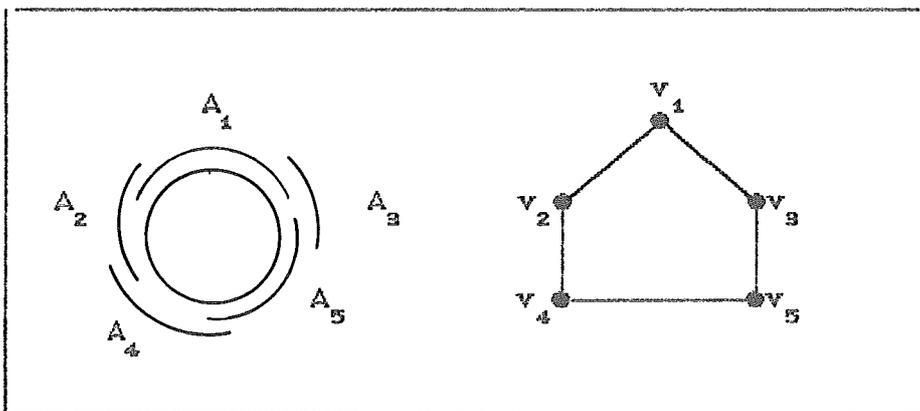


Figura II.16.

**Observações II.8:**

Nem todo grafo arco-circular é perfeito.

Existem grafos que são circulares e arco-circulares,

mas não é possível dar uma relação de subconjuntos entre eles.

#### II.5.4. Grafos Triangularizados

Um *grafo cordal* ou *triangularizado* é o grafo interseção de subárvores de uma árvore.

O capítulo III deste trabalho é dedicado por inteiro a esta classe de grafos.

#### II.5.5. Grafos Interseção de Caminhos

Dados uma árvore e caminhos na árvore, tem-se diferentes grafos interseção de caminhos caso se considere que os caminhos compartilhem vértices ou arestas. Também é feita outra classificação ao considerar a árvore direcionada, distinguindo-se se a árvore é não direcionada ou direcionada ou enraizada. Usando os nomes dados por MONMA (1986) tem-se as classes:

UV: grafo interseção de caminhos de vértices da árvore não direcionada.

DV: grafo interseção de caminhos de vértices da árvore direcionada.

RDV: grafo interseção de caminhos de vértices da árvore direcionada e enraizada.

UE: grafo interseção de caminhos de arestas da árvore não direcionada.

DE: grafo interseção de caminhos de arestas da árvore direcionada.

RDE: grafo interseção de caminhos de arestas da árvore direcionada e enraizada.

O capítulo IV deste trabalho apresenta um estudo mais aprofundado destes grafos.

### Capítulo III: GRAFOS TRIANGULARIZADOS

A classe dos grafos triangularizados foi uma das primeiras a ser reconhecida como uma classe de grafos perfeitos. No primeiro encontro internacional sobre teoria de grafos na Hungria em 1959, HAJNAL e SURÁNYI apresentaram que todo grafo triangularizado é  $\alpha$ -perfeito. BERGE ampliou este resultado em 1960, provando que todo grafo triangularizado é  $\chi$ -perfeito. Este resultado também motivou a formulação da conjectura fraca.

Os estudos de FULKERSON e GROSS (1965) e de ROSE (1970) desenvolveram caracterizações e propriedades destes grafos baseados na estrutura particular deles.

#### III.1. Definições

##### Definição III.1:

Um grafo  $G=(V,E)$  é *triangularizado* se cada ciclo de comprimento maior do que três, possui uma corda. Por exemplo, na figura (III.1) o grafo  $G_1$  é triangularizado, entretanto o grafo  $G_2$  não é, pois o ciclo  $b c e f$  não possui nenhuma corda.

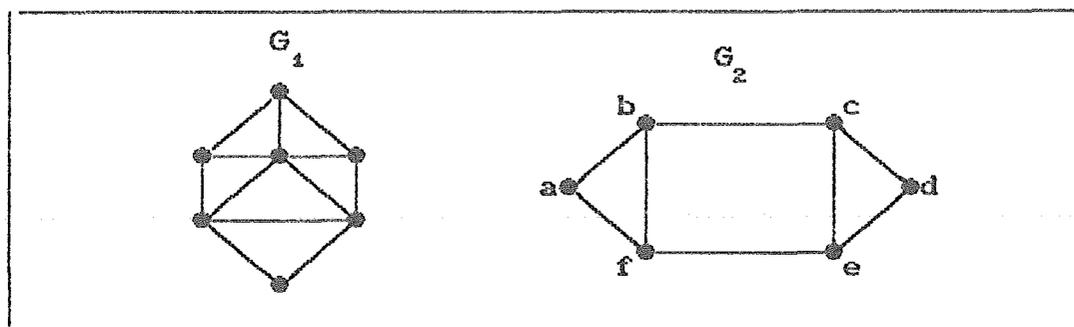


Figura III.1.

Os grafos triangularizados também são chamados *cordais*, *de eliminação perfeita*, *monótono transitivo* ou "*rigid-circuit*".

**Observação III.1:**

Todo grafo completo  $K_n$  é triangularizado.

Todo caminho  $P_n$  é triangularizado.

Todo ciclo  $C_k$ ,  $k > 3$ , sem cordas não é triangularizado.

**Propriedade III.1:**

Se  $G$  é triangularizado, com  $|V| > 1$ , e  $v \in V$  então  $G-v$  é triangularizado.

Isto mostra que o fato de ser um grafo triangularizado é uma propriedade de hereditariedade, ou seja, todo subgrafo induzido de um grafo cordal, é cordal. Entretanto a propriedade ser triangularizado não é fechada por complementação.

**Definição III.2:**

Um grafo  $G$  é *co-triangularizado* se seu grafo complemento  $\bar{G}$  é triangularizado. Por exemplo, na figura (III.2), é dado o grafo  $G_3$  e seu complemento  $\bar{G}_3$ , os dois são triangularizados. Já o grafo  $G_4$  que também é triangularizado, tem como complemento  $\bar{G}_4$  que é exatamente o ciclo  $C_4$ , que não o é.

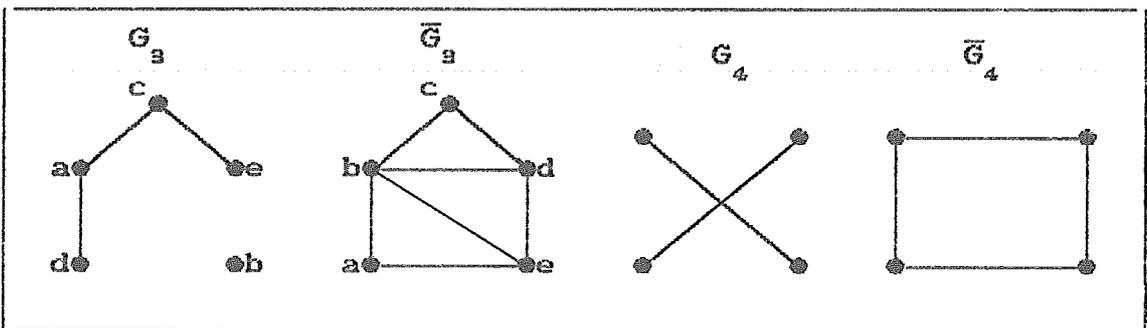


Figura III.2.

### III.2. Caracterização por Vértices Simpliciais

Tem-se várias caracterizações dos grafos triangularizados baseadas nas suas propriedades estruturais, as quais permitem desenvolver algoritmos particulares para esta classe de grafos, para seu reconhecimento e para os problemas clássicos de otimização.

#### Definição III.3:

Um vértice  $v \in V$  é *simplicial* se o subgrafo induzido por  $\text{Adj}(v)$  for completo. Por exemplo, na figura (III.3),  $a$  é simplicial e  $d$  não é simplicial.

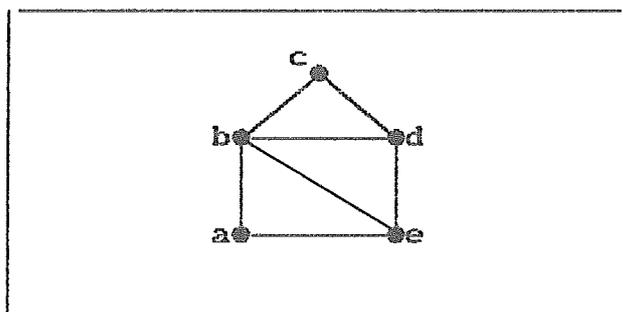


Figura III.3.

#### Propriedade III.2:

Seja  $G$  triangularizado. Se  $G$  é completo, então qualquer um de seus vértices é simplicial. Se  $G$  não é completo, então  $G$  possui um par de vértices não adjacentes que são simpliciais.

#### Definição III.4:

Sejam  $v, w \in V$  tais que  $(v, w) \in E$ ,  $S \subseteq V$ ,  $S$  é um  $v$ - $w$  separador se  $v$  e  $w$  pertencem a componentes conexas distintas de  $G-S$ .

Se  $S$  não contiver  $S' \subseteq S$  tal que  $S'$  é  $v$ - $w$  separador, então  $S$  é  $v$ - $w$  separador minimal.

No exemplo da figura (III.3),  $S = \{b, e\}$  é  $a$ - $c$  separador minimal, porque o grafo é biconexo.

**Propriedade III.3:**

Seja  $G$  grafo não completo conexo então  $G$  é triangularizado se e somente se todo separador minimal induz uma clique.

**Definição III.5:**

Dado  $G = (V, E)$  com  $|V| = n$ , uma ordenação de  $V$  é uma bijeção  $s: \{1, 2, \dots, n\} \rightarrow V$ , tal que a  $i$  associa  $s(i) = v_{si}$ ,  $i = s^{-1}(v)$  indica a posição de  $v$  na ordenação.

A ordenação escreve-se como a sequência  $s = v_{s1}, v_{s2}, \dots, v_{sn}$ .

Denota-se  $G_s = (G, s)$  ao grafo ordenado associado a  $G$ .

Para  $v \in V$  define-se o conjunto dos vértices monotonamente adjacentes a ele:

$$MAdj(v) = Adj(v) \cap \{w / s^{-1}(w) > s^{-1}(v)\}$$

ou equivalentemente

$$MAdj(v) = \{w \in Adj(v) / w \text{ está à direita de } v \text{ em } S\}.$$

**Definição III.6:**

O grafo ordenado  $G_s$  é monótono transitivo se  $\forall v \in V$ :  
 $w \in MAdj(v)$ ,  $z \in MAdj(v) \Rightarrow w \in Adj(z)$ .

**Propriedade III.4:**

$G$  é triangularizado se e somente se existe uma ordenação  $s$  de  $V$  tal que  $G_s$  é monótono transitivo.

**Definição III.7:**

Seja  $S = v_1, v_2, \dots, v_n$  uma sequência de ordenação dos vértices de  $G$ .  $S$  é um *esquema de eliminação perfeita* (EEP), se  $\forall v_i \in V$ , o subgrafo induzido por  $\text{Adj}(v_i) - \{v_1, \dots, v_{i-1}\}$  é completo, ou seja,  $v_i$  é um vértice simplicial de  $G - \{v_1, \dots, v_{i-1}\}$ .

**Observação III.2:**

$$\text{Adj}(v_i) - \{v_1, \dots, v_{i-1}\} = M\text{Adj}(v_i)$$

Os esquemas de eliminação perfeita dão uma caracterização muito importante dos grafos triangularizados, a qual será utilizada no algoritmo de reconhecimento apresentado na seção (III.6).

**Propriedade III.5:**

$S$  é um esquema de eliminação perfeita para  $G$  se e somente se  $G_s$  é monótono transitivo.

**Corolário III.1:**

$G$  é triangularizado se e somente se  $G$  possuir um esquema de eliminação perfeita. Mais ainda, qualquer vértice simplicial pode iniciar um esquema de eliminação perfeita.

No exemplo da figura (III.3)  $c a b d e$  é um EEP.

Os grafos triangularizados têm sido caracterizados também com respeito à aciclicidade de hipergrafos. Considera-se a função  $f$  que, associa cada grafo  $G$  ao hipergrafo  $f(G)$  de suas cliques maximais. D'ATRI e MOSCARINI (1988) apresentam o resultado:  $G$  é cordal se e somente se  $f(G)$  é  $\alpha$ -acíclico.

### III.3. Grafo Triangularizado é um Grafo Interseção

Chama-se *grafo subárvore* ao grafo interseção de uma família de subárvores de uma árvore, isto é, grafo subárvore é o grafo obtido a partir de uma árvore e um conjunto de subárvores dela, associando-se a cada subárvore um vértice e existindo uma aresta entre dois vértices se as subárvores correspondentes compartilham um vértice.

Se nenhuma das subárvores está contida em alguma outra, o grafo chama-se *grafo subárvore próprio*.

GRAVRIL (1974) provou que um grafo  $G$  é cordal se e somente se  $G$  é um grafo subárvore. Mais ainda, é um grafo subárvore próprio.

A árvore  $T$  associada a  $G$  é  $T=(\mathcal{K}, \mathcal{V})$ , onde  $\mathcal{K}$  é o conjunto de cliques maximais de  $G$  e  $\mathcal{V}$  é tal que cada um dos subgrafos induzidos  $T_{K_v}$ ,  $v \in V$ , é conexo, onde  $K_v$  consiste de todas as cliques maximais que contêm  $v$ .  $T$  chama-se *árvore característica* de  $G$ .

Esta caracterização implicitamente usa um resultado de FULKERSON e GROSS (1965) que diz que todo grafo triangularizado tem no máximo  $n$  cliques maximais.

A idéia do *algoritmo sequencial* para determinar a árvore característica é: dado o grafo e seu conjunto de cliques maximais, construir a árvore característica  $T$  com um processo recursivo, associando a cada clique maximal de  $G$  um

vértice na árvore, e a cada vértice  $v$  de  $G$  uma subárvore  $T_{Kv}$  em  $T$ ; as arestas de  $T$  são determinadas de maneira que os subgrafos induzidos  $T_{Kv}$  sejam conexos. O algoritmo sequencial é apresentado com o número (III.1).

**Algoritmo III.1: ÁRVORE CARACTERÍSTICA DE  $G$**

- Dado  $G=(V,E)$  triangularizado
- se  $G$  é completo então  $T$  é um só vértice  $A$
- caso contrário efetuar
  - se  $G$  é desconexo então
  - /existe  $T_i$  associado a cada componente conexa  $G_i$  de  $G$ /
    - aplica-se o algoritmo a cada  $G_i$
    - liga-se um ponto de  $T_i$  com um de  $T_{i+1}$
  - caso contrário efetuar
    - determinar  $v$  vértice simplicial
    - $A=\langle v \rangle \cup \text{Adj}(v)$
    - $U=\langle u \in A / \text{Adj}(u) \subseteq A \rangle$
    - $Y=A-U$
    - $V'=V-U$
    - $G'=G_{V'}$
    - aplicar o algoritmo a  $G'$  /subárvore  $T'$ /
    - determinar  $B$  clique maximal de  $G'$  tal que  $Y \subseteq B$
    - se  $B=Y$  então efetuar
      - $T$  é obtido de  $T'$  trocando  $B$  por  $A$
      - $K=K'+\langle A \rangle - Y$
      - $K_y=K'_y+\langle A \rangle - \langle B \rangle \quad \forall y \in Y$
    - caso contrário efetuar
      - $T$  é obtido de  $T'$  ligando o novo vértice  $A$  a  $B$
      - $K=K'+\langle A \rangle$
      - $K_y=K'_y+\langle A \rangle \quad \forall y \in Y$
  - $K_u=\langle A \rangle \quad \forall u \in U$
  - $K_v=K'_v \quad \forall v \in V-A$
- devolver  $T, K_v \quad \forall v \in V-A$

A complexidade para este algoritmo é:  $O(n^2)$ .

## Observação III.3:

No algoritmo (III.1) definem-se as seguintes variáveis:

Os conjuntos  $A$  são os vértices de  $T$  associados às cliques maximais de  $G$ .

Os  $K_v$ ,  $\forall v \in V$  induzem as subárvores de  $T$  associadas a cada  $v$  de  $V$ .

$K$  corresponde ao conjunto de cliques maximais de  $G$ .

Os elementos de  $U$  são os vizinhos de  $v$  que somente pertencem à clique  $A$  determinada por ele.

Os elementos de  $Y$  são vértices adjacentes a  $v$  que pertencem a  $A$  e a outras cliques maximais, que serão determinadas em chamadas posteriores do algoritmo.

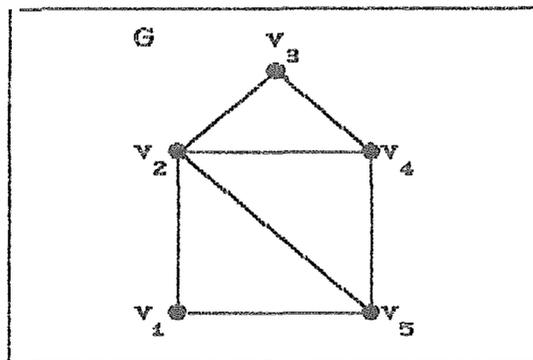


Figura III.4.

Para o grafo  $G$  da figura (III.4) tem-se os seguintes resultados ao aplicar o algoritmo (III.1):

$G$  cordal conexo não completo

Cliques maximais de  $G$ :  $C_1 = \langle v_2, v_3, v_4 \rangle$   
 $C_2 = \langle v_2, v_4, v_5 \rangle$   
 $C_3 = \langle v_1, v_2, v_5 \rangle$

$v_3$  vértice simplicial

$A_1 = \langle v_2, v_3, v_4 \rangle$ ,  $U_1 = \langle v_3 \rangle$ ,

$Y_1 = \langle v_2, v_4 \rangle$ ,  $V_1 = \langle v_1, v_2, v_4, v_5 \rangle$

$T_1$  : aplica-se o algoritmo a  $G_{V_1}$

$v_4$  vértice simplicial

$$A_2 = \langle v_2, v_4, v_5 \rangle, \quad U_2 = \langle v_4 \rangle,$$

$$Y_2 = \langle v_2, v_5 \rangle, \quad V_2 = \langle v_1, v_2, v_5 \rangle$$

$T_2$  : aplica-se o algoritmo a  $G_{V_2}$

$G_{V_2}$  é completo:  $K^2 = V_2, T_2: A_3 \bullet$

$$K_v^2 = V_2 \quad \forall v \in \langle v_1, v_2, v_5 \rangle$$

volta de  $T_2$  a  $G_{V_1}$ :

$$B_2 = \langle v_1, v_2, v_5 \rangle$$

$T_1$  é obtido de  $T_2: A_3 \bullet \text{---} \bullet A_2$

$$K^1 = K^2 + \langle A_2 \rangle = V_2 + \langle A_2 \rangle$$

$$K_{v_2}^1 = K_{v_2}^2 + \langle A_2 \rangle = V_2 + \langle A_2 \rangle$$

$$K_{v_5}^1 = K_{v_5}^2 + \langle A_2 \rangle$$

$$K_{v_4}^1 = \langle A_2 \rangle$$

$$K_{v_1}^1 = K_{v_1}^2 = V_2$$

volta de  $T_1$  a  $G$ :

$$B_1 = \langle v_2, v_4, v_5 \rangle$$

$T$  é obtido de  $T_1: A_3 \bullet \text{---} \overset{A_2}{\bullet} \text{---} \bullet A_1$

$$K = K^1 + \langle A_1 \rangle = V_2 + \langle A_2 \rangle + \langle A_1 \rangle = \langle A_1, A_2, A_3 \rangle$$

$$K_{v_1} = K_{v_1}^1 = V_2 = \langle A_3 \rangle$$

$$K_{v_2} = K_{v_2}^1 + \langle A_1 \rangle = \langle A_1, A_2, A_3 \rangle$$

$$K_{v_3} = \langle A_1 \rangle$$

$$K_{v_4} = K_{v_4}^1 + \langle A_1 \rangle = \langle A_1, A_2 \rangle$$

$$K_{v_5} = K_{v_5}^1 = \langle A_2, A_3 \rangle$$

A árvore característica é dada na figura (III.5).

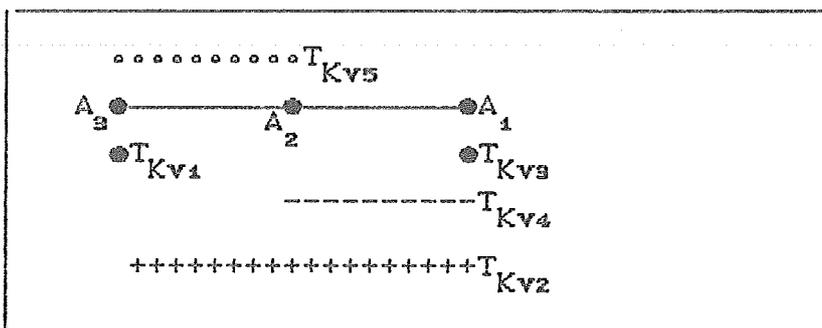


Figura III.5.

BUNEMAN (1974) apresenta um algoritmo sequencial baseado na propriedade que um grafo cordal  $G$  é o grafo interseção de um conjunto de subárvores de uma árvore  $T$ , cujos vértices correspondem às cliques maximais de  $G$ , de forma tal que se  $\bar{c}_i$  é um vértice de  $T$  associado à clique maximal  $C_i$  de  $G$ , então a subárvore  $T_v$  correspondente a  $v$  em  $G$ , contém  $\bar{c}_i$  se e somente se  $C_i$  contém  $v$ .

SHIBATA (1988) desenvolve outro algoritmo sequencial e caracteriza os grafos triangularizados conexos definindo uma *clique simplicial*, que é uma clique maximal que contém um vértice simplicial e *clique fronteira*, que é uma clique simplicial que contém todos os vértices simpliciais ou os vértices dela que não são simpliciais, estão contidos em outra clique maximal.

NAOR et alii (1987) desenvolveram um *algoritmo paralelo* para determinar a árvore característica de um grafo cordal usando um método de *divisão e conquista* para dividir a árvore e o grafo em forma tal que, o número de cliques maximais diminua em cada passo num fator constante. Para isto é necessário um vértice denominado *centróide*.

Numa árvore, a eliminação de um vértice interno dela produz uma separação da árvore numa floresta com árvores de diferentes tamanhos; um *centróide* é um vértice tal que quando é eliminado produz uma floresta onde o tamanho da maior árvore é minimizado.

Dada uma clique maximal  $C$  em  $G$ , o vértice associado em  $T$  denota-se  $\bar{c}$ .

### Propriedade III.6:

Se duas cliques maximais  $C_1$  e  $C_2$  associadas a vértices

que estão em diferentes componentes conexas de  $T - \bar{c}$ , compartilham um vértice  $v$ , então  $v \in C$ .

*Propriedade III.7:*

Ao remover uma clique maximal  $C$  associada a um vértice que não é folha em  $T$ ,  $G-C$  é desconectado.

*Propriedade III.8:*

Se o grafo  $G$  é separado por uma clique maximal  $C$  em componentes conexas  $G_1, \dots, G_s$ , então é possível construir a árvore característica  $T$  de  $G$  em forma tal que, para cada  $i$ , as cliques maximais de  $G_i \cup C$  correspondem aos vértices de uma componente da floresta  $T - \bar{c}$  mais um vértice associado a  $C$ .

A idéia do algoritmo é proceder em forma recursiva dividindo o problema em subproblemas que contêm no máximo a metade mais uma das cliques maximais do problema original, determinando um centróide na árvore. A recursão termina quando tem-se um grafo com duas cliques maximais com interseção não vazia e cuja árvore são dois vértices ligados.

O problema é determinar um centróide, para isto procede-se em paralelo com toda clique maximal  $C$ , contando o número de cliques maximais que estão em  $G_i \cup C$ , onde  $G_i$  é uma componente conexa de  $G-C$ . Determina-se o centróide como o vértice associado àquela clique  $C$  que minimiza o maior número de cliques maximais numa componente  $G_i \cup C$ .

O algoritmo para determinar a árvore característica  $T$  de um grafo cordal  $G$ , em forma paralela, é:

**Algoritmo III.2: ALGORITMO PARALELO PARA ÁRVORE CARACTERÍSTICA**

- Dado  $G=(V,E)$  grafo cordal
- determinar  $c_1, c_2, \dots, c_r$  cliques maximais de  $G$
- se  $r=2$  e  $C_1 \cap C_2 \neq \emptyset$  então  
 devolver a árvore  $T = \{(\bar{c}_1, \bar{c}_2), (\bar{c}_1, \bar{c}_2)\}$
- caso contrário efetuar
  - para toda clique maximal  $C_k$  de  $G$  efetuar em paralelo
    - determinar as componentes conexas  $G_1, \dots, G_s$  de  $G - C_k$
    - para todo  $i=1$  até  $s$  efetuar
      - $N_{ki}$  = número de cliques maximais contidas em  $G_i \cup C_k$
      - determinar  $N_k = \max\{N_{ki} / 1 \leq i \leq s\}$
  - determinar  $j$  tal que  $N_j = \max\{N_k / 1 \leq k \leq r\}$
  - centróide  $= \bar{c}_j$
  - para toda componente conexa  $G_i$  de  $G - C_j$  efetuar
    - aplicar o algoritmo a  $G_i \cup C_j$  obtendo a árvore  $T_i$
- devolver  $T = \bigcup_{i=1}^s T_i$

A complexidade deste algoritmo está dada pela determinação das componentes conexas, o qual é feito num tempo  $O(\log^2 n)$  usando  $O(n^3)$  processadores.

### III.4. Grafo Triangularizado é Perfeito

**Propriedade III.9:**

Todo grafo triangularizado é perfeito.

Todo grafo co-triangularizado é perfeito.

### Prova

A idéia da prova é dar uma coloração nos vértices e provar que ela é ótima, ou seja, que o número máximo de cores usadas é igual ao número cromático e logo, provar que este é igual ao tamanho da maior clique.

Primeiro, dado  $G=(V,E)$ , ordenam-se linearmente os vértices do grafo cordal, em forma inversa à ordem obtida por um esquema de eliminação perfeita.

Depois, dado um subgrafo induzido  $F$  de  $G$ , associa-se a cada vértice  $w$  de  $F$  um inteiro positivo  $f(w)$  com a seguinte regra:  $f(w)$  é o menor inteiro positivo distinto de todos os  $f(v)$  para todo  $v$  em  $F$ , com  $v < w$  e  $(v,w)$  em  $F$ .

Tem-se que  $f$  é uma coloração, pois  $f(v) \neq f(w)$  se  $(v,w) \in E$ . Somente falta provar que existe uma clique de tamanho igual ao número de Grundy de  $(F, <)$ . Seja  $k$  este número e seja  $w$  tal que  $f(w) = k$  então, existem  $k-1$  vértices em  $F$  adjacentes a  $w$  com  $v_i < w$ ,  $f(v_i) = i$  para  $i=1,2, \dots, k-1$ . Mas  $w$  é um vértice simplicial, logo  $(v_i, v_{i+1})$  é uma aresta em  $F$ , para  $i=1,2, \dots, k-2$ , isto é, o conjunto  $\{v_1, \dots, v_{k-1}, w\}$  é uma clique em  $F$ .

Logo  $\omega(F) = k$ .

Pela equação (II.11)  $\chi(F) \leq k$

Sempre para qualquer grafo tem-se  $\omega(F) \leq \chi(F)$ .

Juntando estas igualdades e desigualdades tem-se  $k = \omega(F) \leq \chi(F) \leq k$ , logo,  $\omega(F) = \chi(F) = k$ .

A prova para os grafos co-triangularizados é similar, considerando nos vértices a ordem dada por um esquema de eliminação perfeita do grafo complemento.

### III.5. Grafo Triangularizado é Perfeitamente Ordenável

Os grafos triangularizados têm as seguintes importantes propriedades:

#### *Propriedade III.10:*

Se  $G$  é triangularizado então existe uma ordem linear  $<$  em  $V$  tal que:

Não existem  $x, y, z \in V$  com  $x < y < z$  e  $(x, z), (y, z) \in E$ , mas  $(x, y) \notin E$ .

Esta propriedade é uma generalização sugerida por Lovász ao trabalho de GYÁRFÁS e LEHEL (1970), e permite caracterizar os grafos cordais:

Dado  $G=(V,E)$  com  $|V| \geq 3$  e  $|E| > 2$ ,  $G$  é triangularizado se e somente se seus vértices podem ser ordenados em forma linear tal que, nenhum subgrafo induzido com vértices  $x, y$  e  $z$  e arestas  $(x, z), (y, z)$ , tem-se  $x < y < z$ .

A ordem nos vértices é obtida ao posicionar um vértice simplicial  $v$ , último na ordem, considerar  $G-v$  e continuar em forma recursiva, levando em consideração que, ser cordal é uma propriedade de hereditariedade.

#### *Propriedade III.11:*

A ordem linear dos vértices obtida a partir do inverso de um esquema de eliminação perfeita, é uma ordem linear perfeita.

*Prova*

Seja  $f(G)$  = número de Grundy de  $G$  com ordem  $\prec$ .

Em geral  $\omega(G) \leq \chi(G) \leq f(G)$ .

Se  $f(G) = k$ , então existe  $v \in V$  tal que  $f(v) = k$  e existe um subconjunto de  $V$ , digamos  $\{w_1, w_2, \dots, w_{k-1}\}$ , tal que  $f(w_i) = i$  e  $(w_i, v) \in E$ , (caso contrário  $f(v) < k$ ), ou seja, existem  $k-1$  vértices de  $V$  ligados a  $v$  e coloridos com  $k-1$  cores distintas, mas  $v$  é vértice simplicial, então todos os  $w_i$  estão ligados entre si, logo  $\{w_1, \dots, w_{k-1}, v\}$  é uma clique de tamanho  $k$  e assim  $\omega(G) = f(G)$ , logo  $f(G) = \chi(G)$ , isto é, a ordem é perfeita.

*Propriedade III.12:*

Os grafos co-triangularizados são perfeitamente ordenáveis.

*Prova:*

$G$  é co-triangularizado se e somente se  $\bar{G}$  é triangularizado.

Em  $\bar{G}$  considere-se a ordem inversa de um esquema de eliminação perfeita, isto é,  $v_1, v_2, \dots, v_n$  onde  $v_i$  é um vértice simplicial de  $G - \{v_n, v_{n-1}, \dots, v_{i+1}\}$ .

Em  $G$  considera-se a ordem contrária de  $\bar{G}$ , ou seja, a ordem dada por um esquema de eliminação perfeita de  $\bar{G}$ .

Continua-se em forma similar à prova da propriedade (II.11), considerando que esta ordem satisfaz em  $G$  a seguinte propriedade: se  $G$  é co-triangularizado, então existe uma ordem em  $V$  tal que não existem  $x, y, z \in V$  com  $x \prec y \prec z$  e  $(x, y), (x, z) \notin E$ , mas  $(y, z) \in E$ .

### III.6. Reconhecimento de Grafos Triangularizados

Até agora não se sabe se um grafo perfeito pode ser reconhecido ou não em tempo polinomial. Sabe-se somente que decidir se um grafo é imperfeito é um problema da classe  $\mathcal{NP}$ , segundo LOVÁSZ (1983). Mas os grafos triangularizados podem ser reconhecidos em tempo polinomial, o qual foi provado por ROSE et alii (1976), que desenvolveram um algoritmo sequencial para este objetivo.

#### III.6.1. Algoritmo Sequencial de Reconhecimento

A idéia é aproveitar os seguintes fatos: ser cordal é uma propriedade de hereditariedade e, todo grafo cordal tem pelo menos um vértice simplicial.

O algoritmo resume-se em: pegar um vértice simplicial  $v_1$ . Se  $G$  é cordal, então  $G - v_1$  é cordal. Repetir até que não haja mais vértices simpliciais, ou até que não haja mais vértices a serem retirados.

Surgem os seguintes problemas: se o processo esgotar todos os vértices,  $G$  é cordal? E, como determinar uma forma eficiente de localizar vértices simpliciais? Para isto faz-se uso do seguinte lema:

##### Lema III.1:

Num grafo cordal, a sequência de vértices ordenados decrescentemente segundo a largura definida por uma busca em largura lexicográfica, é um esquema de eliminação perfeita.

O processo de reconhecimento de um grafo triangularizado fica do seguinte modo:

Aplicar busca em largura lexicográfica, obtém-se uma sequência  $S$  dos vértices ordenados em forma decrescente segundo a largura deles.

Aplicar a  $S$  um processo de reconhecimento de esquema de eliminação perfeita. Se  $S$  o for, então  $G$  é cordal, senão  $G$  não é cordal.

Quando a busca em largura lexicográfica é utilizada para construir uma sequência de ordenação dos vértices do grafo e reconhecer se este é cordal ou não, não é preciso fazer todos os passos do algoritmo (II.2), já que não é necessário marcar os vértices e obter a árvore geradora para percorrer todos os vértices. O algoritmo resultante é:

**Algoritmo III.3: BUSCA EM LARGURA LEXICOGRÁFICA PARA RECONHECIMENTO DE GRAFOS CORDAIS**

- Dado  $G(V,E)$  conexo e  $|V|=n$
- para todo  $v \in V$  efetuar
  - $R(v) = \emptyset$
- para  $j = n, n-1, \dots, 1$  efetuar
  - escolher  $v \in V$  com  $S^{-1}(v)$  não definido e  $R(v)$  lexicograficamente máximo
  - $S(j) = v$
  - para  $w \in \text{Adj}(v)$  com  $S^{-1}(w)$  não definido efetuar
    - $R(w) = R(w) \cup \{j\}$
- devolver  $S$ .

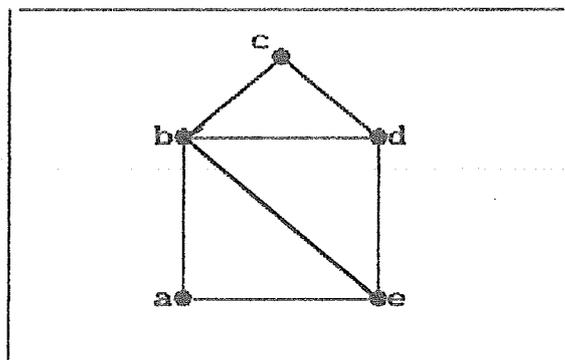


Figura III.6.

Por exemplo, para o grafo da figura (III.6), a

aplicação do algoritmo (III.3) apresenta os seguintes resultados:

$$R(a) = R(b) = R(c) = R(d) = R(e) = \emptyset.$$

$$j = 5: v=a, S^{-1}=(5, \dots, \dots), S=(\dots, \dots, a) \\ R(b) = \langle 5 \rangle, R(e) = \langle 5 \rangle$$

$$j = 4: v=b, S^{-1}=(5, 4, \dots, \dots), S=(\dots, \dots, b, a) \\ R(c) = \langle 4 \rangle, R(d) = \langle 4 \rangle, R(e) = \langle 5, 4 \rangle$$

$$j = 3: v=e, S^{-1}=(5, 4, \dots, 3), S=(\dots, e, b, a) \\ R(d) = \langle 4, 3 \rangle$$

$$j = 2: v=d, S^{-1}=(5, 4, \dots, 2, 3), S=(\dots, d, e, b, a) \\ R(c) = \langle 4, 2 \rangle$$

$$j = 1: v=c, S^{-1}=(5, 4, 1, 2, 3), S=(c, d, e, b, a)$$

*Processo de Reconhecimento de um  
Esquema de Eliminação Perfeita*

Dada uma ordem dos vértices  $S = v_1, v_2, \dots, v_n$ , o processo de reconhecimento pode ser feito segundo a definição pela "força bruta", isto é, pegar cada vértice  $v_i$  da sequência  $S$  e verificar se ele é simplicial de  $G - \langle v_1, v_2, \dots, v_{i-1} \rangle$ . Se é obtido um vértice que não é simplicial, então o processo para, e a sequência não é um esquema de eliminação perfeita. Assim temos o algoritmo:

*Algoritmo III.4: VERIFICAÇÃO ESQUEMA DE ELIMINAÇÃO PERFEITA*

- Dados  $G, S$  e  $n = |V|$
- se  $v_1$  não é vértice simplicial de  $G$  então  
  devolver Falso
- para  $j = 2, \dots, n$  efetuar

•se  $v_j$  não é vértice simplicial de  $G - \{v_1, \dots, v_{j-1}\}$   
 então devolver Falso  
 •devolver Verdadeiro.

No pior caso a complexidade do algoritmo é  $O(mn)$ .

Aproveitando a ordenação  $S$  dos vértices e considerando os conjuntos dos vértices monotonamente adjacentes  $MAdj(v_j)$ , que são os conjuntos  $\{v \in Adj(v_j) / v \text{ está à direita de } v_j \text{ em } S\} \forall v_j \in V$ , é possível melhorar o algoritmo.

A idéia deste algoritmo é: para cada vértice escolher o primeiro vértice adjacente a ele e que está posterior a ele na sequência e controlar os vértices adjacentes deste último, os quais devem estar todos ligados. Este algoritmo apresenta-se a seguir:

#### Algoritmo III.5: ESQUEMA DE ELIMINAÇÃO PERFEITA

•Dados  $G$ ,  $S$  e  $n = |V|$   
 •para  $j = 1, \dots, n$  efetuar  
   • $L(v_j) = \emptyset$   
 •para  $j = 1, \dots, n$  efetuar  
   • $v_j = S(j)$   
   •determinar  $v_k \in MAdj(v_j)$  mais próximo a  $v_j$  em  $S$   
   • $L(v_k) = L(v_k) \cup [MAdj(v_j) - \{v_k\}]$   
 •para  $j = 1, \dots, n$  efetuar  
   •se  $L(v_j) - Adj(v_j) \neq \emptyset$  então  
     •devolver Falso  
 •devolver Verdadeiro.

A complexidade do algoritmo reduz-se a  $O(n+m)$ .

#### Observação III.4:

Se depois que os conjuntos  $L(v_j)$  estiverem todos

construídos e  $L(v_j) - \text{Adj}(v_j) \neq \emptyset$  para algum  $v_j$ , então  $v_j$  não é vértice simplicial de  $G - \langle v_1, v_2, \dots, v_{j-1} \rangle$ .

No exemplo do grafo da figura (III.6) foi obtida a sequência  $S = (c, d, e, b, a)$ , aplicando o algoritmo (III.5) para verificar se  $S$  é um esquema de eliminação perfeita, ocorreram os seguintes passos:

$$L(a) = L(b) = L(c) = L(d) = L(e) = \emptyset$$

$$j = 1: M\text{Adj}(c) = \langle b, d \rangle, L(d) = \langle b \rangle$$

$$j = 2: M\text{Adj}(d) = \langle b, e \rangle, L(e) = \langle b \rangle$$

$$j = 3: M\text{Adj}(e) = \langle a, b \rangle, L(b) = \langle a \rangle$$

$$j = 4: M\text{Adj}(b) = \langle a \rangle, L(a) = \emptyset$$

$$j = 5: M\text{Adj}(a) = \emptyset$$

$$j = 1: \text{Adj}(c) = \langle b, d \rangle$$

$$j = 2: \text{Adj}(d) = \langle b, c, e \rangle$$

$$j = 3: \text{Adj}(e) = \langle a, b, d \rangle$$

$$j = 4: \text{Adj}(b) = \langle a, c, d, e \rangle$$

$$j = 5: \text{Adj}(a) = \langle b, e \rangle$$

Verdadeiro.

Logo  $S$  é um esquema de eliminação perfeita e  $G$  é cordal.

### III.6.2. Algoritmos Paralelos de Reconhecimento

Com o novo e crescente desenvolvimento dos computadores paralelos, tem-se identificado muitos algoritmos que podem ser implementados realizando várias operações em forma simultânea, o que resulta na solução dos problemas em uma forma mais rápida, a complexidade de tempo é menor mas deve considerar-se o número de processadores necessários para realizar o processo completo.

O algoritmo que constrói um esquema de eliminação

perfeita e aquele que o reconhece, parecem difíceis de serem executados com processadores em paralelo, porque cada vez é escolhido um vértice dependendo do que aconteceu até esse momento.

NAOR et alii (1987) apresentam algoritmos paralelos com leitura concorrente e escrita exclusiva numa máquina PRAM ("Parallel Random Access Machine"), para o reconhecimento e para vários problemas em grafos triangularizados, baseados no estudo dos conjuntos de adjacências e na caracterização como grafo de interseção de subárvores.

CHANDRASEKHARAN e IYENGAR (1988) desenvolveram um algoritmo paralelo com leitura e escrita concorrente numa máquina PRAM, para o reconhecimento de grafos cordais, baseado nas cliques separadoras.

KLEIN (1988) desenvolveu um algoritmo paralelo eficiente para determinar um esquema de eliminação perfeita, em uma máquina PRAM com leitura e escrita concorrente, com um menor número de processadores.

#### *Algoritmo de Naor*

O algoritmo de Naor et alii baseia-se na seguinte caracterização:

#### *Propriedade III.13:*

$G$  não é cordal se e somente se para algum vértice  $v$ , uma componente conexa do grafo  $(G-v)-Adj(v)$  é adjacente a pelo menos dois vértices  $v_i$  e  $v_j \in Adj(v)$  tais que  $v_i$  e  $v_j$  não são adjacentes.

A idéia do algoritmo é verificar se para cada vértice é satisfeita a condição da propriedade (III.13), fazendo-o

para todos os vértices em forma simultânea. Se algum deles satisfaz a condição, então o grafo não é cordal.

*Algoritmo III.6: RECONHECIMENTO DE NAOR PARA UM GRAFO CORDAL*

- Dado  $G=(V,E)$  conexo
- para todo  $v \in V$  efetuar em paralelo
  - determinar todas as componentes conexas de  $G-v-Adj(v)$
- para todo  $v \in V$  efetuar em paralelo
  - se uma componente conexa de  $G-v-Adj(v)$  é adjacente a pelo menos dois vértices de  $Adj(v)$  que não são adjacentes entre si então devolver  $G$  não é cordal
- devolver  $G$  é cordal.

A complexidade do algoritmo depende da determinação em paralelo das componentes conexas dos  $|V|$  subgrafos. Isto pode ser feito no tempo  $\mathcal{O}(\log^2 n)$  com  $\mathcal{O}(n^2)$  processadores para cada subgrafo, logo o algoritmo de reconhecimento tem complexidade de tempo  $\mathcal{O}(\log^2 n)$  com  $\mathcal{O}(n^3)$  processadores.

Ao aplicar o algoritmo (III.6) ao grafo da figura (III.6), tem-se:

*Passo 1 (em paralelo):*

- $v = a$ : componente conexa =  $\{b, c, d\}$
- $v = b$ : componente conexa =  $\emptyset$
- $v = c$ : componente conexa =  $\{a, b, e\}$
- $v = d$ : componente conexa =  $\{a\}$
- $v = e$ : componente conexa =  $\{c\}$

*Passo 2 (em paralelo):*

para nenhum  $v \in V$  existe uma componente conexa de  $G-v-Adj(v)$  adjacente a dois vértices de  $Adj(v)$  que não sejam adjacentes entre eles. Logo  $G$  é cordal.

*Algoritmo de Chandrasekharan e Iyengar*

O algoritmo de Chandrasekharan e Iyengar estuda para cada par de vértices não adjacentes, os conjuntos separadores deles, os quais devem ser cliques se o grafo é cordal. Isto é feito para todos os pares analisados, em forma paralela. O algoritmo é:

*Algoritmo III.7: RECONHECIMENTO DE CHANDRASEKHARAN  
PARA UM GRAFO CORDAL*

- Dado  $G=(V,E)$  conexo
- se  $G$  é uma clique então devolver  $G$  é CORDAL
- caso contrário efetuar
  - para todo par de vértices  $u,v$  não adjacentes  
efetuar em paralelo
    - $H= G-Adj(u)$
    - $F=$  componente conexa de  $H$  contendo  $v$
    - $M= \emptyset$
    - para todo  $x \in Adj(u)$  e  $y \in F$   
efetuar em paralelo
      - se  $(x,y) \in E$  então  $M = M \cup \{x\}$
      - se  $M$  é clique então devolver 1
      - caso contrário devolver 0
  - se para todo par  $u,v$  não adjacente a  
resposta é 1 então devolver  $G$  é CORDAL
  - caso contrário devolver  $G$  não é CORDAL

A complexidade do algoritmo está dada pelo estudo dos conjuntos separadores dos pares de vértices não adjacentes e a determinação das componentes conexas que é realizado no tempo  $O(\log n)$  com  $O(n+m)$  processadores. Logo o algoritmo de reconhecimento tem complexidade de tempo  $O(\log n)$  com  $O(n^4+n^2m)$  processadores.

Aplicando o algoritmo (III.7) ao grafo  $G$  da figura (III.6) tem-se:  $G$  não é uma clique,

*Passo 1* (em paralelo):

par  $a, c$ :  $H =$  subgrafo induzido por  $\langle a, c, d \rangle$

$F =$  subgrafo induzido por  $\langle c, d \rangle$

$M = \emptyset$

em paralelo  $\begin{cases} (b, c) \in E \text{ então } M = \langle b \rangle \\ (b, d) \in E \text{ então } M = \langle b \rangle \\ (e, c) \notin E \\ (e, d) \in E \text{ então } M = \langle b, e \rangle \end{cases}$

$M$  é uma clique  $\Rightarrow 1$

par  $a, d$ :  $H =$  subgrafo induzido por  $\langle a, c, d \rangle$

$F =$  subgrafo induzido por  $\langle c, d \rangle$

$M = \emptyset$

em paralelo  $\begin{cases} (b, c) \in E \text{ então } M = \langle b \rangle \\ (b, d) \in E \text{ então } M = \langle b \rangle \\ (e, c) \notin E \\ (e, d) \in E \text{ então } M = \langle b, e \rangle \end{cases}$

$M$  é uma clique  $\Rightarrow 1$

par  $c, e$ :  $H =$  subgrafo induzido por  $\langle a, c, e \rangle$

$F =$  subgrafo induzido por  $\langle a, e \rangle$

$M = \emptyset$

em paralelo  $\begin{cases} (b, a) \in E \text{ então } M = \langle b \rangle \\ (b, e) \in E \text{ então } M = \langle b \rangle \\ (d, a) \notin E \\ (d, e) \in E \text{ então } M = \langle b, d \rangle \end{cases}$

$M$  é uma clique  $\Rightarrow 1$

*Passo 2:*

todos os pares devolveram 1 então  $G$  é CORDAL.

### *Algoritmo de Klein*

A idéia do algoritmo de Klein é, dado um conjunto de rótulos inteiros para os vértices do grafo, determinar um esquema de eliminação perfeita consistente com a ordem parcial induzida pela rotulação ou determinar que não existe um esquema de eliminação perfeita consistente. Isto é

realizado de forma iterativa, tal que cada vez a rotulação é melhorada até que cada número é associado somente a um vértice. Nesse instante, é fácil comprovar que a numeração define um esquema de eliminação perfeita. O método assegura que é necessário, no máximo,  $O(\log n)$  iterações.

Para desenvolver o algoritmo paralelo que determina um esquema de eliminação perfeita de um grafo cordal, é generalizada a rotulação dos vértices a uma numeração  $S$  deles que não precisa ser 1-1. Dada uma numeração  $S$  dos vértices, denota-se  $G_s$  ao grafo  $G$  com cada vértice  $v$  rotulado pelo número  $S(v)$ , se  $S(v) > S(w)$  então diz-se que  $v$  é *mais rico* que  $w$ . As classes de  $G_s$  são os subgrafos induzidos por subconjuntos de vértices os quais têm o mesmo rótulo, as *componentes de classes* de  $G_s$  são as componentes conexas das classes de  $G_s$ . Dadas duas numerações  $S$  e  $L$  dos vértices de  $G$ , diz-se que  $S$  é *consistente* com  $L$  se:  $S(v) < S(w) \Rightarrow L(v) < L(w)$ ,  $\forall v, w \in V$ , e diz-se também que  $L$  é uma *refinação* de  $S$ .

Uma *numeração parcial* é uma função que associa números inteiros a um subconjunto próprio dos vértices de  $G$ . Para a numeração  $S$  e a numeração parcial  $P$ , define-se a *refinação* de  $S$  por  $P$ , denotada  $S_p$ , como:  $S_p(v) < S_p(w)$  se  $S(v) < S(w)$  ou  $S(v) = S(w)$ ,  $P(v)$  e  $P(w)$  estão definidos e  $P(v) < P(w)$ . Utiliza-se uma numeração parcial  $P$  de alguma componente de classe  $C$  de  $G_s$ . Ao obter  $S_p$  a partir de  $S$ , fala-se de *estratificar* a componente de classe  $C$  de  $G_s$ , e de *estratificar bem* se cada componente de classe contém no máximo  $(4/5) |C|$  vértices de  $C$ .

Um *caminho para trás* em  $G_s$  é um caminho tal que seus vértices finais são estritamente mais ricos que os vértices internos. Uma numeração  $S$  de  $G$  é *válida* se todos os caminhos para trás em  $G_s$  têm pontos finais adjacentes.

#### **Propriedade III.14:**

Em um grafo  $G$ , se uma rotulação  $S$  é 1-1 então  $S$  é um

esquema de eliminação perfeita de  $G$ .

*Propriedade III.15:*

Seja  $S$  uma numeração válida de  $G$ , para cada componente de classe  $C$  de  $G_s$ , os vizinhos mais ricos de  $G$  constituem uma clique.

O algoritmo paralelo de Klein para reconhecer se um grafo é triangularizado, realiza uma refinação até obter um esquema de eliminação perfeita, para isto é necessário estratificar as componentes de classes.

*Algoritmo III.8: REFINAÇÃO*

- Dado  $G=(V,E)$  conexo
- para todo  $v \in V$  efetuar em paralelo
  - $S(v) = 0$
- $d = 2 \exp(3 \log_{5/4} n)$
- enquanto  $S$  não seja 1-1 efetuar
  - para toda componente de classe  $C$  tal que  $|C| \neq 1$ 
    - efetuar em paralelo
      - chamar ESTRATIFICA( $G_s, C, d$ ) /algoritmo III.9/
  - $d = d/8$
- devolver  $S$ .

*Algoritmo III.9: ESTRATIFICA*

- Dados  $G_s, C$  e  $d$
- para todo  $v$  em  $C$  efetuar em paralelo
  - identificar as arestas que ligam  $v$  a outro vértice de  $C$
  - identificar as arestas que ligam  $v$  a um vértice mais rico
- $B = \langle \text{vizinhos mais ricos de } C \rangle$

- se  $B = \emptyset$  então efetuar
  - $D = \{ v \in C \mid v \text{ tem mais de } (3/5) |C| \text{ vizinhos em } C \}$
  - determinar uma floresta geradora de  $C-D$
  - se alguma componente  $H$  de  $C-D$  tem pelo menos  $(4/5) |C|$  vértices então efetuar
    - $T =$  árvore geradora de  $H$
    - ordenar os vértices de  $H$  em uma ordem consistente com suas distâncias em  $T$ , a partir da raiz:  $v_1, \dots, v_k$
    - para  $j = 1, \dots, k$  efetuar
      - $A_j = \{v_1, \dots, v_j\} \cup \{ \text{vizinhos de } v_t \text{ em } C \mid t = 1, \dots, j \}$
    - determinar em paralelo
      - $M = \max\{ j \mid |A_j| \leq (4/5) |C| \}$
    - para todo  $v \in A_M$  efetuar
      - $S(v) = S(v) + d$
  - caso contrário efetuar
    - se  $D$  é clique então efetuar
      - seja  $D = \{v_1, \dots, v_k\}$
      - para  $i = 1, \dots, k$  efetuar
        - $S(v_i) = S(v_i) + i$
    - caso contrário efetuar
      - determinar  $x$  e  $y$  vértices não adjacentes em  $D$
      - sejam  $v_1, \dots, v_k$  seus vizinhos comuns
      - para  $i = 1, \dots, k$  efetuar
        - $S(v_i) = S(v_i) + i$
- caso contrário efetuar
  - se todos os vértices de  $B$  têm pelo menos  $(2/5) |C|$  vizinhos então efetuar
    - chamar  $\text{ALTOGRAU}(G_s, C, B, d)$  /algoritmoIII.10/
  - caso contrário efetuar
    - chamar  $\text{BAIXOGRAU}(G_s, C, B, d)$  /algoritmoIII.11/
- devolver  $d$ .

*Algoritmo III.10: ALTOGRAU*

- Dados  $G_s, C, B$  e  $d$

- ordenar os vértices de B de forma arbitrária  $v_1, \dots, v_k$
- para  $j = 1, \dots, k$  efetuar
  - $F_j = \{v \in C / v \text{ é adjacente a } v_1, v_2, \dots, v_k\}$
- determinar em paralelo
  - $M = \max \{j / |F_j| \geq (1/5) |C|\}$
- para todo  $v \in F_j$  efetuar
  - $S(v_j) = S(v) + d$
- determinar  $C'$  a componente maior de  $G_{F_j}$
- se  $M = k$  então efetuar
  - $d = d/2$
  - $D = \{v \in C' / v \text{ tem mais de } (3/5) |C| \text{ vizinhos em } C'\}$
  - determinar uma floresta geradora de  $C' - D$
  - se alguma componente  $H$  de  $C' - D$  tem pelo menos  $(4/5) |C'|$  vértices então efetuar
    - determinar  $T$  árvore geradora de  $H$
    - ordenar os vértices de  $H$  em uma ordem consistente com suas distâncias em  $T$ , a partir da raiz:  $v_1, \dots, v_k$
    - para  $j = 1, \dots, j$  efetuar em paralelo
      - $A_j = \{v_1, \dots, v_k\} \cup \{v \text{ vizinhos de } v_t \text{ em } C' / 1 \leq t \leq j\}$
    - determinar em paralelo
      - $M = \max \{j / |A_j| \leq (4/5) |C'|\}$
    - para todo  $v \in A_M$  efetuar
      - $S(v) = S(v) + d$
  - caso contrário efetuar
  - se  $D$  é clique então efetuar
    - seja  $D = \{v_1, \dots, v_k\}$
    - para  $i = 1, \dots, k$  efetuar
      - $S(v_i) = S(v_i) + i$
  - caso contrário efetuar
    - determinar  $x$  e  $y$  de  $D$  não adjacentes
    - sejam  $v_1, \dots, v_k$  seus vizinhos comuns
    - para  $i = 1, \dots, k$  efetuar
      - $S(v_i) = S(v_i) + i$
- devolver  $S$ .

*Algoritmo III.11: BAIIXO GRAU*

- Dados  $G_g$ ,  $C$ ,  $B$  e  $d$
- $D = \{v \in C \cup B / v \text{ tem mais de } (3/5) |C| \text{ vizinhos em } C\}$
- seja  $H$  a componente conexa do subgrafo induzido por  $C \cup B - D$  que contém vértices de  $B$
- determinar uma árvore geradora  $T$  de  $H$  cuja raiz seja um vértice de  $B$
- ordenar os vértices de  $T$  em uma ordem consistente com suas distâncias em  $T$  a partir da raiz:  $v_1, \dots, v_k$
- para  $j = 1, \dots, k$  efetuar
  - $A_j = \{v_1, \dots, v_j\} \cup \{\text{vizinhos de } v_t \text{ em } C / 1 \leq t \leq j\}$
- determinar em paralelo
  - $M = \max \{ j / |A_j \cap C| \leq (4/5) |C| \}$
- para todo  $v \in A_M \cap C$  efetuar
  - $S(v) = S(v) + d$
- determinar  $C'$  a componente maior de  $C - A_M$
- se  $|C'| > (4/5) |C|$  então efetuar
  - chamar  $\text{ESTRATIFICA}(G_g, C', d/2)$
- devolver  $S$ .

A determinação em paralelo do máximo dos índices de uma sequência de conjuntos com uma restrição, é feita pelo procedimento dado por LADNER e FISCHER (1980).

O algoritmo paralelo para determinar uma refinação de uma rotulação dos vértices de um grafo, tem no máximo  $\mathcal{O}(\log n)$  etapas do processo iterativo, o procedimento de estratificação tem complexidade  $\mathcal{O}(\log k)$  usando  $k$  processadores onde  $k$  é o número de arestas que têm pelo menos um vértice final em uma componente, logo ao executá-lo para todas as componentes de classes, a complexidade é  $\mathcal{O}(\log m)$  usando  $\mathcal{O}(m)$  processadores. Logo, a complexidade do algoritmo para determinar um esquema de eliminação perfeita é  $\mathcal{O}(\log^2 n)$  com  $\mathcal{O}(n+m)$  processadores.

Ao aplicar o algoritmo (III.8) ao grafo da figura (III.6), tem-se no começo uma rotulação de todos os vértices

em zero. No primeiro passo são mudados os rótulos dos vértices  $a$ ,  $b$ ,  $e$  e  $d$  ficando no valor  $d$ , logo são incrementados os rótulos de  $a$ ,  $b$  e  $e$  em  $d$ , finalmente o rótulo de  $a$  é incrementado em 1, o de  $b$  em 2 e o de  $e$  em 3. A partir destes valores obtém-se a ordem dos vértices  $c$ ,  $d$ ,  $a$ ,  $b$  e  $e$  segundo de menor a maior valor do rótulo, esta ordem é um esquema de eliminação perfeita.

### III.7. Problemas em Grafos Triangularizados

Existem vários problemas clássicos em grafos, os quais são *NP-completo* para um grafo qualquer. Alguns deles, quando restritos aos grafos cordais e em outras classes especiais de grafos, são *polinomiais*.

Serão desenvolvidos alguns destes problemas e os algoritmos existentes na literatura para grafos cordais. Os quatro problemas clássicos são *clique*, *número cromático*, *conjunto estável* e *cobertura por cliques*. Estes problemas e seus algoritmos são descritos a seguir.

#### III.7.1 Clique

Dado  $G=(V,E)$  determinar  $\omega(G)$  = tamanho da maior clique.

FULKERSON e GROSS (1965) provaram que, dado um esquema de eliminação perfeita  $S$  para  $G$  grafo cordal, toda clique maximal é da forma  $\langle v \rangle \cup M_{\text{Adj}}(v)$ , ou seja, todos os vértices adjacentes a  $v$  e que estão à direita dele na sequência  $S$ .

Outro resultado importante para fazer um algoritmo para determinar  $\omega(G)$  diz que o número de cliques maximais num grafo cordal é menor ou igual ao número de vértices do grafo.

Com isto, é possível modificar o algoritmo (III.4) de

reconhecimento de um esquema de eliminação perfeita para obter as cliques maximais escrevendo  $\{v_j\} \cup \text{MA}_{\text{Adj}}(v_j)$ ,  $\forall v_j \in V$ . Pode-se ter que alguma dessas cliques não seja maximal. Isto vai acontecer no caso em que  $\text{MA}_{\text{Adj}}(v_j)$  seja unido a algum  $L(v_k)$ . Logo, neste caso, essa clique não deve ser considerada.

### III.7.2. Número Cromático

Dado  $G$ , determinar  $\chi(G)$ , o menor número de cores necessárias para colorir os vértices de  $G$  de forma tal que, dois vértices adjacentes possuam cores diferentes.

Dado que, qualquer grafo cordal é perfeito, e que a característica dos grafos perfeitos é a igualdade do número cromático e o tamanho da maior clique em todo subgrafo induzido, então é possível determinar o número cromático com o mesmo algoritmo das cliques maximais.

#### Algoritmo III.12: NÚMERO CROMÁTICO E CLIQUES MAXIMAIS

- Dados  $G$  e  $S$
- $\chi(G) = 1$
- para  $j = 1, \dots, n$  efetuar
  - $L(v_j) = 0$
- para  $j = 1, \dots, n$  efetuar
  - $v = S(j)$
  - se  $\text{Adj}(v) = \emptyset$  então devolver  $\{v\}$
  - caso contrário efetuar
    - $\text{MA} = \langle w \in \text{Adj}(v) / S^{-1}(v) < S^{-1}(w) \rangle$
    - se  $\text{MA} \neq \emptyset$  então efetuar
      - $u = S(\min\{S^{-1}(w) / w \in \text{MA}\})$
      - $L(u) = \max\{L(u), |\text{MA}| - 1\}$
      - se  $L(v) < |\text{MA}|$  então efetuar
        - devolver  $\{v\} \cup \text{MA}$
        - $\chi(G) = \max\{\chi(G), 1 + |\text{MA}|\}$
- devolver  $\chi(G)$ .

A complexidade no pior caso é  $O(n+m)$ .

No grafo da figura (III.6) com  $S = (c,d,e,b,a)$  tem-se:

$$L(a) = L(b) = L(c) = L(d) = L(e) = 0$$

$$\chi(G) = 1$$

$$j = 1: v = c, MA = \langle b, d \rangle \Rightarrow \langle c, b, d \rangle, \chi(G) = 3$$

$$j = 2: v = d, MA = \langle b, e \rangle \Rightarrow \langle d, b, e \rangle, \chi(G) = 3$$

$$j = 3: v = e, MA = \langle b, a \rangle \Rightarrow \langle e, b, a \rangle, \chi(G) = 3$$

$$j = 4: v = b, MA = \langle a \rangle, \chi(G) = 3$$

$$j = 5: v = a, MA = \emptyset, \chi(G) = 3$$

Resultado:  $\chi(G) = \omega(G) = 3$ , cliques maximais  $\langle c, b, d \rangle$ ,  $\langle d, b, e \rangle$  e  $\langle e, b, a \rangle$ .

### III.7.3. Conjunto Estável

Dado  $G = (V, E)$ , determinar o tamanho do maior conjunto estável de  $G$  ( $\alpha(G)$ ).

A solução do problema foi dada por GAVRIL (1972), usando o mesmo algoritmo do número cromático. A idéia é determinar vértices não adjacentes aproveitando os conjuntos de vértices monotonamente adjacentes.

Considerando um esquema de eliminação perfeita  $S$  para  $G$ , define-se a sequência  $y_1 = S(1)$ ,  $y_i$  = primeiro vértice em  $S$  posterior a  $y_{i-1}$  tal que não pertence a  $MAdj(y_1) \cup \dots \cup MAdj(y_{i-1})$  para  $i = 1, \dots, t$  onde  $t$  é tal que todos os vértices após  $y_t$  em  $S$  pertencem a  $MAdj(y_1) \cup \dots \cup MAdj(y_t)$ . Tem-se que  $t = \alpha(G)$  e  $\langle y_1, y_2, \dots, y_t \rangle$  sendo o maior conjunto estável de  $G$ .

No grafo da figura (III.6) constata-se com  $S = (c, d, e, b, a)$ :  $y_1 = c$ ,  $MAdj(y_1) = \langle b, d \rangle$   
 $y_2 = e$ ,  $MAdj(y_2) = \langle a, b \rangle$ .

Resultado: conjunto estável =  $\langle c, e \rangle$  e  $\alpha(G) = 2$ .

### III.7.4. Cobertura por Cliques

Dado  $G$  determinar  $k = \kappa(G)$  como sendo o menor número de cliques maximais  $C_1, C_2, \dots, C_k$  tal que  $G \subseteq C_1 \cup C_2 \cup \dots \cup C_k$ .

Para o caso que  $G$  é cordal (e portanto perfeito), tem-se  $\kappa(G) = \alpha(G)$ . GAVRIL (1972) aproveitando os conjuntos estáveis provou que, dado o maior conjunto estável  $\langle y_1, y_2, \dots, y_t \rangle$ , a coleção de conjuntos  $C_i = \langle y_i \rangle \cup MAdj(y_i)$ ,  $i = 1, 2, \dots, t$ , é a menor cobertura por cliques de  $G$ .

No grafo da figura (III.6)  $\kappa(G) = 2$  e a cobertura por cliques está formada pelas cliques maximais  $C_1 = \langle b, c, d \rangle$  e  $C_2 = \langle a, b, e \rangle$ .

Estes quatro problemas são *NP-completo* para um grafo qualquer, se é considerada apenas a classe dos grafos perfeitos, os quatro problemas podem ser resolvidos em tempo *polinomial* com algoritmos baseados no *método do elipsoide*, desenvolvidos por GROTSCHHELL et alii (1981). Os algoritmos sequenciais desenvolvidos aqui são somente para o caso de grafos cordais, e o tempo fica linear no número de vértices e no número de arestas do grafo, portanto linear no tamanho do grafo.

NAOR et alii (1987) além de desenvolverem um algoritmo paralelo para o reconhecimento de grafos cordais, eles apresentam algoritmos paralelos para os quatro problemas clássicos, baseados em outros elementos desses grafos.

#### **Definição III.8.:**

Um grafo é uma *biclique* se seu conjunto de vértices pode ser particionado em dois subconjuntos  $A$  e  $B$  de forma tal que os subgrafos induzidos  $G_A$  e  $G_B$  são cliques.

Para determinar todas as cliques maximais e calcular o número cromático de um grafo cordal qualquer, Naor et alii reduzem o problema à resolução deste numa biclique cordal pela técnica de *divisão e conquista*. O algoritmo pode ser escrito na seguinte forma geral:

**Algoritmo III.13: ALGORITMO PARALELO PARA CLIQUES MAXIMAIS**

- Dado  $G=(V,E)$  e  $n=|V|$
- se  $n$  par então dividir  $V= A + B$  tal que  $|A|= |B|$ 
  - caso contrário dividir  $V= A+B$  tal que  $||A|-|B||=1$
- determinar todas as cliques maximais de  $G_A$  e  $G_B$
- para toda clique maximal  $P \subseteq A$  efetuar
  - para toda clique maximal  $Q \subseteq B$  efetuar
    - determinar todas as cliques maximais no subgrafo induzido por  $P \cup Q$  que é uma biclique
- eliminar duplicados.

Assim, precisa-se então de um algoritmo para determinar as cliques maximais de uma biclique.

Seja  $G$  um grafo biclique com o conjunto de vértices  $V$  particionado na forma  $V= A + B$  tal que  $G_A$  e  $G_B$  são cliques. Para  $v \in A$  chama-se  $N_B(v) = \langle w \in B / w \in \text{Adj}(v) \rangle$  e para  $x \in B$  denota-se  $N_A(x) = \langle y \in A / y \in \text{Adj}(x) \rangle$ . Tem-se, para estes grafos, a seguinte propriedade:

**Propriedade III.16:**

Os conjuntos  $N_B(v)$ ,  $v \in A$  e  $N_A(x)$ ,  $x \in B$ , são totalmente ordenados por inclusão.

**Propriedade III.17:**

Seja  $Q= Q_A \cup Q_B$  uma clique maximal na biclique cordal,

induzida por  $A$  e  $B$ , com  $Q_A \subseteq A$  e  $Q_B \subseteq B$ ,  $Q_A$  e  $Q_B$  não vazios; então  $Q_A$  é o menor conjunto de  $\langle N_A(x) / x \in Q_B \rangle$  e  $Q_B$  é o menor conjunto de  $\langle N_B(v) / v \in Q_A \rangle$ .

Portanto, o algoritmo é:

*Algoritmo III.14: CLIQUES MAXIMAIS DE UMA BICLIQUE CORDAL*

- Dado  $G=(V,E)$  biclique cordal,  $V=A + B$
- efetuar em paralelo
  - para todo  $v \in A$  determinar  $N_B(v) = \{w \in B / (v,w) \in E\}$
- efetuar em paralelo
  - para todo  $x \in B$  determinar  $N_A(x) = \{y \in A / (x,y) \in E\}$
- ordenar os vértices de  $A$  em ordem lexicográfica não decrescente de  $N_B(v)$
- ordenar os vértices de  $B$  em ordem lexicográfica não decrescente de  $N_A(x)$
- determinar os vértices  $v_1, \dots, v_r$  de  $A$  tais que o seu predecessor na ordem tem uma vizinhança  $N_B(v)$  estritamente contida na sua vizinhança
- determinar os vértices  $x_1, \dots, x_s$  de  $B$  tais que o seu predecessor na ordem tem uma vizinhança  $N_A(x)$  estritamente contida na sua vizinhança
- para todo  $v_i \in A$  selecionado efetuar
  - $Q_A(v_i) = \{v_i\} \cup \{v_j / N_B(v_i) \subseteq N_B(v_j)\}$
  - $Q_B(v_i) = \{N_B(v_i) / N_B(v_i) \subseteq N_B(v_j) \ \forall v_j \in A\}$
  - $Q(v_i) = Q_A(v_i) \cup Q_B(v_i)$
- devolver  $Q$ .

A complexidade deste algoritmo é  $\mathcal{O}(\log n)$  pela ordenação dos vértices, usando  $\mathcal{O}(m)$  processadores. Logo, a complexidade para determinar as cliques maximais de um grafo cordal dependerá das chamadas a este algoritmo. Em cada chamada há  $\mathcal{O}(n^2)$  bicliques, que podem produzir  $\mathcal{O}(n)$  cliques maximais em cada chamada, mas no máximo há  $n$  cliques maximais diferentes. A eliminação de duplicados pela "força bruta" precisa de um tempo  $\mathcal{O}(n)$  usando  $\mathcal{O}(n^5)$  processadores. Então, a complexidade é  $\mathcal{O}(\log^2 n)$  com  $\mathcal{O}(n^5)$  processadores.

Como o grafo é perfeito, então o número cromático pode ser obtido com este mesmo algoritmo, determinando o tamanho da maior clique.

Para os problemas do conjunto estável e da cobertura por clique, os algoritmos paralelos usam a representação do grafo por sua árvore característica.

O algoritmo para determinar o maior conjunto estável procede em etapas, a idéia em cada passo é processar e logo eliminar, em paralelo, todos os ramos terminais da árvore, onde um ramo terminal é um caminho de três vértices com grau dois ou menor e que contem uma folha.

GAVRIL (1972) provou que o primeiro conjunto estável maximal lexicograficamente, com respeito a um esquema de eliminação perfeita, é um conjunto estável máximo; então basta determinar este primeiro conjunto e o problema se reduz a determinar, com um algoritmo paralelo, um esquema de eliminação perfeita, o qual é descrito no algoritmo (III.11). Para cada ramo  $B$ , define-se  $U(B)$  o conjunto dos vértices que não estão em nenhuma clique com vértice associado em  $T$  fora de  $B$ .

*Algoritmo III.15:* ALGORITMO PARALELO PARA DETERMINAR  
ESQUEMA DE ELIMINAÇÃO PERFEITA

- Dado  $G$  cordal
- determinar a árvore característica  $T$  pelo algoritmo (III.3)
- enquanto  $T \neq \emptyset$  efetuar
  - determinar os ramos terminais  $B_1, \dots, B_k$
  - para todo  $B_i$  efetuar em paralelo
    - determinar esquema de eliminação perfeita em  $U(B_i)$
  - anexar os vértices dos esquemas de  $U(B_1), \dots, U(B_k)$
  - anexar este esquema àquele já obtido

•  $T = T - \{v \in B_i / 1 \leq i \leq k\}$   
 • devolver  $T$

A complexidade deste algoritmo é  $O(\log^2 n)$  usando  $O(n^2)$  processadores.

Para determinar a cobertura por cliques, a idéia é a mesma que aquela usada por Gavril, só que neste caso, é aproveitado o conjunto estável máximo determinado com o algoritmo paralelo. Se  $y_1, y_2, \dots, y_t$  é um conjunto estável máximo determinado a partir de uma ordem nos vértices dado por um esquema de eliminação perfeita, então os conjuntos  $\{y_i\} \cup \text{MAdj}(y_i)$  para  $i = 1, \dots, t$ , constituem uma cobertura mínima por cliques para o grafo.

KLEIN (1988) desenvolveu algoritmos paralelos para resolver os problemas da clique máxima, conjunto estável máximo e coloração ótima para um grafo triangularizado, baseados no seu algoritmo para determinar um esquema de eliminação perfeita ( algoritmo (III.8) ).

### III.7.5. Número de Intervalo

O número de intervalo  $i(G)$  de um grafo  $G$  é o menor inteiro não negativo, tal que  $G$  é o grafo interseção de conjuntos, onde cada um deles é a união de, no máximo,  $i(G)$  intervalos reais. Isto é, cada vértice de  $G$  está associado a um conjunto de  $i(G)$  intervalos, e dois vértices  $v$  e  $w$  de  $G$  são adjacentes se algum intervalo associado a  $v$  intercepta pelo menos um dos intervalos correspondentes a  $w$ .

Trotter, citado por SCHEINERMAN (1988), apresenta exemplos onde o número de intervalo pode ser arbitrariamente grande.

ANDREAE (1987) determinou um limite superior para o

número de intervalo de um grafo cordal, em função do tamanho  $\omega(G)$  da maior clique do grafo. Ele provou que:

$$i(G) \leq \left\lceil \frac{\omega(G)}{2} \right\rceil + 1 \quad (\text{III.1})$$

Considerando que  $h(t) = \max\{i(G) \mid G \text{ é cordal e } \omega(G) = t\}$ , então:

$$\sqrt{t} < h(t) \leq \left\lceil \frac{t}{2} \right\rceil + 1 \quad (\text{III.2})$$

Andreae demanda por um limite superior melhor para  $i(G)$ . SCHEINERMAN (1988) estreita este limite com o resultado:

$$i(G) \leq \left[ 1 + o(1) \right] \frac{\omega(G)}{\ln \omega(G)} \quad (\text{III.3})$$

Onde  $o(t)$  é tal que limite é 1 se  $t \rightarrow \infty$ . (III.4)

### III.7.6. Problemas sem Vantagens

Os quatro problemas estudados tem a vantagem de que, se a classe dos grafos é restrita aos grafos triangularizados, estes problemas podem ser resolvidos em tempo polinomial. Se for considerado um grafo qualquer não existe algoritmo polinomial para resolve-los.

Para outros problemas famosos de grafos não existe esta vantagem. Por exemplo, nos problemas que se seguem.

#### *Ciclo Hamiltoniano*

Dado um grafo  $G$ , a pergunta é se  $G$  tem um ciclo Hamiltoniano. Isto é, um ciclo que passa por cada vértice do grafo exatamente uma vez e volta ao vértice inicial.

Este problema permanece  $\mathcal{NP}$ -completo quando restrito à

classe dos grafos cordais.

### *Índice Cromático*

Dado um grafo  $G$  e um inteiro positivo  $j$ , a pergunta é: o índice cromático de  $G$  é menor ou igual a  $j$ ? Isto é, é possível colorir as arestas de  $G$ , de tal forma que duas arestas adjacentes tenham cores diferentes, com no máximo  $j$  cores?

Este é um problema aberto para um grafo perfeito qualquer e o mesmo acontece para um grafo triangularizado. Apenas é sabido que, em geral para qualquer grafo  $G$ , se  $\rho = \max \{ \text{grau}(v) \mid v \in V \}$  então  $\rho \leq \text{índice cromático} \leq \rho + 1$ .

### *Conjunto dominante*

Dado um grafo  $G=(V,E)$  e um inteiro não negativo  $j$  com  $j \leq |V|$ , pergunta-se se existe  $V' \subseteq V$  com  $|V'| \leq j$  tal que  $\forall v \in V-V' \exists w \in V', w$  adjacente a  $v$ .

BOOTH e JOHNSON (1982) provaram que este problema permanece *NP-completo* para o caso dos grafos triangularizados.

### *Corte Máximo (sem peso)*

Dado um grafo  $G=(V,E)$ , um inteiro não negativo  $j$ , uma função de peso nas arestas  $p(v,w) = 1 \quad \forall (v,w) \in E$ , a pergunta é se existe uma partição do conjunto de vértices  $V = V_1 + V_2$  tal que:

$$\sum_{\substack{v \in V \\ v_1 \in V_1 \\ v_2 \in V_2}} p(v_1, v_2) \geq j \quad (\text{III.5})$$

Este problema está em aberto para um grafo qualquer e o fato de se restringir à classes dos grafos triangularizados não apresenta vantagens pois continua em aberto para esta classe.

### *Problema de Steiner*

Dado um grafo  $G=(V,E)$ , um subconjunto dos vértices  $R \subseteq V$  e um inteiro não negativo  $j$  com  $j \leq |V|-1$ , pergunta-se se existe uma subárvore  $T$  de  $G$ ,  $T=(V_T, E_T)$ , tal que  $R \subseteq V_T$  e  $E_T \leq j$ .

Este problema é *NP-completo* para um grafo qualquer e continua sendo *NP-completo* para os grafos cordais, o que foi provado por WHITE et alii (1985).

### *Isomorfismo de Grafos*

Dados dois grafos  $G_1=(V_1, E_1)$  e  $G_2=(V_2, E_2)$  pergunta-se se eles são isomorfos, isto é se existe uma função unívoca  $f: V_1 \rightarrow V_2$  tal que  $(v,w) \in E_1 \iff (f(v), f(w)) \in E_2$ .

Este problema está em aberto para os grafos perfeitos e em particular para os triangularizados, e é equivalente em complexidade ao problema de isomorfismo de grafos em geral, de acordo com GAREY e JOHNSON (1979).

## **III.8. Aplicações dos Grafos Triangularizados**

### **III.8.1. Método de Eliminação de Gauss**

Dada uma matriz  $A$  não singular com  $n$  linhas e  $n$  colunas, pode-se transformar esta matriz na matriz identidade  $I$  pelo denominado *Método de Eliminação de Gauss*. A idéia deste método é determinar um elemento  $a_{ij}$  não nulo na matriz

o qual será o pivote e atualizar a matriz fazendo operações elementares nas linhas e colunas, de forma tal que  $a_{ij}$  fique na diagonal e passe a ter o valor um e, os outros elementos da linha  $i$  e da coluna  $j$  tomem o valor zero. O processo é repetido até obter-se a matriz identidade.

Ao realizar este processo numa matriz esparsa, com uma eleição arbitrária do pivote, pode acontecer num passo que elementos que são zero passem a tomar um valor não nulo. Um melhoramento do método é, determinar um método de seleção dos pivotes em forma tal que ao transformar  $A$  em  $I$  nenhum zero transforma-se num elemento distinto de zero, também é importante determinar em que matrizes é possível fazê-lo.

Um *esquema de eliminação perfeita* para  $A$  é uma sequência de pivotes que permitem transformar  $A$  em  $I$  sem que em nenhum passo um zero passe a ter um valor diferente de zero.

A matriz  $A$  pode representar-se num *grafo*  $G(A)$  associando um vértice a cada linha (ou coluna) da matriz e a aresta  $(v_i, v_j)$  para  $i \neq j$ , existe se o valor  $a_{ij}$  é distinto de zero. Se  $A$  é simétrica então  $G(A)$  é não direcionado.

Também associa-se um grafo bipartite  $B(A)$  à matriz  $A$ , associando as linhas de  $A$  a um conjunto de vértices  $\{v_1, v_2, \dots, v_n\}$  e as colunas ao conjunto  $\{w_1, w_2, \dots, w_n\}$ , o vértice  $v_i$  é adjacente a  $w_j$  se  $a_{ij}$  é distinto de zero.

Se a matriz  $A$  é simétrica então  $a_{ii}$  não zero é aceito como pivote se e somente se  $v_i$  é um vértice simplicial de  $G(A)$ . Com isto, os esquemas de eliminação perfeita para  $A$ , sob a restrição que os pivotes são escolhidos na diagonal a qual tem todos seus elementos diferentes de zero, correspondem aos esquemas de eliminação perfeita para  $G(A)$ .

**Propriedade III.18:**

Seja  $A$  uma matriz simétrica com todos os elementos da diagonal não nulos, então as três condições seguintes são equivalentes:

- a)  $A$  tem um esquema de eliminação perfeita.
- b)  $A$  tem um esquema de eliminação perfeita no qual os pivotes são escolhidos na diagonal.
- c)  $G(A)$  é um grafo triangularizado.

ROSE (1970) formulou e provou esta propriedade, que caracteriza os esquemas de eliminação para matrizes simétricas. ROSE et alii (1976) apresentam um algoritmo linear no tamanho do grafo, baseado no método de busca em largura lexicográfica. HASKINS e ROSE (1973) estudaram o caso de matrizes não simétricas. COLUMBIC e GOSS (1978) introduziram e estudaram o modelo com os grafos bipartite.

O método de Gauss é um método muito conhecido para resolver um sistema linear:

$$A \vec{x} = \vec{b} \quad (\text{III.6})$$

DENNIS e STEIHAUG (1986) introduzem permutações nas colunas de  $A$  quando é usado o método de Gauss Seidel para resolver o sistema linear.

**III.8.2. Minimização de uma Função com Matriz Hessiana Esparsa**

Para minimizar uma função  $f: \mathbb{R}^n \longrightarrow \mathbb{R}$  duas vezes continuamente diferenciável, os métodos mais usados são os métodos tipo Newton, os quais precisam a solução em cada iteração de um sistema linear da forma:

$$H(x) \vec{d} = - \nabla f(x) \quad (\text{III.7})$$

Onde  $\nabla f$  é o gradiente da função  $f$ ,  $H$  é a matriz

Hessiana de  $f$ ,  $x$  é o ponto atual e  $\vec{d}$  é a direção de descida a ser determinada.

Se  $x$  é contíguo a um mínimo local  $\hat{x}$  de  $f$ , então pode-se esperar que  $H$  seja definida positiva, neste caso pode-se usar o *método do gradiente conjugado* para resolver a equação (III.7). Caso contrário,  $H$  poderia não ser definida positiva e podendo-se aplicar gradiente conjugado até obter uma indefinição. STEIHAUG (1983) apresenta uma modificação do método de Powell que permite resolver a equação (III.7) no caso de matriz Hessiana não definida.

O método do gradiente conjugado pode ser melhorado em termos da sua eficiência, usando um *precondicionador*. COLEMAN (1988) apresenta a determinação de *matrizes precondicionadoras cordais* tais que ao aplicar o método de fatorização de Cholesky, nenhum zero é transformado num elemento não nulo. Uma *matriz é cordal* se seu grafo é cordal.

A idéia é: dada a matriz  $H$ , determinar uma matriz de permutação  $P$  tal que a matriz  $P H P^t$  seja diagonal por blocos e cada um desses blocos seja cordal. Coleman desenvolveu um algoritmo com o *método guloso* para determinar estes blocos.

Para matrizes de grande porte, só são armazenados os elementos distintos de zero. Este algoritmo permite a implementação do gradiente conjugado precondicionado com uma complexidade de espaço  $O(\eta(H))$ , onde  $\eta(H)$  é o número de não zeros de  $H$ .

### III.8.3. "Scheduling" com Tempo de Processamento Unitário

O problema é: dado um conjunto de processos com tempo de processamento unitário, com relações de precedência para os processos dadas num grafo cujo complemento é triangularizado, determinar um "schedule" admissível; isto é

determinar a alocação dos processos aos diferentes processadores e a ordem na qual estes serão processados, de forma tal que nenhum deles deva começar antes da sua data de liberação e que todos os processos estejam processados antes da sua data de término.

PAPADIMITRIOU e YANNAKAKIS (1979) apresentam um algoritmo com complexidade de tempo  $O(n+m)$ , sendo  $n$  o número de vértices e  $m$  o número de arestas do grafo de precedências. A idéia é aproveitar a estrutura do grafo de incomparabilidade (complemento do grafo de precedências) para dar uma ordem nos processos e com base nesta ordem aloca-los nos processadores.

### III.9. Subclasses dos Grafos Triangularizados

Os grafos de comparabilidade com seus complementos e os grafos triangularizados com seus complementos são as classes clássicas de grafos perfeitos, mas não são as únicas. Todo grafo  $G$  pode ser analisado se pertence ou não a uma ou mais dessas classes.

Algumas classes de grafos perfeitos podem ser caracterizadas por uma combinação dessas propriedades. Aqui descrevem-se as classes relacionadas com os grafos triangularizados, para um tratamento mais aprofundado das outras classes pode-se consultar DUCHET (1984), COLUMBIC (1980) e LOVÁSZ (1983).

#### III.9.1. Grafos "Split"

Um grafo  $G$  é "split" se seu conjunto de vértices  $V$ , pode ser particionado em um conjunto estável  $S$  e uma clique  $K$ , com  $V = S + K$ .

A partição pode não ser única. Tem-se que  $\bar{S}$  é uma

clique e  $\bar{K}$  é um conjunto estável de  $\bar{G}$ , logo:  $\bar{G}$  é "split".

A propriedade de ser grafo "split" é de hereditariedade, por isso é possível de caracterizar por subgrafos proibidos.

No grafo da figura (III.7) os vértices b, c e e formam uma clique e os vértices a,d e f são independentes.

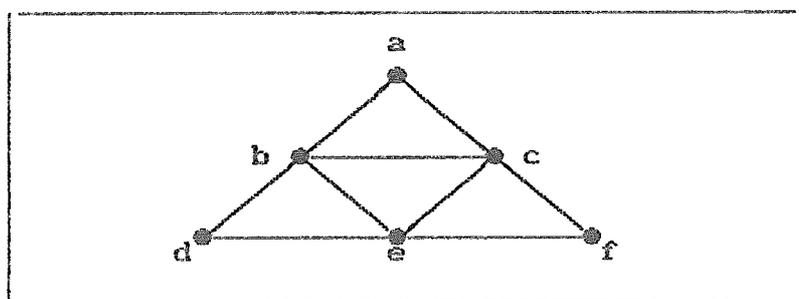


Figura III.7.

**Propriedade III.19:**

As três condições seguintes são equivalentes:

- a)  $G$  é um grafo "split".
- b)  $G$  e  $\bar{G}$  são triangularizados.
- c)  $G$  não contém nenhum subgrafo induzido isomorfo a  $2K_2$ ,  $C_4$  ou  $C_5$ .

**Corolário III.2:**

Todo grafo "split" é perfeito.

Os grafos "split" não se diferenciam dos cordais no que refere-se à complexidade do algoritmo de reconhecimento e dos algoritmos para os problemas descritos na seção (III.7).

### III.9.2. Grafos de Intervalo

Um grafo  $G$  é de intervalo se seu conjunto de vértices  $V$ , pode ser posto em correspondência um a um com um conjunto de intervalos da reta, de forma que dois vértices são adjacentes se seus correspondentes intervalos se interceptam.

Na figura (III.8) apresenta-se um grafo de intervalo e o conjunto de intervalos reais associados a ele, tem-se as seguintes únicas interseções entre os intervalos:  $I_a \cap I_b$ ,  $I_b \cap I_c$ ,  $I_c \cap I_d$ ,  $I_d \cap I_e$ .

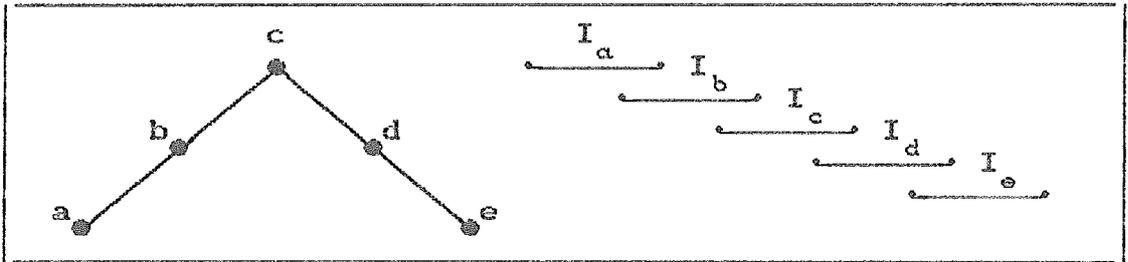


Figura III.8.

#### Propriedade III.20:

Todo grafo de intervalo é triangularizado.

A relação de inclusão é própria, considere-se o grafo da figura (III.9) que é triangularizado mas não é de intervalo.

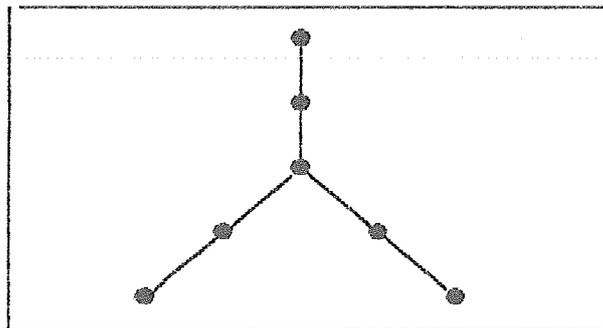


Figura III.9.

**Corolário III.3:**

Todo grafo de intervalo é perfeito.

**Propriedade III.21:**

As três condições seguintes são equivalentes:

- $G$  é um grafo intervalo.
- $G$  é um grafo triangularizado e de co-comparabilidade.
- $G$  não contém nenhum subgrafo induzido isomorfo a  $C_4$ ,  $C_5$ ,  $C_6$ ,  $T_1$ ,  $T_2$  ou  $T_3$ ; onde  $T_1$  é o grafo da figura (III.7),  $T_2$  é o grafo da figura (III.9) e  $T_3$  apresenta-se na figura (III.10).

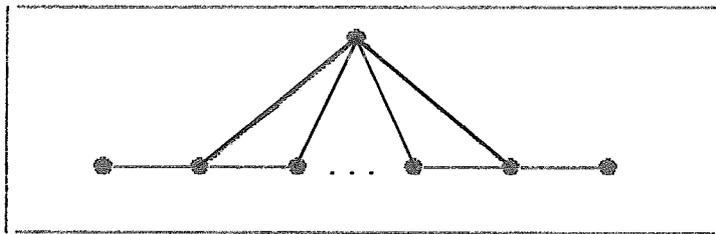


Figura III.10.

Ser grafo de intervalo é propriedade de hereditariedade, mas não é autocomplementar.

O complemento de um grafo de intervalo também é, de fato, perfeito pois é de comparabilidade e co-triangularizado.

Booth e Leuker, descrito em GOLUMBIC (1980), desenvolveram um algoritmo de reconhecimento para estes grafos, em tempo linear, aproveitando uma caracterização deles pela matriz de clique.

GUPTA et alii (1982) desenvolveram algoritmos para os quatro problemas clássicos de otimização, para o caso

particular dos grafos de intervalo. Se a representação por intervalos é conhecida, a complexidade é  $O(n \log n)$ , caso contrário é  $O(n^2)$ .

BOOTH e JOHNSON (1982) apresentam um algoritmo linear para o problema do conjunto dominante para grafos de intervalo. LUEKER e BOOTH (1979) desenvolveram um algoritmo linear para o isomorfismo de grafos de intervalo. WHITE (1985) introduz um algoritmo polinomial para o problema de Steiner nestes grafos.

KLEIN (1988) desenvolveu um algoritmo paralelo para o reconhecimento destes grafos, baseado no algoritmo de Booth e Lueker, com complexidade de tempo  $O(\log^2 n)$  usando  $O(n+m)$  processadores. Baseado nas mesmas idéias, apresentou um algoritmo paralelo para o isomorfismo de grafos de intervalo com complexidade de tempo  $O(\log^2 n)$  usando  $O(n+m)$  processadores em uma máquina PRAM com leitura e escrita concorrente.

Os problemas de índice cromático e de corte máximo permanecem em aberto para os grafos de intervalo.

### III.9.3. Grafos "Threshold"

Um grafo "threshold" é construído a partir de um conjunto de números reais  $\{a_1, a_2, \dots, a_n\}$ , associa-se a cada um destes números um vértice e a aresta  $(a_i, a_j)$  existe se e somente se  $a_i + a_j > 0$ .

Ser um grafo "threshold" é uma propriedade de hereditariedade.

Na figura (III.11) apresenta-se um grafo "threshold".

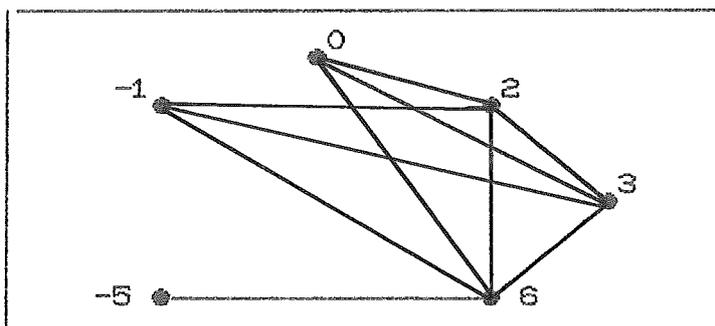


Figura III.11.

**Propriedade III.22:**

Se  $G$  é "threshold" então  $G$  e  $\bar{G}$  são triangularizados e de comparabilidade.

A recíproca não é verdadeira,  $P_3$  é triangularizado, co-triangularizado, de comparabilidade e de co-comparabilidade mas não é "threshold".

**Corolário III.4:**

Todo grafo "threshold" é perfeito.

**Propriedade III.23:**

$G$  é um grafo "threshold" se e somente se  $G$  não contém nenhum subgrafo induzido isomorfo a  $2K_2$ ,  $C_4$  ou  $P_4$ .

**III.9.4. Grafos Fortemente Triangularizados**

Um grafo  $G$  é *fortemente triangularizado* ou *fortemente cordal* se ele é cordal e todo ciclo par de comprimento maior ou igual a seis, tem uma *corda forte*, isto é, uma corda que liga vértices separados por um número ímpar (maior que um) de arestas no ciclo.

Na figura (III.12) tem-se o grafo  $G_1$  que é fortemente cordal e o grafo  $G_2$  que é cordal mas não é fortemente cordal.

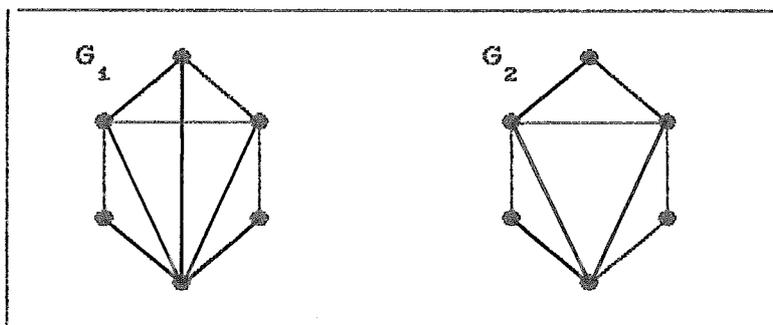


Figura III.12.

**Propriedade III.24:**

$G$  cordal é fortemente cordal se e somente se  $G$  não tem como subgrafo induzido um trampolim. Um *trampolim* é um grafo cordal  $G$  com  $2n$  vértices,  $n \geq 3$ , tal que seu conjunto de vértices pode ser particionado em  $V = W + U$ , cada um deles com  $n$  vértices tal que,  $\forall w_i \in W$  e  $u_j \in U$  tem-se  $(w_i, u_j) \in V$  se e somente se  $i = j$  ou  $i = j + 1 \pmod{n}$ , o subgrafo induzido  $G_U$  é completo e  $W$  é um conjunto estável. Por exemplo, o grafo da figura (III.13) é um trampolim com a partição dos vértices indicada.

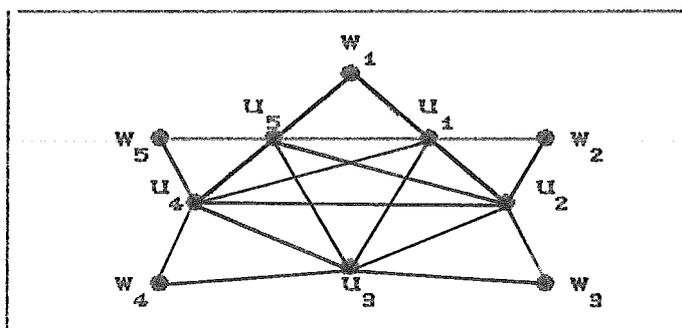


Figura III.13.

**Corolário III.5:**

Os grafos fortemente cordais tem várias caracterizações similares às dos grafos cordais, sendo que os primeiros tem uma restrição dos vértices simpliciais e dos esquemas de eliminação perfeita. Com isto, FARBER (1983) desenvolveu um algoritmo de reconhecimento, em tempo polinomial.

**Definição III.9:**

Seja  $G=(V,E)$ ,  $N(v) = \{v\} \cup \text{Adj}(v) \quad \forall v \in V$ , um *esquema de eliminação forte* é uma ordenação dos vértices  $v_1, v_2, \dots, v_n$  tal que para todo  $i, j, k, l$  se  $i < j, k < l, v_k, v_l \in N(v_i), v_k \in N(v_j)$  então  $v_l \in N(v_j)$ .

**Observação III.5:**

Todo esquema de eliminação forte é um esquema de eliminação perfeita.

**Definição III.10:**

Os vértices  $u$  e  $v$  são *compatíveis* se  $N(u) \subseteq N(v)$  ou  $N(v) \subseteq N(u)$ . Caso contrário, eles são *incompatíveis*.

**Definição III.11:**

Um vértice  $v$  de  $G$  é *simples* se os vértices de  $N(v)$  são compatíveis dois a dois.

**Observação III.6:**

Todo vértice simples é um vértice simplicial.

**Propriedade III.25:**

As três condições seguintes são equivalentes:

- a)  $G$  é fortemente cordal.
- b)  $G$  admite um esquema de eliminação forte.
- c) Todo subgrafo induzido de  $G$  tem um vértice simples.

FARBER (1984) apresenta um algoritmo com complexidade de tempo  $O(n+m)$  para o problema do conjunto dominante. WHITE et alii (1985) expõem um algoritmo com tempo  $O(n^2)$  para o problema de Steiner, para grafos fortemente triangularizados. Estes dois problemas são  $NP$ -completos nos grafos triangularizados.

**III.9.5. Grafos Complemento de "Threshold" com Tolerâncias**

Um grafo  $G$  é um *grafo "threshold" com tolerâncias* se é possível associar peso  $w_v$  e tolerância  $t_v$  a cada vértice  $v$  de  $G$ , tais que dois vértices são adjacentes se a soma dos pesos é maior ou pelo menos igual a cada uma das tolerâncias dos vértices.

Todo grafo "threshold" é um grafo "threshold" com tolerâncias, basta considerar todas as tolerâncias iguais a um.

O complemento de um grafo "threshold" com tolerâncias é chamado *grafo coTT*, e são estes grafos os de interesse aqui.

**Propriedade III.26:**

$G$  é coTT se existem números  $a_v$  e  $b_v$  para cada vértice  $v$  de  $G$  tais que  $(v,w) \in E$ , se e somente se  $a_v \leq b_w$  e  $a_w \leq b_v$ .

**Propriedade III.27:**

Todo grafo "threshold" é um grafo coTT.

Todo grafo de intervalo é um grafo coTT.

Estas inclusões são estritas, o grafo da figura (III.14) é coTT mas não é "threshold" e não é de intervalo,

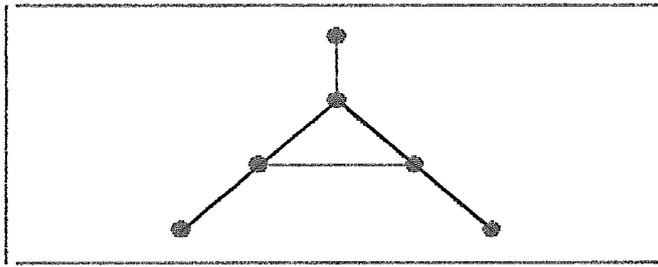


Figura III.14.

**Propriedade III.28:**

Todo grafo coTT é fortemente cordal.

MONMA et alii (1988) definem estes grafos, apresentam caracterizações e um algoritmo polinomial de reconhecimento.

**III.9.6. Grafos Cordais de Interseção de Caminhos**

Dada uma árvore e caminhos na árvore, chama-se *grafo UV* ao grafo interseção de caminhos de vértices. Se a árvore é direcionada, o grafo interseção de caminhos denomina-se *grafo DV*.

Um estudo mais aprofundado destes grafos é apresentado no capítulo IV.

**Observação III.7:**

Nem todo grafo perfeito deve pertencer a uma das classes até agora consideradas: triangularizados, co-triangularizados, comparabilidade, co-comparabilidade. Como estas quatro classes são de hereditariedade, basta observar que  $C_6$  não é triangularizado, nem co-triangularizado e não é de co-comparabilidade, e o grafo da figura (III.14) não pertence a nenhuma das quatro classes especificadas, então o grafo da figura (III.15) também não pertence a nenhuma daquelas classes embora seja perfeito.

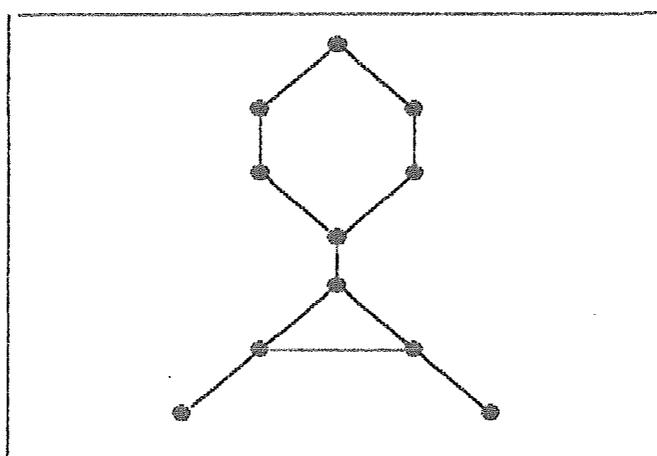


Figura III.15.

### III.10. Extensões dos Grafos Triangularizados

Sabendo que são perfeitos os grafos tais que todos os ciclos com pelo menos três vértices possuem uma corda, uma relaxação na procura de classes de grafos perfeitos é diminuir o número de cordas necessárias para obter a perfeição. Com esta idéia nasceram os grafos apresentados a seguir.

#### III.10.1. Grafos de Gallai

Dado  $G=(V,E)$ , diz-se que duas cordas  $(v_1, v_2)$  e  $(w_1, w_2)$ , de um ciclo se cruzam se os vértices  $v_1, v_2, w_1$  e  $w_2$  são

todos distintos e aparecem na ordem  $v_1, w_1, v_2, w_2$  no ciclo.

**Definição III.12:**

$G$  é um *grafo de Gallai* se todo ciclo ímpar com pelo menos cinco vértices tem duas cordas que não se cruzam.

O grafo da figura (III.16) é Gallai e triangularizado.

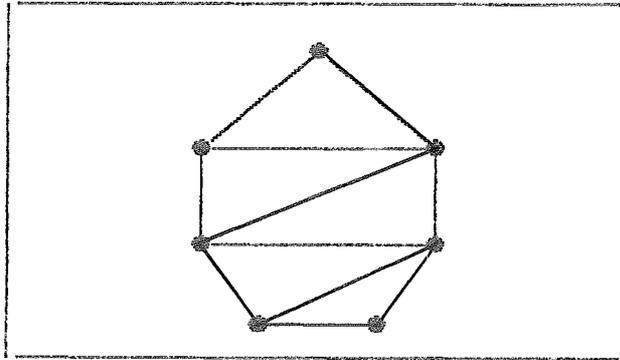


Figura III.16.

Os grafos de Gallai também são denominados *grafos  $i$ -triangularizados*.

**Propriedade III.29:**

Todo grafo triangularizado é de Gallai.

A inclusão é estrita, o grafo da figura (III.17) é Gallai mas não é cordal.

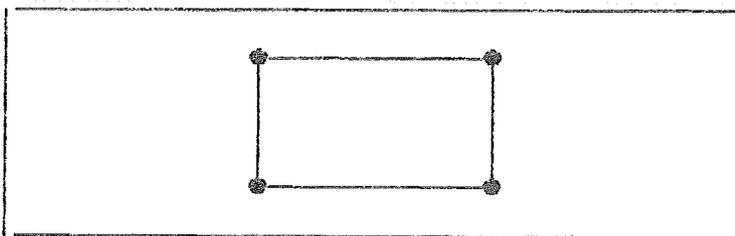


Figura III.17.

**Propriedade III.30:**

Todo grafo de Gallai é perfeito.

Em relação a esta classe, WHITESIDES (1984) apresenta um algoritmo para resolver os quatro problemas clássicos de otimização em tempo *polinomial*, baseados na decomposição do grafo por cliques maximais. O reconhecimento é feito em tempo  $O(n^5)$ .

**III.10.2. Grafos de Paridade**

$G$  é um *grafo de paridade* se todo ciclo ímpar tem duas cordas que se cruzam.

Na figura (III.18) apresenta-se um grafo de paridade.

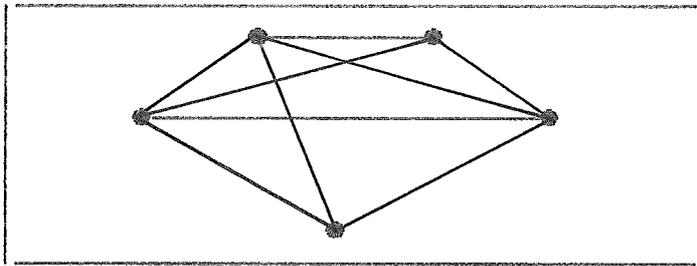


Figura III.18.

**Propriedade III.31:**

Todo grafo cordal é de paridade.

A inclusão é estrita, o grafo da figura (III.19) é de paridade, mas não é cordal.

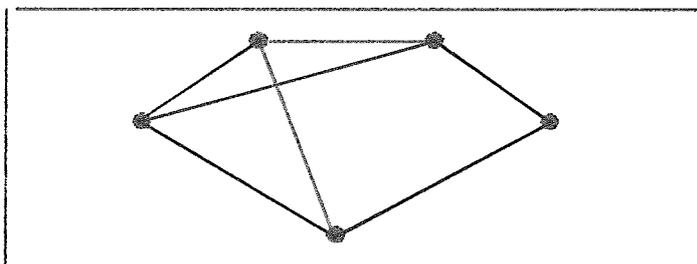


Figura III.19.

**Propriedade III.32:**

Todo grafo de paridade é perfeito.

Estes grafos foram inicialmente estudados por Olaru que demonstrou que eles são perfeitos, sem dar-lhes um nome especial mas como uma analogia aos grafos de Gallai que precisam que as cordas não se cruzem. Depois, BURLET e UHRY (1984) definiram estes grafos usando uma propriedade de paridade de caminhos, e desenvolveram algoritmos *polinomiais* para o reconhecimento e os quatro problemas clássicos de otimização, aproveitando as propriedades estruturais deles.

**III.10.3. Grafos de Meyniel**

$G$  é um *grafo de Meyniel* se todo ciclo ímpar com mais de três vértices, tem pelo menos duas cordas.

**Propriedade III.33:**

Todo grafo de paridade é de Meyniel.

Todo grafo de Gallai é de Meyniel.

A inclusão é estrita, o grafo da figura (III.20) é Meyniel, mas não é de Gallai e não é de paridade.

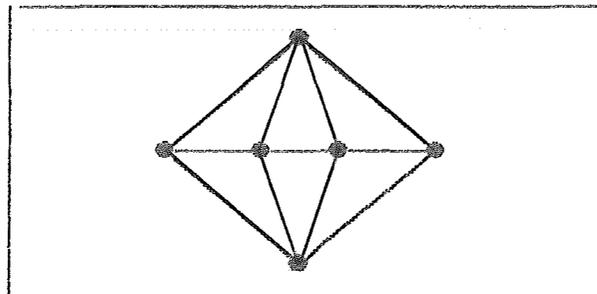


Figura III.20.

**Propriedade III.34:**

Todo grafo de Meyniel é perfeito.

MEYNIEL (1976) provou que estes grafos são perfeitos com base na definição deles que não é uma caracterização que permite um reconhecimento fácil, pois é preciso enumerar todos os ciclos ímpares.

BURLET e FONLUPT (1984) apresentam um algoritmo de reconhecimento dos grafos de Meyniel, em tempo *polinomial*. Este algoritmo baseia-se numa operação em dois grafos de Meyniel denominada *amalgama*, a qual preserva a perfeição. No mesmo artigo, eles anunciam algoritmos para os quatro problemas de otimização usando a estrutura dos grafos de Meyniel, mas ainda não estão publicados.

**III.10.4. Grafos Separáveis por Cliques**

Considerando que a operação de unir dois grafos perfeitos através de uma clique maximal contida em ambos, ou seja, identificando os vértices da clique, superpondo-os e copiando o resto dos dois grafos em apenas um grafo, preserva a perfeição, na procura para aproximar-se da conjectura forte, parece interessante estudar os grafos que podem decompor-se por suas cliques maximais.

Um grafo  $G=(V,E)$  é *separável por cliques* se satisfaz a condição a) ou b).

a)  $G$  tem uma das duas formas básicas seguintes:

a.1) Tipo 1:  $V$  pode ser particionado em dois conjuntos disjuntos,  $V=V_1+V_2$ , tal que  $G_{V_1}$  é um grafo bipartite conexo com  $|V_1| \geq 3$ ,  $G_{V_2}$  é uma clique e em  $G$ , todo vértice de  $V_1$  é adjacente a todo vértice de  $V_2$ .

a.2) Tipo 2:  $G$  é um grafo  $k$ -partite completo, para algum  $k$ . Isto é,  $V=V_1+V_2+\dots+V_k$ ,  $V_i \cap V_j = \emptyset \quad \forall i \neq j$ , todo  $V_i$  é um conjunto independente e para todo  $i \neq j$ , todo

vértice de  $V_i$  é adjacente a todo vértice de  $V_j$ .

b)  $G$  tem uma clique maximal  $K$  tal que o subgrafo induzido  $G_{V-K}$  é desconexo e cada uma das componentes conexas dele é um grafo separável por cliques.

O grafo da figura (III.21) satisfaz a condição b), considerando a clique  $K = \{b, c, g\}$ , o subgrafo induzido por  $\{a, b, c, e, f, g\}$  é do tipo 2 e o subgrafo induzido por  $\{b, c, d, g, h\}$  é do tipo 1.

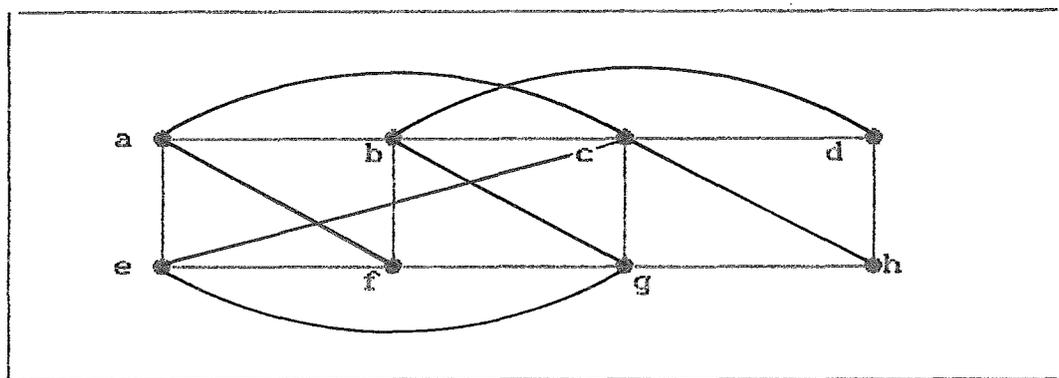


Figura III.21.

**Propriedade III.35:**

Todo grafo de Gallai é separável por cliques.

A inclusão é estrita, o grafo da figura (III.21) é separável por cliques, mas não é de Gallai.

**Propriedade III.36:**

Todo grafo separável por cliques é perfeito.

GAVRIL (1977) definiu estes grafos e desenvolveu um algoritmo de reconhecimento e algoritmos para resolver os problemas do número cromático e da maior clique, em tempo *polinomial*.

WHITESIDES (1984) dá para estes grafos uma generalização dos algoritmos usados nos grafos de Gallai, para o reconhecimento e os quatro problemas clássicos, baseado na decomposição do grafo por cliques. Esta decomposição é feita em tempo  $O(n^5)$ .

### III.10.5. Grafos Fracamente Triangularizados

Um grafo  $G$  é *fracamente triangularizado* (ou *fracamente cordal*) se ele e seu complemento não contem un subgrafo induzido isomorfo a um ciclo sem cordas com cinco vértices ou mais.

Na figura (III.22), apresenta-se um grafo fracamente cordal.

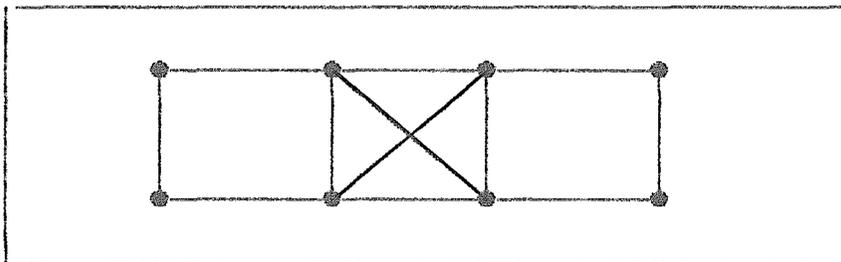


Figura III.22.

#### *Propriedade III.37:*

Todo grafo triangularizado é fracamente triangularizado.

Todo grafo co-triangularizado é fracamente triangularizado.

#### *Propriedade III.38:*

Todo grafo fracamente triangularizado é perfeito.

HAYWARD (1985) provou a perfeição destes grafos baseado num *conjunto separador-estrela* que é um conjunto separador  $C$  de  $G$ , tal que existe um vértice em  $C$  que é adjacente a todos os outros vértices de  $C$ .

HAYWARD et alii (1988) desenvolveram algoritmos polinomiais para os quatro problemas clássicos de otimização baseados em outra caracterização dos grafos fracamente triangularizados. Eles definem um "two-pair" que é um par de vértices não adjacentes tais que todo caminho sem cordas entre eles, tem exatamente duas arestas. O teorema fundamental para isto diz que todo grafo fracamente triangularizado que não é uma clique, tem um "two-pair". O algoritmo para o problema da maior clique e do número cromático tem complexidade de tempo  $O((n+m)n^3)$ . Ao aplicar este algoritmo ao grafo complemento obtém-se o conjunto estável máximo e uma cobertura mínima por cliques para o grafo.

No apêndice A expõe-se um esquema com as relações de inclusão entre as diferentes classes de grafos apresentadas.

No apêndice II apresenta-se uma tabela com as diferentes classes de grafos consideradas e alguns dos problemas clássicos em grafos. A entrada da tabela é a complexidade do algoritmo para resolver esse problema na classe de grafos correspondente, se existe um algoritmo *polinomial*. Caso contrário, indica-se se o problema pertence à classe *NP-completo* ou se está em aberto.

## Capítulo IV: GRAFOS INTERSEÇÃO DE CAMINHOS

### IV.1. Definições

Dada uma árvore e diferentes caminhos na árvore, pode-se definir vários tipos de *grafos interseção de caminhos*, dependendo se é considerado como interseção que os caminhos compartilham vértices ou arestas. Também distingue-se se a árvore, e portanto, os caminhos, é direcionada ou não. Se é direcionada, considera-se o caso especial em que a árvore é enraizada, isto é, possui somente um vértice com grau de entrada zero, o qual é denominado raiz.

No estudo destes grafos interseção, os pesquisadores têm dado diversos nomes às suas classes. Primeiro RENZ (1970) chamou *grafos de caminhos* aos *grafos interseção de caminhos de vértices de uma árvore não direcionada*, caracterizou esta classe e deixou em aberto a construção de um algoritmo de reconhecimento. Gavril aceitou o desafio e desenvolveu primeiro um algoritmo *polinomial* para os *grafos interseção de caminhos de vértices de uma árvore direcionada enraizada* (GAVRIL,1975), os quais denominou *grafos de caminhos direcionados*. Logo depois, desenvolveu um algoritmo *polinomial* para os grafos de caminhos (GAVRIL,1978). Estes dois algoritmos baseiam-se na mesma idéia com que ele demonstrou que os grafos triangularizados são os grafos de interseção de subárvores de uma árvore não direcionada (GAVRIL,1974).

Golumbic e Jamison estudaram os *grafos interseção de caminhos de arestas de uma árvore não direcionada*, denominando-os *grafos EPT*. Eles provaram que o problema de reconhecimento desses grafos é *NP-completo* (GOLUMBIC e JAMISON,1985b) e mostraram que as classes dos grafos interseção em arestas e em vértices, coincidem no caso onde todos seus vértices têm grau 3. Também provaram que a conjectura forte dos grafos perfeitos é válida para os grafos EPT (GOLUMBIC e JAMISON,1985a).

Ao mesmo tempo, SYSLO (1985) estudou a subclasse dos *grafos EPT triangularizados* e provou que esta subclasse é um subconjunto dos *grafos VPT*, que são os *grafos interseção em vértices*.

MONMA e WEI (1986) fizeram um estudo unificado de todos estes grafos interseção de caminhos, dando relações entre as diferentes classes e também entre elas e os grafos triangularizados e os grafos perfeitos. Eles apresentam a seguinte classificação :

- UV: grafos interseção de caminhos de vértices da árvore não direcionada
- DV: grafos interseção de caminhos de vértices da árvore direcionada
- RDV: grafos interseção de caminhos de vértices da árvore direcionada e enraizada
- UE: grafos interseção de caminhos de arestas da árvore não direcionada
- DE: grafos interseção de caminhos de arestas da árvore direcionada
- RDE: grafos interseção de caminhos de arestas da árvore direcionada e enraizada.

Considerando esta classificação, os grafos EPT correspondem aos grafos UE, os grafos VPT aos grafos UV e os grafos de caminhos direcionados aos RDV.

Dado que, não é possível, em geral, estudar os grafos UE em conjunto com as outras classes de grafos interseção de caminhos, define-se uma nova classe de grafos quando a coleção  $\mathcal{P}$  de caminhos satisfaz a propriedade de Helly que diz: Para toda subcoleção  $\mathcal{P}'$  de  $\mathcal{P}$ , se  $P_i \cap P_j \neq \emptyset \quad \forall P_i, P_j \in \mathcal{P}'$  então  $\bigcap_{P_i \in \mathcal{P}'} P_i \neq \emptyset$ .

Os grafos UE que satisfazem esta propriedade são chamados *grafos UEH*. Uma família de subárvores sempre satisfaz a propriedade de Helly, logo uma família de caminhos de vértices também.

Por exemplo, na figura (IV.1) apresenta-se a árvore  $T_1$  com os caminhos  $P_1, P_2, \dots, P_6$  indicados e seu grafo UV correspondente.

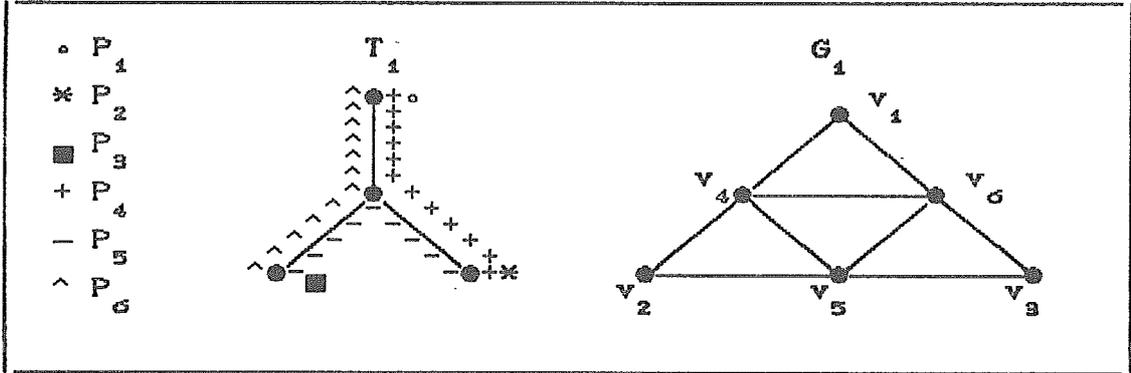


Figura IV.1.

Na figura (IV.2) mostra-se a árvore direcionada  $T_2$  e seu grafo DV.

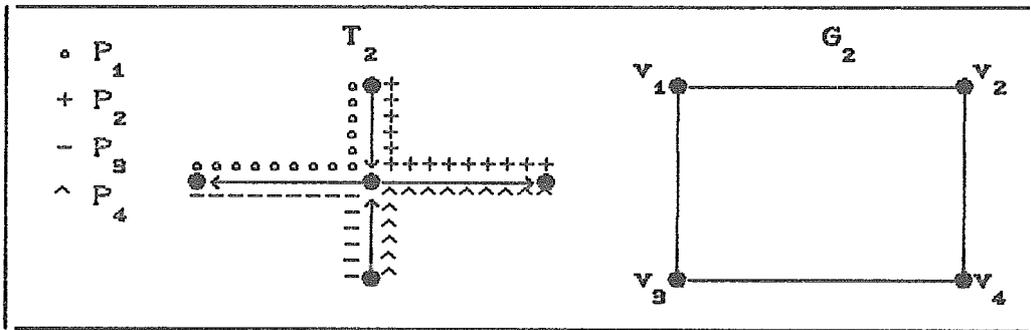


Figura IV.2.

O grafo  $G_3$  da figura (IV.3) é o grafo RDV da árvore direcionada e enraizada  $T_3$ .

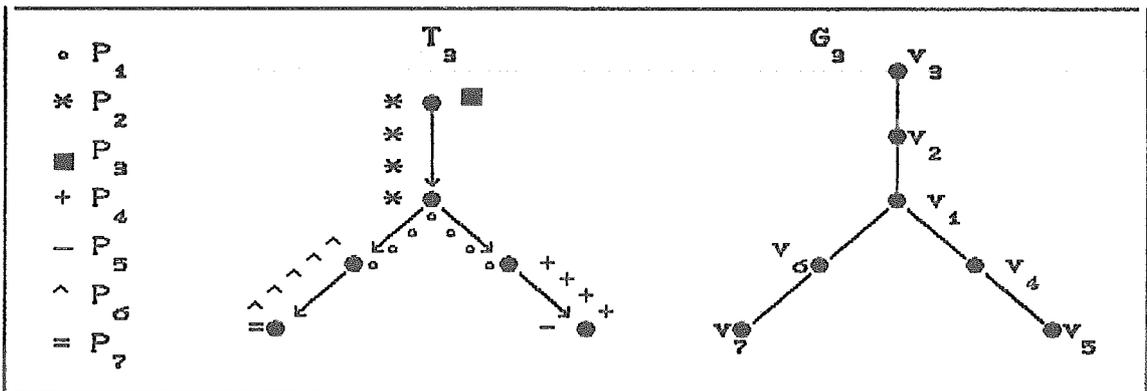


Figura IV.3.

Na figura (IV.4) tem-se a árvore  $T_4$  e seu grafo UE.

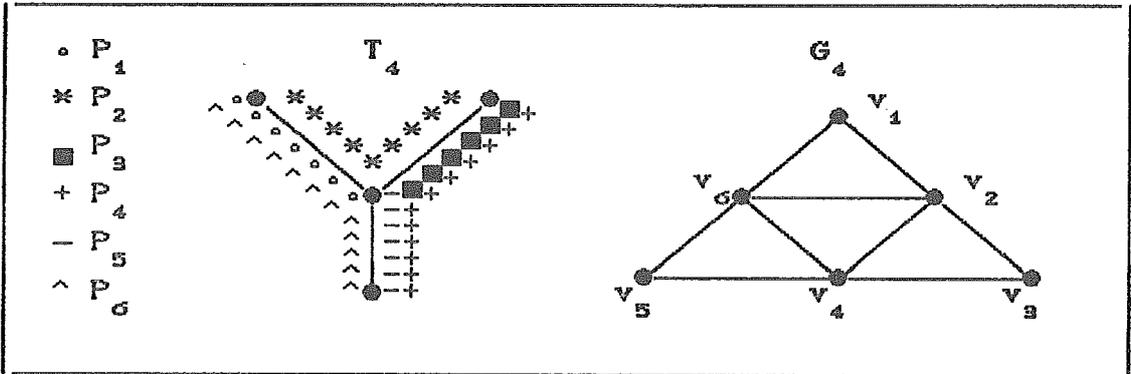


Figura IV.4.

Na figura (IV.5) apresenta-se a árvore direcionada  $T_5$  e seu grafo DE correspondente.

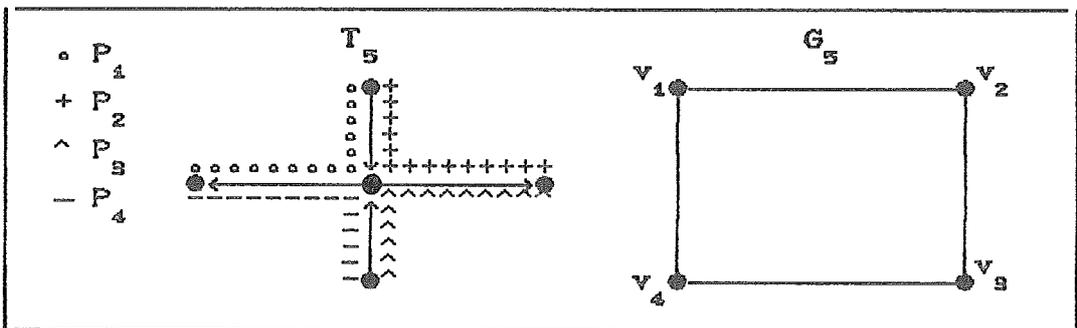


Figura IV.5.

**Propriedade IV.1:**

Existem várias relações entre as classes de grafos interseção de caminhos :

- a)  $UV \subseteq \text{Cordial}$
- b)  $\text{RDV} \subset UV$
- c) Todo grafo intervalo é RDV
- d)  $(\text{UE} \cap \text{cordal}) \subseteq UV$
- e)  $\text{RDV} = \text{RDE}$
- f)  $\text{RDV} \subseteq \text{DV}$
- g)  $\text{DV} \subseteq \text{UV}$
- h)  $\text{DV} \subseteq \text{DE}$

- i)  $DE \subseteq UEH$
- j)  $UEH \subseteq UE$
- k)  $DV = \text{Cordial} \cap UEH$
- l)  $(UE \cap \text{Cordial}) = (UE \cap UV)$
- m)  $DE = \text{Perfeito} \cap UEH$
- n)  $DE$  são perfeitos
- o)  $UE$  satisfaz conjectura forte.

As inclusões em b e c são estritas, na figura (IV.6) tem-se o grafo  $G_1$  que é UV mas não é RDV e o grafo  $G_2$  que é RDV mas não é de intervalo.

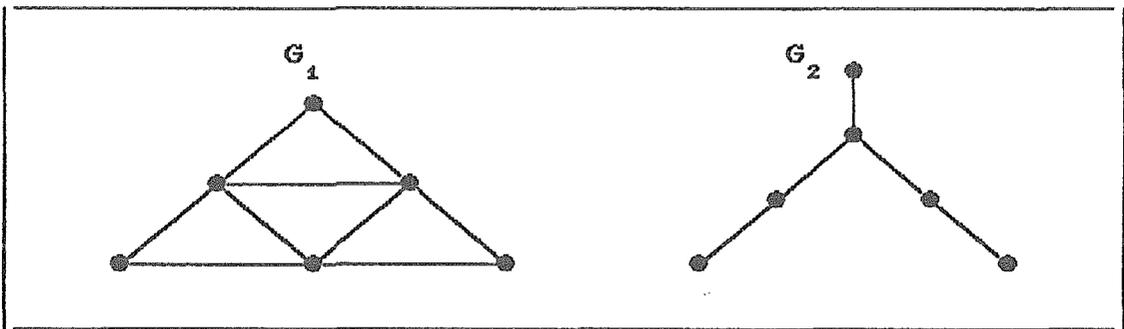


Figura IV.6.

## IV.2. Representação pela Árvore de Cliques

Apresenta-se nesta seção uma maneira de caracterizar a árvore associada a cada uma das classes de grafos interseção apresentadas. Dado um grafo  $G=(V,E)$ , chama-se  $\mathcal{C}$  ao conjunto de cliques maximais de  $G$ . Para um vértice  $v \in V$  denomina-se  $\mathcal{C}_v$  ao conjunto de cliques maximais que contém  $v$ ,  $T(\mathcal{C}_v)$  é uma subárvore de uma árvore  $T$  com vértices ou arestas correspondentes a  $\mathcal{C}_v$ .

### Propriedade IV.2: Teorema da Árvore de Cliques

a)  $G$  é cordal se e somente se existe uma árvore  $T$  com conjunto de vértices  $\mathcal{C}$  tal que, para todo  $v \in V$ ,  $T(\mathcal{C}_v)$  é uma subárvore em  $T$ .

b)  $G$  é UV se e somente se existe uma árvore  $T$  com conjunto de vértices  $\mathcal{E}$  tal que, para todo  $v \in V$ ,  $T(\mathcal{E}_v)$  é um caminho em  $T$ .

c)  $G$  é DV se e somente se existe uma árvore direcionada  $T$  com conjunto de vértices  $\mathcal{E}$  tal que, para todo  $v \in V$ ,  $T(\mathcal{E}_v)$  é um caminho direcionado em  $T$ .

d)  $G$  é RDV se e somente se existe uma árvore enraizada  $T$  com conjunto de vértices  $\mathcal{E}$  tal que, para todo  $v \in V$ ,  $T(\mathcal{E}_v)$  é um caminho direcionado em  $T$ .

e)  $G$  é UEH se e somente se existe uma árvore  $T$  com conjunto de arestas  $\mathcal{E}$  tal que, para todo  $v \in V$ ,  $T(\mathcal{E}_v)$  é um caminho em  $T$ .

f)  $G$  é DE se e somente se existe uma árvore direcionada  $T$  com conjunto de arestas direcionadas  $\mathcal{E}$  tal que, para todo  $v \in V$ ,  $T(\mathcal{E}_v)$  é um caminho direcionado em  $T$ .

g)  $G$  é RDE se e somente se existe uma árvore enraizada  $T$  com conjunto de arestas direcionadas  $\mathcal{E}$  tal que, para todo  $v \in V$ ,  $T(\mathcal{E}_v)$  é um caminho direcionado em  $T$ .

h) Se  $G$  é UE então existe uma árvore  $T$  e uma coleção de caminhos  $\{P_v / v \in V\}$ , tais que dois caminhos diferentes não têm pontos finais comuns e todos os vértices finais dos caminhos têm grau dois.

As provas desta propriedade são feitas com base em um algoritmo de reconhecimento para o grafo de interseção correspondente. A prova da parte a pode ser consultada em GAVRIL (1974), da parte b encontra-se em GAVRIL (1978). GAVRIL (1975) apresenta a parte d, MONMA e WEI (1986) desenvolvem as provas das partes c, e, f. SYSLO (1985) mostra a parte h desta propriedade.

Note-se que não é possível caracterizar os grafos UE de forma similar aos outros grafos de interseção de caminhos, nos quais a caracterização é análoga, apenas é necessário ver se os caminhos são de vértices ou arestas e se a árvore é enraizada, direcionada ou não direcionada.

A árvore  $T$  que satisfaz a propriedade (IV.2) para um grafo  $G$  dado, denomina-se *árvore de cliques* ou *árvore característica* de  $G$ . Esta árvore tem o menor número de arestas possíveis, pois a família de caminhos não contém nenhum caminho que seja subconjunto de outro elemento da família.

### IV.3. Decomposição por Cliques Separadoras

Dado um grafo conexo  $G=(V,E)$ , uma clique maximal  $K$  de  $G$  é um *separador* se o subgrafo induzido  $G_{V-K}$  é desconexo. O separador produz o grafo  $G_{V-K}$  que tem pelo menos duas componentes conexas. Sejam  $V_1, V_2, \dots, V_s, s \geq 2$ , os conjuntos de vértices das componentes conexas de  $G_{V-K}$ , então  $V$  foi particionado por  $K$  nos  $s+1$  conjuntos  $V_1, V_2, \dots, V_s$  e  $K$ , e diz-se que  $K$  separa  $G$  nos  $s$  subgrafos induzidos  $G_{V_i \cup K}$ , com  $i=1,2, \dots, s$ . Este processo pode repetir-se em cada um dos subgrafos  $G_{V_i \cup K}$  e continuar até que nenhuma componente conexa gerada tenha um separador. Os grafos conexos que não têm separador denominam-se *átomos*, por exemplo qualquer grafo completo e qualquer ciclo é um átomo.

Para representar a decomposição de  $G$  por suas cliques separadoras, constrói-se uma árvore  $T_G$  com raiz  $G$ , os nós filhos de  $G$  são os subgrafos  $G_{V_i \cup K}, i=1, 2, \dots, s$ . Repete-se este processo de construção considerando cada um dos  $G_{V_i \cup K}$  como raiz de uma subárvore se o subgrafo tem separador. Continua-se desta forma até que todas as folhas de  $T_G$  sejam átomos.

A partir da árvore  $T_G$  pode-se construir novamente  $G$ , pois  $V=K+V_1+V_2+\dots+V_s$  e todos os subgrafos  $G_{V_i \cup K}$  são

induzidos, logo preservam as arestas de  $G$ , sendo somente necessário fazer a união dos nós filhos através da clique que os separou e está contida em ambos, superpondo a clique e conservando o resto de cada um dos subgrafos.

Um processo para determinar a decomposição por cliques de um grafo pode ser realizado em forma recursiva, como é dado no algoritmo (IV.1).

*Algoritmo IV.1: DECOMPOSIÇÃO POR CLIQUES*

- Dado  $G$
- se  $G$  é uma clique então devolver  $G$  é um átomo
- caso contrário efetuar
  - determinar o conjunto  $\mathcal{C}=\{c_1, \dots, c_r\}$  de cliques maximais de  $G$
  - chamar o algoritmo  $\text{DECOMP}(G, \mathcal{C})$

*Algoritmo IV.2: DECOMP( $G, \mathcal{C}$ )*

- Dados  $G, \mathcal{C}$  e  $r=|\mathcal{C}|$
- $K = \emptyset$
- $i = 1$
- enquanto  $K$  não seja separador e  $i \leq r$  efetuar
  - $K = c_i$
  - $i = i + 1$
- se  $i = r + 1$  então devolver  $G$  é um átomo
- caso contrário efetuar / $c_i$  é separador de  $G$ /
  - determinar as componentes conexas  $v_1, \dots, v_s$  de  $G - K$
  - para  $j = 1, \dots, s$  efetuar
    - determinar  $\mathcal{C}_j = \{\text{cliques de } G \text{ que estão em } G_j\}$
    - para  $k = 1, \dots, i - 1$  efetuar
      - se  $c_k \subseteq \mathcal{C}_j$  então  $\mathcal{C}_j = \mathcal{C}_j - \{c_k\}$
    - $\mathcal{C}_j = \mathcal{C}_j - \{K\}$
    - chamar  $\text{DECOMP}(G_j, \mathcal{C}_j)$

A determinação das cliques maximais do grafo pode fazer-se em tempo  $O(nmr)$  com o algoritmo de TSUKIYAMA et alii (1977).

O grafo da figura (IV.7), tem as cliques maximais  $c_1 = \{a, b, e\}$ ,  $c_2 = \{b, d, e\}$  e  $c_3 = \{b, c, d\}$ .  $K = c_2$  é uma clique separadora gerando os subgrafos  $G_1$  e  $G_2$ .

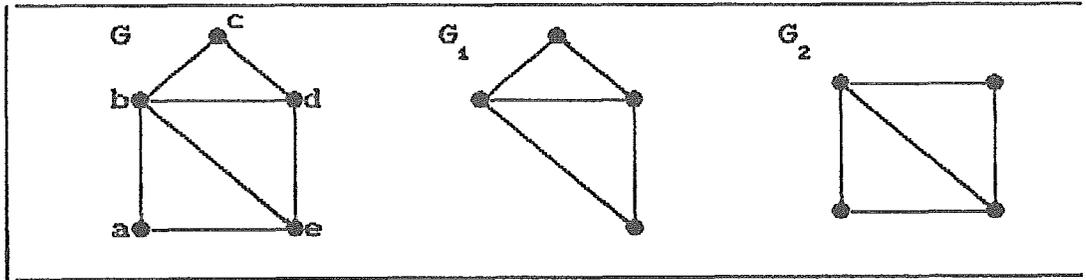


Figura IV.7.

**Propriedade IV.3:**

Tem-se quatro propriedades das cliques separadoras, as quais obtem-se diretamente da definição:

- Dada  $K$  uma clique maximal separadora de  $G$ ,  $K$  não é separador de nenhum  $G_{V_i \cup K}$ .
- Se uma clique maximal  $C$  de  $G$  não é separador de  $G$ , então  $C$  não é separador de nenhum  $G_{V_i \cup K}$ .
- O conjunto de cliques maximais de  $G$  é a união das cliques maximais dos  $G_{V_i \cup K}$ , para  $i = 1, 2, \dots, s$ .
- $K$  é uma clique maximal em cada  $G_{V_i \cup K}$ , qualquer outra clique de  $G$  está em um e somente em um dos subgrafos  $G_{V_i \cup K}$ .

GAVRIL (1977) utilizou a decomposição por cliques para determinar o número cromático e a maior clique de grafos separáveis por cliques.

É possível caracterizar os átomos de alguns grafos interseção de caminhos, através de multigrafos.

**Definição IV.1:**

Uma árvore-estrela é um grafo bipartite conexo, onde um dos subconjuntos de vértices contém somente um vértice. É comum representá-lo na forma da figura (IV.8).

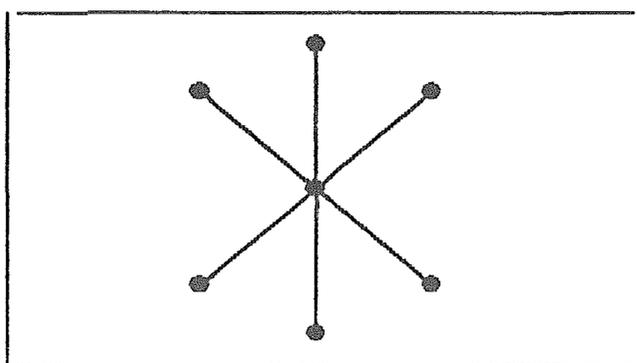


Figura IV.8.

**Propriedade IV.4: Teorema dos Átomos**

a) Os grafos linha de multigrafos são exatamente os grafos UE com representação numa árvore característica que é uma árvore-estrela. Todo átomo UE tem representação numa árvore-estrela.

b) Os grafos linha de multigrafos livres de triângulos são exatamente os grafos UEH, cuja árvore característica é uma árvore-estrela. Todo átomo UEH tem uma representação por uma árvore-estrela.

c) Os grafos linha de multigrafos bipartite são exatamente os grafos DE com representação em árvore-estrela. Todo átomo DE tem uma representação numa árvore-estrela.

d) Um grafo que consiste somente de uma clique maximal ou de duas cliques maximais que se interceptam, é um grafo RDV.

Todo átomo cordal consiste de uma clique ou de duas cliques que se interceptam. A árvore característica destes átomos é um vértice ou dois vértices ligados por uma aresta, respectivamente.

Com isto, tem-se que os átomos de grafos UE, UEH, DE e cordais, produzidos por uma separação do grafo através de cliques maximais, podem ser reconhecidos de forma eficiente. Este fato, junto com uma caracterização dos diferentes grafos em relação a sua decomposição por cliques maximais, dão a idéia para um algoritmo de reconhecimento dos grafos interseção de caminhos. A caracterização será dada a seguir no teorema de separação, mas antes é preciso dar algumas definições.

#### Definição IV.2:

Seja  $K$  uma clique maximal que separa  $G$  em  $G_i = G_{V_i} \cup K$  para  $i = 1, 2, \dots, s$ .

Uma clique maximal  $C$  que intercepta  $K$ , mas  $C \neq K$ , é chamada *clique relevante*.

Duas cliques relevantes  $C_1$  e  $C_2$  são *soltas* se  $(C_1 \cap K) \cap (C_2 \cap K) = \emptyset$ . Denota-se  $C_1 | C_2$ .

$C_1$  *domina*  $C_2$  se  $(C_2 \cap K) \subseteq (C_1 \cap K)$ . Denota-se  $C_1 \geq C_2$ .

$C_1$  e  $C_2$  são *antipodais* se elas não são soltas e nenhuma delas domina a outra. Denota-se  $C_1 \leftrightarrow C_2$ .

Dois subgrafos filhos  $G_1$  e  $G_2$  são *soltos* se  $C_1 | C_2 \forall C_1$  em  $G_1$  e  $\forall C_2$  em  $G_2$ . Denota-se  $G_1 | G_2$ .

$G_1$  *domina*  $G_2$  se eles são soltos e  $\forall C_1$  em  $G_1$  tem-se ou  $C_1 \geq C_2 \forall C_2$  em  $G_2$  ou  $C_1 | C_2 \forall C_2$  em  $G_2$ . Denota-se  $G_1 \geq G_2$ .

$G_1$  e  $G_2$  são um *par antipodal* se eles não são soltos e

nenhum deles domina ao outro. Denota-se  $G_1 \leftrightarrow G_2$ .

$G_i$  é um subgrafo vizinho de  $v \in K$  se existe pelo menos um vértice em  $V_i$  que seja adjacente a  $v$ .

*Propriedade IV.5: Teorema de Separação*

Seja  $K$  uma clique maximal de  $G$  que o separa em  $G_1, G_2, \dots, G_s$  com  $s \geq 2$ .

a)  $G$  é um grafo DV se e somente se cada  $G_i$  é DV e na árvore característica os  $G_i$  podem ser 2-coloridos de forma tal que, para nenhum par antipodal os dois elementos têm a mesma cor.

b)  $G$  é um grafo DE se e somente se cada  $G_i$  é DE e os  $G_i$  podem ser 2-coloridos de forma que em nenhum par antipodal os dois elementos tenham a mesma cor.

c)  $G$  é um grafo UEH se e somente se cada  $G_i$  é UEH e os  $G_i$  podem ser 2-coloridos de forma tal que, para nenhum par antipodal os dois elementos tenham a mesma cor.

d)  $G$  é um grafo UV se e somente se cada  $G_i$  é UV e os  $G_i$  podem ser coloridos de forma que, para nenhum par antipodal os dois elementos têm a mesma cor e para cada  $v \in K$ , o conjunto de subgrafos vizinhos de  $v$  é 2-coloravel.

e)  $G$  é um grafo RDV se e somente se cada  $G_i$  é RDV e os  $G_i$  podem ser coloridos de forma que, para nenhum par antipodal os dois elementos têm a mesma cor e que numa cor todo subgrafo tem uma árvore característica RDV enraizada em  $K$  e na outra cor nenhum par de subgrafos são soltos e todo subgrafo (com possivelmente uma exceção) tem uma árvore característica enraizada numa clique relevante. O subgrafo exceção, se existe, é dominado por qualquer outro subgrafo da mesma cor e tem uma árvore característica RDV na qual o vértice  $K$  tem grau de saída zero.

f)  $G$  é cordal se e somente se cada  $G_i$  é cordal.

g)  $G$  é um grafo cordal e UE se e somente se cada  $G_i$  é cordal e UE e os  $G_i$  podem ser 3-coloridos de forma tal que, para nenhum par antipodal os dois elementos têm a mesma cor e para cada  $v \in K$  o conjunto de subgrafos vizinhos de  $v$  é 2-coloravel.

Um algoritmo de reconhecimento para os grafos interseção de caminhos é:

*Algoritmo IV.3: RECONHECIMENTO GRAFOS INTERSEÇÃO DE CAMINHOS*

- Dado  $G=(V,E)$
- obter uma árvore  $T_G$  de decomposição por cliques
- verificar quais os átomos que satisfazem o teorema dos átomos
- verificar se cada separação na decomposição satisfaz o teorema de separação.

Para verificar as condições do teorema de separação, este passo pode fazer-se na seguinte forma:

Determinar as cliques maximais dos subgrafos.

Decidir antipolaridade entre todos os pares de cliques maximais.

Decidir antipolaridade entre subgrafos.

A complexidade do algoritmo de reconhecimento será: passo 1 é executado pelo algoritmo (IV.1) em tempo  $O(nmr)$ , o passo 2 pode ser realizado em tempo polinomial. O passo 3 deve ser realizado no máximo  $n$  vezes, pois em cada separação há no máximo  $|V|$  subgrafos. Logo, a complexidade dependerá da dificuldade para verificar o teorema de separação em cada tipo de grafo.

O teorema dos átomos dá uma caracterização somente para os átomos de grafos UE, UEH, DE e triangularizados.

Considerando o fato de que a classe dos grafos DV é a interseção dos grafos triangularizados e os grafos UEH, também é possível a caracterização para estes grafos. Com isto, o algoritmo geral de reconhecimento fica totalmente explicitado para grafos destas classes. Para os grafos UV e RDV são considerados os algoritmos desenvolvidos por Gavril, os quais são *polinomiais* no tamanho do grafo.

#### IV.4. Reconhecimento de Grafos Interseção de Caminhos

##### IV.4.1. Grafos UV

A cada clique maximal de  $G$  está associado um vértice de  $T$ . A idéia do algoritmo é construir esta árvore com um processo iterativo. Ele começa com uma árvore  $T_1$  sem vértices. Em cada passo contrói-se  $T_{m+1}$  a partir de  $T_m$  acrescentando vértices de  $T$  ainda não considerados e constrói-se os caminhos característicos que satisfazem as condições do teorema da árvore de cliques. Para isto, é aproveitado o fato de que se  $G$  é UV então  $\forall v \in V$ , o subgrafo induzido por  $\{v\} \cup \text{Adj}(v)$  é um grafo intervalo. Além disso, como todo grafo UV é cordal, o algoritmo primeiro verifica que o grafo de interesse seja cordal.

O algoritmo pode escrever-se em forma geral, da seguinte maneira:

#### Algoritmo IV.4: ALGORITMO GERAL RECONHECIMENTO GRAFOS UV

- Dado  $G=(V,E)$  e  $n=|V|$
- se  $G$  não é cordal então devolver  $G$  não é UV
- caso contrário efetuar
  - determinar conjunto de cliques maximais de  $G$
  - associar a cada clique maximal um vértice em  $T$
  - partir com um vértice  $v$  qualquer
  - $T_1 = \emptyset$
  - $m = 1$

- enquanto tiver vértices de  $T$  a acrescentar a  $T_m$
- efetuar
  - construir o caminho associado de  $\bar{v}$ 
    - se não for possível então devolver
    - $G$  não é UV
  - construir  $T_{m+1}$  a partir de  $T_m$  e  $\bar{v}$
  - $m = m + 1$
- devolver  $G$  é grafo UV com árvore característica  $T_m$ .

Para detalhar mais os passos do algoritmo e as condições que devem satisfazer os caminhos característicos para que o grafo seja UV, é necessário dar algumas definições e estabelecer umas restrições.

**Definição IV.3:**

$\mathcal{C} = \{c_1, c_2, \dots, c_r\}$ , conjunto de cliques maximais de  $G$ .

$\bar{\mathcal{C}} = \{\bar{c}_1, \bar{c}_2, \dots, \bar{c}_r\}$ , conjunto de vértices da árvore característica  $T$ .

Todo  $\bar{\chi} \in \bar{\mathcal{C}}$  está associado a uma clique  $\chi \in \mathcal{C}$ , todo  $v \in V$  está associado a um caminho  $\bar{v}$  em  $T$ .

Para todo  $v \in V$  define-se:

$\mathcal{C}_v = \{c \in \mathcal{C} / v \in c\}$ , conjunto de cliques maximais de  $G$  que contém ao vértice  $v$ ,  $V_v = \{v\} \cup \text{Adj}(v)$ ,  $G_v = G_{V_v}$ , subgrafo induzido por  $V_v$ . Tem-se que  $\mathcal{C}_v$  é exatamente o conjunto de cliques maximais de  $G_v$ ,  $\bar{\mathcal{C}}_v = \{\bar{c} \in \bar{\mathcal{C}} / v \in c\}$ .

No passo  $m$  tem-se:

$T_m =$  árvore no passo  $m$ ,  $\bar{\mathcal{C}}_m = \{\text{vértices de } T_m\}$ ,  $V_m = \bigcup_{\alpha \in \mathcal{C}_m} \bar{a}$ ,  $G_m = G_{V_m}$ , subgrafo induzido por  $V_m$ ,  $\bar{\mathcal{C}}_v = \bar{\mathcal{C}}_v \cap \bar{\mathcal{C}}_m$ .

$A = \{\bar{a} \in \bar{\mathcal{C}} - \bar{\mathcal{C}}_v / \bar{a} \text{ é adjacente em } T_m \text{ a um elemento de } \bar{\mathcal{C}}_v\}$ .

$\forall \bar{a} \in A$  define-se:

$B_a = \{\bar{b} \in \bar{\mathcal{C}}_v / b \cap a \neq \emptyset\}$ ,  $U_a = \{\bar{b} \in B_a / \forall \bar{b}' \in B_a: (b' \cap a) \subseteq (b \cap a)\}$ ,  $\bar{U}_a = B_a - U_a$ .

$$A_1 = \{ \bar{a} \in A \mid \exists \bar{b}_1, \bar{b}_2 \in B_\alpha \text{ tal que } (b_1 \cap a) \cap (b_2 \cap a) = \emptyset \}.$$

$$A_2 = \{ \bar{a} \in A \mid \bar{U}_\alpha = \emptyset \text{ e } \bar{b}_1', \bar{b}_2' \in B_\alpha \text{ tal que } (b_1 \cap a) \cap (b_2 \cap a) = \emptyset \}.$$

$$A_3 = \{ \bar{a} \in A \mid \bar{U}_\alpha = \emptyset \text{ e } \bar{b}_1', \bar{b}_2' \in B_\alpha \text{ tal que } (b_1 \cap a) \cap (b_2 \cap a) = \emptyset \}.$$

$$Z = \{ \bar{a} \in \bar{\mathcal{E}}_m - \bar{c}_v \mid \bar{a} \text{ é adjacente em } T_m \text{ a um elemento de } \bar{c}_v \}.$$

$$H = G(A, E_h) \text{ onde } E_h = \{ (\bar{a}, \bar{b}) \mid \text{tem-se } (a) \text{ ou } (b) \} \text{ com:}$$

(a)  $(a \cap V_v) \subseteq (b \cap V_v)$ ,  $(b \cap V_v) \subseteq (a \cap V_v)$  e  $\exists u \in a \cap b \cap V_v$  tal que  $|\mathcal{E}_u \cap \mathcal{E}_v| > 1$ .

(b)  $\bar{a} \in Z$ ,  $(b \cap V_v) \subseteq (a \cap V_v)$  e  $\exists u \in a \cap b \cap V_v$  tal que  $|\mathcal{E}_u \cap \mathcal{E}_v| > 1$ .

Para  $\bar{a}, \bar{b} \in A$  tem-se um caminho em  $H$ :  $\bar{a} = \bar{a}_0, \bar{a}_1, \dots, \bar{a}_k$  e  $\bar{a}_k = \bar{b}$ ,  $k \geq 1$ , tal que  $\forall 1 \leq i \leq k-1$ :  $\bar{a}_i \in A_3$ .

Se  $k$  é ímpar, então  $\bar{a}$  e  $\bar{b}$  estão em relação  $\alpha(a, b)$ , se  $k$  é par, então  $\bar{a}$  e  $\bar{b}$  estão em relação  $\beta(a, b)$ .

Para todo  $\bar{a} \in A_1$ :  $b_\alpha =$  único elemento de  $U_\alpha$ .

Para todo  $\bar{a} \in Z$ :  $b_\alpha =$  ponto união de  $\bar{a}$  com  $T_m(\bar{c}_v)$ .

$$P_1 = \{ (\langle \bar{b} \rangle, \bar{c}_v) \mid \bar{c} \in \bar{\mathcal{E}}_v - \bar{c}_v \}.$$

$$P_2 = \{ (U_\alpha, \bar{U}_\alpha) \mid \bar{a} \in A_2 \}.$$

$$P_3 = \{ (\langle \bar{b}_\alpha \rangle, U_c), (\langle b_\alpha \rangle, \bar{U}_c) \mid \bar{c} \in A_2, \bar{a} \in A_1 \cup Z: \alpha(a, c), b_\alpha \in U_c \}.$$

$$P_4 = \{ (\langle b_\alpha \rangle, U_c), (\langle b_\alpha \rangle, \bar{U}_c) \mid \bar{c} \in A_2, \bar{a} \in A_1 \cup Z: \beta(a, c) \}.$$

$$P_5 = \{ (\langle b_\alpha \rangle, \mathcal{E}_u \cap \mathcal{E}_v) \mid \bar{a} \in Z, u \in \bar{a} \cap b_\alpha \}.$$

$$Q_1 = \{ (\langle U_\alpha, \bar{U}_\alpha \rangle, \langle U_c, \bar{U}_c \rangle) \mid \bar{a}, \bar{c} \in A_2: \alpha(a, c) \}.$$

$$Q_2 = \{ (\langle U_a, \bar{U}_a \rangle, \langle U_c, \bar{U}_c \rangle) / \bar{a}, \bar{c} \in A_2: \beta(a, c) \}.$$

$$Q_3 = \{ (\langle \langle b_a \rangle, U_c \rangle, \langle \langle b_a \rangle, \bar{U}_c \rangle) / \bar{c} \in A_2, \bar{a} \in A_1 \cup Z: \alpha(a, c), b_a \in U_c \}.$$

$$Q_4 = \{ (\langle \langle b_a \rangle, \bar{U}_c \rangle, \langle \bar{U}_c, \langle b_a \rangle \rangle) / \bar{c} \in A_2, \bar{a} \in A_1 \cup Z: \beta(a, c) \}.$$

Para todo  $\bar{a} \in A$ ,  $H_a$ , subgrafo de  $H$  com conjunto de vértices  $V_{H_a} = \{\bar{a}\} \cup \{\bar{b} \in A_3 / \bar{b} \text{ é atingível a partir de } \bar{a} \text{ só com vértices de } A_3\}$ , é bipartite:  $V_{H_a} = M_a + N_a$  com  $\bar{a} \in M_a$ .

Dado  $v \in V$ ,  $a \in \mathcal{E}_v$  e  $G_v$  com caminho característico  $p$ ,  $u \subseteq a$ :  $p(\mathcal{E}_u \cap \mathcal{E}_v)$  subcaminho de  $p$  com vértice em  $\mathcal{E}_u \cap \mathcal{E}_v$ ,  $f(p, a) = \langle p(\mathcal{E}_u \cap \mathcal{E}_v) / u \in a \cap V_v \rangle$ .

*Observação IV.1:*

Nas definições feitas para desenvolver o algoritmo tem-se as seguintes interpretações:

O conjunto  $A$  corresponde às cliques maximais de  $G$  que contém o vértice  $v$  corrente e que já foram incluídas na árvore.

$\bar{C}_v$  são os vértices da árvore associados às cliques maximais que contém  $v$  e que já foram incorporados a ela.

$B_a$  são vértices da árvore atual, cujas cliques associadas contém  $v$  e interceptam a clique  $a$ .

Os elementos  $(\bar{x}, \bar{y})$  dos conjuntos  $P_i$ ,  $i = 1, 2, \dots, s$ , são tais, que no caminho característico, todos os elementos de  $\bar{x}$  devem estar à esquerda de cada um dos elementos de  $\bar{y}$ .

Os elementos  $(\langle \bar{x}, \bar{y} \rangle, \langle \bar{z}, \bar{w} \rangle)$  dos conjuntos  $Q_i$ ,  $i = 1, 2, 3$  e  $4$ , são tais que, no caminho característico, todos os elementos de  $\bar{x}$  devem estar a um mesmo lado de todos os elementos de  $\bar{y}$  (todos à esquerda ou todos à direita) e todos os elementos de  $\bar{z}$  devem estar do lado contrário (que estão

os elementos de  $\bar{x}$  com respeito aos de  $\bar{y}$ ) de todos os elementos de  $\bar{w}$ .

Para todo  $\bar{a} \in A$ , existe um único caminho minimal em  $T$ , que liga  $\bar{a}$  com  $p$ , denotado  $s(p, a)$ . Se  $\bar{a} \in A_1$ , então existem  $b_1$  e  $b_2 \in B_a$  tais que  $(b_1 \cap a) \cap (b_2 \cap a) = \emptyset$ . No caminho  $p$ ,  $b_1$  deve estar a um lado de  $s(p, a)$  e  $b_2$  deve estar ao outro lado. Se  $\bar{a} \in A_2 \cup A_3$ , então todos os elementos de  $B_a$  devem estar a um mesmo lado de  $s(p, a)$ .

Se  $\bar{a}$  e  $\bar{c} \in A$  são adjacentes no grafo  $H$ , então  $s(p, a) \neq s(p, c)$ . Caso contrário, com a existência de  $\alpha(a, c)$ ,  $s(p, a) \neq s(p, c)$  e de  $\beta(a, c)$ ,  $s(p, a) = s(p, c)$ .

Se  $\bar{a} \in A_1$ , então o único elemento de  $U_a$  é o único ponto final comum para  $f(p, a)$ . Se  $\bar{a} \in A_2$ , o ponto final de  $p(U_a)$ , que é diferente daquele que une  $p(U_a)$  e  $p(\bar{U}_a)$ , é o único ponto final comum para  $f(p, a)$ . Se  $\bar{a} \in A_3$ , então os dois vértices finais de  $p(U_a)$  são pontos finais de  $f(p, a)$ .

São estabelecidas as seguintes restrições:

a) Para todo  $\bar{a} \in A$ :  $|U_a| \geq 1$ .

b) Se  $\bar{a} \in A_1$ , então  $|U_a| = 1$  e  $B_a$  contém  $\bar{b}_1$  e  $\bar{b}_2$  tais que,  $\forall \bar{b} \in B_a - U_a$  tem-se  $(b \cap a) \subseteq (b_1 \cap a)$  ou  $(b \cap a) \subseteq (b_2 \cap a)$ .

c) Para todo  $\bar{a}, \bar{c} \in A$ :  $\alpha(a, c)$  e  $\beta(a, c)$  não ocorrem simultaneamente.

Para todo  $\bar{a}, \bar{c} \in A_1$ : se  $\alpha(a, c)$ , então  $b_a \neq b_c$ ,  
se  $\beta(a, c)$ , então  $b_a = b_c$ .

Para  $\bar{a} \in A_1$  e  $\bar{c} \in Z$ : se  $\alpha(a, c)$ , então  $b_a \neq b_c$ ,  
se  $\beta(a, c)$ , então  $b_a = b_c$ .

d) Para  $\bar{a} \in A_1 \cup Z$  e  $\bar{c} \in A_2$ , se  $\beta(a, c)$ , então  $b_a \in U_c$ .

**Algoritmo IV.5: RECONHECIMENTO GRAFOS UV**

- Dado  $G=(V,E)$ ,  $n=|V|$
- se  $G$  não é cordal então devolver  $G$  não é UV
- caso contrário efetuar
  - determinar  $\bar{C} = \{\bar{c}_1, \dots, \bar{c}_r\}$
  - para  $i = 1, \dots, n$  efetuar
    - determinar  $\mathcal{C}_{V_i} = \{c_k \in \mathcal{C} / v_i \in c_k\}$
    - contruir  $V_{V_i} = \{v_i\} \cup \text{Adj}(v_i)$
    - construir  $\bar{\mathcal{C}}_{V_i} = \{\bar{c}_k / c_k \in \mathcal{C}_{V_i}\}$
  - $m = 1$
  - enquanto  $\bar{\mathcal{C}}_m \neq \bar{\mathcal{C}}$  efetuar
    - se  $m = 1$  efetuar
      - escolher  $v \in V$  qualquer
      - $\bar{c}_v = \bar{\mathcal{C}}_v$
      - $A = \bar{\mathcal{C}} - \bar{\mathcal{C}}_v$
    - caso contrário efetuar
      - escolher  $v \in V$  tal que  $\bar{\mathcal{C}}_m \cap \bar{\mathcal{C}}_v \neq \emptyset$  e  $\bar{\mathcal{C}}_v \subset \bar{\mathcal{C}}_m$
      - $\bar{c}_v = \bar{\mathcal{C}}_v \cap \bar{\mathcal{C}}_m$
    - determinar  $A$
    - para todo  $\bar{a} \in A$  determinar  $B_a, U_a, \bar{U}_a$
    - determinar  $A_1, A_2, A_3, Z, H$ , relações  $\alpha$  e  $\beta$
    - se alguma das restrições não se verifica então devolver  $G$  não é grafo UV
    - para todo  $\bar{a} \in A$ , determinar  $b_a$
    - para todo  $\bar{a} \in Z$  determinar  $b_a$
    - construir  $P_1, P_2, P_3, P_4, P_5, Q_1, Q_2, Q_3, Q_4$
    - $P = \bigcup_{i=1}^5 P_i$
    - $Q = \bigcup_{i=1}^4 Q_i$
    - para todo  $(x,y) \in P$  efetuar
      - se  $(y,x) \notin P$  então  $P = P \cup \{(y,x)\}$
    - para todo  $(\langle x,y \rangle, \langle z,w \rangle) \in Q$  efetuar
      - $Q = Q \cup \{(\langle z,w \rangle, \langle x,y \rangle), (\langle y,x \rangle, \langle w,z \rangle), (\langle w,z \rangle, \langle y,x \rangle)\}$
    - determinar  $\bar{G}_v =$  grafo complemento de  $G_{V_v}$
    - construir  $p$  com algoritmo (IV.6)

- se não for possível então devolver  $G$  não é grafo UV
- $\bar{T} = T_m$
- $\bar{p} = \bar{T}(\bar{z}_v)$
- $X = \{w \in \bar{z} - \bar{z}_v \mid w \text{ adjacente, em } \bar{T}, \text{ a elementos de } \bar{p}\}$
- para todo  $\bar{a} \in X \cap (A_1 \cup A_2 \cup Z)$  efetuar
  - determinar  $H_a, M_a, N_a$
  - substituir  $T_m(\bar{c}_v)$  por  $p$  usando o algoritmo (IV.7) / devolve  $T_{m+1}$
- $m = m+1$
- determinar  $\bar{z}_m$
- devolver  $G$  é grafo UV com árvore característica  $T_m$

A complexidade deste algoritmo é  $O(n^4)$ .

#### Algoritmo IV.6: CONSTRUÇÃO DO CAMINHO P

- Dado  $\bar{G} = (V_v, \bar{E}_v)$ ,  $T_m$ ,  $P$ ,  $Q$ ,  $\bar{c}_v$
- $E_1 = E_2 = E_3 = R = \emptyset$
- para todo  $(u, w) \in \bar{E}_v$  efetuar
  - se  $\exists \bar{a}, \bar{c} \in \bar{c}_v$  tal que  $u \in \bar{a}, w \in \bar{c}$  e  $w$  está à esquerda de  $u$  em  $T_m(\bar{c}_v)$  então efetuar
    - para  $i = 1, 2, 3$  efetuar
      - $E_i = E_i \cup \{u \rightarrow w\}$
- enquanto existirem arestas de  $\bar{E}_v$  não orientadas efetuar
  - escolher  $(u, w) \in \bar{E}_v$  tal que  $u \rightarrow w \notin E_1$  e  $w \rightarrow u \notin E_1$
  - para  $i = 1, 2, 3$  efetuar
    - $E_i = E_i \cup \{u \rightarrow w\}$
- enquanto acrescentar arestas a  $E_3$  efetuar
  - enquanto  $E_3 \neq \emptyset$  efetuar
    - escolher  $s \rightarrow u \in E_3$
    - para toda  $(u, w) \in \bar{E}_v$  não orientada tal que  $(s, w) \notin \bar{E}_v$  efetuar
      - se  $w \rightarrow u \notin E$ , então efetuar
        - para  $i = 1, 2, 3$  efetuar
          - $E_i = E_i \cup \{w \rightarrow u\}$

- para toda  $(s, t) \in \bar{E}_V$  não orientada tal que  $(t, u) \notin \bar{E}_V$  efetuar
  - se  $s \rightarrow t \notin E_1$  então efetuar
    - para  $i = 1, 2, 3$  efetuar
      - $E_i = E_i \cup \{s \rightarrow t\}$
  - $E_3 = E_3 - \{s \rightarrow u\}$
- enquanto  $E_2 \neq \emptyset$  efetuar
  - escolher  $u \rightarrow w \in E_2$
  - para todo  $(\bar{x}, \bar{y}) \in P$  tal que  $\exists u \in a \in X, w \in b \in Y$  efetuar
    - $R = R \cup \{(\bar{x}, \bar{y})\}$
    - $P = P - \{(\bar{x}, \bar{y})\}$
    - $E_2 = E_2 - \{u \rightarrow w\}$
  - para todo  $(\langle \bar{x}, \bar{y} \rangle, \langle \bar{z}, \bar{w} \rangle) \in Q$  tal que  $(\bar{x}, \bar{y}) \in R$  efetuar
    - $R = R \cup \{(\bar{w}, \bar{z})\}$
    - $Q = Q - \{(\langle \bar{x}, \bar{y} \rangle, \langle \bar{z}, \bar{w} \rangle)\}$
  - se  $R \neq \emptyset$  então efetuar
    - escolher  $(\bar{x}, \bar{y}) \in R$
    - para todo  $(u, w) \in \bar{E}_V$  tal que  $u \in a \in X, w \in b \in Y$  efetuar
      - se  $u \rightarrow w \notin E_1$  então efetuar
        - para  $i = 1, 2, 3$  efetuar
          - $E_i = E_i \cup \{u \rightarrow w\}$
    - se existe  $(u, w) \in \bar{E}_V$  tal que  $u \rightarrow w, w \rightarrow u \in E_1$  então devolver  $p$  não pode ser construído
    - para todo  $(\bar{x}, \bar{y}) \in R$  efetuar
      - para todo  $a \in \bar{x}$  e todo  $b \in \bar{y}$ ,  $a$  deve estar à esquerda de  $b$  em  $p$
    - devolver  $p$ .

A complexidade deste algoritmo é  $O(n^4)$ .

#### Observação IV.2:

Em cada passo do algoritmo,  $E_1$  contém as arestas já orientadas de  $\bar{G}_V$ ,  $E_3$  contém as arestas às quais será aplicada a regra de orientação transitiva,  $R$  contém pares de subconjuntos  $(\bar{x}, \bar{y})$  de  $\mathcal{S}_V$ , tais que todo elemento de  $\bar{x}$  deve

estar à esquerda de todo elemento de  $\bar{y}$  no caminho característico e  $E_2$  contem as arestas orientadas pelas quais são determinados os pares do caminho com relação a R.

*Algoritmo IV.7: SUBSTITUIÇÃO DE CAMINHO*

- Dado  $\bar{T}$ ,  $\bar{p}$ ,  $p$ ,  $X$ ,  $H_a$ ,  $Z$ ,  $A_1$ ,  $A_2$
- apagar de  $\bar{T}$  o caminho  $\bar{p}$  e as arestas incidentes a ele e substituí-lo pelo caminho  $p$
- para todo  $\bar{a} \in X \cap Z$ , ligar  $\bar{a}$  com o elemento de  $p$  que o liga com  $\bar{T}$
- para todo  $\bar{a} \in X \cap (A_1 \cap A_2)$ , ligar  $\bar{a}$  com o único ponto final comum de  $f(p, a)$
- para todo  $\bar{a} \in X \cap (A_1 \cap A_2 \cap Z)$  efetuar
  - para todo  $\bar{c} \in X \cap M_a - \{a\}$ , ligar  $\bar{c}$  com  $b_a$
  - para todo  $\bar{c} \in X \cap N_a$ , ligar  $\bar{c}$  ao ponto final de  $p(U_c)$  distinto de  $b_a$
  - $X = X - (M_a \cup N_a)$
- enquanto  $X \neq \emptyset$  efetuar
  - se  $\exists \bar{a} \in X$  tal que um ponto final de  $p(U_a)$  está ligado a  $\bar{c} \in X$  com  $c \cap a \cap V_v \neq \emptyset$  então efetuar
    - para todo  $\bar{d} \in X \cap N_a$ , ligar  $\bar{d}$  a  $b_c$
    - para todo  $\bar{d} \in X \cap M_a$ , ligar  $\bar{d}$  ao outro ponto final de  $p(U_a)$
    - $X = X - (M_a \cup N_a)$
  - caso contrário efetuar
    - escolher  $\bar{a} \in X$
    - para todo  $\bar{d} \in X \cap M_a$  ligar  $\bar{d}$  a um ponto final de  $p(U_a)$
    - para todo  $\bar{d} \in X \cap N_a$  ligar  $\bar{d}$  ao outro ponto final de  $p(U_a)$
    - para todo  $\bar{d} \in X \cap N_a$  ligar  $\bar{d}$  ao outro ponto final de  $p(U_a)$
    - $X = X - (M_a \cup N_a)$
    - devolver  $\bar{T}$ .

A complexidade é  $O(n)$ .

O grafo da figura (IV.9) tem as cliques maximais  $c_1 = \{v_2, v_3, v_4\}$ ,  $c_2 = \{v_2, v_4, v_5\}$  e  $c_3 = \{v_1, v_2, v_5\}$ . A árvore característica terá três vértices  $\bar{c}_1$ ,  $\bar{c}_2$  e  $\bar{c}_3$ . No primeiro passo do algoritmo seleciona-se o vértice  $\bar{c}_3$ , logo ele é ligado com  $\bar{c}_2$  e finalmente  $\bar{c}_2$  é ligado com  $\bar{c}_1$ . A árvore característica  $T$  é dada na mesma figura (IV.9).

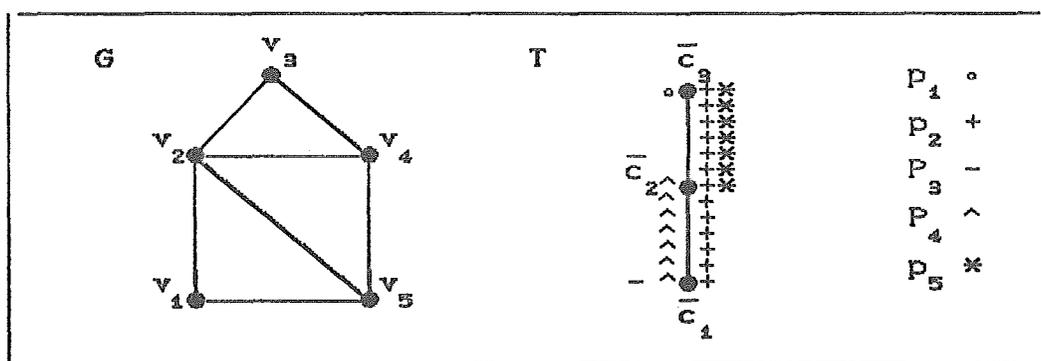


Figura IV.9.

#### IV.4.2. Grafos RDV e RDE

O algoritmo de reconhecimento para estes grafos é análogo ao algoritmo para grafos UV, levando em consideração as diferenças entre as duas classes de grafos e entre suas árvores características.

Considerando que todo grafo RDV é cordal, o algoritmo, ao começar, verifica que o grafo de interesse seja cordal.

A idéia do algoritmo é, associar a cada clique maximal  $c$  de  $G$ , um vértice  $\bar{c}$  na árvore característica  $T$  e construir esta árvore em forma iterativa, a partir de um vértice raiz e da árvore  $T_1$  que contém apenas este vértice. Em cada passo, constrói-se  $T_{m+1}$  a partir de  $T_m$ , examinando os possíveis vértices de  $T$  a acrescentar a  $T_m$  e determinando as arestas correspondentes de tal forma que satisfaçam o teorema da árvore de cliques. Se em algum passo, não é possível construir esta subárvore de acordo com as

restrições do teorema, então o grafo não é RDV. Outra condição de parada é se nenhum vértice pode ser raiz para construir a árvore, neste caso o grafo também não é RDV.

O algoritmo pode ser escrito da forma geral seguinte:

**Algoritmo IV.8: ALGORITMO GERAL RECONHECIMENTO DE GRAFO RDV**

- Dado  $G$
- se  $G$  não é cordal então devolver  $G$  não é RDV
- caso contrário efetuar
  - determinar conjunto de cliques maximais de  $G$
  - associar a cada clique maximal um vértice em  $T$
  - começar com um vértice  $\bar{c}$  como raiz
  - $m=1$
  - enquanto tiver vértices de  $T$  a acrescentar a  $T_m$  efetuar
    - construir  $T_{m+1}$  se é possível
    - se não for possível então começar com outro vértice como raiz
    - $m= m+1$
  - se não tiver mais vértices para ser raiz e nenhum deles é viável então devolver  $G$  não é RDV
  - caso contrário devolver  $G$  é grafo RDV com árvore característica  $T_m$ .

Para explicitar mais o algoritmo e as condições que a árvore característica deve satisfazer, são necessárias algumas definições e desenvolver a construção dos caminhos.

**Definição IV.4:**

No desenvolvimento do algoritmo, considera-se:

$\mathcal{C} = \{c_1, c_2, \dots, c_r\}$ , conjunto de cliques maximais de  $G$ ,  
 $\bar{\mathcal{C}} = \{\bar{c}_1, \bar{c}_2, \dots, \bar{c}_r\}$ , conjunto de vértices da árvore  $T$ .

Todo  $\bar{x} \in \bar{\mathcal{E}}$  está associado a uma clique maximal  $x \in \mathcal{E}$ , todo  $v \in V$  está associado a um caminho  $\bar{v} \in T$ .

No passo  $m$  do algoritmo tem-se:

$T_m$ : árvore característica no passo  $m$ ,  $\bar{\mathcal{E}}_m = \{\text{vértices de } T_m\}$ ,  $J = \{\text{vértices de } T_m \text{ a partir dos quais é possível continuar a construção de } T_{m+1}\}$ .

Dado  $\bar{x} \in J$ :

$$A_x^- = \{ \bar{b} \in \bar{\mathcal{E}} / \bar{b} \notin \bar{\mathcal{E}}_m, b \cap x \neq \emptyset, \forall \bar{a} \in \bar{\mathcal{E}}_m \text{ sucessor de } \bar{x} \text{ em } T_m : a \cap x = \emptyset \text{ e } b \cap x \text{ é maximal no conjunto } \{a \cap x / a \in \mathcal{E} - \mathcal{E}_m\} \}.$$

Partição de  $A_x^-$ :  $A_x^- = A_1 + A_2 + \dots + A_s$ , tal que  $\bar{a}$  e  $\bar{b}$  pertencem à mesma classe  $A_i$  se  $a \cap x = b \cap x$ , para algum  $i$  com  $1 \leq i \leq s$ .

Para cada classe  $A_k$  subconjunto de  $A_x^-$  e para cada  $\bar{a}_j^k \in A_k$ , define-se  $D_j^k = \{ \bar{b} \in \bar{\mathcal{E}} - (\bar{\mathcal{E}}_m \cup A_x^-) / b \cap a_j^k \neq \emptyset \}$ .

Para a subárvore  $T_m$  tem-se as seguintes restrições:

- Os elementos de  $A_k \cup \{\bar{x}\}$  formam um caminho direcionado  $p_k$ , para  $k = 1, 2, \dots, s$ . Deve-se verificar que  $\forall \bar{a} \in A_{k_1}, \bar{b} \in A_{k_2}$  com  $k_1 \neq k_2$ :  $a \cap b = \emptyset$ .
- Para todo  $\bar{b} \in T_m$  que não seja predecessor de  $\bar{x}$  e todo  $\bar{a} \in A_k$  com  $k = 1, 2, \dots, s$ , deve-se ter  $a \cap b = \emptyset$ .
- Para todos  $\bar{a}_{j_1}^k, \bar{a}_{j_2}^k \in A_k$  com  $k = 1, 2, \dots, s$ , deve-se ter que  $\forall \bar{b} \in D_{j_1}^k \cap D_{j_2}^k : \exists (a_{j_1}^k \cap b) \subseteq (a_{j_2}^k \cap b)$  ou que  $\forall \bar{b} \in D_{j_1}^k \cap D_{j_2}^k : \exists (a_{j_2}^k \cap b) \subseteq (a_{j_1}^k \cap b)$ .

### Construção do Caminho $p_k$ :

Seja  $\bar{p}_k$  o subconjunto de  $p_k$  que conterá apenas elementos de  $A_k$ ,  $p_k = \{(\bar{a}_{j_1}^k, \bar{a}_{j_2}^k) \mid \exists \bar{b} \in D_{j_1}^k \cap D_{j_2}^k \text{ tal que } (\bar{a}_{j_1}^k \cap \bar{b}) \subset (\bar{a}_{j_2}^k \cap \bar{b})\}$  e os elementos de  $D_{j_1}^k \cap D_{j_2}^k$  são sucessores de  $\bar{a}_{j_1}^k$  e de  $\bar{a}_{j_2}^k$ , logo  $\bar{a}_{j_1}^k$  deve ser predecessor de  $\bar{a}_{j_2}^k$  em  $\bar{p}_k$ .

Se  $(\bar{a}_{j_1}^k, \bar{a}_{j_2}^k) \notin P_k$  então  $(\bar{a}_{j_2}^k, \bar{a}_{j_1}^k) \notin P_k$ . Considera-se  $p_k$  uma reta da esquerda à direita.

Seja  $G_k$  o subgrafo de  $G$  definido por  $V_k = \bigcup_{c \in A_k} c$ , para cada  $v \in V$ :  $A_k^v = \{\bar{a} \in A_k \mid v \in \bar{a}\}$ ,  $T(A_k^v)$  deve ser um caminho direcionado em  $\bar{p}_k$ , logo  $G_k$  deve ser o grafo de intervalo da família  $F_k = \{T(A_k^v) \mid v \in V\}$ .

O problema é determinar uma representação por intervalos para  $G_k$ , tal que  $\forall (\bar{a}_{j_1}^k, \bar{a}_{j_2}^k) \in P_k$ , o intervalo associado a  $\bar{a}_{j_1}^k$  deve estar à esquerda daquele de  $\bar{a}_{j_2}^k$ . Para isto, determina-se uma orientação transitiva do complemento de  $G_k$ , na seguinte forma: para  $v$  e  $w$  adjacentes em  $\bar{G}_k$  tais que  $u \in \bar{a}_{j_1}^k$ ,  $v \in \bar{a}_{j_2}^k$  e  $(\bar{a}_{j_1}^k, \bar{a}_{j_2}^k) \in P_k$ , orienta-se  $u \rightarrow v$ . Se não for possível obter esta orientação então não é possível construir  $p_k$ . Obtida a orientação transitiva de  $\bar{G}_k$ , obtém-se uma representação  $F_k$  por intervalos num conjunto linearmente ordenado  $L_k$  tal que se  $\bar{a}_{j_1}^k, \bar{a}_{j_2}^k \in A_k$  e  $(\bar{a}_{j_1}^k, \bar{a}_{j_2}^k)$  está em  $P_k$ , então o ponto associado a  $\bar{a}_{j_1}^k$  está à esquerda do ponto de  $\bar{a}_{j_2}^k$  em  $L_k$ .

Seja  $\bar{a}_{j_1}^k, \bar{a}_{j_2}^k, \dots, \bar{a}_{j_k}^k$  a ordem dos elementos de  $A_k$  em  $L_k$ , da esquerda à direita, então  $\bar{p}_k$  é o caminho  $\bar{a}_{j_1}^k \rightarrow \bar{a}_{j_2}^k \rightarrow \dots \rightarrow \bar{a}_{j_k}^k$  e está agregado a  $T_m$  pela aresta  $x \rightarrow \bar{a}_{j_1}^k$ .

O algoritmo de reconhecimento para os grafos RDV é:

#### Algoritmo IV. 9: RECONHECIMENTO GRAFOS RDV

• Dado  $G=(V,E)$

• se  $G$  não é cordal então devolver  $G$  não é RDV

◦ caso contrário efetuar

◦ determinar  $\mathcal{E} = \{c_1, \dots, c_r\}$

◦ fazer  $\bar{\mathcal{E}} = \{\bar{c}_1, \dots, \bar{c}_r\}$

◦  $i = 0$

◦ enquanto  $i < r$  e possa existir árvore característica com raiz  $\bar{c}_i$  efetuar

◦  $i = i + 1$

◦  $m = 1$

◦ seja  $\bar{c}_i$  a raiz de  $T_m$

◦  $\bar{\mathcal{E}}_m = \{\bar{c}_i\}$

◦  $J = \{\bar{c}_i\}$

◦ enquanto  $\bar{\mathcal{E}}_m \neq \bar{\mathcal{E}}$  efetuar

◦ escolher  $\bar{x} \in J$

◦ construir  $A_{\bar{x}}^-$

◦ se  $A_{\bar{x}}^- = \emptyset$  então efetuar

◦  $\bar{\mathcal{E}}_{m+1} = \bar{\mathcal{E}}_m$

◦  $T_{m+1} = T_m$

◦  $J = J - \{\bar{x}\}$

◦  $m = m + 1$

◦ caso contrário efetuar

◦ particionar  $A_{\bar{x}}^- = A_1, \dots, A_s$

◦ para todo  $k = 1, \dots, s$  efetuar

◦ para todo  $\bar{a}_j^k \in A_k$  efetuar

◦ determinar  $D_j^k$

◦ se não se verificam as condições para  $T_m$  então devolver  $\bar{\mathcal{A}}$  árvore com raiz  $\bar{c}_i$

◦ caso contrário efetuar

◦ para todo  $k = 1, \dots, s$  efetuar

◦ construir caminho  $p_k$

◦ se não é possível então devolver  $G$  não é RDV

◦  $T_{m+1} = T_m \cup \{p_k / 1 \leq k \leq s\}$

◦  $\bar{\mathcal{E}}_{m+1} = \bar{\mathcal{E}}_m \cup A_{\bar{x}}^-$

◦  $m = m + 1$

◦  $T_m = T_{m+1}$

◦  $J = J - \{\bar{x}\} \cup A_{\bar{x}}^-$

◦ devolver  $G$  é RDV com árvore característica  $T_m$

◦ devolver  $G$  não é RDV.

O reconhecimento do grafo cordal é feito em tempo  $O(n)$ , a orientação transitiva em tempo  $O(n^3)$  e o número de cliques maximais em  $O(n)$ . Logo a complexidade do algoritmo de reconhecimento de grafos RDV é  $O(n^4)$ .

Para o grafo da figura (IV.9) tem-se que ele é um grafo RDV com a árvore característica  $T$  dada na figura, obtida a partir do algoritmo (IV.9) para o reconhecimento de grafos RDV.

O grafo  $G$  da figura (IV.10) tem as cliques maximais  $c_1 = \{v_2, v_3\}$ ,  $c_2 = \{v_1, v_2\}$ ,  $c_3 = \{v_1, v_4\}$ ,  $c_4 = \{v_4, v_5\}$ ,  $c_5 = \{v_1, v_6\}$  e  $c_6 = \{v_6, v_7\}$ . A árvore característica tem o conjunto de vértices  $\bar{c} = \{\bar{c}_1, \bar{c}_2, \dots, \bar{c}_6\}$ . No primeiro passo do algoritmo (IV.9) é selecionado  $\bar{c}_1$  como raiz e seu sucessor imediato  $\bar{c}_2$ . No segundo passo, obtém-se um caminho direcionado a partir de  $\bar{c}_2$  que contém  $\bar{c}_3$  e  $\bar{c}_5$ , sendo que nenhum destes dois vértices é predecessor do outro. Depois obtém-se um caminho direcionado a partir de  $\bar{c}_3$  que contém  $\bar{c}_4$  e finalmente um caminho direcionado a partir de  $\bar{c}_5$  que contém  $\bar{c}_6$ . A árvore característica  $T$  é dada na figura (IV.10)

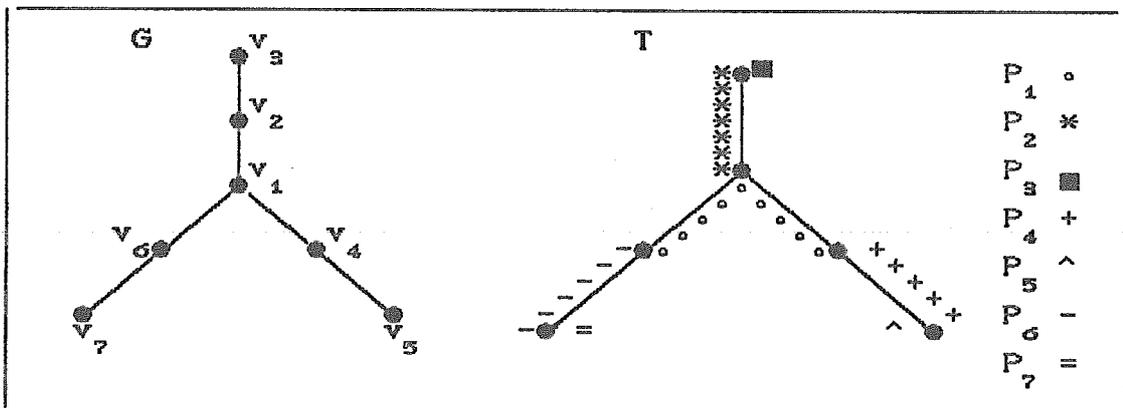


Figura IV.10.

### IV.4.3. Grafos DV

Um grafo DV é o grafo interseção de um conjunto de caminhos direcionados de vértices, numa árvore direcionada.

A árvore característica  $T$  de um grafo DV é uma árvore direcionada, tal que cada vértice  $v_i$  de  $G$  está associado a um caminho direcionado  $p_i$  em  $T$  e cada vértice de  $T$  corresponde a uma clique maximal de  $G$ .

Ao separar  $G$ , por uma das suas cliques maximais  $K$ , nos subgrafos  $G_i = G_{V_i \cup K}$ , para  $i = 1, 2, \dots, s$ , tem-se pela propriedade (IV.5), que  $G$  é um grafo DV se e somente se cada  $G_i$  é DV e, na árvore característica, os  $G_i$ 's podem ser 2-coloridos de maneira que em nenhum par antipodal os dois elementos têm a mesma cor. Esta separação de  $G$  pela clique  $K$  produz a divisão da sua árvore pelo vértice  $\bar{k}$  associado a  $K$ , em várias partes. Os subgrafos na separação podem ser classificados em dois grupos, dependendo se a parte da árvore onde eles estão localizados, tem uma aresta ligando o vértice  $\bar{k}$ . Dois subgrafos antipodais devem ficar em grupos diferentes.

Para fazer o reconhecimento de acordo com o processo geral apresentado no algoritmo (IV.3), é necessário caracterizar os átomos destes grafos. A propriedade (IV.4) não dá uma caracterização deles, mas observando a parte  $k$  da propriedade (IV.1), tem-se que um grafo DV é cordal e UEH. Logo, da propriedade (IV.4), deduz-se que um átomo DV deve ter somente um ou dois vértices ligados.

Com isto o algoritmo de reconhecimento fica:

#### Algoritmo IV.10: RECONHECIMENTO DE GRAFOS DV

- Dado  $G$
- determinar as cliques maximais de  $G$
- determinar  $T_G$  árvore de decomposição por cliques de  $G$

- determinar as cliques maximais que permanecem em cada subgrafo gerado
- determinar a árvore característica de cada subgrafo
- verificar que cada átomo de  $T_G$  é cordal e UEH
- se algum átomo de  $T_G$  não é cordal ou não é UEH então
  - devolver  $G$  não é DV
- decidir antipodalidade entre todos os pares de cliques maximais
- decidir antipodalidade entre todos os pares de subgrafos
- se algum subgrafo não é DV então devolver  $G$  não é DV
- se algum subgrafo não pode ser 2-colorido, de forma que para nenhum par antipodal os dois elementos tenham a mesma cor, então devolver  $G$  não é DV
- devolver  $G$  é grafo DV

A complexidade do algoritmo é  $\mathcal{O}(nmr)$ , onde  $r$  é o número de cliques maximais de  $G$ .

O grafo da figura (IV.11) tem as cliques maximais  $c_1 = \{a, b, h\}$ ,  $c_2 = \{b, c, d\}$ ,  $c_3 = \{d, e, f\}$ ,  $c_4 = \{f, g, h\}$  e  $c_5 = \{b, d, f, h\}$ . A clique  $c_5$  separa o grafo gerando as componentes conexas induzidas por  $V_1 = \{a\}$ ,  $V_2 = \{c\}$ ,  $V_3 = \{e\}$  e  $V_4 = \{g\}$ .

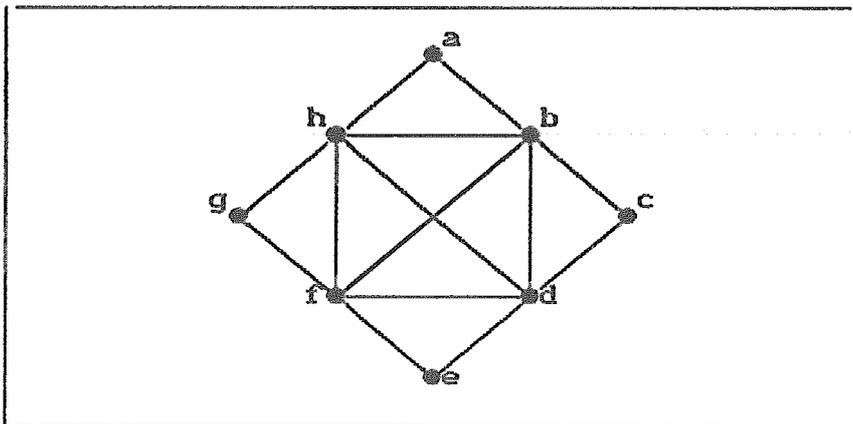


Figura IV.11.

Os subgrafos induzidos por  $V_i \cup c_5$ , para  $i=1,2,3,4$ , são todos átomos, da forma do grafo  $G_i$  mostrado na figura (IV.12) junto com sua árvore característica  $T_i$ , na qual para cada vértice  $k$  de  $G_i$  indica-se seu caminho associado pelo ponto inicial  $x_k$  e o ponto final  $y_k$ , tem-se que  $x_1=y_1=x_2=x_5$ ,  $x_4=x_3$ ,  $y_2=y_3$  e  $y_4=y_5$ . O grafo  $G_i$  é cordal e UEH, sua árvore característica  $T_i$  é uma árvore-estrela.

Para o grafo original  $G$  da figura (IV.11) tem-se que os pares de cliques  $c_1$  e  $c_2$ ,  $c_1$  e  $c_4$ ,  $c_2$  e  $c_3$ ,  $c_3$  e  $c_4$  são antipodais. Os pares de subgrafos  $G_1$  e  $G_2$ ,  $G_1$  e  $G_4$ ,  $G_2$  e  $G_3$ ,  $G_3$  e  $G_4$  são antipodais.

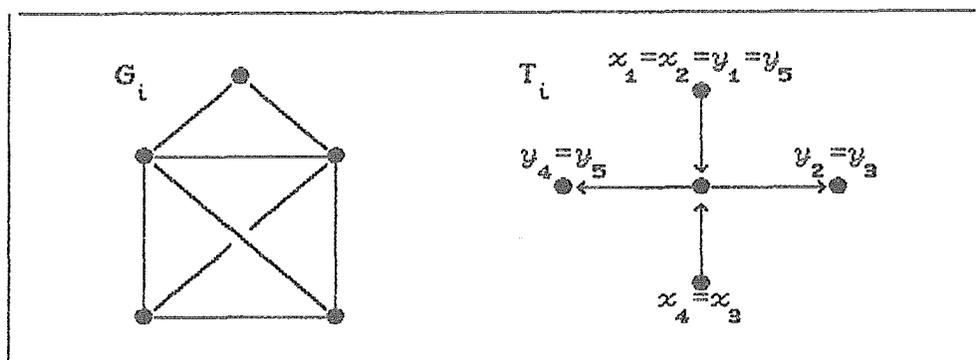


Figura IV.12.

Na figura (IV.13) apresenta-se a árvore característica  $T$  do grafo  $G$ , tem-se que  $x_1 = y_1 = x_2 = x_8 = \bar{c}_1$ ,  $y_2 = x_3 = y_3 = y_4 = \bar{c}_2$ ,  $x_4 = x_5 = y_5 = x_6 = \bar{c}_3$  e  $y_6 = x_7 = y_7 = y_8 = \bar{c}_4$ .

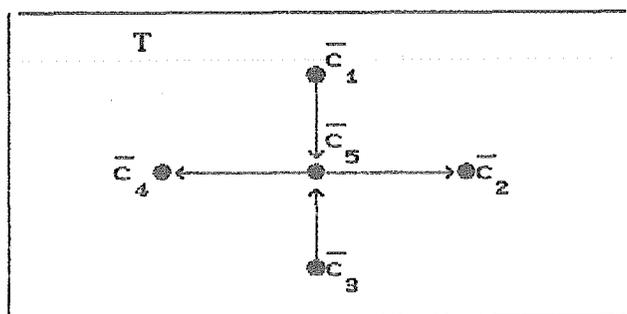


Figura IV.13.

## IV.4.4. Grafos DE

Um grafo DE é o grafo interseção de um conjunto de caminhos direcionados de arestas, numa árvore direcionada.

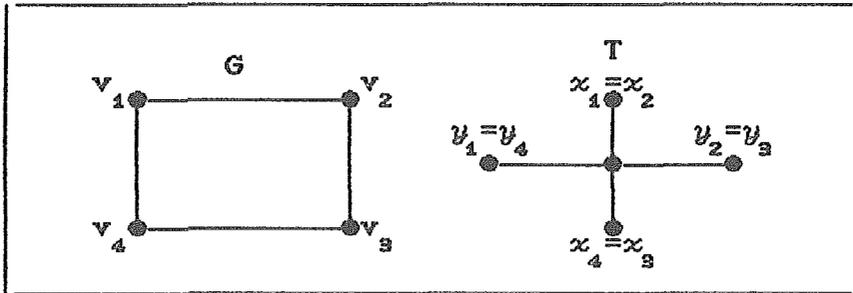


Figura IV.14.

Na figura (IV.14), apresenta-se o grafo  $G$ , que é um átomo DE mas não é cordal, e sua árvore característica  $T$  onde os pontos  $x_i$  e  $y_i$  são os vértices iniciais e finais do caminho  $p_i$  respectivamente.

A árvore característica de um grafo DE é uma árvore direcionada, tal que cada vértice  $v_i \in V$  está associado a um caminho direcionado  $p_i$  em  $T$  e cada aresta  $\bar{c}$  de  $T$  corresponde a uma clique maximal  $C$  de  $G$ .

De acordo com a propriedade (IV.4), todo átomo DE tem uma árvore característica que é uma árvore-estrela, como pode ser visto no exemplo da figura (IV.14).

Considerando a árvore característica  $T$  de um grafo DE e analisando a propriedade (IV.5) para ele, tem-se que, separar  $G$  por uma clique  $K$  corresponde a separar  $T$  pela aresta associada a  $K$  em várias partes. As arestas associadas às cliques de um subgrafo formam uma subárvore que permanece inteira em uma parte. Cada parte pode conter mais de uma subárvore. Subárvores de subgrafos antipodais devem ficar em lados opostos da aresta  $\bar{k}$ , isto produz a necessidade de duas cores.

*Algoritmo IV.11: RECONHECIMENTO DE GRAFOS DE*

- Dado  $G=(V,E)$
- determinar as cliques maximais de  $G$
- determinar  $T_G$  árvore de decomposição por cliques de  $G$
- determinar as cliques maximais dos subgrafos gerados na decomposição
- determinar a árvore característica de cada subgrafo
- se algum átomo não tem representação numa árvore-estrela então devolver  $G$  não é DE
- decidir antipodalidade entre todos os pares de cliques maximais
- decidir antipodalidade entre os subgrafos
- se algum subgrafo não é DE ou não pode ser 2-colorido, de forma que em todo par antipodal os dois elementos têm cor diferente, então devolver  $G$  não é DE
- devolver  $G$  é DE.

A complexidade do algoritmo é  $O(nmr)$ .

O grafo da figura (IV.15) é um grafo DE, com cliques maximais  $c_1=\{a,b,c\}$ ,  $c_2=\{c,d,e\}$ ,  $c_3=\{a,c,e,f\}$ ,  $c_4=\{f,g,h\}$ ,  $c_5=\{f,k,l\}$ ,  $c_6=\{f,h,j,k\}$  e  $c_7=\{h,i,j\}$ .

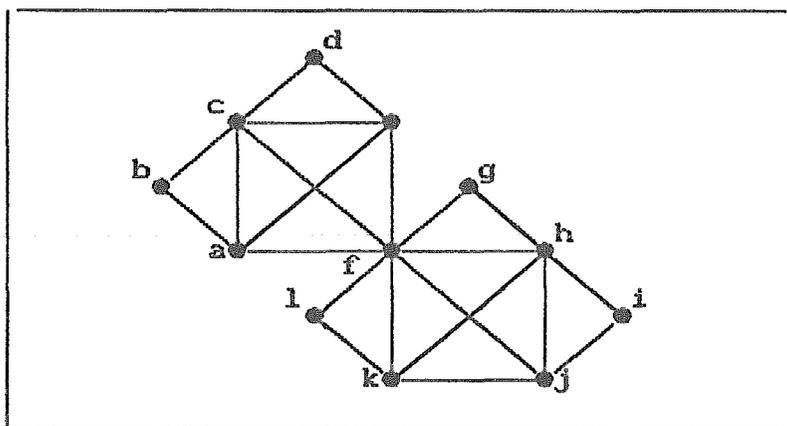


Figura IV.15.

A clique  $c_6$  separa o grafo da figura (IV.15) produzindo as componentes conexas induzidas por  $V_1=\{a,b,c,d,e\}$ ,  $V_2=\{g\}$ ,  $V_3=\{i\}$  e  $V_4=\{l\}$ .

O subgrafo  $G_1$  induzido por  $V_1 \cup c_6$  é separado pela clique  $c_3$ , as componentes conexas são induzidas por  $V_{11}=\{B\}$ ,  $V_{12}=\{d\}$  e  $V_{13}=\{h,j,k\}$ . Os restantes subgrafos gerados são átomos. Os átomos  $G_i$  induzidos por  $V_i \cup c_6$  para  $i=2,3,4$ , e os átomos  $G_{1j}$  gerados por  $V_{1j}$  para  $j=1,2$ , têm todos a mesma forma que é dada na figura (IV.16) junto com sua árvore característica. Para cada vértice  $k$  em  $G$ , os caminhos são indicados na árvore por seu ponto inicial  $x_k$  e seu ponto final  $y_k$ , tem-se que  $x_2=x_3=x_4$ ,  $x_1=x_5=y_3$ ,  $y_1=y_2=y_4=y_5$ .

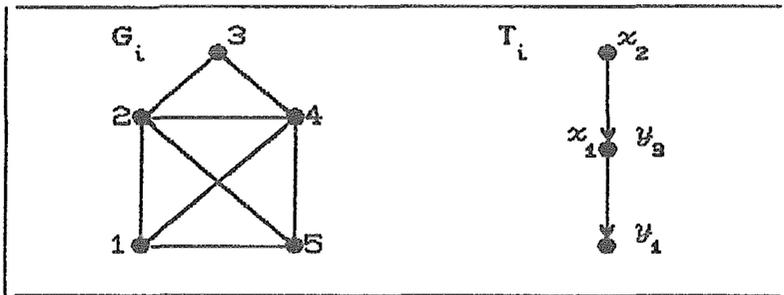


Figura IV.16.

A árvore do subgrafo induzido por  $V_{13} \cup c_6$  tem a mesma forma, mas neste caso o primeiro vértice corresponde a  $x_a$ ,  $x_c$ ,  $x_e$  e  $x_f$ , o segundo a  $y_a$ ,  $y_c$ ,  $y_o$ ,  $y_h$ ,  $y_j$  e  $y_k$  e o terceiro a  $y_i$ ,  $y_h$ ,  $y_j$  e  $y_k$ .

Na figura (IV.17) apresenta-se a árvore característica  $T$  para o grafo original da figura (IV.15), a sequência de letras ao lado de cada aresta indica os vértices cujo caminho associado contém essa aresta.

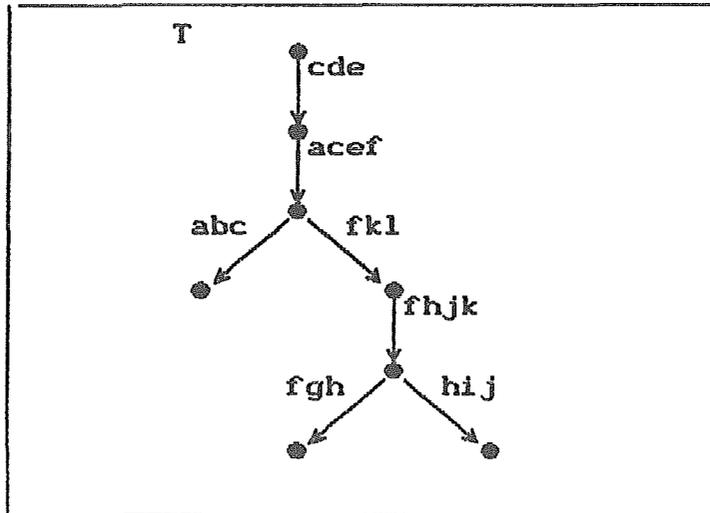


Figura IV.17.

#### IV.4.5. Grafos UE

Um grafo UE é o grafo interseção de um conjunto de caminhos de arestas, numa árvore não direcionada. Os átomos da árvore característica de um grafo UE são árvores-estrela.

Dada uma decomposição de  $G$  nos subgrafos  $G_1, G_2, \dots, G_s$  através de uma das suas cliques maximais, a propriedade (IV.5) diz que  $G$  é cordal e UE se e somente se, cada um dos  $G_i$  é cordal e UE e estes subgrafos podem ser 3-coloridos de forma que, em nenhum par antipodal, os dois elementos tenham a mesma cor e para cada vértice da clique separadora, o conjunto de subgrafos vizinhos dele é 2-colorável. Com base nessa caracterização, GOLUMBIC e JAMISON (1985a) provaram que o reconhecimento dos grafos (UE  $\cap$  cordal) é *NP-completo*, como os grafos cordais são reconhecidos em tempo *polinomial*, podemos concluir que o reconhecimento dos grafos UE é *NP-completo*.

## Capítulo V: CONCLUSÕES

A classe de *grafos triangularizados* se diferencia dos grafos perfeitos porque é possível fazer o reconhecimento de um grafo triangularizado em tempo *polinomial*, mas ainda não se conhece a natureza do problema de reconhecer se um grafo é perfeito ou não.

Uma característica importante dos grafos triangularizados e de todo grafo perfeito é que os problemas da determinação do *tamanho da maior clique*, do *maior conjunto estável*, de uma *cobertura mínima por cliques* e do *número cromático* de um grafo, podem ser resolvidos em tempo *polinomial*. Para todo grafo perfeito prova-se isto em base ao *método do elipsoide*, através da formulação de um problema de programação linear. Para os grafos *triangularizados*, tem-se desenvolvido algoritmos de tempo *linear* no tamanho do grafo, baseados nas características estruturais deles.

Alguns outros problemas não apresentam vantagens ao restringir o estudo à classe dos triangularizados, mas ao considerar alguma subclasse particular deles, é possível desenvolver algoritmos *polinomiais*. Por exemplo, para os *grafos de intervalo* tem-se desenvolvido algoritmos *polinomiais* para o *problema de Steiner*, o *isomorfismo de grafos* e o *problema do conjunto dominante*; no caso dos *grafos fortemente triangularizados*, é possível resolver o *problema do conjunto dominante* e o *problema de Steiner*, em tempo *polinomial*.

Também tem-se estudado classes de grafos que contém os triangularizados, procurando algoritmos *polinomiais* para seu reconhecimento e para resolver os problemas clássicos de otimização em grafos. Dentro desta linha de pesquisa tem-se desenvolvido algoritmos *polinomiais* de reconhecimento para os *grafos de Gallai*, de *paridade*, de *Meyniel*, *grafos separáveis por cliques* e *fracamente triangularizados*. Além destes algoritmos de reconhecimento, tem-se desenvolvido algoritmos *polinomiais* para os quatro problemas clássicos no

caso de todos estes grafos.

A pesquisa dentro desta área pode dividir-se em duas linhas: continuar na procura de classes de grafos para os quais seja possível desenvolver algoritmos *polinomiais* para o reconhecimento e para diferentes problemas em grafos, isto pode tentar-se considerando relaxações nos grafos triangularizados obtendo novas classes que os contenham ou, dentro das subclasses dos triangularizados já conhecidas ou construindo novas subclasses, desenvolver algoritmos *polinomiais* para outros problemas em grafos. Outra linha de pesquisa é continuar explorando extensões de grafos triangularizados tentando aproximar-se à caracterização dos grafos perfeitos que permita decidir respeito se seu reconhecimento pode ser feito em tempo *polinomial* ou a qué classe de problemas pertence.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ANDREAE, T. (1987), "On the Interval Number of a Triangulated Graph", *Journal of Graph Theory*, vol. 11, pp. 273-280.
- BARROSO, M. (1987), *Orientações em Grafos Circulares e suas Aplicações*, Tese de Doutorado, COPPE/UFRJ.
- BOOTH, K. e JOHNSON, J. (1982), "Dominating Sets in Chordal Graphs", *SIAM Journal on Computing*, vol. 11, pp. 191-199.
- BUCKINGHAM, M. e GOLUMBIC, M. (1983), "Partitionable Graphs, Circle Graphs and the Berge Strong Perfect Graph Conjecture", *Discrete Mathematics*, vol. 44, pp. 45-54.
- BUNEMAN, P. (1974), "A Characterization of Rigid Circuit Graphs", *Discrete Mathematics*, vol. 9, pp. 205-212.
- BURLET, M. e FONLUPT, J. (1984), "Polynomial Algorithm to Recognize a Meyniel Graph", em *Annals of Discrete Mathematics* 21, Berge, C. e Chvátal, V. (eds.), North-Holland, New York, pp. 225-252.
- BURLET, M. e UHRY, J. (1984), "Parity Graphs", em *Annals of Discrete Mathematics* 21, Berge, C. e Chvátal, V. (eds.), North-Holland, New York, pp. 253-277.
- CHANDRASEKHARAN, N. e IYENGAR, S. (1988), "NC Algorithms for Recognizing Chordal Graphs and k-trees", *IEEE Transactions on Computers*, vol. 37, pp. 1178-1183.
- CHVÁTAL, V. (1984a), "Perfectly Ordered Graphs", em *Annals of Discrete Mathematics* 21, Berge, C. e Chvátal, V. (eds.), North-Holland, New York, pp. 63-65.

- CHVÁTAL, V. (1984b), "A Semi-strong Perfect Graph Conjecture", em *Annals of Discrete Mathematics* 21, Berge, C. e Chvátal, V. (eds.), North-Holland, New York, pp. 279-280.
- CHVÁTAL, V. (1984c), "Notes on Perfect Graphs", em *Progress in Combinatorial Optimization*, Pulleyblanc, W. (ed.), Academic Press, Canada, pp. 107-115.
- COLEMAN, T. (1988), "A Chordal Preconditioner for Large-scale Optimization", *Mathematical Programming*, vol. 40, pp. 265-287.
- D'ATRI, A. e MOSCARINI, M. (1988), "On Hipergraph Aciclycity and Graph Chordality", *Information Processing Letters*, vol. 29, pp. 271-274.
- DENNIS, J. e STEIHAUG, T. (1986), "On the Successive Projections Approach to Least-squares Problems", *SIAM Journal on Numerical Analysis*, vol. 23, pp. 717-733.
- DUCHET, P. (1984), "Classical Perfect Graphs", em *Annals of Discrete Mathematics* 21, Berge, C. e Chvátal, V. (eds.), North-Holland, New York, pp. 67-96.
- FARBER, M. (1983), "Characterizations of Strongly Chordal Graphs", *Discrete Mathematics*, vol. 43, pp. 173-189.
- FARBER, M. (1984), "Domination, Independent Domination and Duality in Strongly Chordal Graphs", *Discrete Applied Mathematics*, vol. 7, pp. 115-130.
- FULKERSON, D. e GROSS, O. (1965), "Incidence Matrices and Interval Graphs", *Pacific Journal of Mathematic*, vol. 15, pp. 835-855.

- GAREY, R. e JOHNSON, D. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. Freeman e co., New York.
- GAVRIL, F. (1972), "Algorithms for Minimum Coloring, Maximum Clique, Minimum Covering by Cliques and Maximum Independent Set of a Chordal Graph", *SIAM Journal on Computing*, vol. 1, pp. 180-187.
- GAVRIL, F. (1974), "The Intersection Graph of Subtrees in a Tree are Exactly the Chordal Graphs", *Journal of Combinatorial Theory B*, vol. 16, pp. 47-56.
- GAVRIL, F. (1975), "A Recognition Algorithm for the Intersection Graphs of Directed Paths in Directed Trees", *Discrete Mathematics*, vol. 13, pp. 237-249.
- GAVRIL, F. (1977), "Algorithms on Clique Separable Graphs", *Discrete Mathematics*, vol. 19, pp. 159-165.
- GAVRIL, F. (1978), "A Recognition Algorithm for the Intersection Graphs of Paths in Trees", *Discrete Mathematics*, vol. 23, pp. 211-217.
- GOLUMBIC, M. e GOSS, C. (1978), "Perfect Elimination and Chordal Bipartite Graphs", *Journal of Graph Theory*, vol. 2, pp. 153-163.
- GOLUMBIC, M. (1980), *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York.
- GOLUMBIC, M. e JAMISON, R. (1985a), "The Edge Intersection Graphs of Paths in a Tree", *Journal of Combinatorial Theory B*, vol. 38, pp. 8-22.
- GOLUMBIC, M. e JAMISON, R. (1985b), "Edge and Vertex Intersection of Paths in a Tree", *Discrete Mathematics*, vol. 55, pp. 151-159.

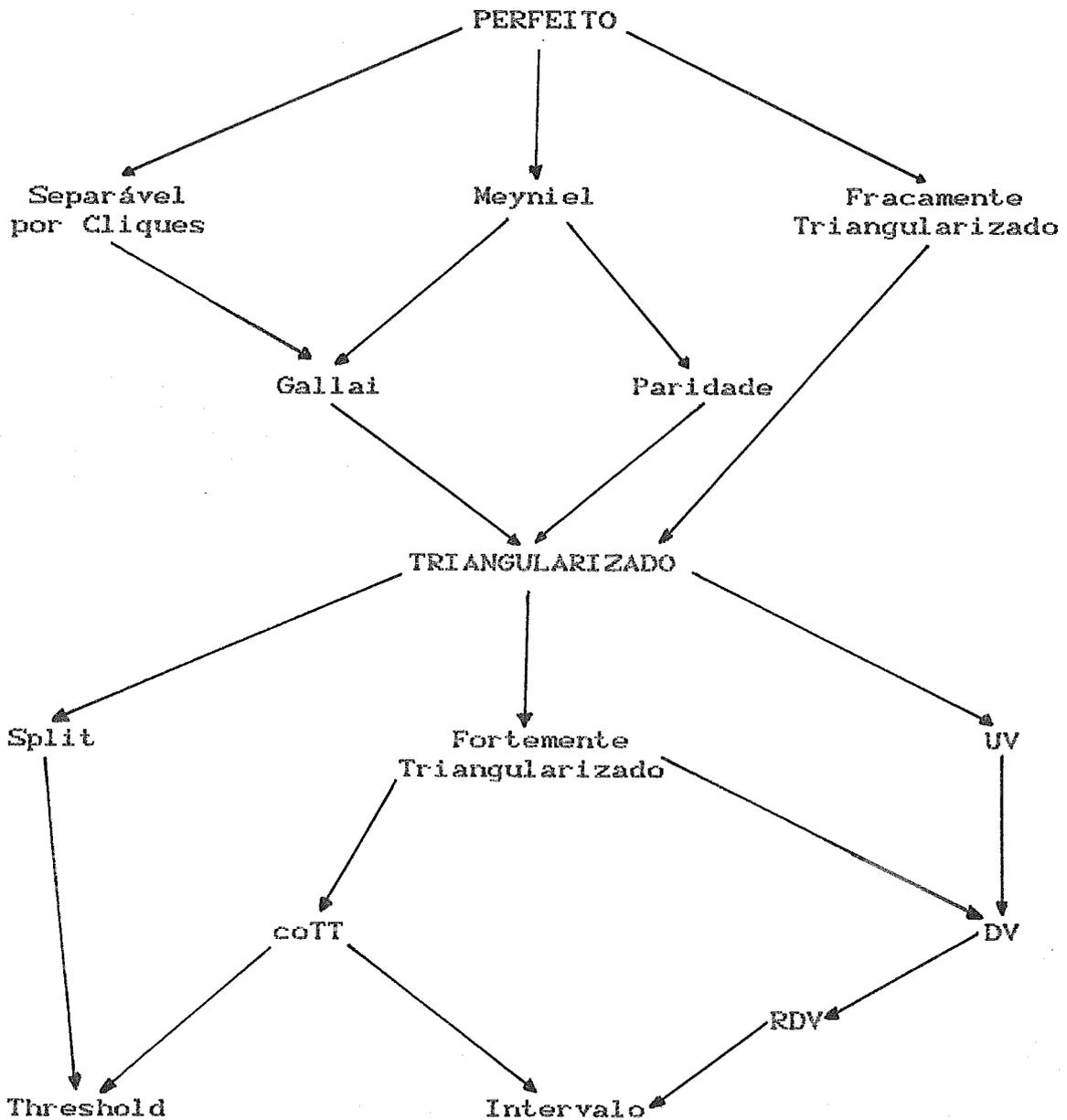
- GROTSCHEL, M., LOVÁSZ, L. e SCHRIJVER, A. (1981), "The Ellipsoid Method and its Consequences in Combinatorial Optimization", *Combinatorica*, vol. 1, pp. 169-197.
- GUPTA, U., LEE, D. e LEUNG, J. (1982), "Efficient Algorithms for Interval Graphs and Circular-arc Graphs", *Networks*, vol. 12, pp. 459-467.
- GYÁRFAS, A. e LEHEL, J. (1970), "A Helly-type Problem in Trees", em *Combinatorial Theory and its Applications II*, Erdos, P. et alii (eds.), North-Holland, Amsterdam, pp. 571-584.
- HASKINS, L. e ROSE, D. (1973), "Toward a Characterization of Perfect Elimination Digraphs", *SIAM Journal on Computing*, vol. 2, pp. 217-224.
- HAYWARD, R. (1985), "Weakly Triangulated Graphs", *Journal of Combinatorial Theory B*, vol. 39, pp. 200-209.
- HAYWARD, R., HOÀNG, C. e MAFFRAY, F. (1988), "Optimizing Weakly Triangulated Graphs", *manuscripto*.
- HSU, W. (1987), "Decomposition of Perfect Graphs", *Journal of Combinatorial Theory B*, vol. 43, pp. 70-94.
- JOHNSON, D. (1985), "The NP-Completeness Column: An Ongoing Guide", *Journal of Algorithms*, vol. 6, pp. 434-451.
- KLEIN, N. (1988), "Efficient Parallel Algorithms for Chordal Graphs", *Proceedings of the 29<sup>th</sup> Annual Symposium on Foundations of Computer Science*, New York, pp. 150-166.
- LADNER, R. e FISHER, M. (1980), "Parallel Prefix Computation", *Journal of Association Computer Machinery*, vol. 27, pp. 831-838.

- LOVÁSZ, L. (1972), "A Characterization of Perfect Graphs", *Journal of Combinatorial Theory B*, vol. 13, pp. 95-98.
- LOVÁSZ, L. (1979), "On the Shannon Capacity of a Graph", *IEEE Transactions on Information Theory*, vol. 25, pp. 1-7.
- LOVÁSZ, L. (1983), "Perfect Graphs", em *Select Topics in Graph Theory 2*, Beineke, L. e Wilson, R. (eds.), Academic Press, New York, pp. 55-87.
- LUEKER, G. e BOOTH, K. (1979), "A Linear Time Algorithm for Deciding Interval Graphs Isomorfism", *Journal of Association Computer Machinery*, vol. 26, pp. 183-195.
- MEYNIEL, H. (1976), "On the Perfect Graph Conjecture", *Discrete Mathematics*, vol. 16, pp. 339-342.
- MONMA, C. e WEI, V. (1986), "Intersection Graphs of Paths in a Tree", *Journal of Combinatorial Theory B*, vol. 41, pp. 141-181.
- MONMA, C., REED, B. e TROTTER, W. Jr. (1988), "Threshold Tolerance Graphs", *Journal of Graph Theory*, vol. 12, pp. 343-362.
- NAOR, J., NAOR, M. e SCHAFFER, A. (1987), "Fast Parallel Algorithms for Chordal Graphs", *Proceedings of the 19<sup>th</sup> Annual ACM Symposium on Theory of Complexity*, New York, pp. 355-364.
- NETO, C. (1987), *Sobre Grafos Perfeitos*, Tese de Mestre em Ciência da Computação, UNICAMP.
- PAPADIMITRIOU, C. e YANNAKAKIS, M. (1979), "Scheduling Interval-ordered tasks", *SIAM Journal on Computing*, vol. 8, pp. 405-409.

- PARTHASARATY, K. e RAVINDRA, G. (1976), "The Strong Perfect Graph Conjecture is True for  $K_{1,3}$ -free Graphs", *Journal of Combinatorial Theory B*, vol. 21, pp. 212- 223.
- PARTHASARATY K. e RAVINDRA, G. (1979), "The Validity of the Strong Perfect Graph Conjecture for  $(K_4-e)$  Free Graphs", *Journal of Combinatorial Theory B*, vol. 26, pp. 98-100.
- PNUELI, A., LEMPEL, A. e EVEN, S. (1971), "Transitive Orientation of Graphs and Identification of Permutation Graphs", *Canadian Journal of Mathematics*, vol. 23, pp. 160-175.
- REED, B. (1987), "A Semi-strong Perfect Graph Theorem", *Journal of Combinatorial Theory B*, vol. 43, pp. 223-240.
- REED, B. (1988), "Perfectly Orderable Graphs", *WORKSHOP em Aspectos Teóricos da Computação*, USP, São Paulo.
- RENZ, P. (1970), "Intersection Representation of Graphs by Arcs", *Pacific Journal of Mathematics*, vol. 34, pp. 501-510.
- ROSE, D. (1970), "Triangulated Graphs and the Elimination Process", *Journal of Mathematical Analysis and Applications*, vol. 32, pp. 597-609.
- ROSE, D., TARJAN, R. e LEUKER, G. (1976), "Algorithmic Aspects of Vertex Elimination on Graphs", *SIAM Journal on Computing*, vol. 5, pp. 266-283.
- SCHEINERMAN, E. (1988), "On the Interval Number of a Chordal Graph", *Journal of Graph Theory*, vol.12, pp. 311-316.

- SHIBATA, Y. (1988), "On the Tree Representation of Chordal Graphs", *Journal of Graph Theory*, vol. 12, pp. 421-428.
- STEIHAUG, T. (1983), "The Conjugate Gradient Method and Trust Regions in Large Scale Optimization", *SIAM Journal on Numerical Analysis*, vol.20, pp. 626-637.
- SYSLO, M. (1985), "Triangulated Edge Intersection Graphs of Paths in a tree", *Discrete Mathematics*, vol. 55, pp. 217-220.
- SZWARCFITER, J. (1986), *Grafos e Algoritmos Computacionais*, Editora Campus, Rio de Janeiro.
- TSUKIYAMA, S., IDE, M, ARIYOSHI, M. e SHIRAWAKA, I. (1977), "A New Algorithm for Generating all the Maximal Independent Sets", *SIAM Journal on Computing*, vol. 6, pp. 505-517.
- TUCKER, A. (1973), "The Strong Perfect Graph Conjecture for Planar Graphs", *Canadian Journal of Mathematics*, vol. 25, pp. 103-114.
- TUCKER, A. (1975), "Coloring a Family of Circular Arcs", *SIAM Journal on Applied Mathematics*, vol. 29, pp. 493-502.
- WHITE, K., FARBER, M. e PULLEYBLANC W. (1985), "Steiner Trees, Connected Domination and Strongly Chordal Graphs", *Networks*, vol. 15, pp. 109-124.
- WHITESIDES, S. (1984), "A Method for Solving Certain Graph Recognition and Optimization Problems, with Applications to Perfect Graphs", em *Annals of Discrete Mathematics* 21, Berge, C. e Chvátal V. (eds.), North-Holland, New York, pp. 281-297.

## Apêndice A: Relações Gerais entre classes de grafos.

Obs:  $a \rightarrow b$  é  $b \subseteq a$ 

UV: Grafo interseção de caminhos de vértices de uma árvore não direcionada.

coTT: Grafo complemento de threshold com tolerâncias.

DV: Grafo interseção de caminhos de vértices de uma árvore direcionada.

RDV: Grafo interseção de caminhos de vértices de uma árvore direcionada e enraizada.

Apêndice B: Tabela de complexidade por classes.

GRAFO	Reconhecimento	Clique	Número Cromático	Conjunto Estável	Cobertura Cliques
Perfeito	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$
Fraca. Triang.	$\mathcal{P}$	$n^4 + mn^3$	$n^4 + mn^3$	$n^4 + mn^3$	$n^4 + mn^3$
Meyniel	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$
Separável Cliques	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$
Gallai	$n^5$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$
Paridade	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$
Triang.	$n + m$	$n + m$	$n + m$	$n + m$	$n + m$
Forte. Triang.	$n + m$	$n + m$	$n + m$	$n + m$	$n + m$
Split	$n + m$	$n + m$	$n + m$	$n + m$	$n + m$
UV	$n^4$	$n + m$	$n + m$	$n + m$	$n + m$
DV	$nmr$	$n + m$	$n + m$	$n + m$	$n + m$
coTT	$\mathcal{P}$	$n + m$	$n + m$	$n + m$	$n + m$
RDV	$n^4$	$n + m$	$n + m$	$n + m$	$n + m$
Intervalo	$n + m$	$n \log n$	$n \log n$	$n \log n$	$n \log n$
Threshold	$\mathcal{P}$	$n + m$	$n + m$	$n + m$	$n + m$

Na entrada da tabela é dada a complexidade do pior caso para o algoritmo sequencial, se existir, em função de número de vértices ( $n$ ), número de arestas ( $m$ ) e número de cliques maximais ( $r$ ). Caso contrário,  $\mathcal{P}$  indica que o problema pode ser resolvido em tempo polinomial para essa classe de grafos.  $\mathcal{A}$  indica que o problema está em aberto.