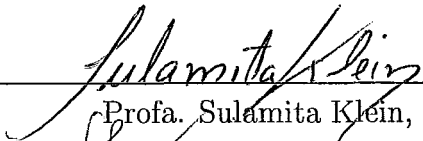


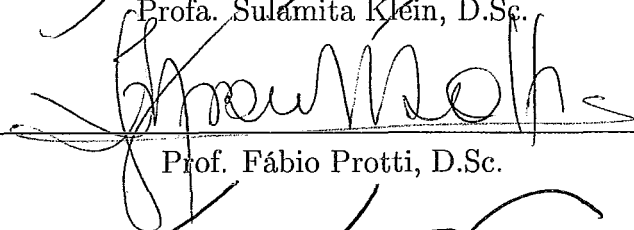
CONJUNTO INDEPENDENTE E COBERTURA POR CLIQUES EM  
GRAFOS DE DISCO UNITÁRIO E MOEDA UNITÁRIA

Talita de Oliveira Ferreira

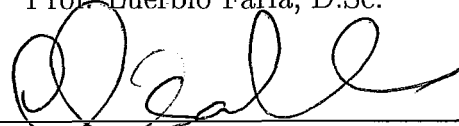
TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO  
DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA  
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS  
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE  
MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E  
COMPUTAÇÃO.

Aprovada por:

  
\_\_\_\_\_  
Prof. Sulamita Klein, D.Sc.

  
\_\_\_\_\_  
Prof. Fábio Protti, D.Sc.

  
\_\_\_\_\_  
Prof. Luércio Faria, D.Sc.

  
\_\_\_\_\_  
Prof. Cláudia Linhares Sales, Docteur

RIO DE JANEIRO, RJ - BRASIL  
JUNHO DE 2003

FERREIRA, TALITA DE OLIVEIRA

Grafos de disco unitário [Rio de Janeiro]  
2003

XI, 94 p., 29,7 cm (COPPE/UFRJ, M.Sc.,  
Engenharia de Sistemas e Computação, 2003)

Tese – Universidade Federal do Rio de  
Janeiro, COPPE

1 - Grafos de interseção de disco

2 - Conjunto independente

3 - Cobertura por cliques

I. COPPE/UFRJ II. Título (série)

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

## CONJUNTO INDEPENDENTE E COBERTURA POR CLIQUES EM GRAFOS DE DISCO UNITÁRIO E MOEDA UNITÁRIA

Talita de Oliveira Ferreira

Junho/2003

Orientadores: Márcia Rosana Cerioli  
Fábio Protti

Programa: Engenharia de Sistemas e Computação

Este trabalho descreve um estudo dos problemas de encontrar um conjunto independente máximo e uma cobertura por cliques mínima em grafos de disco unitário e de moeda unitária. Um grafo é de disco unitário quando seus vértices estão associados a discos no plano de mesmo diâmetro e existe aresta entre dois deles se houver interseção entre os respectivos discos. Quando os discos não podem se sobrepor, temos a subclasse dos grafos de moeda unitária.

Os principais resultados dessa tese são o estabelecimento da complexidade dos problemas citados para a classe dos grafos de moeda unitária, ambos NP-completos, e dois algoritmos aproximativos. Um algoritmo para o problema da cobertura por cliques em grafos de moeda unitária, e um algoritmo para o problema da cobertura por cliques em grafos de disco unitário.

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

INDEPENDENT SET AND PARTITION INTO CLIQUES OF UNIT  
DISK GRAPHS AND PENNY GRAPHS

Talita de Oliveira Ferreira

June/2003

Advisors: Márcia Rosana Cerioli  
Fábio Protti

Department: Computing and Systems Engineering

This work describes a study on the maximum independent set problem and the minimum partition into cliques problem in unit disk graphs and penny graphs. A graph is a unit disk graph if its vertices are associated to discs on the plane with the same diameter and there is an edge between two of them if and only if the corresponding discs intersect. When the disks are not allowed to overlap each other, we have the class of penny graphs.

The main results of this thesis are the establishment of the complexity of the above problems, both NP-complete, and two approximation algorithms. An algorithm to find a minimum clique cover in penny graphs and an algorithm to find a minimum clique cover in unit disk graphs.

# Agradecimentos

A Deus.

A meus pais e irmãos, Maria José, Manoel Célio, Ednardo e Ícaro Matheus, pelo amor incondicional, pelo apoio e ensinamentos. A toda a minha família, em especial meus tios Oscar e Valdísia e a Valéria.

Aos amigos Cláudio Prata Santiago e Izabela de Fátima Bellini Neves pelo companheirismo e apoio que me deram nos momentos mais difíceis de minha jornada no Rio. Onde quer que a vida nos leve, onde for minha morada será a de vocês também.

Aos meus orientadores Márcia Cerioli e Fábio Protti, pelo apoio e atenção dados a mim.

Aos professores Cláudia Linhares e Marcos Negreiros, que me assitiram durante a graduação.

Aos integrantes da república da Farinha, da república Dragão do Mar e da Bancada Cearense. Em especial, Henrique Limaverde e Sérgio Assunção.

Aos amigos da UECE. Em especial, João Fernando Machry Sarubbi, Michele Nogueira Lima e todos aqueles com quem tive o prazer de participar do grupo PET.

Aos amigos do mestrado. Em especial, àqueles que mais conviveram comigo durante esses anos: Ana Lúcia Pimentel, Luidi Simonetti e Rosa Figueiredo.

Aos amigos do Colégio 7 de Setembro. Em especial, Simone Santos Sousa.

Aos amigos que fizeram parte do grupo de desbravadores Conquistadores do Rei, o qual para mim foi exemplo de organização, seriedade e companheirismo acima de tudo.

Aos amigos da "Peleja".

Aos "irmãos" na fé da Igreja de Bom Sucesso, Fortaleza, e da Igreja do Méier, Rio de Janeiro.

Aos professores da UECE.

Aos professores e funcionários da COPPE/Sistemas.

Ao povo brasileiro que fomentou os meus estudos através da CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior).

*Aos meus pais e Ednardo.*

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Definições e classes</b>	<b>5</b>
2.1	Notação e definições . . . . .	5
2.2	Modelos e aplicações . . . . .	10
2.3	Subclasses de grafos ID e DU . . . . .	13
2.4	Outras classes . . . . .	17
2.5	Problemas clássicos restritos a subclasses de grafos ID . . . . .	19
<b>3</b>	<b>Conjunto independente em grafos DU e MU</b>	<b>25</b>
3.1	Definições e propriedades . . . . .	26
3.2	Conjunto independente para DU e MU . . . . .	29
3.3	Algoritmos . . . . .	33
3.3.1	Algoritmo IS . . . . .	34
3.3.2	Algoritmo FMIS . . . . .	36
3.3.3	Algoritmos para grafos $k$ -faixa . . . . .	49

3.4	Outras classes . . . . .	58
<b>4</b>	<b>Cobertura por cliques</b>	<b>61</b>
4.1	O problema COBERTURA DE DISCOS . . . . .	63
4.2	Cobertura por cliques para grafos de Moeda Unitária . . . . .	70
4.3	Algoritmos . . . . .	80
4.3.1	Grafos faixa . . . . .	81
4.3.2	Algoritmo aproximativo para DU . . . . .	82
4.3.3	Algoritmo aproximativo para MU . . . . .	83
4.4	Outras classes . . . . .	84
<b>5</b>	<b>Conclusões</b>	<b>85</b>



# Lista de Figuras

2.1	Transformação entre os modelos de interseção e contenção. . .	11
2.2	Transformação entre os modelos de proximidade e interseção. .	11
2.3	Representações do grafo $K_4$ . . . . .	14
2.4	Representações do grafo $G$ . . . . .	15
2.5	Diagrama de classes para os grafos ID. . . . .	18
3.1	CV para grafos DU . . . . .	31
3.2	Alteração no número de discos. . . . .	32
3.3	Grafo $G'$ desenhado a partir do grafo $G$ da Figura 3.1. . . . .	33
3.4	Os números indicam a ordem em que os vértices foram escolhidos.	35
3.5	Uma execução de FMIS para $r = 2$ . . . . .	46
3.6	Exemplo do algoritmo de Matsui para $k = 3$ . . . . .	55
3.7	Exemplo do algoritmo de Matsui para $k = 3$ . . . . .	56
3.8	Exemplo do algoritmo aproximativo de Matsui para $r = 3$ . . .	59
3.9	Exemplo do algoritmo aproximativo de Matsui para $r = 3$ . . .	60

4.1	Estruturas da redução . . . . .	63
4.2	Redução a partir da instância $E = \{E_1 \wedge E_2 \wedge E_3\}$ . Onde $E_1 = (x_1 \vee \bar{x}_3 \vee \bar{x}_2)$ , $E_2 = (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$ e $E_3 = (x_1 \vee x_2 \vee x_3)$ . . . . .	65
4.3	Cláusula. . . . .	66
4.4	Cobertura de circuitos com cruzamentos. . . . .	68
4.5	Estruturas da redução. . . . .	71
4.6	Cláusula. . . . .	73
4.7	Circuito. . . . .	73
4.8	Grafo $G_E$ gerado a partir da instância $E = \{E_1 \wedge E_2 \wedge E_3 \wedge E_4\}$ . Onde $E_1 = (x_1 \vee \bar{x}_2 \vee \bar{x}_3)$ , $E_2 = (x_3 \vee \bar{x}_4 \vee \bar{x}_5)$ , $E_3 = (x_4 \vee \bar{x}_5 \vee x_2)$ e $E_4 = (\bar{x}_1 \vee x_3 \vee \bar{x}_4)$ . . . . .	74
4.9	Construção de $G'$ . . . . .	75
4.10	Cobertura de circuito: representação dos circuitos de $G'$ . Cliques representadas por pontos. . . . .	76
4.11	Cobertura de uma cláusula: a) $E_j$ não é satisfeita; b) $E_j$ satisfeita por $x$ , $y$ e $z$ ; c) $E_j$ satisfeita por $x$ ; d) $E_j$ satisfeita por $y$ ; e) $E_j$ satisfeita por $x$ e $z$ ; f) $E_j$ satisfeita por $y$ e $z$ . Os demais casos, $E_j$ satisfeita por $z$ e $\bar{E}_j$ satisfeita por $x$ e $y$ , são simétricos a $E_j$ satisfeita por $x$ e $\bar{E}_j$ satisfeita por $y$ e $z$ . Os pontos pretos representam as cliques que cobrem somente vértices da cláusulas e os pontos cinzas cliques que cobrem vértices de circuito. . . . .	77

4.12 Cobertura de uma cláusula: a)  $E_j$  não é satisfeita; b)  $E_j$  satisfeita por  $x$ ,  $y$  e  $z$ ; c)  $E_j$  satisfeita por  $x$ ; d)  $E_j$  satisfeita por  $y$ ; e)  $E_j$  satisfeita por  $x$  e  $z$ ; f)  $E_j$  satisfeita por  $y$  e  $z$ . Os demais casos,  $E_j$  satisfeita por  $z$  e  $E_j$  satisfeita por  $x$  e  $y$ , são simétricos a  $E_j$  satisfeita por  $x$  e  $E_j$  satisfeita por  $y$  e  $z$ . As linhas cheias delimitam as cliques que cobrem somente vértices de cláusula e as linhas tracejadas delimitam cliques que cobrem vértices de circuito. . . . . 78

4.13 Cobertura do disco  $D_x$ : a)  $\tau(u_i) = V$  b)  $\tau(u_i) = F$  . . . . . 79

# Capítulo 1

## Introdução

Grafos de interseção de objetos geométricos no plano têm sido amplamente utilizados na modelagem de diversos problemas reais [21, 29, 31, 39, 44, 55, 56]. Em geral, estes problemas retratam situações onde diversos objetos interagem e, após uma certa distância entre eles, a interação é tão pequena que pode ser ignorada. Exemplos atuais deste tipo de problema surgem com as redes de difusão e de telefonia móvel. Nestes casos, as antenas de transmissão são os objetos e seu alcance é o limitador de distância. Como este alcance se dá igualmente em todas as direções, para certos propósitos, a modelagem de uma rede como esta pode ser feita representando-se cada antena com o seu alcance por um disco no plano e suas interferências pela interseção dos discos.

Dada uma família de discos no plano, seu grafo de interseção é obtido associando-se um vértice a cada disco e dois vértices são adjacentes exatamente quando os discos correspondentes têm interseção não vazia. Os grafos

que podem ser obtidos por este processo são chamados de grafos ID. Em nosso trabalho, abordamos duas subclasses dos grafos ID: os **grafos de disco unitário**, ou grafos DU, e os **grafos de moeda unitária**, ou grafos MU. Os grafos DU são os grafos de interseção de famílias cujos discos possuem todos o mesmo tamanho. Os grafos MU são os grafos de interseção de discos de mesmo tamanho que não se sobrepõem. Assim, todo grafo MU é DU.

O interesse pelo estudo dos grafos de interseção de objetos geométricos acentuou-se nas décadas de 70 e 80. Durante este período os grafos ID aparecem na modelagem de diversos problemas [21, 29, 45, 55], mas somente na década de 90 são publicados os primeiros trabalhos a focar os grafos DU [6, 8, 20, 40]. Esses trabalhos apresentam alguns resultados relativos aos grafos MU, mas não encontramos um trabalho que tratasse exclusivamente dessa classe.

Um grafo  $G$  é **perfeito** [18] se o número cromático de cada subgrafo induzido  $H$  de  $G$  é igual ao tamanho de uma clique máxima de  $H$ . Como o complemento de um grafo perfeito também é perfeito, no complemento de  $G$  temos que um grafo é perfeito se o tamanho da menor cobertura por cliques de cada subgrafo induzido é igual ao tamanho do seu conjunto independente máximo. Muitas das famílias de grafos de interseção formam subclasses dos grafos perfeitos [4]. No entanto, os grafos MU não são necessariamente perfeitos, pois todo ciclo ímpar é grafo MU.

Quanto ao comportamento das classes de grafos DU e MU em relação

a estes quatro problemas clássicos (número cromático, clique máxima, conjunto independente máximo e cobertura por cliques mínima), temos que o problema do número cromático foi vastamente estudado [8, 39, 38, 20]. O problema da clique máxima em grafos DU e, conseqüentemente, em grafos MU, é polinomial [8]. Nosso principal objetivo é estudar o comportamento dos grafos DU e MU em relação aos dois outros destes problemas: o do conjunto independente máximo e o da a cobertura por cliques mínima. Uma das contribuições desta tese foi mostrar que ambos são NP-completos quando restritos à classe dos grafos MU. Em [8], Clark et al. já haviam provado que o problema do conjunto independente máximo é NP-completo quando restrito a grafos DU. O trabalho prossegue com a investigação de algoritmos aproximativos ou exatos quando restritos a subclasses, para a solução destes problemas.

Outro fato importante é que os grafos DU não são, necessariamente, planares. Na verdade, a classe dos planares é uma subclasse da dos grafos ID e contém a classe dos grafos MU [32]. Devido a esta relação estreita entre as classes, muitas das técnicas utilizadas para resolver problemas em grafos ID são adaptações de técnicas utilizadas nos grafos planares [11, 27].

Este texto está organizado da seguinte forma:

No Capítulo 2 apresentamos algumas definições básicas em teoria dos grafos, apresentamos detalhadamente a classe dos grafos DU, enumeramos modelos e aplicações destes grafos, descrevemos suas principais subclasses e

a relação entre elas e enumeramos outras subclasses de grafos ID.

No Capítulo 3 abordamos o problema do conjunto independente máximo nos grafos DU e MU. Além das provas de que o problema é NP-completo quando restrito a grafos DU e a MU, apresentamos também um estudo do problema complementar (cobertura de vértices), uma lista das propriedades dos grafos DU e MU e três algoritmos polinomiais da literatura: dois aproximativos para os grafos DU e um exato para uma de suas subclasses, a dos grafos  $k$ -faixa. Encerramos enumerando resultados em outras subclasses.

No Capítulo 4 abordamos o problema da cobertura por cliques em grafos MU e o problema da cobertura de discos. Este problema se assemelha ao problema da cobertura por cliques em grafos DU, mas não é equivalente ao mesmo. Além da prova de que o problema da cobertura por cliques é NP-completo quando restrito a grafos MU, apresentamos também dois novos algoritmos aproximativos que elaboramos, um de razão  $\frac{3}{2}$  para grafos MU e um de razão 3 para grafos DU. Por fim, enumeramos resultados em outras subclasses.

No Capítulo 5 apresentamos as considerações finais e direções futuras para este trabalho.

# Capítulo 2

## Definições e classes

O objetivo deste capítulo é apresentar as principais subclasses da classe dos grafos de interseção de discos e como elas se relacionam. Na Seção 2.1 descrevemos a notação adotada e algumas definições em teoria dos grafos, em especial, definições específicas aos grafos de interseção. Na Seção 2.2 descrevemos os modelos utilizados para representar os grafo DU. Na Seção 2.3 descrevemos as principais subclasses dos grafos DU e o relacionamento entre elas. Finalmente, na Seção 2.4 enumeramos a complexidade de alguns problemas clássicos quando restritos às classes apresentadas.

### 2.1 Notação e definições

Um **grafo**  $G$  é um par  $(V, E)$  onde  $V$  é um conjunto finito e não-vazio cujos elementos são chamados de **vértices** e  $E$  um conjunto finito de pares não ordenados de elementos de  $V$ . Os elementos de  $E$  são chamados de



**arestas**. Denotaremos por  $n$  o número de vértices ( $|V|$ ) e  $m$  o número de arestas ( $|E|$ ). Se  $e \in E$ ,  $e = (u, v)$  e  $u, v \in V$ , então dizemos que  $u$  e  $v$  são **adjacentes** ou **vizinhos**,  $u$  e  $v$  são **incidentes** a  $e$  e  $e$  **incide** em  $u$  e  $v$ . Chamamos de **vizinhança** de um vértice  $v \in V$ ,  $N(v)$ , o conjunto de vértices adjacentes a  $v$ . O **grau** de um vértice é igual ao tamanho de sua vizinhança e o grau de um grafo é o grau máximo entre seus vértices. O grau de um vértice  $v$  é denotado por  $\delta(v)$  e o grau de um grafo  $G$  por  $\Delta(G)$ . O **complemento** de um grafo  $G = (V, E)$  é o grafo  $\overline{G} = (V, \overline{E})$ , onde  $\overline{E} = \{(u, v) : (u, v) \notin E \text{ e } u \neq v\}$ .

Um grafo é **orientado** quando suas arestas são orientadas; nesse caso, as arestas  $(a, b)$  e  $(b, a)$  representam duas arestas distintas. Do contrário, o grafo é dito **não orientado**. Quando o grafo for orientado, chamaremos as arestas de **arcos**, e o conjunto de arcos será  $A$ . Uma **orientação** de um grafo  $G = (V, E)$  é a especificação de uma direção para cada aresta de  $G$  originando um grafo orientado. Uma orientação é **transitiva** quando a seguinte propriedade vale:  $(a, b) \in A$  e  $(b, c) \in A$  implica  $(a, c) \in A$ ,  $\forall a, b, c \in V$ . Um grafo é dito de **comparabilidade** se admite uma orientação transitiva. O complemento de um grafo de comparabilidade é um grafo de **cocomparabilidade**. Um vértice é dito **fonte** se não é extremidade final de nenhum arco que incide nele e **sumidouro** se não é extremidade inicial de nenhum arco que incide nele.

Um **caminho** em um grafo é uma seqüência de vértices distintos

$(v_0, v_1, \dots, v_k)$  tal que  $(v_{i-1}, v_i) \in E$  para  $i = 1, 2, \dots, k$ . Um caminho  $(v_1, v_2, \dots, v_k, v_{k+1})$  é um **ciclo** quando  $(v_1, v_2, \dots, v_k)$  é um caminho,  $v_1 = v_{k+1}$  e  $k > 2$ . Um grafo que não possui ciclos é chamado **acíclico**. Um caminho  $C = (v, \dots, w)$  entre dois vértices  $v, w \in V$  em um grafo  $G = (V, E)$  é **caminho mínimo** se não existir outro caminho entre  $v$  e  $w$  com menor número de vértices (arestas).

Um grafo  $G = (V, E)$  é **bipartido** quando seu conjunto de vértices pode ser particionado em dois subconjuntos  $V'$  e  $V''$  tais que toda aresta de  $G$  une um vértice de  $V'$  a um vértice de  $V''$ . Em um grafo bipartido, quando existe uma aresta para cada par de vértices  $v, w$  tais que  $v \in V'$  e  $w \in V''$ , o grafo é dito **bipartido completo**. Denotamos um grafo bipartido completo por  $K_{n_1, n_2}$ , onde  $n_1 = |V'|$  e  $n_2 = |V''|$ .

Um **subgrafo**  $G' = (V', E')$  de um grafo  $G = (V, E)$  é um grafo tal que  $V' \subseteq V$  e  $E' \subseteq E$ . O subgrafo  $G'$  de  $G$  é **subgrafo induzido** se para quaisquer  $v, w \in V'$  e  $(v, w) \in E$ ,  $(v, w) \in E'$ . Neste caso, denotamos  $G'$  por  $G[V']$ .

Grafos de interseção são grafos onde objetos são associados a vértices e dois vértices são adjacentes se a interseção entre os respectivos objetos é não vazia. Agora daremos uma definição mais formal de grafos de interseção de uma família de objetos.

**Definição 2.1 [4]** *Seja  $F = \{S_1, \dots, S_n\}$  uma família de conjuntos. O*

*grafo de interseção de  $F$* , denotado por  $\Omega(F)$ , é o grafo com conjunto de vértices  $F$  tal que  $S_i$  é adjacente a  $S_j$  se e somente se  $i \neq j$  e  $S_i \cap S_j \neq \emptyset$ . Um grafo  $G$  é simplesmente um **grafo de interseção** se existe uma família  $F$  tal que  $G \simeq \Omega(F)$ ; neste caso, dizemos que  $F$  é uma **representação** de  $G$ .

Os **grafos de interseção de objetos geométricos** são grafos de interseção onde a família de conjuntos  $F$  é formada por objetos geométricos tais como: quadrados, retângulos, discos, etc. Para uma família  $D$  de discos no plano, o grafo de interseção de  $D$  é o grafo onde cada disco no plano corresponde a um vértice e existe uma aresta entre dois vértices se e somente se os respectivos discos se interceptam (incluindo tangência). Tais grafos de interseção são chamados de **grafos de interseção de discos** ou grafos ID. Os grafos ID são definidos por uma família  $D = \{D_1, \dots, D_n\}$  de pontos no plano e uma função diâmetro  $d : V \rightarrow \mathfrak{R}^+$ , onde  $D_i$  tem centro  $c_i = (x_i, y_i)$  e diâmetro  $d(D_i)$ . Seja  $G$  o grafo com conjunto de vértices  $V$  e arestas  $\{(v, w) \mid v \neq w \text{ e } |\overline{vw}| \leq \frac{1}{2}(d(v) + d(w))\}$ , onde  $|\overline{vw}|$  é a distância euclidiana entre  $v$  e  $w$ .  $G$  é ID se e somente se  $G \simeq (V, d)$  para um certo conjunto de pontos  $V$  e uma função diâmetro  $d$ . Chamamos  $(V, d)$  de **modelo ID** de  $G$ .

Quando os discos de um modelo de um grafo ID têm mesmo tamanho, ou, equivalentemente, quando a função diâmetro for constante, o grafo é chamado **grafo de disco unitário** ou grafo DU. Alguns autores definem como discos

unitários aqueles de raio igual a 1, outros consideram o diâmetro igual a 1. A maioria dos resultados independe desta escolha, pois a unidade de distância não é relevante uma vez que uma família de discos no plano representa o mesmo grafo independentemente da escala. Neste texto denominamos como discos unitários aqueles que têm diâmetro 1.

Os problemas abordados neste trabalho são descritos como problemas de otimização. Um problema de otimização é composto de um conjunto de **instâncias**  $I$ , um conjunto de **soluções viáveis** para cada instância  $I$  e uma função que atribui um valor a cada solução viável. Em um problema de **minimização** estaremos interessados nas soluções viáveis de valor mínimo e em um problema de **maximização** procuramos soluções viáveis de valor máximo. A solução cujo valor é ótimo, mínimo ou máximo, é chamada de **solução ótima**, denotada por  $\text{OPT}(I)$ .

Um algoritmo  $\rho$ -**aproximativo** para um problema de otimização  $\Pi$  é um algoritmo de tempo polinomial que tem uma razão de aproximação  $\rho$ , isto é, para cada instância  $I$  de  $\Pi$  o valor da solução obtida pelo algoritmo está dentro de uma razão  $\rho$  do valor ótimo. Quando não especificamos  $\rho$ , dizemos simplesmente *algoritmo aproximativo*. Um **esquema polinomial de aproximação** para o problema  $\Pi$  é um conjunto de algoritmos polinomiais  $(1 + \varepsilon)$ -aproximativos para cada  $\varepsilon > 0$ . Note que um algoritmo  $A$   $\alpha$ -aproximativo para um problema de minimização devolve uma solução  $A(I)$  para qualquer instância  $I$  tal que  $\text{valor}(A(I)) \leq \alpha \text{OPT}(I)$ , onde  $\alpha \geq 1$ .

Se o problema for de maximização, então  $\text{valor}(A(I)) \geq \alpha \text{OPT}(I)$ , para  $0 < \alpha \leq 1$ . Um algoritmo 1-aproximativo é um **algoritmo exato**.

## 2.2 Modelos e aplicações

Além da definição padrão de grafo DU como grafo ID, existem outras definições que serão analisadas nessa seção.

Chamaremos de modelo de um grafo DU ao mapeamento dos vértices de um grafo DU (a discos ou pontos, conforme o tipo de modelo adotado.) Quando definida como a interseção de discos unitários no plano, a representação do grafo DU é chamada **modelo de interseção**. Existem outras definições na literatura, como os modelos de contenção e proximidade. No **modelo de contenção**, dado uma família de  $n$  discos no plano de mesmo tamanho, o grafo de contenção da família é o grafo com  $n$  vértices, onde existe aresta entre dois vértices se um dos discos contém o centro do outro. Já o **modelo de proximidade** é uma definição puramente geométrica: dados  $n$  pontos no plano, forme um grafo com  $n$  vértices onde cada vértice corresponde a um ponto e existe aresta entre dois deles se a distância euclidiana entre os pontos correspondentes é no máximo um limite  $d$  especificado. Transformações entre modelos são simples. Entre os modelos de interseção e contenção a transformação se resume a dobrar ou dividir pela metade o diâmetro dos discos. Dado um modelo de interseção, dobramos o diâmetro dos discos e obtemos o modelo de contenção. Para obter o modelo de inter-

seção a partir de um modelo de contenção dividimos pela metade o diâmetro dos discos (veja Fig. 2.1). Entre os modelos de interseção e proximidade a transformação se dá pela identificação dos centros dos discos com os pontos (veja Fig. 2.2). Logo, dado um dos modelos, a obtenção dos outros pode ser feita em tempo linear.

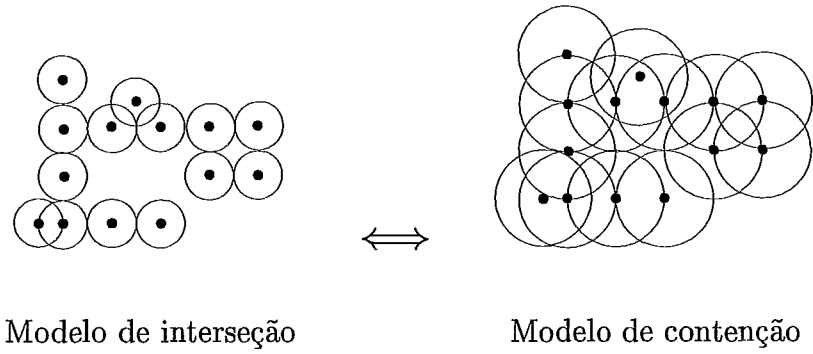


Figura 2.1: Transformação entre os modelos de interseção e contenção.

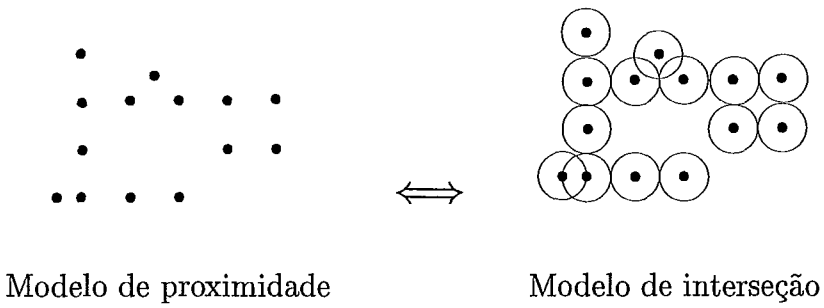


Figura 2.2: Transformação entre os modelos de proximidade e interseção.

Os grafos DU descrevem situações onde há interação entre objetos. Portanto, o modelo a ser usado está intimamente relacionado à aplicação ou problema a ser estudado. Nos problemas relacionados a difusão os trans-

missores e receptores são modelados por pontos e a área de alcance destes por discos centrados nos pontos correspondentes. Se cada estação possuir a mesma potência, temos discos de raios aproximadamente iguais. O modelo é simplificado, pois não considera alguns problemas práticos como obstáculos físicos (montanhas, prédios, etc) ou o clima da região. Apesar disso, este modelo é empregado na solução de vários problemas relacionados a sistemas de telefonia celular [33, 34, 35], atribuição de frequências [21, 37, 49] e mensageiros de emergência (*emergency senders*) [52]. No problema de atribuição de frequências, deve-se alocar diferentes frequências a transmissores cujas áreas de alcance se interceptam. Este problema corresponde ao problema da coloração de vértices em grafos ID. O problema dos mensageiros de emergência consiste em achar um conjunto mínimo de transmissores com os quais é possível transmitir em caso de emergência a todos os demais. Este problema equivale ao problema do conjunto dominante mínimo no modelo de contenção em grafos DU. Temos ainda o problema de agrupar o máximo de pontos tais que dois deles distam um do outro em no máximo  $d$ . Este é o problema da clique máxima no modelo de proximidade em grafos DU.

Além dos problemas descritos, os grafos DU são também utilizados para modelar outros problemas. Bentley, Stanat e William [29] mostram que o problema de encontrar a vizinhança de um vértice restrita a uma distância fixa (em grafos DU) tem aplicação em gráficos moleculares e decodificação de dados corrompidos. Outro problema seria a alocação de facilitadores [55], que visa cobrir a maior área possível evitando que estes estejam próximos.

Além destes, os problemas de empacotamento e cobertura são aplicados em processamento de imagens [24] e na modelagem de circuitos VLSI [44].

## 2.3 Subclasses de grafos ID e DU

Dentre as diversas subclasses dos grafos ID e DU, concentramo-nos nas mais estudadas: grade (*grid*), moeda (*coin*), moeda unitária (*penny*) e faixa (*strip*). Grafos **grade** são os grafos DU nos quais, no modelo de interseção, todos os discos possuem coordenadas inteiras. Grafos de **moeda** ou de contato são grafos ID para os quais existe um modelo onde os discos não podem se sobrepor e grafos de **moeda unitária**, ou grafos MU, são grafos que admitem uma representação cujos discos têm mesmo tamanho e não se sobrepõem. Apesar dos grafos MU serem subclasse dos grafos de moeda e dos grafos DU, a classe dos grafos MU não corresponde à interseção entre Moeda e DU. Como exemplo temos o grafo  $K_4$  que possui uma representação moeda e em discos unitários, mas não admite uma representação em discos unitários que não se sobrepõem (veja Fig. 2.3). Grafos  **$k$ -faixa** são grafos DU que admitem representações onde os centros dos discos estão restritos a uma faixa do plano de largura  $k$  dada. Quando essa largura não é especificada, admite-se  $k = \frac{\sqrt{3}}{2}$  e o grafo é chamado de grafo **faixa**. Observe que todo grafo  $k$ -faixa é  $k'$ -faixa para  $k \leq k'$  e todo grafo DU é  $k$ -faixa para algum  $k$ .

Quanto à relação entre os grafos ID e as demais classes, Koebe demonstrou que um grafo  $G$  é um grafo de moeda se e somente se  $G$  é planar, ou seja, os



grafos de moeda e os planares formam a mesma classe. A Figura 2.4 mostra um exemplo de grafo planar e sua representação por moedas.

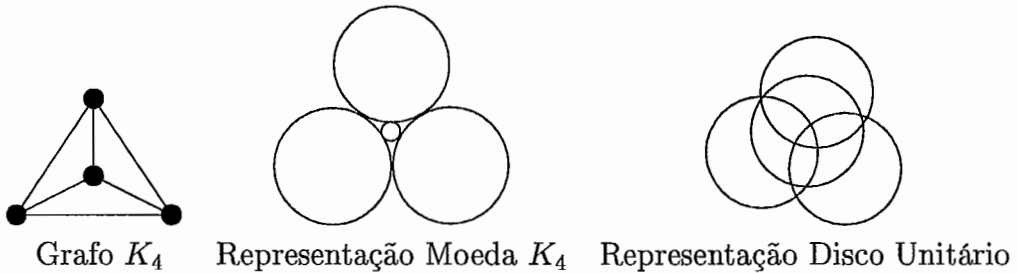


Figura 2.3: Representações do grafo  $K_4$ .

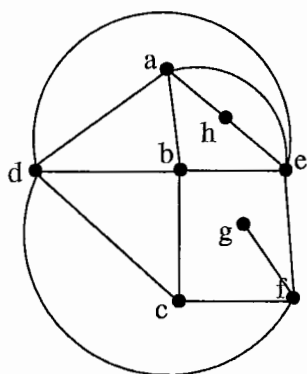
**Teorema 2.1** (Koebe, 1936 [47, 50]) *Dado um grafo planar  $G$  com um conjunto de vértices  $V(G) = \{v_1, \dots, v_n\}$  e um conjunto de arestas  $E(G)$ , podemos achar um empacotamento de  $n$  (não necessariamente congruentes) discos no plano  $C = \{C_1, \dots, C_n\}$  no plano com a propriedade que  $C_i$  e  $C_j$  se tocam se e somente se  $(v_i, v_j) \in E(G)$  para  $1 \leq i \leq n$ .*

**Prova (resumo):** É suficiente provar o teorema para grafos planares maximais. Seja  $G$  um grafo com todas as faces triangulares, conjunto de vértices  $V = \{v_1, \dots, v_n\}$ , conjunto de arestas  $E$  e conjunto de faces  $F$ . Pela fórmula de Euler temos que

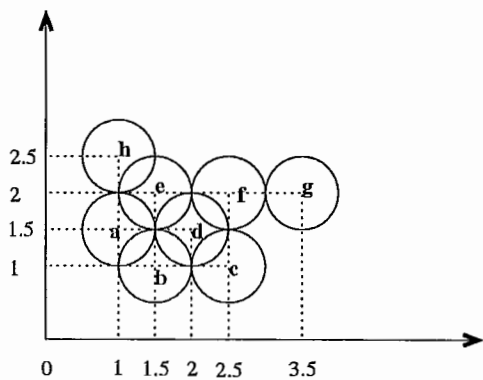
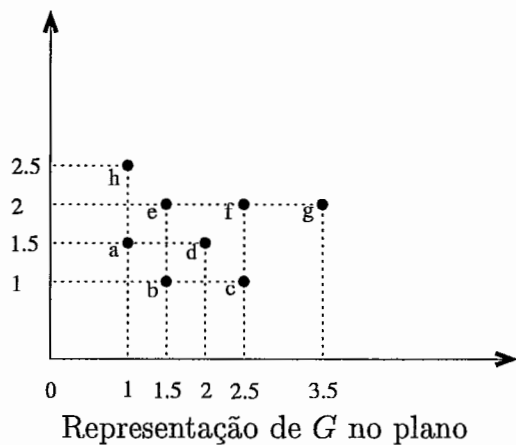
$$n - m + |F| = 2.$$

Como o grafo só possui faces triangulares,

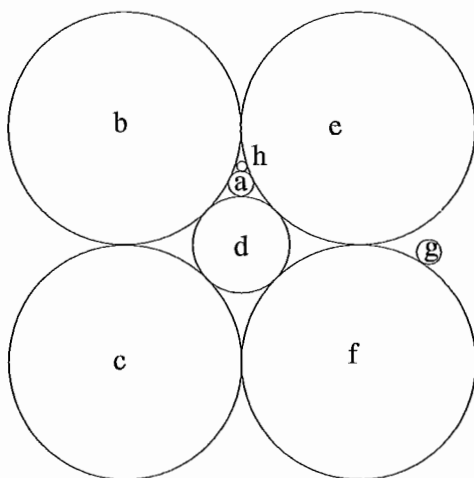
$$|F| = 2n - 4 = 2n - 4. \tag{2.1}$$



Grafo planar  $G$



Família de discos que representa  $G$



Representação por moedas de  $G$

Figura 2.4: Representações do grafo  $G$ .

Seja  $r = (r_1, \dots, r_n)$  um vetor de  $n$  reais positivos com  $r_1 + \dots + r_n = 1$ . Para cada face  $v_i v_j v_k$  de  $G$ , associamos um triângulo tal que os vértices deste são centros de discos mutuamente tangentes de raios  $r_i, r_j$  e  $r_k$  respectivamente. Tentamos então unir estes triângulos pelas arestas da mesma maneira que as respectivas faces estão unidas em  $G$ . Para cada vértice  $v_i$ , denotamos por  $\sigma_r(v_i)$  a soma dos ângulos de  $v_i$  em todos os triângulos em que ele é vértice. Se para todo vértice  $\sigma_r(v_i) = 2\pi$ , os triângulos irão se encaixar perfeitamente e teremos uma representação de  $G$  por  $n$  discos no plano de raios  $r_1, \dots, r_n$ , respectivamente. Supondo que isso não é verdade, a equação 2.1 implica que

$$\sum_{i=1}^n \sigma_r(v_i) = |F|\pi = (2n - 4)\pi.$$

Seja  $S \subseteq \mathfrak{R}^n$ . Modelamos esse problema com o simplex de dimensão  $n - 1$  definido por

$$S = \{(r_1, \dots, r_n) \mid r_i > 0 \text{ para todo } i \text{ e } \sum_{i=1}^n r_i = 1\},$$

e

$$H = \{(x_1, \dots, x_n) \mid \sum_{i=1}^n x_i = (2n - 4)\pi\},$$

considerando o mapeamento  $f : S \rightarrow H$ , onde  $f(r) = (\sigma_r(v_1), \dots, \sigma_r(v_n))$ .

Podemos assumir, sem perda de generalidade, que  $v_1, v_2$  e  $v_3$  são os vértices da face externa. Como vimos, é suficiente mostrar que, por exemplo,

$$x^* = \left(\frac{2\pi}{3}, \frac{2\pi}{3}, \frac{2\pi}{3}, 2\pi, 2\pi, \dots, 2\pi\right) \tag{2.2}$$

está na imagem de  $f$ . Para isso, prova-se que  $f : S \rightarrow P^*$  é bijetora, onde  $P^*$  é o politopo convexo definido por:

$$\sum_{i=1}^n x_i = (2n - 4)\pi \quad , \quad \sum_{i \in I} x_i < |F(I)|\pi \quad (2.3)$$

onde  $I$  varia entre todos os possíveis subconjuntos próprios e não vazios de  $\{1, \dots, n\}$ .

Para completar a prova é suficiente mostrar que o ponto  $x^* = (x_1^*, \dots, x_n^*)$ , definido em (2.2), pertence a  $P^*$ . □

Baseado nas definições das classes construímos o Diagrama 2.3

## 2.4 Outras classes

Encontramos outras subclasses dos grafos DU que aparecem com menos frequência:  $\lambda$ -precisão, *planar*  $(r, s)$ -civilized e disco duplo. Um grafo  $G$  DU é de  $\lambda$ -precisão [27], para um  $\lambda > 0$ , quando os centros dos discos de  $G$  distam em pelo menos  $\lambda$ . Para um par de reais  $r > 0$  e  $s > 0$ , um grafo é dito  $(r, s)$ -civilized se pode ser mapeado a bolas em  $\mathbb{R}^d$  cujos diâmetros são maiores ou iguais a  $r$  e menores ou iguais a  $s$ . Quando o grafo pode ser imerso em  $\mathbb{R}^2$ , o grafo é chamado de *planar*  $(r, s)$ -civilized [27]. Os grafos de **disco duplo** [39] são definidos por dois raios, interno e externo. Nos grafos de disco duplo só existirá aresta entre dois vértices quando o disco interno (disco delimitado pelo raio interno) que representa um dos vértices é

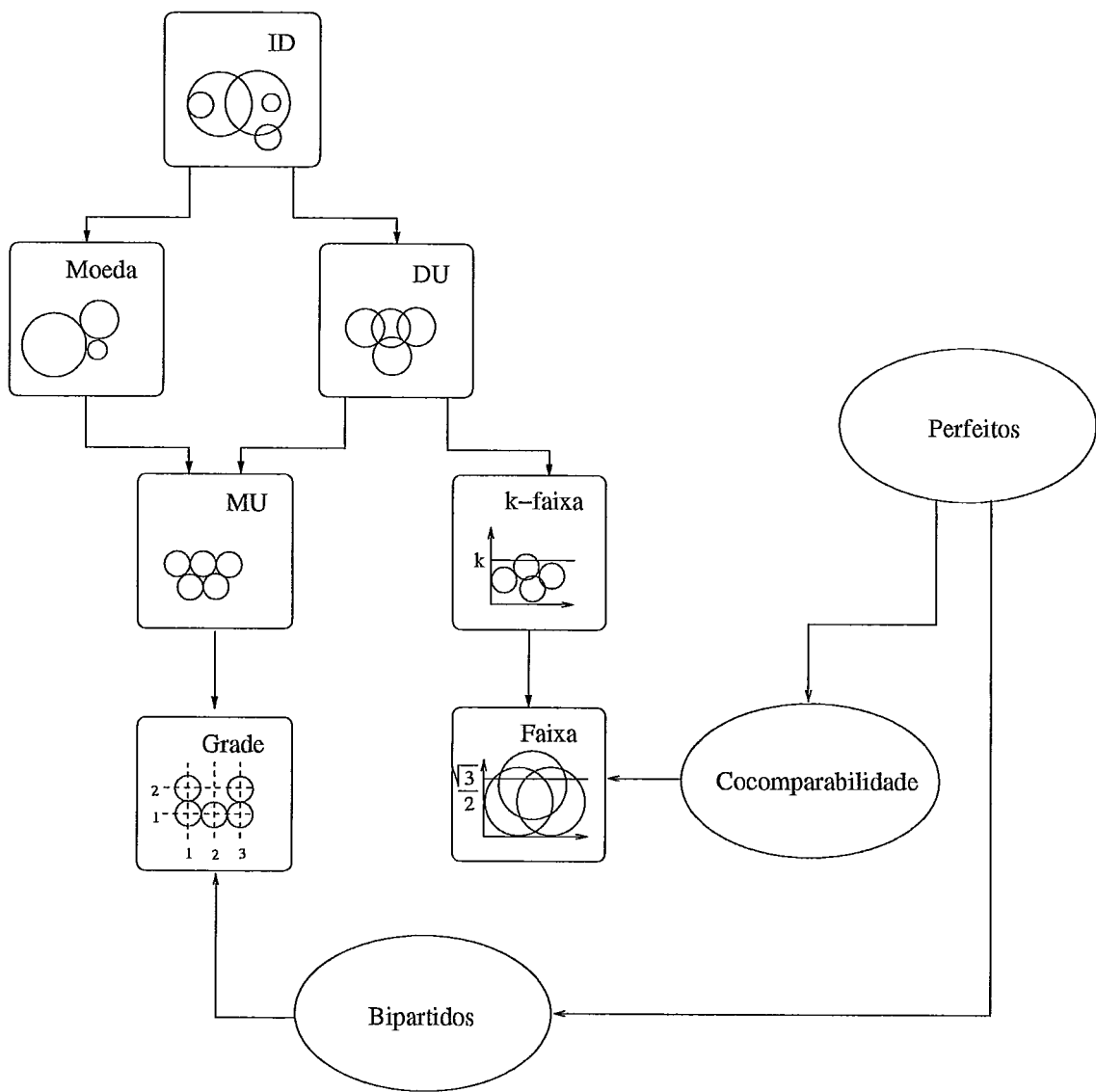


Figura 2.5: Diagrama de classes para os grafos ID.

interceptado pelo disco que representa o outro vértice, ou seja, um grafo  $G$  é de disco duplo (DD) se ele possui um modelo  $DD(V, d, D)$  que consiste em um conjunto de vértices  $V$  e duas funções diâmetro  $d$  e  $D$ , onde  $D(v) \geq d(v)$  para todo  $v \in V$ . Dois vértices  $v, w$  são adjacentes se e somente se  $v \neq w$  e  $|\overline{vw}| \leq \frac{1}{2} \max\{d(v) + D(w), D(v) + d(w)\}$ .

Existe ainda uma especificação que descreve os grafos DU de maneira hierárquica [27, 41] usando uma linguagem apresentada por Bentley, Ottmann e Windmayer em [3]. Essa especificação, conhecida como especificação BOW, é utilizada para descrever problemas relacionados à modelagem de circuitos VLSI.

## 2.5 Problemas clássicos restritos a subclasses de grafos ID

A Tabela 2.5 no final desta seção mostra a complexidade de alguns problemas clássicos quando restritos aos grafos DU, grade, moeda, MU e faixa. Os resultados relativos aos grafos grade, moeda e faixa foram obtidos em trabalhos não relacionados a grafos de interseção de discos, pois: os grafos grade são subclasse dos grafos bipartidos; os grafos faixa são de cocomparabilidade; e os grafos de moeda coincidem com os planares. Quando o problema é polinomial, é listada a complexidade do melhor algoritmo exato conhecido, e quando o problema é NP-completo, é listada a razão de aproximação e a complexidade de todos os algoritmos de aproximação conhecidos, seguidos

da referência bibliográfica. Os resultados referenciados por [\*] foram obtidos neste trabalho, os campos preenchidos com “?” indicam resultado desconhecido ou em aberto e os resultados não referenciados são consequência de trabalhos anteriores.

Dado um grafo  $G = (V, E)$ , um subconjunto  $V'$  de  $V$  é chamado de **cobertura de vértices** se para cada aresta  $(x, y)$  em  $E$  pelo menos um dos vértices,  $x$  ou  $y$ , está em  $V'$ . Um **conjunto independente máximo** é o maior subconjunto  $V'$  de  $V$  tal que não existe aresta entre quaisquer dois vértices em  $V'$ . Um **conjunto dominante** é um subconjunto  $V' \subseteq V$  tal que cada vértice em  $V - V'$  é vizinho de pelo menos um vértice em  $V'$ . Um conjunto dominante  $V' \subseteq V$  é **conexo** quando  $G[V']$  é conexo. O **número cromático** de  $G$  é o menor número de cores necessárias para se atribuir aos vértices de maneira que dois vértices vizinhos não possuam a mesma cor. Um caminho contendo todos os vértices de um grafo é chamado de **caminho hamiltoniano**. Um conjunto de cliques de  $G$  tal que todo vértice pertence a pelo menos uma das cliques e tal que seu tamanho é mínimo é uma **cobertura mínima por cliques**. Na Tabela 2.5 indicamos por  $\omega$  o tamanho da maior clique de  $G$ , por  $n$  o número de vértices e  $m$  o número de arestas. A entrada dos algoritmos, ou as instâncias dos problemas, será um grafo  $G = (V, E)$ . Com exceção do problema de reconhecimento, os algoritmos referentes aos grafos DU e MU exigem como entrada a representação geométrica do grafo.

Resumimos a seguir um esquema da prova da NP-dificuldade do reconhe-

cimento de grafos DU apresentada por Breu [6], e que pode ser estendida aos grafos ID e MU [5]: dada uma instância  $C$  de SATISFATIBILIDADE tal que cada cláusula possui no máximo três literais e cada variável aparece em no máximo três cláusulas, constrói-se um grafo  $G_C$  a partir de  $C$ . É definida uma noção de orientação (que chamaremos de “ $\pi$ -orientação”) sobre grafos gerados pela redução, e prova-se que  $C$  é satisfatível se e somente se  $G_C$  admite uma  $\pi$ -orientação. Finalmente, descreve-se como imergir  $G_C$  numa grade e mostra-se que  $G_C$  admite uma  $\pi$ -orientação se e somente se a imersão de  $G_C$  é um grafo DU.

O problema da coloração de vértices em grafos DU foi provado ser NP-completo por Gráf, Stumpf e Weissnfels, em 1998 [20]. Apesar de ter sido publicado em revista somente em 1998, uma versão anterior deste artigo pode ser encontrada em [19]. Para grafos DU temos dois algoritmos aproximativos de coloração. O primeiro, para grafos  $k$ -faixa [20], particiona o grafo de entrada em subgrafos a partir da representação do grafo. O plano é dividido em faixas de largura  $\frac{\sqrt{3}}{2}$  dividindo os discos em subconjuntos tais que um disco pertence ao subconjunto correspondente a uma faixa se seu centro encontra-se dentro da faixa. Cada subconjunto de discos corresponde a um subgrafo induzido pelos vértices representados pelos discos. Os grafos gerados são de cocomparabilidade e, portanto, perfeitos. Para gerar a coloração final, os grafos gerados são coloridos de maneira ótima e as colorações obtidas são combinadas. O segundo algoritmo foi apresentado em [40]. Na verdade, este “algoritmo” nada mais é do que o estabelecimento da razão de aproximação



de um algoritmo para grafos gerais descrito em [22] quando aplicado aos grafos DU.

O capítulo 4 da tese de Malesińska [38] estuda o problema da coloração em grafos geométricos. Os resultados estabelecem a relação entre o tamanho da maior clique e o número cromático de alguns grafos geométricos, entre eles, os grafos ID e DU.

Em [8] Clark, Colburn e Johnson apresentam um algoritmo para o problema da clique máxima em grafos DU: seja  $G$  a representação em pontos no plano de um grafo DU e  $r$  o raio usado no modelo. Para cada dois pontos  $a$  e  $b$ , considere o conjunto de pontos que distam de no máximo  $r$  de  $a$  e  $b$ . O grafo  $H_{ab}$  induzido pelos pontos compreendidos nessa área é complemento de um grafo bipartido. Achar o conjunto independente num grafo bipartido pode ser resolvido em  $O(n^{2.5})$ . Logo, para cada grafo  $H_{ab}$  calculamos o maior conjunto independente no seu complemento, que corresponde à maior clique de  $H_{ab}$ . O maior conjunto independente encontrado no complemento corresponderá à clique máxima no grafo original. Este algoritmo é exato e portanto o problema da clique máxima em grafos DU é polinomial, dada a representação em discos do grafo. Breu e Malesińska apresentam variações deste algoritmo em [5] e [38], que melhoram a complexidade do original.

Os problemas do conjunto independente e da cobertura por cliques serão abordados nos Capítulos 3 e 4.

Para o problema de achar o conjunto dominante mínimo, temos dois algoritmos. O primeiro usa a estratégia de deslocamento adaptada por Hunt III, Marathe e Radhakrishnan [27] para os grafos DU, e o segundo, desenvolvido por Marathe et al. [40] seleciona um vértice qualquer do grafo e o adiciona ao conjunto resultante, removendo sua vizinhança em seguida. Aplicando este procedimento recursivamente encontramos um conjunto independente maximal que, nos grafos de disco, tem tamanho de no máximo  $5OPT$ , onde  $OPT$  é o tamanho de um conjunto dominante mínimo. No mesmo artigo, temos um algoritmo para o conjunto dominante conexo [40]: a partir de um vértice  $v$  qualquer é construída uma árvore em largura  $T$  de  $G$  de raiz  $v$ . Seja  $k$  a altura da árvore.  $T$  é então particionada em  $k + 1$  conjuntos  $S_i$ ,  $0 \leq i \leq k$ , onde  $S_i$  é o conjunto de vértices no nível  $k$ . Para cada  $G[S_i]$  é gerado um conjunto independente  $Y_i$ . O conjunto produzido pelo algoritmo é a união de dois conjuntos: um conjunto independente maximal ( $\bigcup_{i=0}^k Y_i$ ) e um conjunto de nós escolhidos para garantir a conexidade.

PROBLEMA	DU	Grade	Moeda	MU	Faixa
Reconhecimento	NP-difícil [6]	P	$O(n)$ [26]	NP-difícil [5]	?
Número cromático	NP-completo [20] $3, O(n\omega^2 + r\omega^{2.5})$ [20] $3, O(n+m)$ [40]	Trivial	NP-completo [51]	P [1]	$O(n\omega^2)$ [20]
Clíque máxima	$O(n^{4.5})$ [8] $O(n^{3.5} \log n)$ [5] $O(n\omega^3)$ [38]	Trivial	$O(n)$ [48]	$O(n)$	$O(n^3)$ [46]
Conjunto independente	NP-completo [8] $\frac{1}{3}, O(n \log n)$ [40] $(\frac{r+1}{r})^2, nO(r^2)$ [27] $(1 - \frac{1}{r}), O(r P ^{4/2(r-1)/\sqrt{3}})$ [43]	$O(n^{2.5})$ [10]	NP-completo [2, 7] $r^{\pm 1}, O(r8^r n)$ [2]	NP-completo [*]	$O(n^2)$ [46]
Cobertura por cliques	NP-difícil $3, O(n + \overline{E})$ [*]	Trivial	NP-completo [9]	NP-difícil [*] $\frac{3}{2}, O(n \log n + m)$ [*]	$O(m + \overline{E})$ [*]
Cobertura de vértices	NP-completo [8] $(\frac{r+1}{r})^2, nO(r^2)$ [27] $1.5, O(m)$ [40]	$O(n^{2.5})$ [10]	NP-completo [2] $r^{\pm 1}, O(r8^r n)$ [2] $\frac{8}{5}, O(n^2 \log n)$ [23]	NP-completo [*]	$O(n^2)$ [12]
Conjunto dominante	NP-completo [42] $(\frac{r+1}{r})^2, nO(r^2)$ [27] $5, O(m)$ [40]	NP-completo [30]	NP-completo [2] $r^{\pm 1}, O(r8^r n)$ [2]	NP-completo	$O(nm^2)$ [5]
Conjunto dom. conexo	NP-completo [36] $10, O(m)$ [40]	NP-completo [8]	NP-completo	NP-completo	$O(nm)$ [5]
Caminho hamiltoniano	NP-completo	NP-completo [28]	NP-completo [36]	NP-completo	$O(n^3)$ [5]
Arvore de Steiner	NP-completo	NP-completo [15]	NP-completo	NP-completo	?

Tabela 2.1: Problemas em classes de grafos ID.

## Capítulo 3

# Conjunto independente em grafos DU e MU

Neste capítulo abordamos os problemas do conjunto independente máximo e da cobertura mínima de vértices nos grafos DU e MU. Por serem polinomialmente equivalentes, estabelecemos a complexidade do problema do conjunto independente máximo através da prova de NP-completude para o problema da cobertura de vértices. Na Seção 3.1 temos a descrição do problema e suas equivalências, e algumas propriedades dos grafos DU e MU. Na Seção 3.2 estão as provas de NP-completude para os dois problemas nos grafos DU e MU. Na Seção 3.3 descrevemos três algoritmos: dois aproximativos de tempo polinomial para os grafos DU e um exato para os grafos  $k$ -faixa. Descrevemos também como adaptar este último para os grafos DU. Na Seção 3.4 enumeramos resultados relativos a outras classes, descritos na bibliografia.

## 3.1 Definições e propriedades

Os problemas do conjunto independente máximo e da cobertura mínima de vértices são NP-completos para os grafos DU. Antes de mostrar estes resultados, descrevemos estes problemas em geral, mostramos como estão relacionados e enumeramos alguns resultados estruturais de grafos DU necessários para o que segue.

**Problema:** CONJUNTO INDEPENDENTE MÁXIMO (CI)

*Instância:* Grafo  $G = (V, E)$  e um inteiro positivo  $k \leq |V|$ .

*Questão:*  $G$  contém um conjunto independente de tamanho  $k$  ou maior, isto é, um subconjunto  $V' \subseteq V$  tal que  $|V'| \geq k$  e não existe aresta entre dois vértices de  $V'$ ?

**Problema:** COBERTURA MÍNIMA DE VÉRTICES (CV)

*Instância:* Grafo  $G = (V, E)$  e um inteiro positivo  $k \leq |V|$ .

*Questão:* Existe uma cobertura de vértices de tamanho  $k$  ou menor para  $G$ , isto é, um subconjunto  $V' \subseteq V$  com  $|V'| \leq k$  tal que para cada aresta  $(u, v) \in E$  pelo menos um de seus extremos pertence a  $V'$ ?

É fácil ver que, em um grafo  $G = (V, E)$ , um conjunto de vértices  $V'$  é uma cobertura de vértices se e somente se  $V - V'$  é um conjunto independente. Logo, os problemas cobertura mínima de vértices e conjunto

independente máximo são polinomialmente equivalentes, mesmo restritos a uma classe de grafos. Note ainda que o problema do conjunto independente máximo nos grafos DU corresponde a um problema geométrico conhecido como empacotamento (*packing*):

**Problema:** EMPACOTAMENTO DE DISCOS

*Instância:* Coleção  $C = \{c_1, c_2, \dots, c_n\}$  de discos no plano e um inteiro  $K \leq n$ .

*Questão:*  $C$  contém  $k$  ou mais discos tais que dois a dois não possuem pontos em comum?

**Lema 3.1** (Marathe et al., 1995, [40]) *Seja  $c$  um disco de raio  $r$  e  $S$  um conjunto de discos de raio  $r$  tal que cada disco em  $S$  intercepta  $c$  e os discos em  $S$  não se interceptam dois a dois. Então  $|S| \leq 5$ .*

**Prova:** Suponha que  $|S| \geq 6$ . Denotamos por  $s_i$ ,  $1 \leq i \leq 6$ , os centros dos discos em  $S$ ,  $c$  o centro de  $C$  e  $\vec{cs}_i$  o raio  $r_i$  ( $1 \leq i \leq 6$ ), onde  $r_i$  é o segmento de reta com extremos em  $c$  e  $s_i$ . Como existem 6 raios partindo de  $c$ , existe pelo menos um par  $r_j$  e  $r_k$  tal que o ângulo entre eles é de no máximo  $60^\circ$ . A distância entre  $s_j$  e  $s_k$  é no máximo  $2r$ , o que implica que os discos centrados em  $s_j$  e  $s_k$  se interceptam, contradizendo nossa suposição. Portanto,  $|S| \leq 5$ .

□

O lema a seguir é uma consequência imediata do lema anterior:

**Lema 3.2** (Marathe et al., 1995, [40]) *Seja  $G = (V, E)$  um grafo DU. Então  $G$  não pode conter um subgrafo induzido isomorfo a  $K_{1,6}$ .*

Ou seja, nos grafos DU, o tamanho do conjunto independente máximo no subgrafo induzido pela vizinhança de um vértice qualquer é no máximo 5. O lema a seguir especifica um tipo de vértice para o qual este conjunto é menor.

**Lema 3.3** (Marathe et al., 1995, [40]) *Seja  $G$  um grafo DU e  $v$  um vértice tal que o disco correspondente a  $v$  (em alguma representação de  $G$ ) tem a menor abcissa. O tamanho de um conjunto independente em  $G[N(v)]$  é no máximo 3.*

**Prova:** Usando o mesmo raciocínio do Lema 3.1 e admitindo-se que nenhum dos discos restantes tem abcissa maior ou igual à do disco considerado, é fácil ver que a soma dos ângulos entre os raios é no máximo  $180^\circ$ . Logo, o tamanho de um conjunto independente em  $G[N(v)]$  é no máximo 3.  $\square$

**Lema 3.4** (Valiant, 1981, [54]) *Um grafo planar  $G$  com grau máximo 4 pode ser imerso no plano usando uma área  $O(|V|)$  de maneira que seus vértices estão em coordenadas inteiras e suas arestas são desenhadas como segmentos de reta horizontais e verticais que não se cruzam.*

Chamaremos de “tamanho da aresta” a soma dos comprimentos dos segmentos de reta que unem os vértices numa representação planar como a descrita no Lema 3.4.

## 3.2 Conjunto independente para DU e MU

A partir dos resultados enumerados anteriormente Clark provou que cobertura mínima de vértices e conjunto independente máximo são NP-completos para grafos DU.

**Teorema 3.1** (Clark et al., 1990, [8]) *CV para grafos DU é NP-completo.*

**Prova:** A redução é a partir do problema da cobertura de vértices mínima em grafos planares com grau máximo 3, que é um problema NP-completo [15]. Transformamos um grafo planar  $G$  com grau máximo 3 em  $G'$ , um grafo DU, tal que  $G$  possui uma cobertura de vértices  $S$  com  $|S| \leq k$  se e somente se  $G'$  possui uma cobertura de vértices  $S'$  com  $|S'| \leq k'$ .

Usando o resultado do Lema 3.4, desenhamos  $G$  no plano. Para construir  $G'$ , substituímos cada aresta  $(x, y)$  de  $G$  por um caminho par com  $2k_{xy}$  vértices, onde  $k_{xy}$  é o tamanho da aresta entre os vértices  $x$  e  $y$  (veja Figura 3.1). O número de vértices de  $G'$  é  $|V| + \sum_{(x,y) \in E(G)} 2k_{xy}$ . Como o desenho de  $G'$  é planar, em sua representação por discos no plano não há interseção entre os  $\sum_{(x,y) \in E(G)} 2k_{xy}$  discos extras. Logo, é necessário e



suficiente que  $k_{xy}$  vértices sejam acrescentados a  $S$ . Portanto, se  $|S| \leq k$  então  $|S'| \leq k + \sum_{(x,y) \in E(G)} k_{xy}$ . Para que  $S'$  seja mínima, acrescentamos à cobertura  $S$  metade dos novos vértices, escolhendo-os alternadamente nos caminhos que substituíram as arestas. Para obter uma cobertura para  $G$  a partir de  $S'$  fazemos o processo contrário, retiramos de  $S'$  os vértices correspondentes aos discos extras. ■

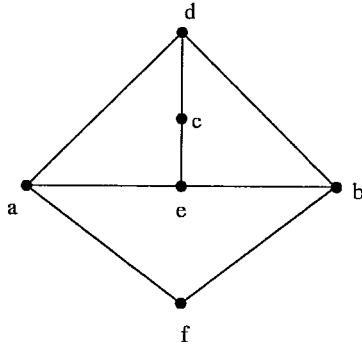
**Colorário 3.1** (Clark et al., [8]) *CI para grafos DU é NP-completo.*

Outra prova para o problema do conjunto independente máximo em grafos DU usa uma redução a partir do problema 3-SAT e pode ser encontrada em [55, 14].

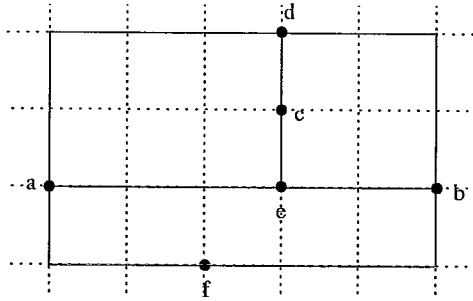
A partir de uma redução semelhante à apresentada no Teorema 3.1, provamos que o problema do conjunto independente máximo para grafos de moeda unitária (MU) também é um problema NP-completo.

**Teorema 3.2** *CV para grafos MU é NP-completo.*

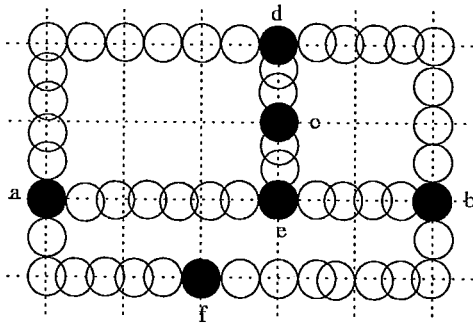
**Prova:** Assim como no teorema anterior, a redução é a partir do problema da cobertura de vértices mínima em grafos planares com grau máximo 3. A partir de um grafo planar  $G$  com grau máximo 3 construiremos  $G'$ , um grafo MU, tal que  $G$  possui uma cobertura de vértices  $S$  com  $|S| \leq k$  se e somente se  $G'$  tem uma cobertura de vértices  $S'$  com  $|S'| \leq k'$ .



a) Grafo planar  $G$



b) Grafo  $G$  em coordenadas inteiras

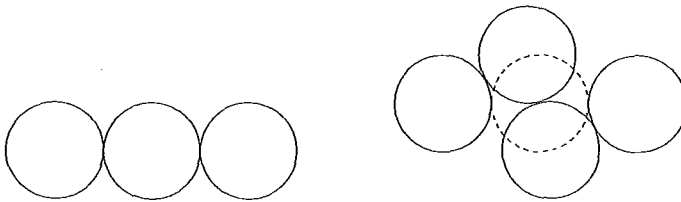


c) Representação de  $G$  em discos

Figura 3.1: CV para grafos DU

Após desenhar  $G$  no plano, construímos  $G'$  substituindo cada aresta  $(x, y)$  por um caminho par com  $4k_{xy}$  vértices, onde  $k_{xy}$  é o tamanho da aresta entre os vértices  $x$  e  $y$ . Na representação por discos de  $G'$ , é necessário e suficiente que o diâmetro dos discos seja no máximo  $\frac{u}{4}$ , onde  $u$  é a unidade de distância usada. Note que na representação de  $G'$ , além dos discos que representam vértices extras, temos os discos que representam vértices de  $G$ , excedendo o comprimento das arestas. Por isso, para cada caminho de discos extras, será necessária a substituição mostrada na Figura 3.2. Além disso, como a representação de  $G'$  é planar, não há interseção entre discos (veja a Figura 3.3).

O número de vértices em  $G'$  é  $|V| + \sum_{(x,y) \in E(G)} 4k_{xy}$ . Logo é necessário e suficiente que  $2 \sum_{(x,y) \in E(G)} k_{xy}$  vértices sejam acrescentados a  $S$ . Portanto, assim como na prova anterior, se  $|S| \leq k$  então  $|S'| \leq k + 2 \sum_{(x,y) \in E(G)} k_{xy}$ . Para obter uma cobertura para  $G$  a partir de  $S'$  fazemos o processo contrário, retiramos de  $S'$  os vértices correspondentes aos discos extras. ●



a) Número ímpar de discos    b) Número par de discos

Figura 3.2: Alteração no número de discos.

**Colorário 3.2** CI para grafos MU é NP-completo.

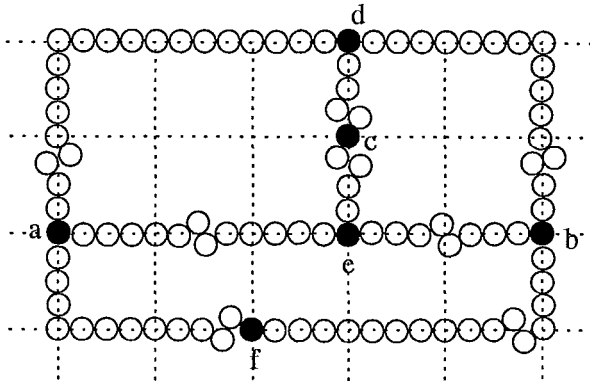


Figura 3.3: Grafo  $G'$  desenhado a partir do grafo  $G$  da Figura 3.1.

### 3.3 Algoritmos

Vimos que o problema do conjunto independente máximo continua NP-completo mesmo quando restrito à classe dos grafos DU. Além disso, sabemos que o reconhecimento dos grafos DU é um problema NP-difícil [6], e a complexidade de construir uma representação para um grafo que se sabe ser DU está em aberto. No entanto, construir um grafo DU a partir de sua representação geométrica (os  $|V|$  pontos no plano) pode ser resolvido em tempo  $O(n \log n + m)$  [5]. Este fato, juntamente com as particularidades geométricas dos grafos DU, fazem com que a maioria dos algoritmos desenvolvidos exija como entrada uma representação geométrica do grafo. Esta seção resume os principais resultados em relação aos grafos DU. Na Sessão 3.3.1 descrevemos um algoritmo simples para determinar um conjunto independente em um

grafo DU com razão de aproximação  $\frac{1}{3}$ . Na Seção 3.3.2 um algoritmo aproximativo mais sofisticado baseado numa estratégia chamada *deslocamento* com razão de aproximação  $(\frac{r}{r+1})^2$ , onde  $r$  é o menor inteiro tal que  $(\frac{r}{r+1})^2 \geq 1 - \epsilon$ , para um  $\epsilon$  dado. Na Seção 3.3.3 descrevemos finalmente um algoritmo exato para os grafos  $k$ -faixa e uma análise deste quando aplicado aos grafos DU. Os algoritmos descritos foram analisados quando aplicados aos grafos DU, mas os mesmos resultados são válidos para os grafos MU.

### 3.3.1 Algoritmo IS

O algoritmo IS descrito em [40] é um algoritmo guloso baseado no Lema 3.3. Este lema nos diz que a vizinhança de um vértice  $v$ ,  $N(v)$ , correspondente ao disco mais à esquerda em uma representação de um grafo DU, possui conjunto independente máximo de cardinalidade não superior a 3. Além disso, sabemos que qualquer subgrafo induzido de um grafo DU também é DU. Estas duas observações resultaram no algoritmo IS a seguir (veja exemplo na Figura 3.4):

---

#### Algoritmo 1 IS

---

**Entrada:** Grafo  $G$

- 1:  $IS \leftarrow \emptyset$  {IS é o conjunto resultante.}
  - 2: **repita**
  - 3:   Achar um vértice  $v \in V$  cuja vizinhança possua conjunto independente máximo de cardinalidade não superior a 3
  - 4:   Inserir  $v$  em IS e retirar  $v \cup N(v)$  de  $V$
  - 5: **até**  $V = \emptyset$
- 

**Teorema 3.3** *Seja  $G$  um grafo DU, e denote por  $IS(G)$  e  $OPT(G)$  o con-*

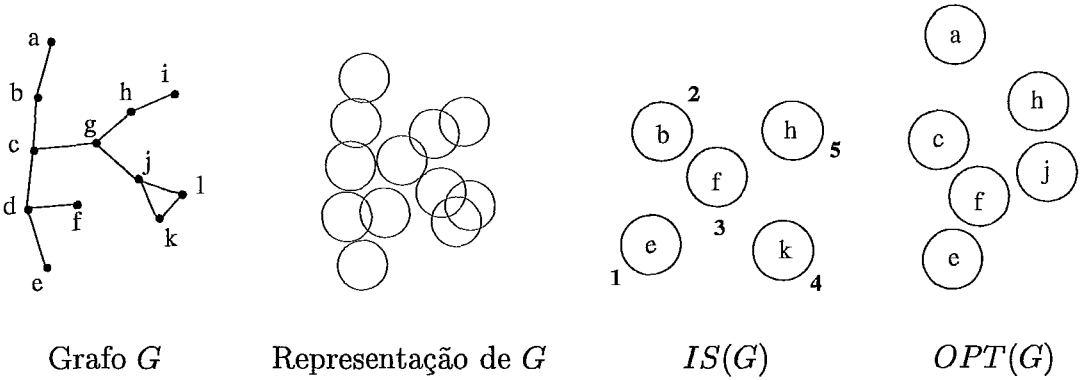


Figura 3.4: Os números indicam a ordem em que os vértices foram escolhidos.

*junto independente produzido pelo algoritmo IS e um conjunto independente máximo de  $G$ , respectivamente. Então  $|IS(G)| \geq \frac{|OPT(G)|}{3}$ .*

**Prova:** Por construção, todo vértice no grafo está na vizinhança de pelo menos um vértice em  $IS(G)$ , inclusive os vértices em  $OPT(G)$ . Também por construção, o tamanho do conjunto independente máximo em cada vizinhança contém no máximo 3 vértices de  $OPT(G)$ . Portanto, para cada vértice  $v$  em  $|IS(G)|$ , ou  $v \in |OPT(G)|$  ou no máximo 3 vértices de sua vizinhança pertencem a  $|OPT(G)|$ . Logo,  $|OPT(G)| \leq 3|IS(G)|$ .  $\square$

O algoritmo IS, como descrito, independe da representação geométrica do grafo. A complexidade de se achar um vértice com uma vizinhança com conjunto independente máximo de tamanho não superior a 3 é  $O(n^5)$ , como explica-se a seguir. Seja  $v$  um vértice. Para cada quatro vértices de  $N(v)$

verifica-se se eles formam um conjunto independente; isto consome tempo  $O(n^4)$ . Como essa busca é feita para cada vértice, a complexidade resultante é de  $O(n^5)$ . No entanto, essa complexidade pode ser melhorada se a representação do grafo for dada. Os autores afirmam que o algoritmo pode ser implementado em  $O(n^2)$ , construindo-se a lista de vizinhos de cada vértice, em tempo  $O(n^2)$ , e uma lista crescente  $L$  dos centros de todos os discos ordenada pelas abcissas, em tempo  $O(n)$ . Como num grafo DU a vizinhança do disco mais à esquerda tem conjunto independente máximo de cardinalidade não superior a 3, os vértices são escolhidos seguindo a ordem da lista  $L$  de centros. No entanto, Breu [5] mostra como listar a vizinhança de todos os vértices em tempo  $O(n \log n)$ , dada a representação do grafo. Logo, o algoritmo pode na verdade ser implementado em tempo  $O(n \log n)$ .

Note que o algoritmo guloso para um grafo DU qualquer tem uma razão de aproximação  $\frac{1}{5}$ . No entanto, dada a representação geométrica do grafo, usamos uma particularidade dos grafos DU que nos permite estabelecer uma ordem de escolha que melhora a razão de aproximação do algoritmo guloso (IS) para  $\frac{1}{3}$ .

### 3.3.2 Algoritmo FMIS

A estratégia de deslocamento (*shifting strategy*) é usada para obter esquemas de aproximação de tempo polinomial para vários problemas: por Baker [2], em problemas restritos a grafos planares; por Hochbaum e Maass

[24, 25], em problemas de cobertura e empacotamento no plano; por Feder e Greene [13], em um problema de localização. Seja  $\Pi$  um problema que pode ser resolvido por uma abordagem de divisão e conquista com uma razão de aproximação  $\rho$ . A estratégia de deslocamento nos permite limitar  $\rho$  ao erro de uma divisão e conquista simples, aplicando-a iterativamente e escolhendo a melhor solução entre as obtidas em cada iteração.

Um esquema NC de aproximação é um esquema de aproximação que tem tempo polilogaritmo num computador paralelo com um número polinomial de processadores. Ou seja, um problema está em NC (Nick's Class) se existem duas constantes  $c$  e  $k$  tais que possa ser resolvido em  $O((\log n)^c)$  usando  $O(n^k)$  processadores em paralelo. O algoritmo FMIS descrito em [27] é um esquema de aproximação NC que usa a estratégia de deslocamento. Dada uma família de discos contidos numa área  $I$  do plano, dividimos o plano em faixas horizontais de largura 1. Dado um  $\epsilon > 0$ , achamos o menor inteiro  $r$  tal que  $(\frac{r}{r+1})^2 \geq 1 - \epsilon$ . Em seguida, para cada  $i$ ,  $0 \leq i \leq r$ , particionamos os discos em  $l$  conjuntos disjuntos  $G_1, \dots, G_l$  removendo os discos que interceptam as linhas divisórias entre faixas congruentes a  $i \pmod{r+1}$ . Cada faixa é fechada à esquerda e aberta à direita. Para cada conjunto  $G_p$ ,  $1 \leq p \leq l$ , achamos um conjunto independente pelo menos  $\frac{r}{r+1}$  vezes o tamanho de um conjunto independente ótimo para  $G_p$ . O conjunto independente de cada partição é a união dos conjuntos independentes de  $G_p$ . E o conjunto resultante será o maior dentre os gerados para as partições. Essa é a versão seqüencial do algoritmo. A versão paralela divide as partições resultantes



removendo os discos congruentes a faixas verticais no plano (este será o passo 7).

No algoritmo FMIS, o passo 4 é executado  $r + 1$  vezes e o passo 6 é  $n^{O(r^2)}$  pois o tamanho máximo de um conjunto independente num grafo de interseção de discos cuja representação está contida num quadrado de lado  $r$  é  $O(r^2)$ . Logo, a complexidade da versão paralela do algoritmo FMIS é  $n^{O(r^2)}$ . Na versão seqüencial, os passos 6 a 13 são suprimidos e usa-se programação dinâmica para encontrar os conjuntos  $IS(G_{i,j})$ . Seja  $R$  uma área retangular de largura  $r$  e comprimento 1 em  $G_{i,j}$ . Seja  $G_R$  o grafo induzido pelos vértices correspondentes aos discos cujos centros estão no interior de  $R$ . Como  $G_{i,j}$  é um grafo DU, o número máximo de discos mutuamente disjuntos em  $R$  é  $O(r)$ . Ao retirarmos os discos em  $R$ , geramos dois subgrafos disjuntos de  $G_{i,j}$ ,  $G'$  e  $G''$ . Para cada combinação de discos disjuntos de  $G_{i,j}$  em  $R$ , temos uma possibilidade de conjunto independente máximo em  $R$ . Seja  $\rho$  o número de vértices de  $G_R$ . Para os subgrafos  $G_R \cup G'$  e  $G_R \cup G''$ , mantemos uma tabela de no máximo  $O(\rho^{O(r)})$  entradas correspondentes a conjuntos independentes máximos, para cada subgrafo e para cada possibilidade de conjunto independente máximo em  $G_R$ . A união dos conjuntos gerados quando os discos selecionados em  $R$  coincidem em ambos os subgrafos nos dá um conjunto independente para o grafo todo. O tempo total do algoritmo FMIS em sua versão seqüencial é  $n^{O(r)}$  e a razão de aproximação é de  $\frac{r}{r+1}$ .

Criamos um exemplo de funcionamento do algoritmo FMIS, que pode

ser visto na Figura 3.5. A razão de aproximação deste algoritmo em sua versão paralela é de  $(\frac{r}{r+1})^2$ . Para mostrar isso, primeiro provamos que pelo menos em uma iteração do passo 4 o número de vértices não considerados na computação do conjunto independente é uma fração pequena do conjunto independente ótimo.

---

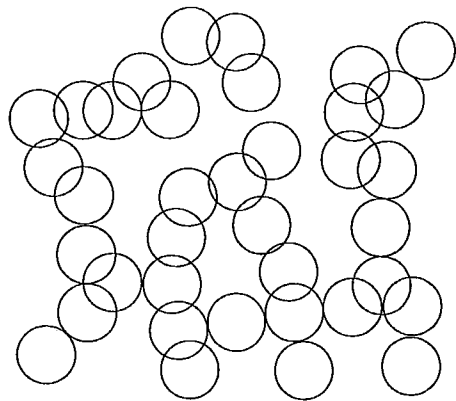
**Algoritmo 2** FMIS (versão paralela)

---

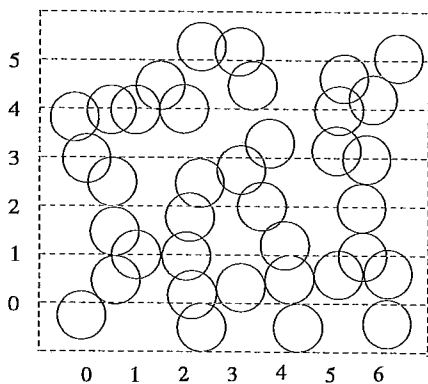
**Entrada:** Representação de  $G$  no plano

- 1: Achar o menor inteiro  $r$  tal que  $(\frac{r}{r+1})^2 \geq 1 - \epsilon$
  - 2: Dividir o plano em faixas horizontais de largura 1
  - 3: Dividir o plano em faixas verticais de largura 1
  - 4: **para** cada  $0 \leq i \leq r$  **faça**
  - 5:   Particionar o conjunto de discos em  $l_i$  conjuntos disjuntos  $G_{i,1}, \dots, G_{i,l_i}$  removendo todos os discos em toda faixa horiz. de índice  $i \bmod (r+1)$   
 $\{G_i = \bigcup_{1 \leq j \leq l_i} G_{i,j}\}$
  - 6:   **para** cada  $1 \leq j \leq l_i$  **faça**
  - 7:     **para** cada  $0 \leq i_1 \leq r$  **faça**
  - 8:      Particionar o conjunto de discos no conjunto  $G_{i,j}$  em  $s_j$  conjuntos disjuntos  $G_{i_1,1}^{i_1}, \dots, G_{i_1,s_j}^{i_1}$  removendo todos os discos em toda faixa vertical de índice  $i_1 \bmod (r+1)$   
 $\{G_{i,j}^{i_1} = \bigcup_{1 \leq j_1 \leq s_j} G_{i,j}^{i_1,j_1}\}$
  - 9:      Resolver o problema de maneira ótima para cada  $G_{i,j}^{i_1,j_1}$
  - 10:       $\{IS(G_{i,j}^{i_1,j_1})\}$  é um conjunto independente ótimo para  $G_{i,j}^{i_1,j_1}$
  - 11:       $IS(G_{i,j}^{i_1}) \leftarrow \bigcup_{1 \leq j_1 \leq s_j} IS(G_{i,j}^{i_1,j_1})$
  - 12:     **fim para**
  - 13:      $IS(G_{i,j}) \leftarrow \max_{0 \leq i_1 \leq r} IS(G_{i,j}^{i_1})$
  - 14:   **fim para**
  - 15:    $IS(G_i) \leftarrow \bigcup_{1 \leq j \leq l_i} IS(G_{i,j})$
  - 16: **fim para**
  - 17:  $IS(G) \leftarrow \max_{0 \leq i \leq r} IS(G_i)$
- 

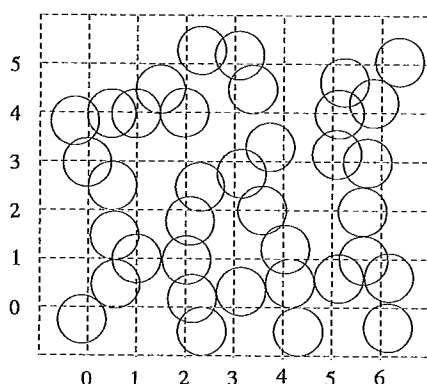
**Lema 3.5**  $\max_{0 \leq i \leq r} |OPT(G_i)| \geq \frac{r}{r+1} |OPT(G)|$ .



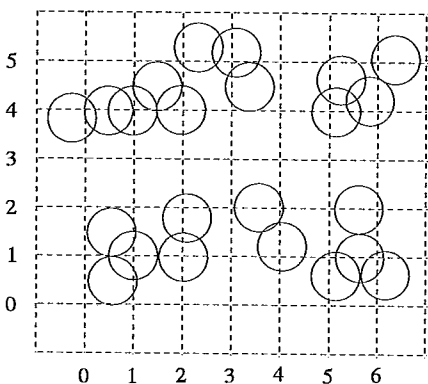
Representação do grafo  $G$



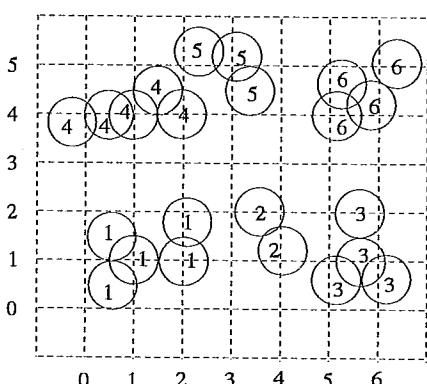
Passo 2



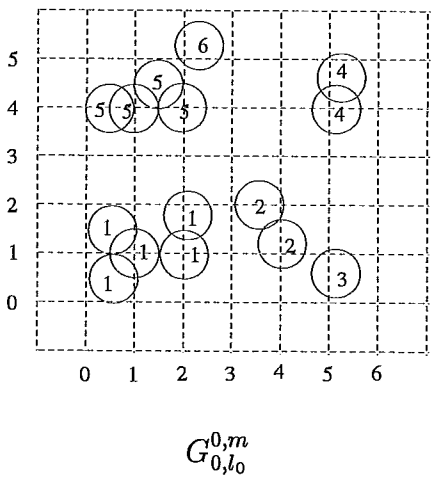
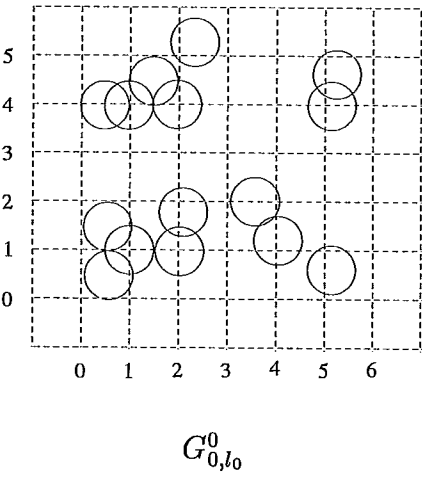
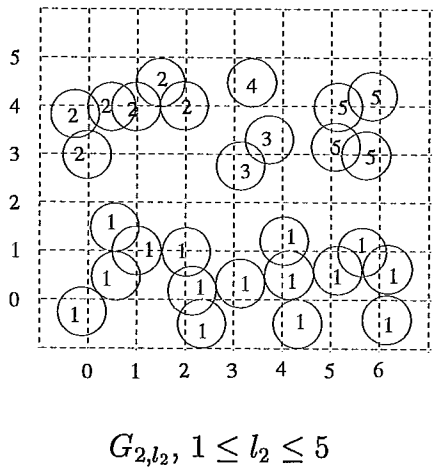
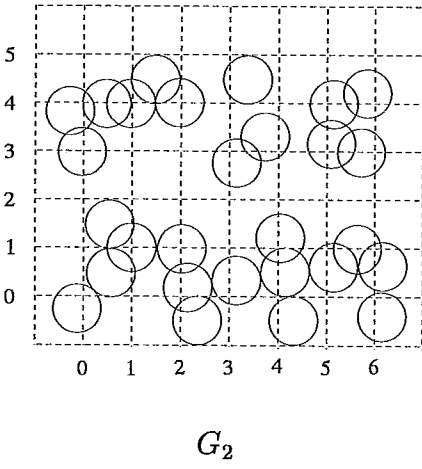
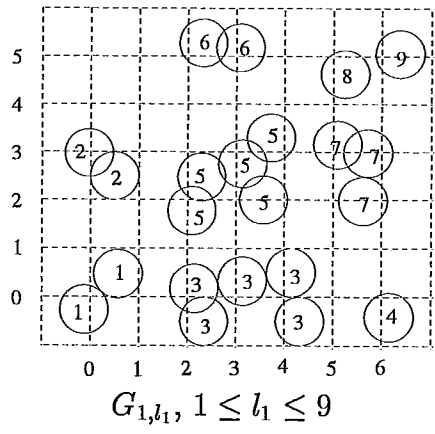
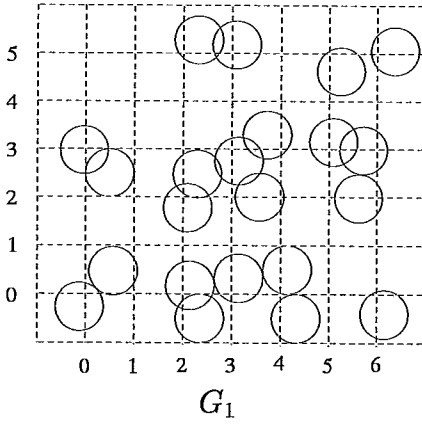
Passo 3

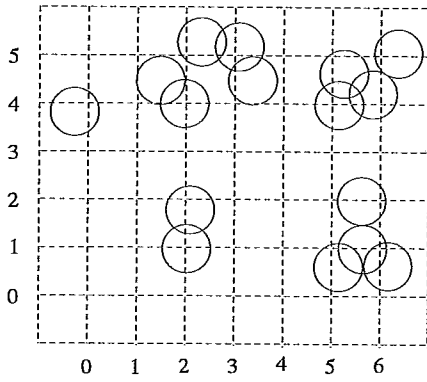
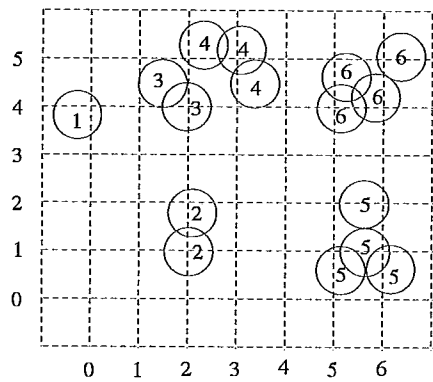
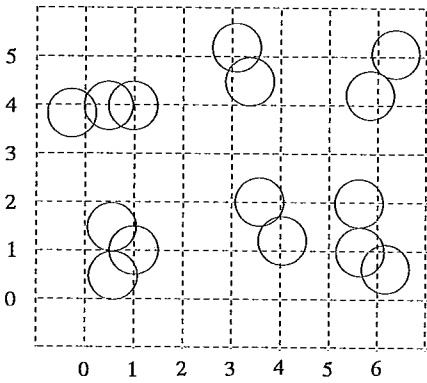
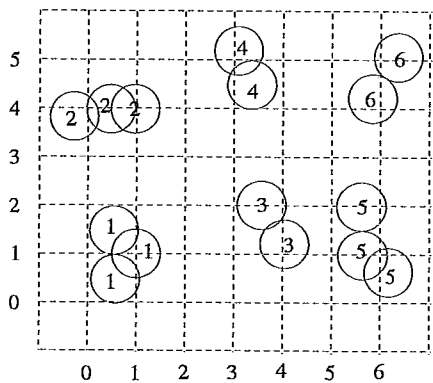
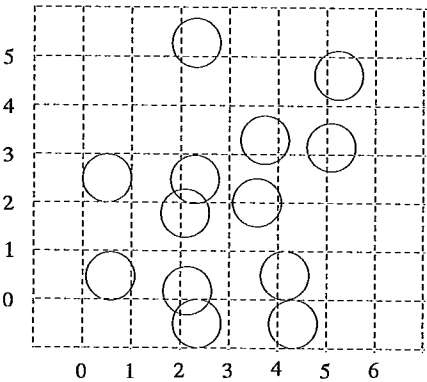
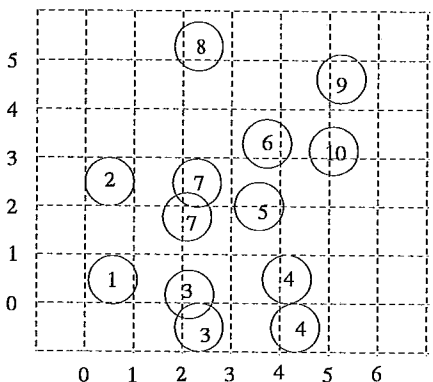


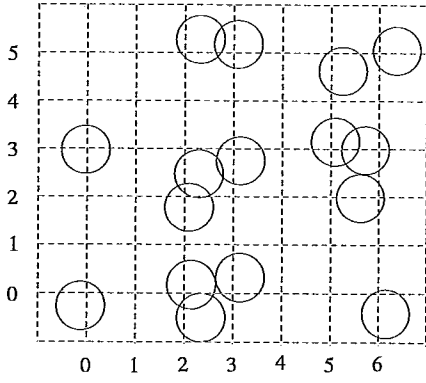
Grafo  $G_0$



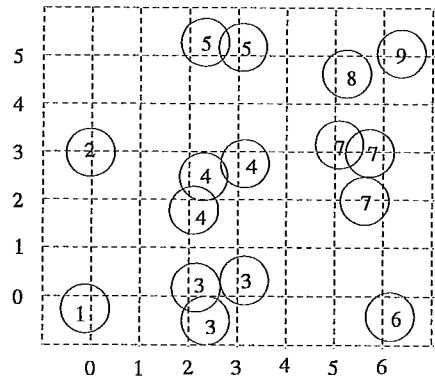
$G_{0,l_0}$ ,  $1 \leq l_0 \leq 6$



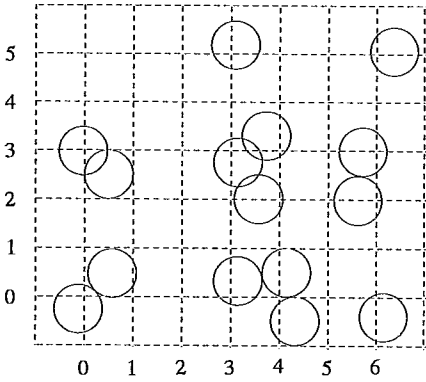

 $G_{0,l_0}^1$ 

 $G_{0,l_0}^{1,n}$ 

 $G_{0,l_0}^2$ 

 $G_{0,l_0}^{2,o}$ 

 $G_{1,l_1}^0$ 

 $G_{1,l_1}^{0,p}$



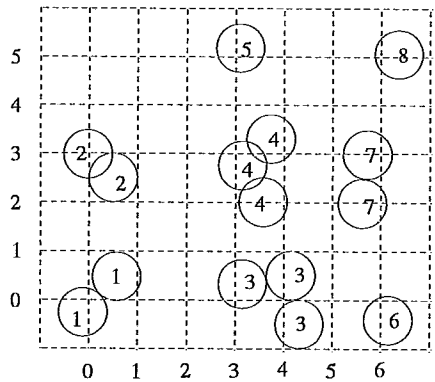
$G_{1,l_1}^1$



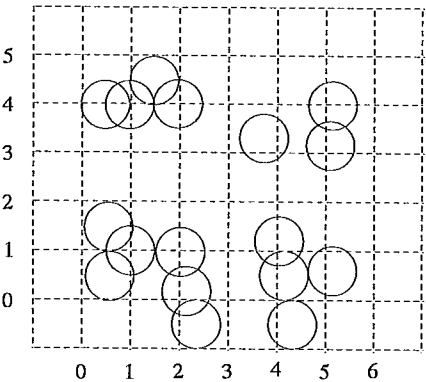
$G_{1,l_1}^{1,q}$



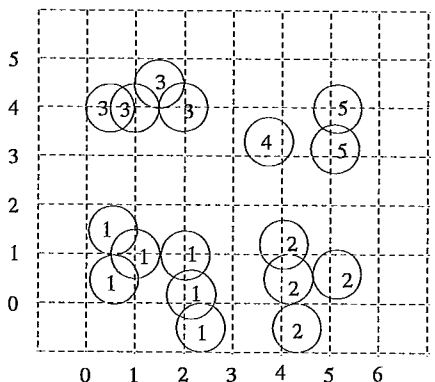
$G_{1,l_1}^2$



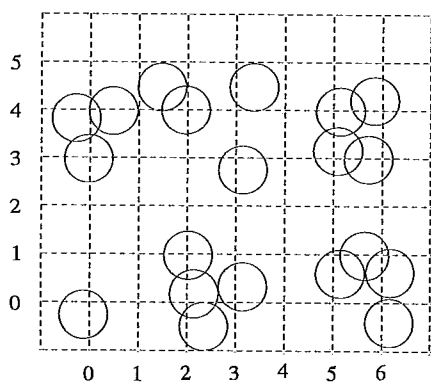
$G_{1,l_1}^{2,r}$



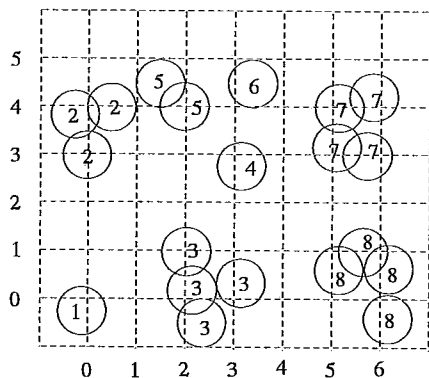
$G_{2,l_2}^0$



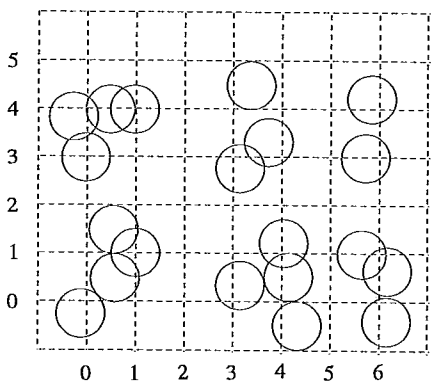
$G_{2,l_2}^{0,s}$



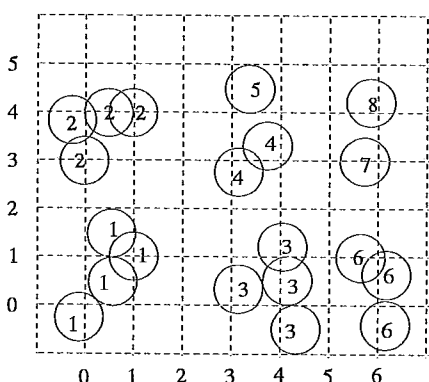
$G_{2,l_2}^1$



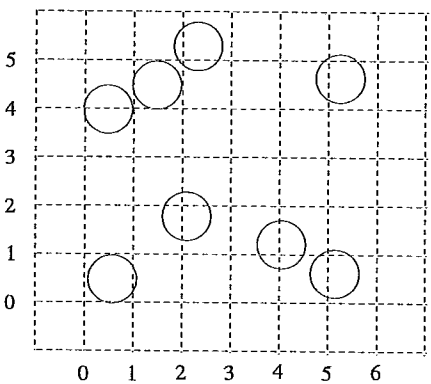
$G_{2,l_2}^{1,t}$



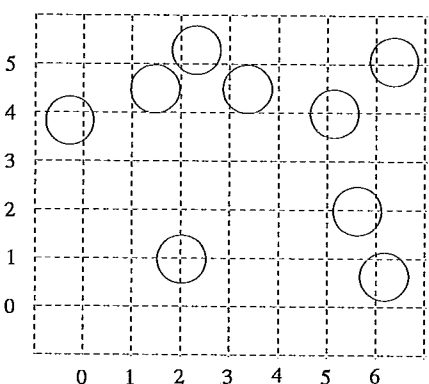
$G_{2,l_2}^2$



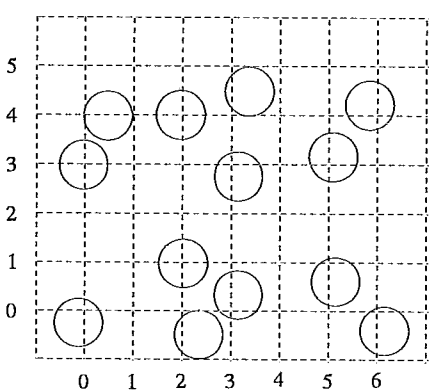
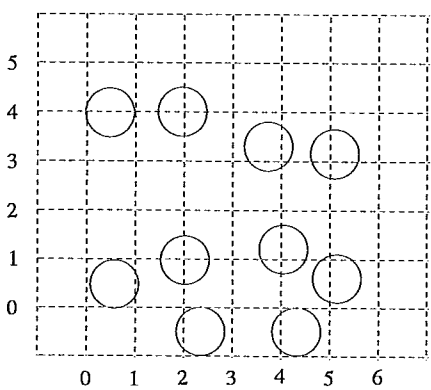
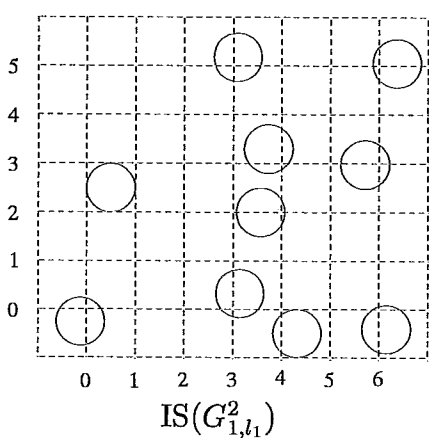
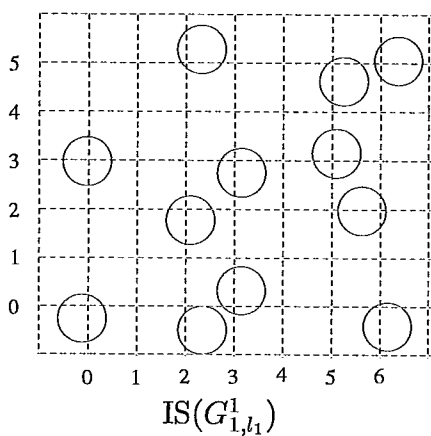
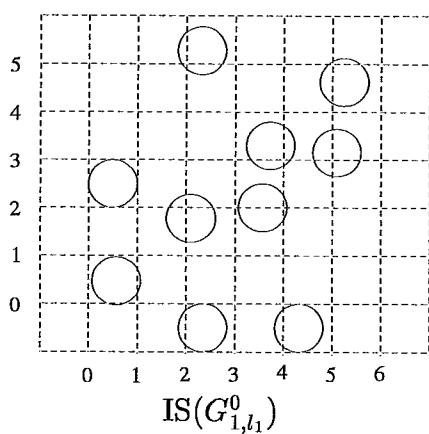
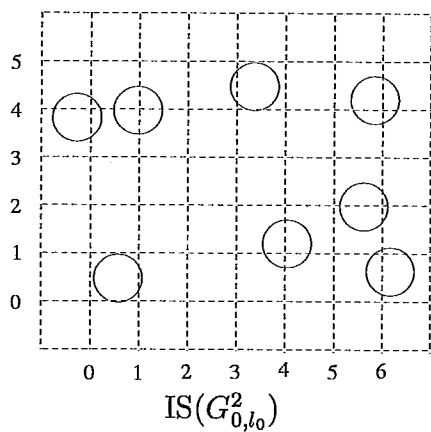
$G_{2,l_2}^{2,u}$



$IS(G_{0,l_0}^0)$



$IS(G_{0,l_0}^1)$





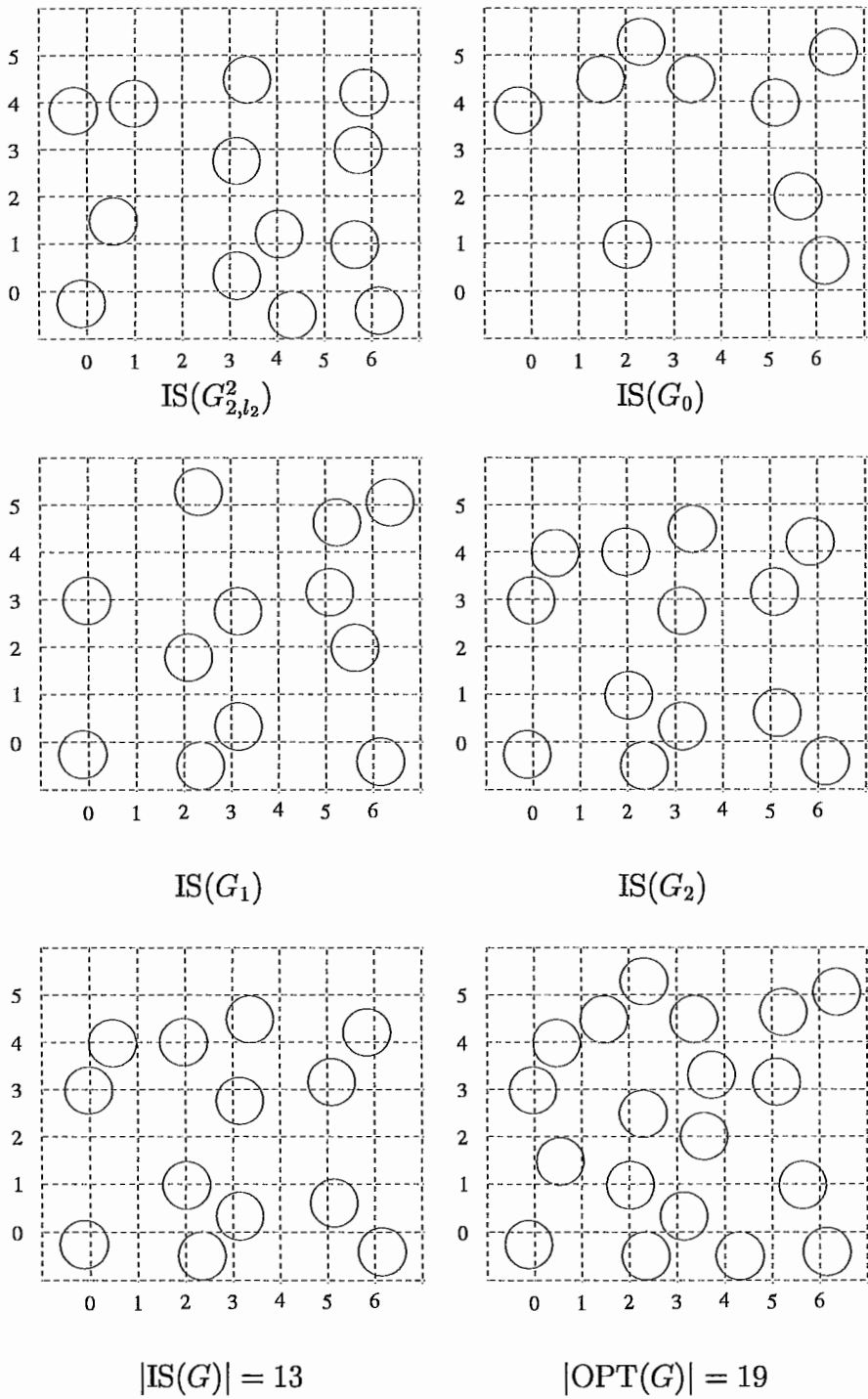


Figura 3.5: Uma execução de FMIS para  $r = 2$

**Prova:** Seja  $S_i$  o conjunto de discos não considerados na iteração  $i$  e  $IS_{opt}(S_i)$  os vértices em  $S_i$  que foram escolhidos no conjunto independente ótimo  $OPT(G)$ . Observe que

$$S_i \cap S_j = \emptyset \text{ para } 0 \leq i, j \leq r \text{ (} i \neq j \text{)} \quad \text{e} \quad \bigcup_{t=0}^{t=r} S_t = V(G).$$

Como diferentes níveis são considerados em diferentes iterações, das equações anteriores temos que

$$|IS_{opt}(S_0)| + |IS_{opt}(S_1)| + \dots + |IS_{opt}(S_r)| = |OPT(G)|.$$

Logo,

$$\begin{aligned} \min_{0 \leq t \leq r} |IS_{opt}(S_t)| &\leq \frac{|OPT(G)|}{(r+1)}, \\ \max_{0 \leq i \leq r} |OPT(G_i)| &\geq |OPT(G)| - \min_{0 \leq t \leq r} |IS_{opt}(S_t)| \\ &\geq \frac{r}{(r+1)} |OPT(G)|. \end{aligned}$$

□

**Lema 3.6**  $|IS(G_{i,j})| \geq (\frac{r}{r+1}) |OPT(G_{i,j})|$ .

**Prova:** Fixando uma iteração  $i$  e aplicando o Lema 3.5 a cada grafo  $G_{i,j}$  individualmente, temos que para algum  $0 \leq i_1 \leq r$

$$|OPT(G_{i_1}^{i_1})| \geq \frac{r}{(r+1)} |OPT(G_{i_1,j})|.$$

Pelo passo 6 do algoritmo FMIS temos

$$\begin{aligned} |OPT(G_{i_1}^{i_1})| &= \sum_{j_1=1}^{j_1=s_j} |OPT(G_{i_1,j}^{i_1,j_1})|, \\ |OPT(G_{i_1,j})| &= \sum_{j=1}^{j=r} |OPT(G_{i_1,j})|. \end{aligned} \tag{3.1}$$

A partir das equações acima temos

$$\begin{aligned} |IS(G_{i_1,j})| &= \max_{0 \leq i_1 \leq r} |IS(G_{i_1}^{i_1})| \quad (\text{passo 18}) \\ &= \max_{0 \leq i_1 \leq r} \sum_{j_1=1}^{j_1=s_j} |IS(G_{i_1,j}^{i_1,j_1})| \quad (\text{passo 16}) \\ &= \max_{0 \leq i_1 \leq r} \sum_{j_1=1}^{j_1=s_j} |OPT(G_{i_1,j}^{i_1,j_1})| \quad (\text{passo 11}) \\ &= \max_{0 \leq i_1 \leq r} |OPT(G_{i_1}^{i_1})| \quad (\text{eq. 2.1}) \\ &\geq \frac{r}{(r+1)} |OPT(G_{i_1,j})| \quad (\text{Lema 3.5}). \end{aligned}$$

□

Aplicando o Lema 3.6 repetidamente temos:

**Teorema 3.4**  $|IS(G)| \geq (\frac{r}{r+1})^2 |OPT(G)|$ .

**Prova:** Pelo Lema 3.5, sabemos que existe uma iteração para a qual o valor de  $i$  é tal que  $|OPT(G_i)| \geq (\frac{r}{r+1}) |OPT(G)|$ . Note ainda que

$$|OPT(G_i)| = \sum_{j=1}^{j=r} |OPT(G_{i,j})|.$$

A partir das equações anteriores temos que

$$\begin{aligned} |IS(G)| &= \max_{0 \leq i \leq r} |IS(G_i)| \\ &= \max_{0 \leq i \leq r} \sum_{j=1}^{j=r} |IS(G_{i,j})| \quad (\text{passo 16}) \\ &\geq \frac{r}{r+1} \max_{0 \leq i \leq r} \sum_{j=1}^{j=r} |OPT(G_{i,j})| \quad (\text{Lema 3.6}) \\ &\geq \frac{r}{r+1} \max_{0 \leq i \leq r} |OPT(G_i)| \quad (\text{passo 16}) \\ &\geq (\frac{r}{r+1})^2 |OPT(G)| \quad (\text{Lema 3.5}) \end{aligned}$$

□

### 3.3.3 Algoritmos para grafos $k$ -faixa

Um grafo DU é  $k$ -faixa quando a família  $D$  de discos associada a seus vértices está definida numa largura  $k$ , ou seja, para todo grafo  $k$ -faixa definido por  $D$ , os centros dos discos em  $D$  estão contidos na região  $S_k = \{(x, y) \in \mathbb{R}^2 | 0 \leq y < k\}$ .

### Algoritmo exato para grafos $k$ -faixa, $k \leq \frac{\sqrt{3}}{2}$

Quando  $k \leq \frac{\sqrt{3}}{2}$ , o problema do conjunto independente pode ser resolvido em tempo polinomial, pois os grafos  $k$ -faixa com  $k \leq \frac{\sqrt{3}}{2}$  são de cocomparabilidade. O lema a seguir esclarece essa afirmação.

**Lema 3.7** (Breu, 1996, [5]) *Se  $D \subseteq S_k$  e  $k \leq \frac{\sqrt{3}}{2}$ , então o grafo DU  $G$  com conjunto de vértices  $D$  é um grafo de cocomparabilidade.*

**Prova:** Seja  $G$  um grafo DU e  $\overline{G}$  seu complemento. Construimos um grafo orientado  $\vec{G}$  direcionando as arestas de  $\overline{G}$  como segue. Sejam  $c_1 = \{x_1, y_1\}$  e  $c_2 = \{x_2, y_2\}$  dois centros de discos em  $D$  cujos vértices correspondentes são adjacentes em  $\overline{G}$ . Denotemos os vértices exatamente como  $c_1$  e  $c_2$ , para não sobrecarregar a notação. Seja  $e = (c_1, c_2)$ . Podemos assumir, sem perda de generalidade, que  $x_1 < x_2$ . Direcionamos então a aresta  $e$  de  $c_1$  para  $c_2$ . Observe que  $x_1 \neq x_2$ , pois como  $y_2 - y_1 \leq \frac{\sqrt{3}}{2}$ , se  $x_1 = x_2$  teríamos que  $c_1$  e  $c_2$  seriam vizinhos em  $G$ . Mostraremos agora que a orientação obtida é transitiva.  $\vec{G}$  é acíclico pois suas arestas são sempre orientadas pela abcissa (da menor à maior). Sejam  $e = (c_1, c_2)$  e  $f = (c_2, c_3)$  arestas de  $\vec{G}$  e  $(x_i, y_i)$  as coordenadas de  $c_i$  para  $i = 1, 2, 3$ . Como  $c_1$  e  $c_2$  não são adjacentes em  $G$ ,  $\|c_1 - c_2\| > 1$ . Além disso,  $k \leq \frac{\sqrt{3}}{2}$  e  $|y_2 - y_1| \leq k$ . Logo,

$$(y_2 - y_1)^2 + (x_2 - x_1)^2 > 1$$

$$\begin{aligned}
x_2 - x_1 &> \sqrt{1 - (y_2 - y_1)^2} \\
&\geq \sqrt{1 - k^2} \\
&\geq \frac{1}{2}.
\end{aligned}$$

Do mesmo modo,  $x_3 - x_2 > \frac{1}{2}$ . Portanto,

$$\begin{aligned}
x_3 - x_1 &= (x_3 - x_2) + (x_2 - x_1) \\
&> \frac{1}{2} + \frac{1}{2} \\
&= 1.
\end{aligned}$$

Isto implica que a aresta  $(c_1, c_3)$  pertence a  $\overline{G}$  e, como  $x_1 < x_3$ , ela está orientada de  $c_1$  para  $c_3$  em  $\vec{G}$ . Assim, a orientação é transitiva e  $\overline{G}$  é grafo de comparabilidade. Isto é, seu complemento  $G$  é um grafo de cocomparabilidade. □

Pelo Lema 3.7, sabemos que os grafos  $k$ -faixa para  $k \leq \frac{\sqrt{3}}{2}$  são de cocomparabilidade. Os grafos de cocomparabilidade[18] são perfeitos e, para esta classe, o algoritmo dos elipsóides resolve o problema do conjunto independente máximo em tempo polinomial. No entanto, para os grafos de cocomparabilidade existe uma outra solução linear por um algoritmo combinatório. Este algoritmo, descrito a seguir, resolve o problema da clique máxima em grafos de comparabilidade. Como vimos, o complemento de

um grafo de comparabilidade é um grafo de cocomparabilidade; conseqüentemente, o algoritmo descrito resolve CI para grafos de cocomparabilidade. Seja  $G = (V, E)$  um grafo de comparabilidade e  $F$  uma orientação acíclica e transitiva de  $G$ . Definimos uma função *altura* que associa a cada vértice  $v \in V$  um valor,  $h(v) = 0$  se  $v$  é uma fonte e  $h(v) = 1 + \max\{h(w) \mid (w, v) \in F\}$  caso contrário. Devido à transitividade de  $F$ , todo caminho em  $\vec{G}$  (grafo  $G$  orientado por  $F$ ) corresponde a uma clique em  $G$ . Logo, o tamanho da maior clique de  $G$  é igual ao número de vértices no maior caminho em  $\vec{G}$ , equivalente a  $1 + \max\{h(v) \mid v \in V\}$ . Cada caminho em  $\vec{G}$  corresponde a um conjunto independente em  $\overline{G}$ , e como  $F$  é uma orientação acíclica de  $G$ , podemos achar o conjunto independente máximo em  $\overline{G}$  resolvendo o problema do caminho máximo em  $\vec{G}$ .

---

**Algoritmo 3** *altura*( $v$ ) - Cálculo de  $h(v)$

---

```

1: Se  $Adj(v) = \emptyset$  então
2:    $\{Adj(v)$  é a lista dos vizinhos de entrada de  $v$  em  $\vec{G}\}$ 
3:    $h(v) \leftarrow 0$ 
4:    $v.h \leftarrow t$ 
5:    $v.prox \leftarrow nil$ 
6: senão
7:   para cada  $x \in Adj(v)$  faça
8:     Se  $x.h \neq t$  então
9:        $altura(x)$ 
10:    fim Se
11:  fim para
12:  Escolher  $y$  tal que  $h(y) = \max\{h(w) \mid w \in Adj(v)\}$ 
13:   $h(v) \leftarrow h(v) + h(y)$ 
14:   $v.prox \leftarrow y$ 
15: fim Se

```

---

Encontrar uma orientação transitiva num grafo de comparabilidade pode

---

**Algoritmo 4** Clique máxima em grafos de comparabilidade

---

**Entrada:** Orientação transitiva de  $G$ **Saída:** Clique máxima  $K$ 

- 1: **para** cada  $v \in V$  **faça**
  - 2:    $altura(v)$
  - 3: **fim para**
  - 4: Escolher  $y$  tal que  $h(y) = \max\{h(v) | v \in V\}$
  - 5:  $K \leftarrow \{y\}$
  - 6:  $y \leftarrow y.prox$
  - 7: **enquanto**  $y \neq nil$  **faça**
  - 8:    $K \leftarrow K \cup \{y\}$
  - 9:    $y \leftarrow y.prox$
  - 10: **fim enquanto**
  - 11: Retorne  $K$
- 

ser resolvido em  $O(\Delta(G)m)$ [18]. No entanto, dada a representação em discos de um grafo  $k$ -faixa, podemos construir uma orientação para seu complemento em  $O(n + m)$  a partir das coordenadas dos centros dos discos (veja prova do Lema 3.7). O algoritmo *altura* faz uma busca em profundidade. Portanto, sua complexidade é  $O(n + m)$ .

**Algoritmo exato para grafos  $k$ -faixa,  $k$  qualquer**

O algoritmo a seguir foi proposto por Matsui em [43]. Para resolver o problema do conjunto independente na classe dos grafos  $k$ -faixa, Matsui constrói um grafo auxiliar orientado  $\vec{G}$  a partir de um grafo  $k$ -faixa  $G$ .

Dividiremos os vértices de  $G$  em  $r$  subconjuntos a partir de sua representação como segue. Chamaremos de  $D_i \subseteq D$ ,  $0 \leq i \leq r - 1$ , o conjunto de vértices representados por discos cujo centro possui abcissa maior que  $i$  e menor ou igual a  $i + 1$ . Denote por  $\min D_i = \min\{ \lfloor x \rfloor \mid (x, y) \in D_i \}$ .



Um *bloco independente* é um subconjunto  $B_i^j \subseteq D_i$  tal que cada ponto  $p = (x, y)$  em  $B_i^j$  satisfaz  $\lfloor x \rfloor = \min D_i$  e esses pontos formam um conjunto independente.  $F$  é a família de todos os blocos independentes de  $D$ . Para construir  $\vec{G}$ , associamos a cada  $B_i^j \in F$  um vértice em  $\vec{G}$  e inserimos mais dois vértices,  $s$  e  $t$ . O conjunto de arestas é definido por  $\{(s, B) \mid B \in F\} \cup \{(B, t) \mid B \in F\} \cup \{(B, B') \in F \times F \mid \min B < \min B' \text{ e } B \cup B' \text{ é um conjunto independente}\}$ . Ou seja, o grafo  $\vec{G}$  é um grafo orientado com  $|F| + 2$  vértices, um vértice correspondente a cada  $B \in F$ , uma fonte ( $s$ ) e um sumidouro ( $t$ ). Existem arcos da fonte a todos os vértices  $B$  e de todos os vértices  $B$  ao sumidouro. Entre os vértices  $B$  só existe aresta se a interseção entre os respectivos conjuntos for vazia. Associa-se a cada vértice de  $\vec{G}$  um peso: a fonte e o sumidouro têm peso 0 e os demais vértices têm peso igual ao tamanho do bloco independente que ele representa. Portanto, a soma dos pesos dos vértices num caminho  $s - t$  é igual ao tamanho do conjunto independente correspondente em  $G$ . Logo, encontrar o conjunto independente máximo de  $G$  se reduz a encontrar o maior caminho de  $s$  a  $t$  em  $\vec{G}$ . Um exemplo do algoritmo de Matsui está nas Figuras 3.6 e 3.7.

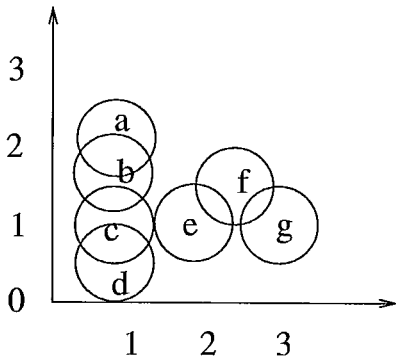
---

**Algoritmo 5**  $k$ -faixa

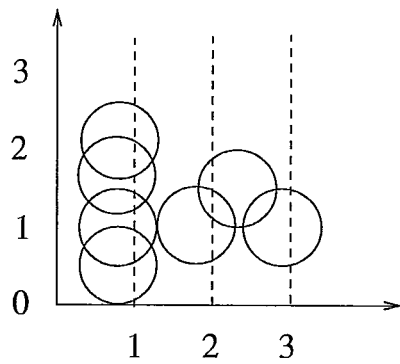
---

**Entrada:** Representação de  $G$  no plano

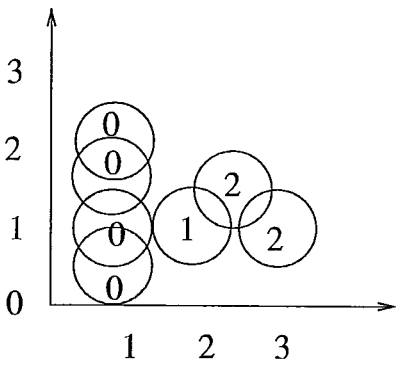
- 1: Gerar a família  $F$  de blocos independentes
  - 2: Construir  $\vec{G}$  como segue: para cada  $B_i$  em  $F$  inserir um vértice correspondente; inserir mais dois vértices,  $s$  e  $t$ ; atribuir à fonte e ao sumidouro peso 0 e aos demais vértices peso igual ao tamanho do bloco independente que ele representa; inserir arestas  $\{(s, B) \mid B \in F\} \cup \{(B, t) \mid B \in F\} \cup \{(B, B') \in F \times F \mid \min B < \min B' \text{ e } B \cup B' \text{ é um conjunto independente}\}$ .
  - 3: Encontrar o caminho máximo de  $s$  a  $t$  em  $\vec{G}$ .
-



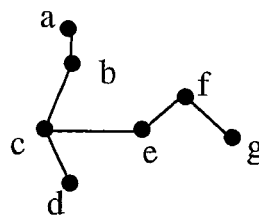
Representação de  $G$



Divisão dos discos pela abcissa

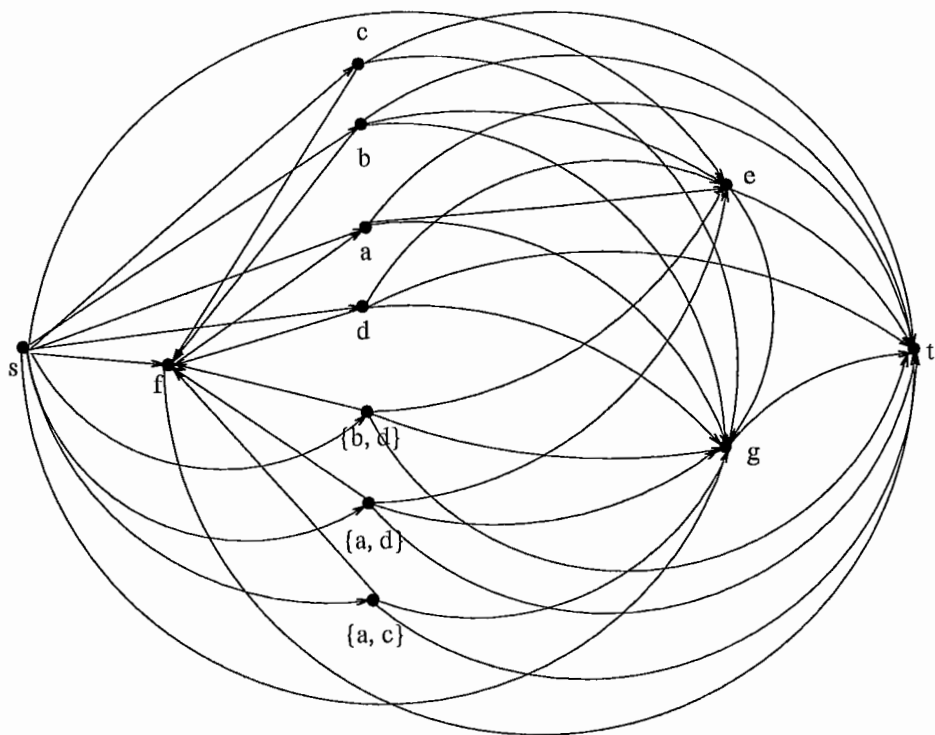


Famílias  $D_i$ ,  $0 \leq i \leq 2$

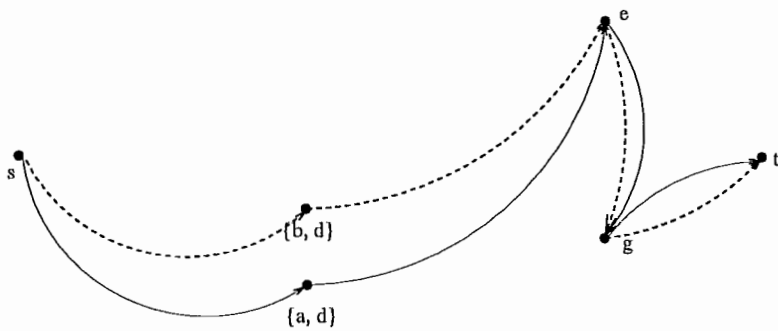


Grafo  $G$

Figura 3.6: Exemplo do algoritmo de Matsui para  $k = 3$ .



Grafo  $\vec{G}$



Caminhos máximos em  $\vec{G}$

Figura 3.7: Exemplo do algoritmo de Matsui para  $k = 3$ .

Visto que só há arestas entre blocos cuja união gera um conjunto independente, para cada caminho de  $s$  a  $t$  em  $\vec{G}$ , a união dos blocos independentes respectivos aos vértices internos do caminho é um conjunto independente de  $G$ . Da mesma maneira, a cada conjunto independente em  $G$  corresponde um caminho de  $s$  a  $t$  em  $\vec{G}$ . A geração dos blocos independentes pode ser feita em tempo  $O(n^3|F|)$  (passo 1) aplicando o algoritmo de enumeração para conjuntos independentes maximais descrito em [53]. Como o grafo auxiliar  $\vec{G}$  é acíclico e orientado, achar o caminho máximo pode ser resolvido em tempo linear em relação ao número de arcos, que é limitado por  $|F|^2$ . Portanto, a complexidade total do algoritmo é  $O(n^3|F| + |F|^2)$ . Seja  $\alpha$  o tamanho do maior bloco independente. Então,  $|F| \leq n^\alpha$ . Desconsiderando o caso trivial (quando  $\alpha \leq 2$ ), o tempo total do algoritmo passa a ser  $O(n^{2\alpha})$ . Na verdade, Matsui prova que  $\alpha \leq 2\lceil 2k/\sqrt{3} \rceil$ . Isto implica uma complexidade  $O(n^{4\lceil 2k/\sqrt{3} \rceil})$  para o algoritmo descrito.

### Aproximativo para DU

Matsui também descreve uma versão aproximativa de seu algoritmo para os grafos DU. O algoritmo tem razão de aproximação  $(1 - \frac{1}{r})$  e complexidade  $O(rn^{4\lceil 2(r-1)/\sqrt{3} \rceil})$ , onde  $r$  é o número de regiões em que a área da representação dada é dividida.

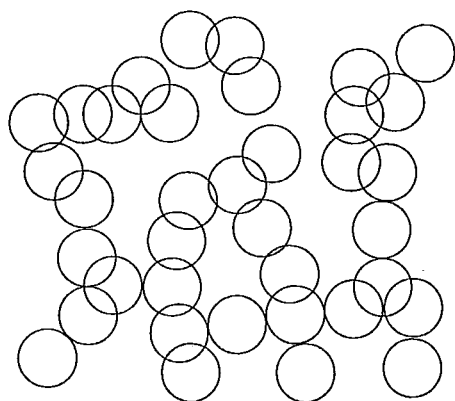
Assim como no algoritmo FMIS, os discos são divididos de acordo com as ordenadas de seu centro. Seja  $r$  o número de regiões em que dividire-

mos o plano. Denotaremos por  $T_s$ , onde  $s \in \{0, 1, \dots, r - 1\}$ , a região  $\{(x, y) \in \mathbb{R}^2 \mid s \leq (y \bmod r) < s + 1\}$ . Seja  $G(D_s)$ ,  $0 \leq s \leq r - 1$ , o grafo definido pelo conjunto de pontos  $D_s = D \setminus T_s$ . Decompomos  $D_s$  em  $i$  grafos  $r$ -faixa (veja exemplo nas Figuras 3.8 e 3.9) e aplicamos a cada componente o algoritmo para grafos  $k$ -faixa. O conjunto independente resultante será o maior conjunto independente encontrado para os  $D_s$ 's. Como para um conjunto independente máximo  $D^*$  vale  $D^* \setminus T_s \subseteq D \setminus T_s$  e  $\max\{|D^* \setminus T_s| \mid s \in \{0, 1, \dots, r - 1\}\} \geq (1 - 1/r)|D^*|$ , o tamanho do conjunto independente computado pelo algoritmo de Matsui é maior ou igual a  $(1 - 1/r)|D^*|$ . Usando o limite superior  $\alpha \leq 2\lceil 2k/\sqrt{3} \rceil$ , tem-se uma complexidade de  $O(rn^{4\lceil 2(r-1)/\sqrt{3} \rceil})$ .

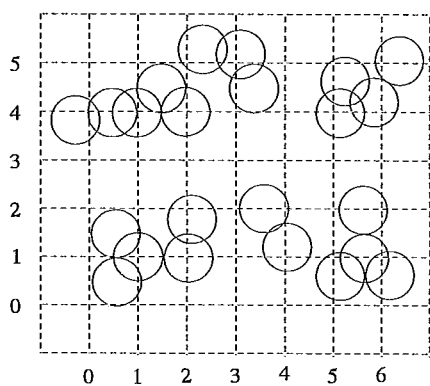
### 3.4 Outras classes

Vimos que nos grafos grade e faixa, resolver o problema do conjunto independente máximo ou a cobertura mínima de vértices pode ser resolvido em tempo polinomial, pois os grafos grade são bipartidos [8] e os grafos faixa são de cocomparabilidade. Um algoritmo  $O(n^{2,5})$  para grafos bipartidos é dado em [10], e um algoritmo  $O(n^2)$  para grafos de cocomparabilidade é descrito em [46].

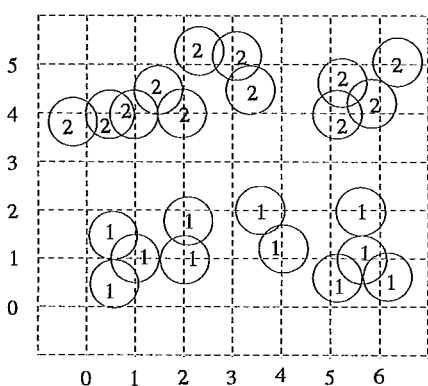
Uma generalização do algoritmo FMIS para os grafos ID é descrita em [11]. O algoritmo principal é o mesmo, mas como há discos de diferentes diâmetros, além da divisão em faixas, faz-se uma divisão em níveis (de acordo



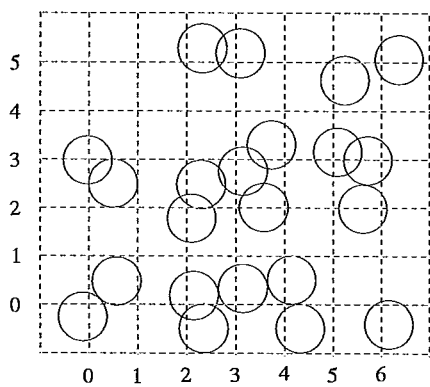
Representação do grafo  $G$



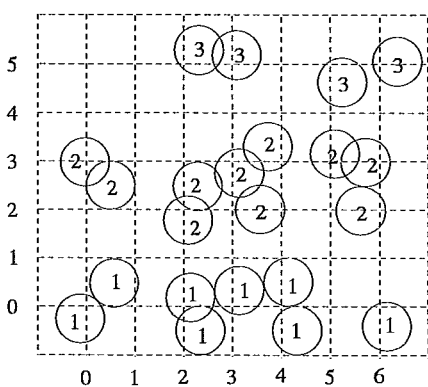
$D_0$



$D_0^1$  e  $D_0^2$



$D_1$



$D_1^1, D_1^2$  e  $D_1^3$

Figura 3.8: Exemplo do algoritmo aproximativo de Matsui para  $r = 3$ .

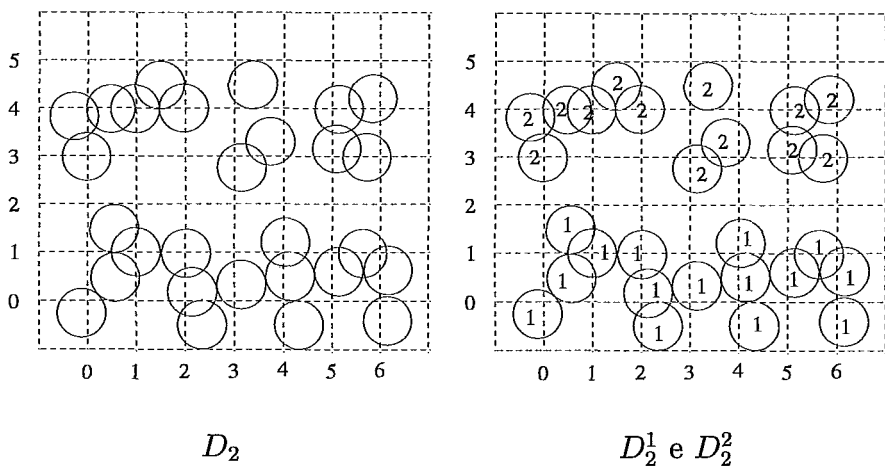


Figura 3.9: Exemplo do algoritmo aproximativo de Matsui para  $r = 3$ .

com os tamanhos dos diâmetros). A estratégia de divisão do plano adotada permite que a estratégia de deslocamento seja aplicada a diferentes níveis simultaneamente. A razão de aproximação da estratégia de deslocamento aplicada aos grafos ID em sua versão paralela é de  $(1 + \frac{1}{r-1})^2$  e a complexidade é de  $r^2 n^{O(r^4)}$ .

# Capítulo 4

## Cobertura por cliques

Neste capítulo abordamos o problema da cobertura por cliques em grafos DU, MU e faixa. Apesar do termo cobertura por cliques também ser aplicado a cobertura das arestas de um grafo, sempre que usarmos a expressão cobertura por cliques, estaremos nos referindo a cobertura dos vértices. Os resultados expostos neste capítulo englobam: uma prova de NP-dificuldade para o problema COBERTURA DE DISCOS (Seção 4.1), que é muito próximo ao problema COBERTURA POR CLIQUES em grafos DU; e uma prova de NP-dificuldade para o problema COBERTURA POR CLIQUES nos grafos MU (Seção 4.2), que implica a NP-dificuldade do mesmo problema para grafos DU. Além disso, são apresentados três algoritmos polinomiais (Seção 4.3). Um algoritmo exato para grafos faixa, um aproximativo para grafos DU e um aproximativo para os grafos MU. Por fim listamos resultados anteriores relativos a outras classes (Seção 4.4).

A seguir enunciamos os principais problemas citados neste capítulo.



**Problema:** COBERTURA POR CLIQUES - CC

*Instância:* Grafo  $G = (V, E)$ , um inteiro positivo  $K \leq |V|$ .

*Questão:* Pode-se particionar os vértices de  $G$  em  $k \leq K$  conjuntos disjuntos  $V_1, V_2, \dots, V_k$  tais que, para  $1 \leq i \leq k$ , o subgrafo induzido por  $V_i$  é completo?

**Problema:** 3-SAT

*Instância:* Conjunto  $U = \{u_1, u_2, \dots, u_q\}$  de variáveis e coleção  $E = E_1 \wedge E_2 \wedge \dots \wedge E_m$  de cláusulas tal que  $E_j = x_j \vee y_j \vee z_j$  para cada  $j = 1, 2, \dots, m$ , onde  $\{x_j, y_j, z_j\} \subseteq \{u_1, \bar{u}_1, u_2, \bar{u}_2, \dots, u_q, \bar{u}_q\}$ .

*Questão:* Existe um conjunto  $S \subseteq \{u_1, \bar{u}_1, u_2, \bar{u}_2, \dots, u_q, \bar{u}_q\}$  tal que valem  $S \cap \{x_j, y_j, z_j\} \neq \emptyset$  para cada  $j = 1, 2, \dots, m$  e  $|S \cap \{u_i, \bar{u}_i\}| = 1$  para cada  $i = 1, 2, \dots, q$ ?

**Problema:** COBERTURA DE DISCOS

*Instância:* Uma família  $D$  de discos de mesmo tamanho no plano e um inteiro  $p > 0$ .

*Questão:* Existem  $p$  pontos tais que cada disco intercepta ou contém pelo menos um destes pontos?

## 4.1 O problema COBERTURA DE DISCOS

Megiddo e Supowit, em [45], provaram a NP-dificuldade do problema COBERTURA DE DISCOS usando uma redução a partir de 3-SAT [16].

**Teorema 4.1** COBERTURA DE DISCOS é NP-difícil.

**Prova:** A redução é a partir de 3-SAT. Dada uma instância de 3-SAT, construiremos uma família de discos distintos, estabelecida a partir de três estruturas: circuitos, cláusulas e junções (Fig. 4.1).

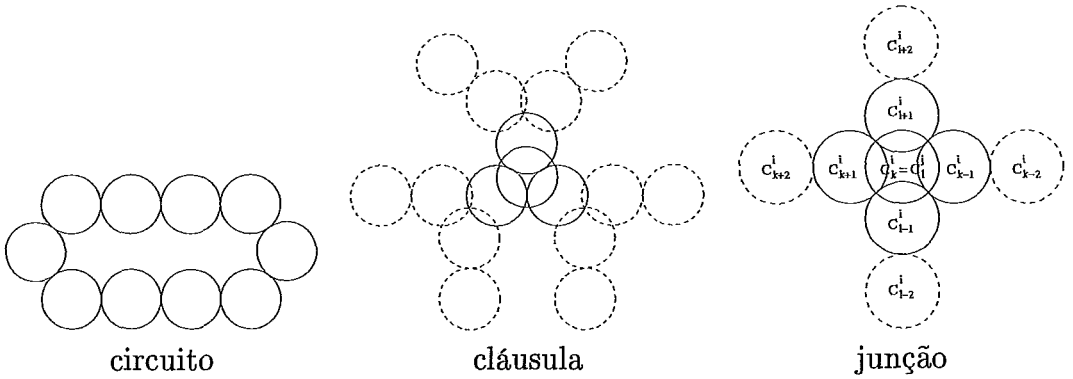
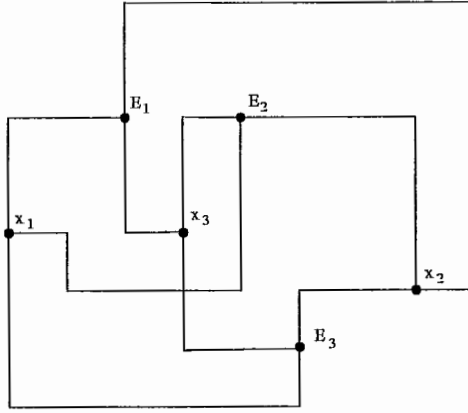


Figura 4.1: Estruturas da redução

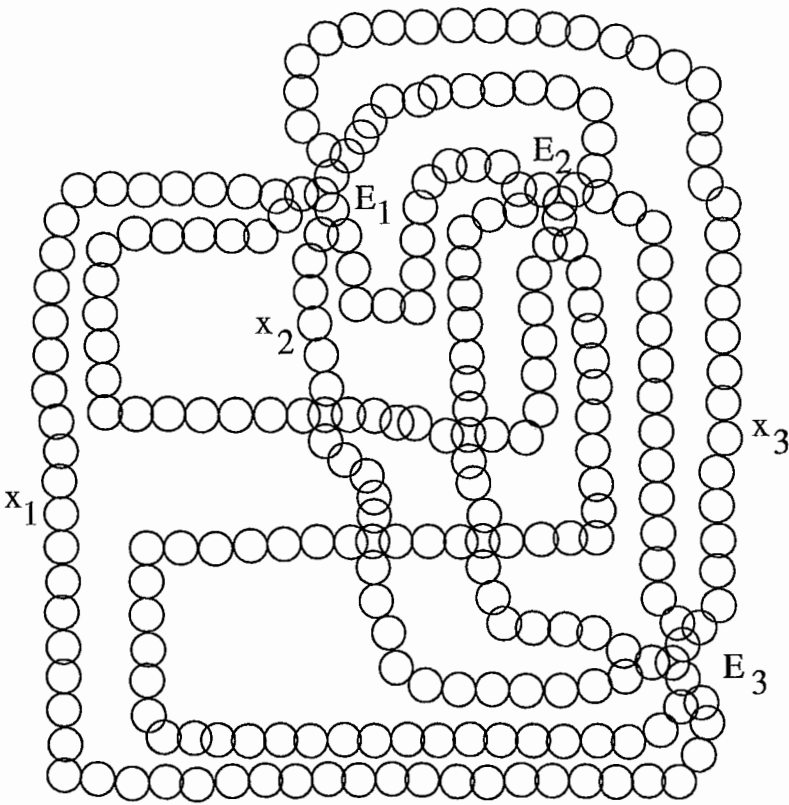
Cada variável  $u_i$  será representada por um *circuito*  $C^i$  de discos tal que  $C^i = \{C_0^i, C_1^i, \dots, C_{r_i}^i\}$ , onde  $C_0^i = C_{r_i}^i$ ,  $C_k^i \cap C_l^i \neq \emptyset$  se  $|k - l| \leq 1 \pmod{r_i}$ ,  $r_i$  par. Ou seja, um circuito é um ciclo de discos de tamanho par onde cada disco é vizinho de exatamente dois outros, os discos anterior e posterior a ele no circuito. As *cláusulas* determinam como os circuitos se relacionam e são formadas por discos distintos dos circuitos. A maneira como esses circuitos

serão dispostos no plano pode acarretar cruzamentos entre eles. Para evitar que esses cruzamentos alterem as propriedades dos circuitos, são definidas as *junções*, que são estruturas formadas por discos dos próprios circuitos.

Cada cláusula será representada por quatro discos de modo que um disco é central e intercepta a interseção de cada dois outros. No entanto, a interseção entre os quatro discos é vazia. Seja  $D^j$  o disco central e  $D_x^j$ ,  $D_y^j$  e  $D_z^j$  os demais, correspondentes aos literais  $x_j$ ,  $y_j$  e  $z_j$ , cada um interceptando exatamente dois discos do circuito correspondente. Uma junção é o cruzamento de dois circuitos distintos. Para construir a junção dos circuitos  $C^i$  e  $C^j$  são escolhidos dois discos  $C_k^i$  e  $C_l^j$  que se interceptem. Estes discos passam a ser idênticos. Além disso, forçamos  $k$  e  $l$  ímpares, garantindo que segmentos de circuitos entre junções consecutivas tenham tamanho ímpar. Um segmento de circuito é um conjunto maximal de discos da forma  $\{C_i^{k+2}, C_i^{k+3}, \dots, C_i^{l-2}\}$ , onde  $k$  e  $l$  são ímpares e para cada  $s$ ,  $k + 2 \leq s \leq l - 2$ ,  $C_s^i$  não está envolvido em nenhuma junção. O disco central da junção ( $C_k^i = C_l^j$ ) intercepta os seguintes conjuntos não vazios:  $C_{k-1}^i \cap C_{l-1}^j$ ,  $C_{k-1}^i \cap C_{k+1}^j$ ,  $C_{k+1}^i \cap C_{l-1}^j$  e  $C_{k+1}^i \cap C_{l+1}^j$ . Note que essas especificações não alteram as propriedades dos circuitos, pois a construção da junção garante que  $C_{k-1}^i \cap C_{k+1}^i = C_{l-1}^j \cap C_{l+1}^j = \emptyset$ . Construimos um exemplo da construção de uma representação de um grafo DU a partir de uma instância de 3-SAT, veja Figura 4.2



Grafo  $G_E$



Representação DU

Figura 4.2: Redução a partir da instância  $E = \{E_1 \wedge E_2 \wedge E_3\}$ . Onde  $E_1 = (x_1 \vee \bar{x}_3 \vee \bar{x}_2)$ ,  $E_2 = (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$  e  $E_3 = (x_1 \vee x_2 \vee x_3)$ .

Seja  $p = \sum_{i=1}^q \frac{r_i}{2} + m - J$ , onde  $J$  é o número de junções e  $m$  o número de cláusulas. A construção da instância para COBERTURA DE DISCOS está encerrada, e toma tempo polinomial.

Os pontos para cobrir cada estrutura serão escolhidos como segue. Para cobrir um circuito  $C^i$  são necessários  $\frac{r_i}{2}$  pontos. Seja  $Z$  uma cobertura mínima de  $C_i$ , existem duas maneiras de obter  $Z$ : ou todos os pontos de  $Z$  pertencem a  $C_k^i \cap C_{k+1}^i$ , para  $k$  par ou pertencem a  $C_k^i \cap C_{k+1}^i$ , para  $k$  ímpar. Nas cláusulas, dois pontos são necessários e suficientes para cobrir toda a estrutura. Um para cobrir o disco central e mais dois outros e um para cobrir o disco restante. Na Figura 4.3, por exemplo,  $D_x^j$  intercepta os círculos  $C_k^i$  e  $C_{k+1}^i$ , onde  $i$  é tal que  $x_j \in \{u_i, \bar{u}_i\}$  e  $D_x^j \cap C_k^i \cap C_{k+1}^i \neq \emptyset$ . Ou seja, existe pelo menos um ponto que intercepta  $D_x^j$  e mais dois discos consecutivos no circuito correspondente a variável  $x$ . As junções fazem parte dos circuitos e obedecem a cobertura dos mesmos.

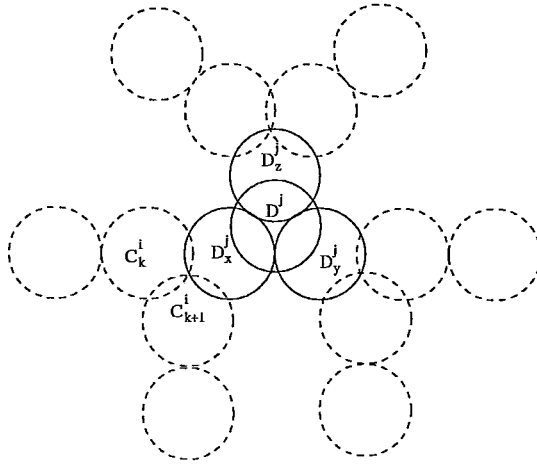


Figura 4.3: Cláusula.

Agora que já sabemos como cobrir cada estrutura, vamos provar que  $E$  é satisfatível se e somente se existe um conjunto  $Z$  com  $p$  pontos que cobre a família de discos construída a partir de  $E$ . Para isso, associamos a atribuição de valor verdadeiro ou falso as variáveis de  $E$  a cobertura de  $G$ . Nos circuitos, a escolha dos pontos em  $C_k^i \cap C_{k+1}^i$  correspondem a atribuição de valor verdadeiro ( $k$  par) ou falso ( $k$  ímpar) a  $u_i$ . Os pontos de  $Z$  serão chamados *pontos verdadeiros* ou *pontos falsos*, respectivamente; nas cláusulas, se  $x_j = u_i$ , então  $k$  é par, senão  $k$  é ímpar. Portanto, se alguma atribuição de valores implicar na escolha de um ponto em  $C_k^i \cap C_{k+1}^i$ , esse ponto também deve pertencer a  $D_x^j$ . O mesmo vale para  $D_y^j$  e  $D_z^j$ . Ou seja, se a cláusula  $E_j$  é satisfeita, pelo menos um dos discos  $D_x^j, D_y^j, D_z^j$  é coberto por um ponto verdadeiro ou falso. Como  $D^j$  não pode ser coberto por esse ponto, é necessário exatamente mais um ponto por cláusula satisfeita para garantir a cobertura. Nas junções, quaisquer que sejam os valores verdade de  $u_i$  e  $u_j$ , um ponto a menos será utilizado na cobertura dos circuitos. A Figura 4.4 deixa mais clara essa situação.

Assumindo que  $E$  é satisfeita pela atribuição  $\tau$ , construímos  $Z$  como segue. Para cada variável  $u_i$  se  $\tau(u_i) = V$ , incluímos em  $Z$  os pontos verdadeiros do circuito  $C^i$ . Se  $\tau(u_i) = F$ , incluímos os pontos falsos. Como a cada junção economizamos um ponto,  $Z$  possui  $\sum_{i=1}^q \frac{r_i}{2} - J$  pontos. Como todas as cláusulas são satisfeitas, incluímos mais um ponto por cláusula para cobrir o disco central e os que eventualmente não foram cobertos. Portanto,  $Z$  possui  $p$  pontos.

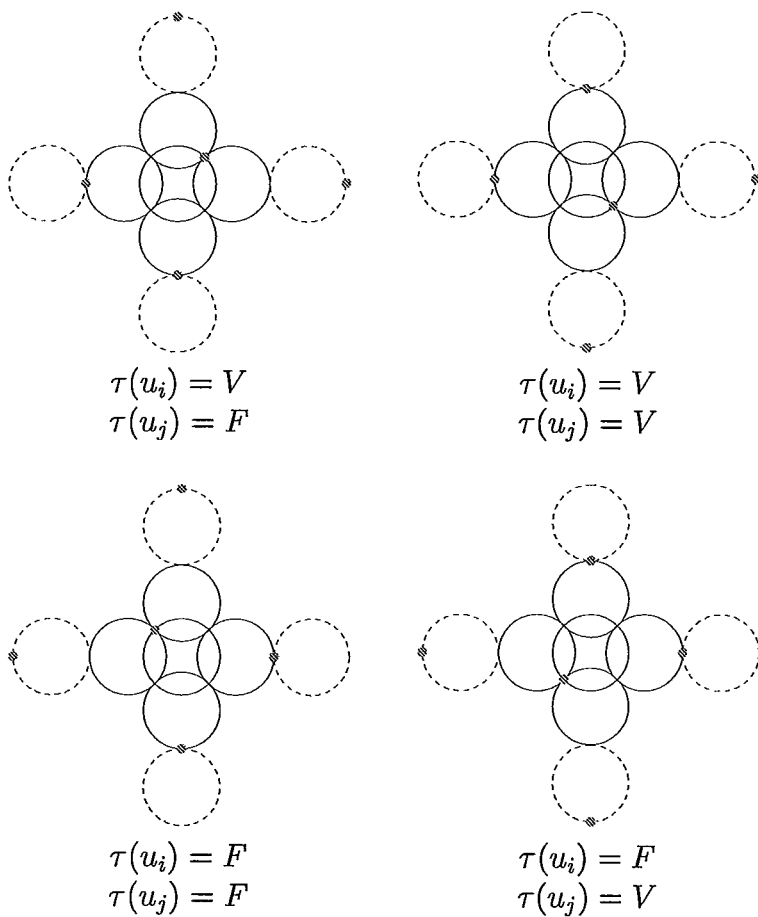


Figura 4.4: Cobertura de circuitos com cruzamentos.

Para provar a recíproca, construiremos uma atribuição  $\tau$  que satisfaz  $E$ , a partir de uma cobertura  $Z$  com  $p$  pontos. Começamos pelos segmentos. Sabemos que os segmentos de circuito entre duas junções consecutivas têm tamanho ímpar\* e são dois a dois disjuntos. Além disso, existem  $2J$  desses segmentos, pois cada junção intercepta quatro segmentos e cada segmento intercepta duas junções. Logo, o tamanho total do conjunto de segmentos é

$$\sum_i^q r_i - 6J,$$

que é o número total de discos de cada circuito ( $\sum r_i$ ) menos o número de junções ( $J$ ) menos os círculos da junção ( $5J$ ). Como existem  $2J$  segmentos e todos tem tamanho ímpar, são necessários

$$\frac{1}{2}(\sum r_i - 6J - 2J) + 2J = \sum \frac{r_i}{2} - 2J$$

pontos para cobrir todos eles. É fácil verificar que para que essa cobertura seja mínima, os pontos escolhidos em cada segmento devem ser todos o mesmo tipo (verdadeiro ou falso). Restam então  $m + J$  pontos para cobrir as junções e as cláusulas. Como os discos centrais dessas estruturas não podem ser cobertos pelos pontos que cobrem os círculos de segmento, precisamos exatamente de um ponto a mais para cada junção e um a mais para cada cláusula. Nas junções, o ponto que cobre o disco central intercepta no máximo mais dois outros. Portanto, os dois discos restantes precisam ser cobertos por pontos que cobrem os segmentos. Como esses pontos interceptam no máximo um círculo da junção, serão necessários mais dois pontos de

---

\*Ao construir a junção, garantimos que  $C_k^i = C_l^j$ ,  $k$  e  $l$  ímpares.



circuitos distintos. Isso implica que em cada junção, segmentos de um mesmo circuito terão o mesmo valor (veja Figura 4.4). A construção das cláusulas também garante  $\tau$ , pois o ponto que cobre o disco central da cláusula cobre no máximo mais dois dos demais. Ou seja, pelo menos um dos discos de cada cláusula é coberto por um ponto da cobertura dos segmentos.

□

## 4.2 Cobertura por cliques para grafos de Moeda Unitária

Para provar a NP-completude desse problema, usamos uma redução a partir de uma variante do problema 3-SAT, o 3-PSAT $_{\bar{3}}$  [16]:

**Problema:** 3-PSAT $_{\bar{3}}$

*Instância:* Conjunto  $U = \{u_1, u_2, \dots, u_q\}$  de variáveis e coleção

$E = E_1 \wedge E_2 \wedge \dots \wedge E_m$  de cláusulas tal que  $E_j = x_j \vee y_j \vee z_j$  para cada  $j = 1, 2, \dots, m$ , onde  $\{x_j, y_j, z_j\} \subseteq \{u_1, \bar{u}_1, u_2, \bar{u}_2, \dots, u_q, \bar{u}_q\}$  e cada cláusula contém no máximo três literais, cada variável ocorre no máximo 3 vezes e o grafo não direcionado  $G_E = (N, M)$  é planar, onde  $N = (U \cup E)$  e  $M = \{u_i E_j \mid u_i \in E_j \text{ ou } \bar{u}_i \in E_j\}$ .

*Questão:*  $E$  é satisfatível?

**Teorema 4.2** *CC para grafos MU é NP-difícil.*

**Prova:** A partir de uma instância  $E$  de um  $3\text{-SAT}_{\bar{3}}$  planar, mostraremos como construir a representação de um grafo de moeda unitária  $G'$  a partir de  $G_E$  através de duas estruturas: *circuito* e *cláusula* (Fig. 4.5).

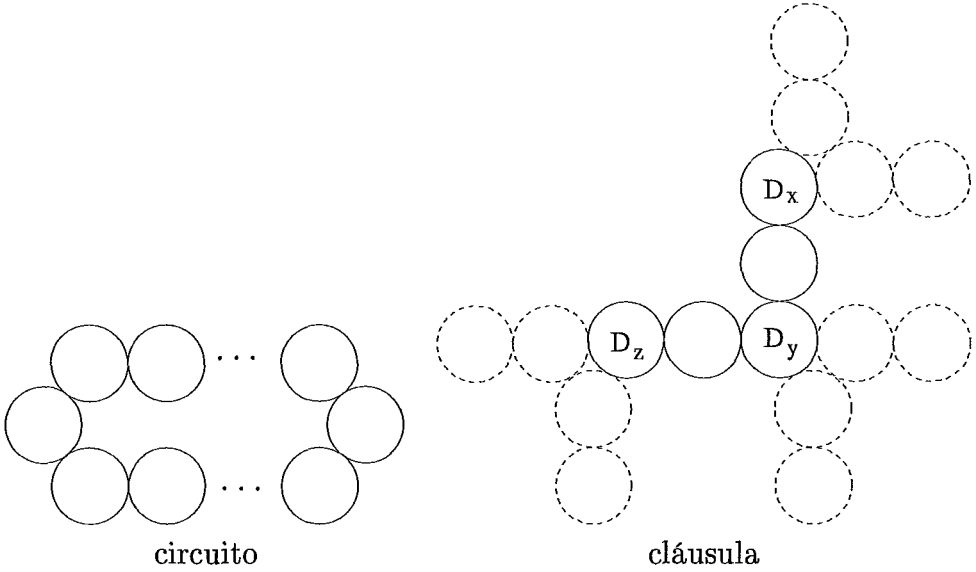


Figura 4.5: Estruturas da redução.

Cada variável  $u_i$  será representada por um *circuito*  $C^i$  de discos tal que  $C^i = \{C_0^i, C_1^i, \dots, C_{r_i}^i\}$ , onde  $C_0^i = C_{r_i}^i$ ,  $C_k^i \cap C_l^i \neq \emptyset$  se  $|k - l| \leq 1 \pmod{r_i}$ ,  $r_i$  par e suficientemente grande. As *cláusulas* são formadas por três discos distintos dos circuitos. Três deles,  $D_x$ ,  $D_y$  e  $D_z$ , correspondentes aos literais  $x$ ,  $y$  e  $z$ , cada um interceptando exatamente dois discos do circuito correspondente.

Sabemos que  $G_E$  é planar e possui grau máximo 3, usaremos então o Lema 3.4 para desenhar  $G_E$  no plano. Para garantir que não haverá sobreposição de discos em  $G'$ , construímos uma representação de  $G'$  a partir de  $G_E$  como segue. Quando substituirmos um vértice cláusula  $E_j$  pela estrutura cláusula, posicionamos o disco  $D_y$  de maneira que seu centro tenha coordenadas iguais às de  $E_j$ . Os demais discos da cláusula devem ser alinhados como mostra a Figura 4.6. Note que nos demais discos uma das coordenadas coincide com a coordenada correspondente do disco  $D_y$ . Os vértices-variáveis e as arestas são substituídos pelos circuitos de discos que formam a estrutura variável. Como cada aresta é formada por segmentos de reta, o circuito será desenhado de maneira a circundar os segmentos (Fig. 4.7). Como os circuitos devem ter tamanho par, sempre que for necessário, pode-se fazer a substituição mostrada na Figura 3.2. Para garantir que discos de circuitos distintos não se interceptem, desenhamos os discos com diâmetro suficientemente pequeno. Um exemplo completo de como desenhar o grafo  $G'$  a partir de  $G_E$  é dado nas Figuras 4.8 e 4.2.

As cliques necessárias à cobertura do grafo  $G'$  serão representadas por pontos desenhados na estrutura. Observe que cada “ponto” aqui empregado não guarda relação com os pontos utilizados na seção anterior para o problema COBERTURA DE DISCOS. Aqui, eles são apenas um artifício para simplificar o desenho das figuras. As cliques de tamanho três serão representadas por um ponto entre os três discos e as cliques de tamanho dois por um ponto comum aos dois discos.

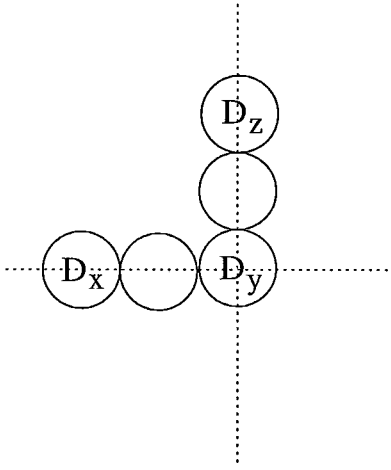


Figura 4.6: Cláusula.

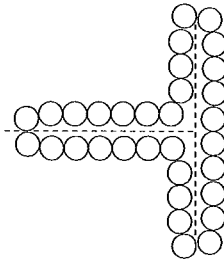


Figura 4.7: Circuito.

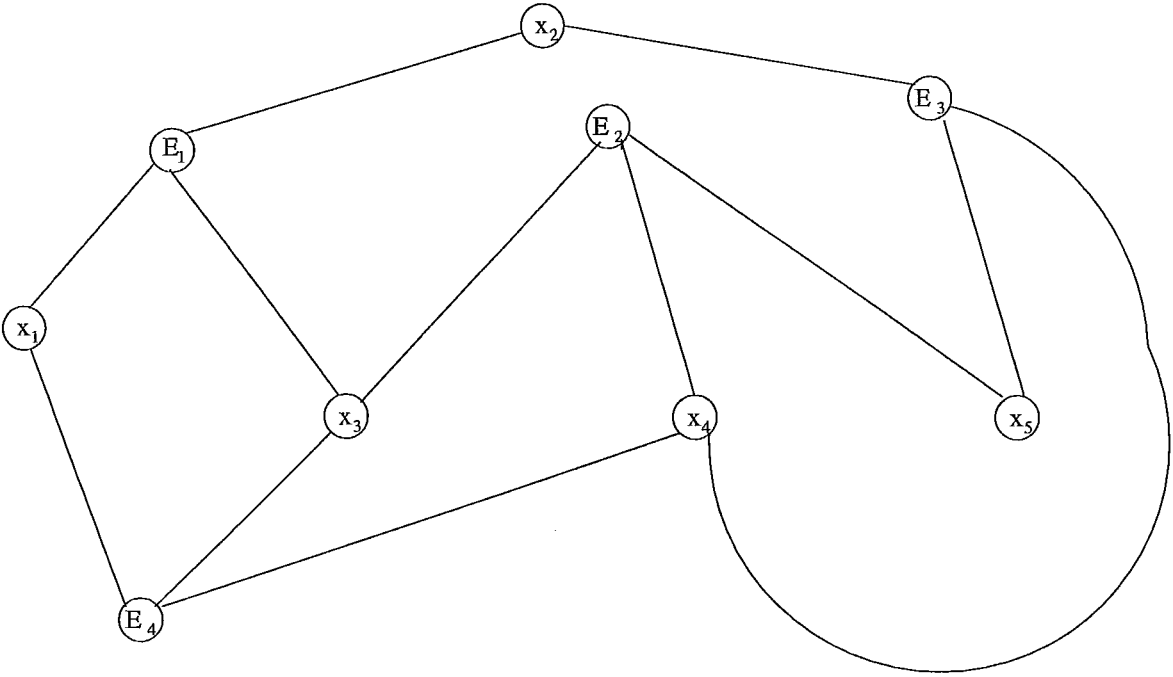
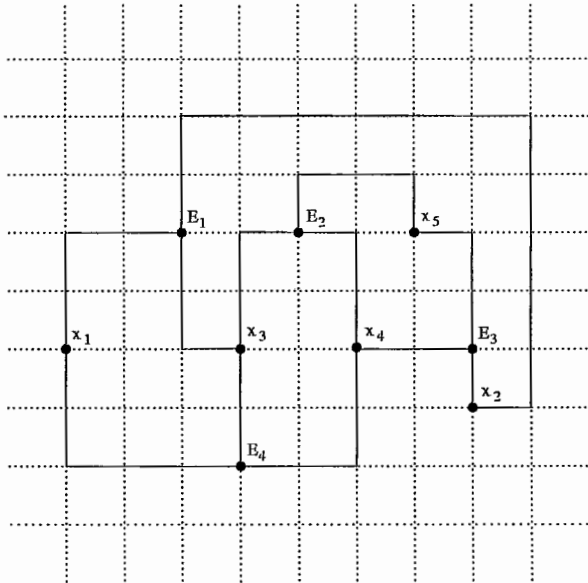
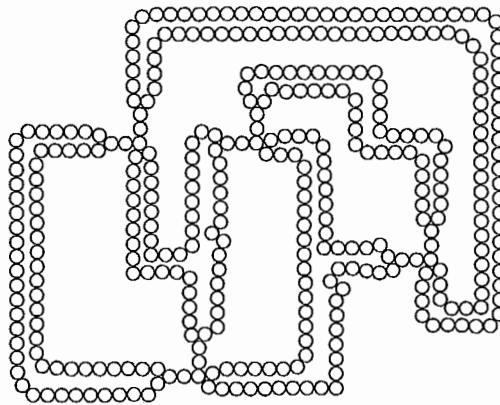


Figura 4.8: Grafo  $G_E$  gerado a partir da instância  $E = \{E_1 \wedge E_2 \wedge E_3 \wedge E_4\}$ . Onde  $E_1 = (x_1 \vee \bar{x}_2 \vee \bar{x}_3)$ ,  $E_2 = (x_3 \vee \bar{x}_4 \vee \bar{x}_5)$ ,  $E_3 = (x_4 \vee \bar{x}_5 \vee x_2)$  e  $E_4 = (\bar{x}_1 \vee x_3 \vee \bar{x}_4)$ .



Transposição no plano do grafo  $G_E$



Representação do grafo  $G'$

Figura 4.9: Construção de  $G'$ .

Seja  $p = \sum_{i=1}^q \frac{r_i}{2} + 2m$ , onde  $m$  é o número de cláusulas e  $G'$  o grafo construído como especificamos. As cliques para cobrir cada estrutura serão escolhidas como segue. Para cobrir um circuito  $C^i$  são necessárias  $\frac{r_i}{2}$  cliques. Nos circuitos, as cliques escolhidas são arestas de extremidades  $C_k^i$  e  $C_{k+1}^i$ , para  $k$  par ou  $C_k^i$  e  $C_{k+1}^i$ , para  $k$  ímpar (veja Fig. 4.10). Nas cláusulas, três cliques são necessárias e suficientes para cobrir toda a estrutura (Figs. 4.11 e 4.12).

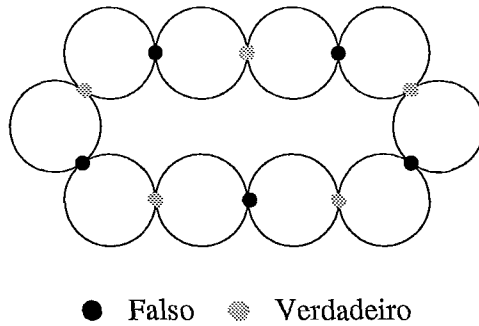


Figura 4.10: Cobertura de circuito: representação dos circuitos de  $G'$ . Cliques representadas por pontos.

Agora vamos provar que  $E$  é satisfatível se e somente se existe um conjunto  $Z$  com  $p$  cliques que cobre a estrutura. Associamos uma atribuição de  $E$  a uma cobertura de  $G'$  como segue. Nos circuitos, a escolha dos pontos em  $C_k^i \cap C_{k+1}^i$  corresponde a atribuição verdadeiro ou falso a  $u_i$ . Nas cláusulas, se  $x_j = u_i$ , então  $k$  é par, senão  $k$  é ímpar. Os **pontos verdadeiros** são os que cobrem os discos na ordem par-ímpar, ou seja, o primeiro disco é de número par e o segundo de número ímpar. Estes pontos representam o literal não negado com valor verdadeiro. Os **pontos falsos** são os que cobrem os discos na ordem ímpar-par e representam o literal negado com valor verdadeiro.

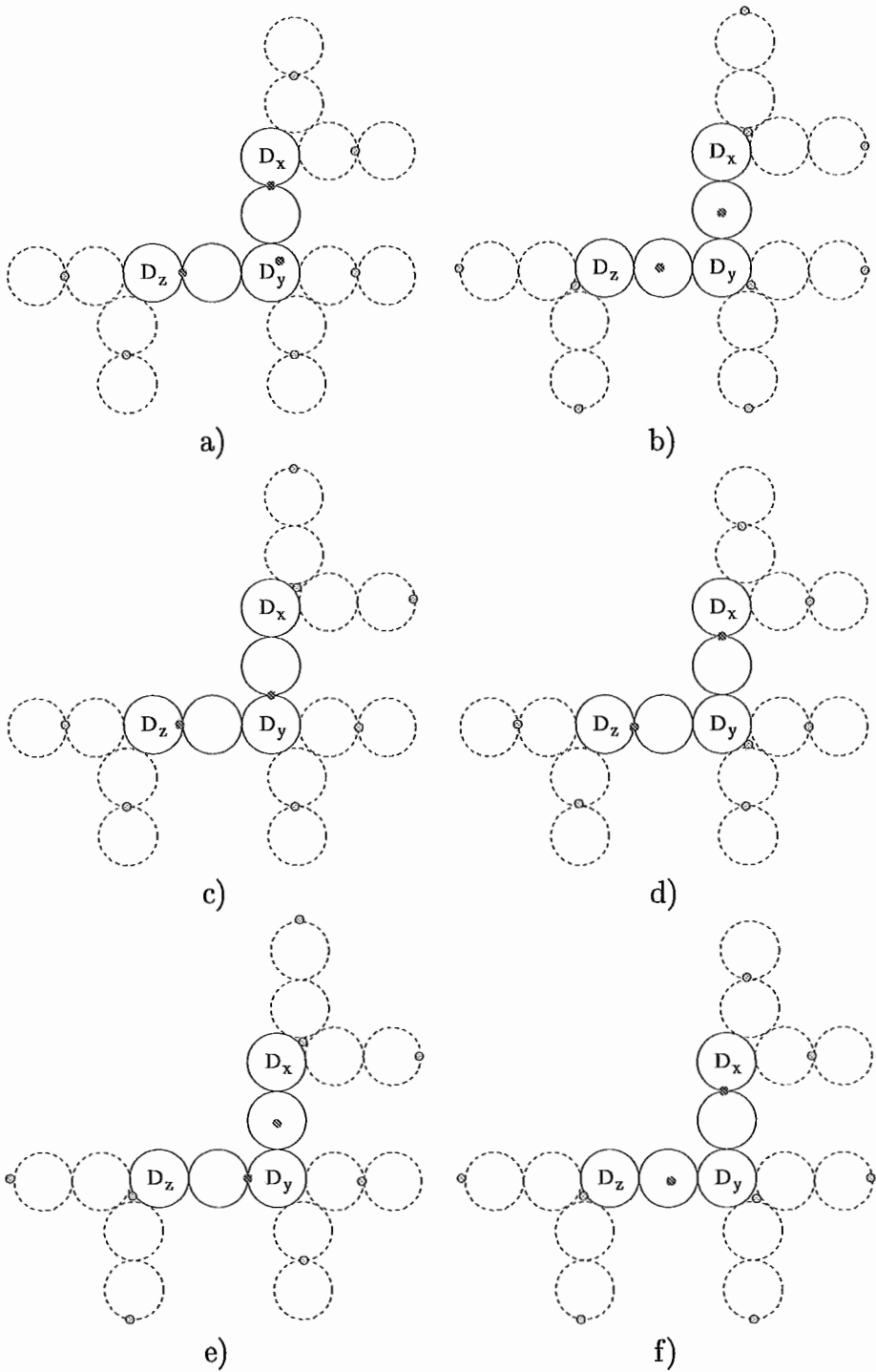


Figura 4.11: Cobertura de uma cláusula: a)  $E_j$  não é satisfeita; b)  $E_j$  satisfeita por  $x$ ,  $y$  e  $z$ ; c)  $E_j$  satisfeita por  $x$ ; d)  $E_j$  satisfeita por  $y$ ; e)  $E_j$  satisfeita por  $x$  e  $z$ ; f)  $E_j$  satisfeita por  $y$  e  $z$ . Os demais casos,  $E_j$  satisfeita por  $z$  e  $E_j$  satisfeita por  $x$  e  $y$ , são simétricos a  $E_j$  satisfeita por  $x$  e  $E_j$  satisfeita por  $y$  e  $z$ . Os pontos pretos representam as cliques que cobrem somente vértices da cláusulas e os pontos cinzas cliques que cobrem vértices de circuito.



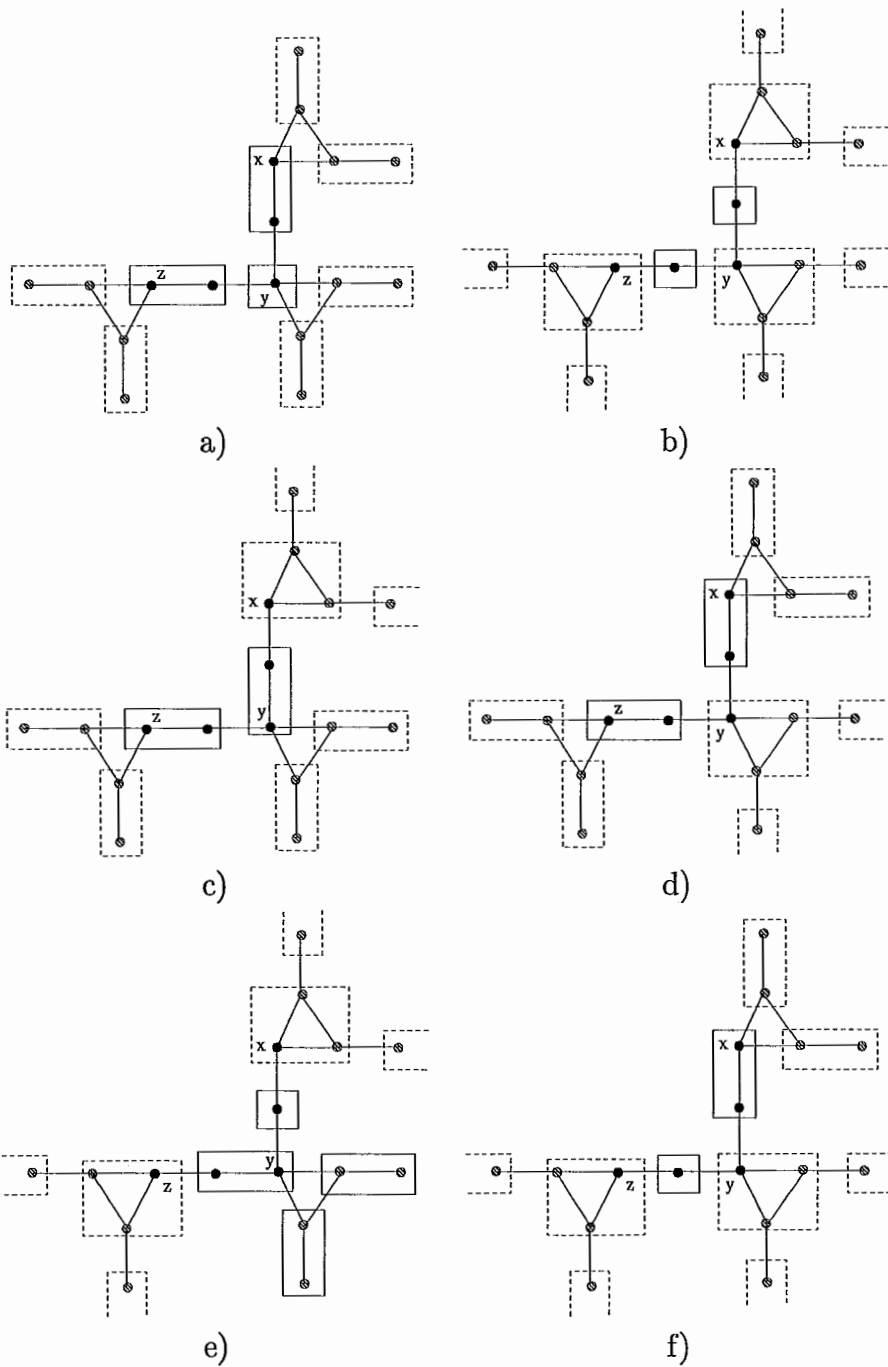


Figura 4.12: Cobertura de uma cláusula: a)  $E_j$  não é satisfeita; b)  $E_j$  satisfeita por  $x, y$  e  $z$ ; c)  $E_j$  satisfeita por  $x$ ; d)  $E_j$  satisfeita por  $y$ ; e)  $E_j$  satisfeita por  $x$  e  $z$ ; f)  $E_j$  satisfeita por  $y$  e  $z$ . Os demais casos,  $E_j$  satisfeita por  $z$  e  $E_j$  satisfeita por  $x$  e  $y$ , são simétricos a  $E_j$  satisfeita por  $x$  e  $E_j$  satisfeita por  $y$  e  $z$ . As linhas cheias delimitam as cliques que cobrem somente vértices de cláusula e as linhas tracejadas delimitam cliques que cobrem vértices de circuito.

Se alguma atribuição verdade implicar na escolha de uma aresta que tenha como extremidade os vértices correspondentes aos dois discos vizinhos a um dos discos que representam literais,  $D_x$  por exemplo, então a clique escolhida será o triângulo formado pelos vértices correspondentes aos três discos (veja Figs. 4.11 a 4.13). No entanto, se as cliques que cobrem  $C_k^i$  e  $C_{k+1}^i$  são distintas, então o vértice que corresponde ao disco  $D_x$  deve ser coberto por uma clique que não pertence à cobertura dos vértices de circuito.

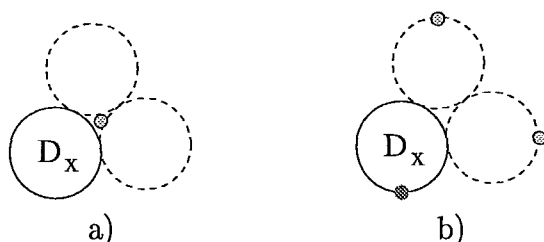


Figura 4.13: Cobertura do disco  $D_x$ : a)  $\tau(u_i) = V$  b)  $\tau(u_i) = F$

Seja  $\tau$  uma atribuição que satisfaz  $E$ . Construímos  $Z$  a partir de  $\tau$  como segue. Para cada variável  $u_i$ , se  $\tau(u_i) = V$  incluímos em  $Z$  as cliques correspondentes aos pontos verdadeiros do circuito  $C^i$ . Se  $\tau(u_i) = F$ , incluímos as cliques correspondentes aos pontos falsos. Como cada cláusula é satisfeita por pelo menos um literal e os vértices correspondentes aos literais de valor verdadeiro são cobertos por cliques pertencentes ao conjunto de cliques que cobrem o circuito, então para cada cláusula são necessários somente duas cliques distintas das cliques que cobrem circuitos.

Para provar a recíproca, construiremos uma atribuição  $\tau$  que satisfaz  $E$  a partir de uma cobertura  $Z$  com  $p$  cliques. Para cobrir cada variável

com o mínimo de cliques necessárias, cada vértice deve pertencer somente a uma clique da cobertura. Ou seja, na representação, os pontos escolhidos devem ser do mesmo tipo (verdadeiro ou falso), pois o número de discos é par e qualquer outra cobertura envolvendo pontos de valores diferentes precisaria de um número maior de pontos. Usamos então  $\sum_{i=1}^q \frac{r_i}{2}$  cliques para cobrir as variáveis. Restam  $2m$  cliques para cobrir as cláusulas. Qualquer que seja a cobertura escolhida, são necessários pelo menos três cliques para cobrir cada cláusula. Observe no entanto que mesmo que todos os vértices correspondentes a discos  $D$  sejam cobertos por cliques que cobrem os vértices correspondentes aos discos de circuito, ainda serão necessários duas cliques para cobrir os vértices restantes (veja Fig. 4.11). Portanto, na cobertura de cada cláusula pelo menos uma clique pertence à cobertura dos circuitos. Logo, pelo menos um dos literais em cada cláusula tem valor verdade.

□

**Colorário 4.1** *CC para grafos DU é NP-difícil.*

## 4.3 Algoritmos

Os algoritmos descritos a seguir assumem que o grafo entrada  $G = (V, E)$  é um grafo DU e sua representação é conhecida.

### 4.3.1 Grafos faixa

No Capítulo 3 vimos que os grafos faixa pertencem à classe dos grafos de cocomparabilidade e que um grafo de cocomparabilidade é complemento de um grafo de comparabilidade. Sabemos ainda que ao resolvermos o problema da coloração de vértices num grafo de comparabilidade, resolvemos cobertura por cliques no seu complemento. Um algoritmo  $O(|V| + |E|)$  para o problema da coloração em grafos de comparabilidade é dado em [18, 17]. A partir desse algoritmo, apresentamos um algoritmo  $O(|E| + |\overline{E}|)$  para cobertura por cliques em grafos de cocomparabilidade, em especial, para grafos faixa.

---

**Algoritmo 6** cobertura por cliques para grafos faixa.

---

**Entrada:** Representação de  $G$

- 1: Seja  $\overline{G} = (V, \overline{E})$  o complemento de  $G$ . Construir uma orientação transitiva  $\pi$  de  $\overline{G}$  da seguinte maneira: para cada aresta  $e \in \overline{E}$  que conecta dois vértices  $a$  e  $b$  de  $V$  temos que  $x_a \neq x_b$ . Direcionar a aresta  $e$  de  $a$  a  $b$  se  $x_a < x_b$ .
  - 2: Atribuir uma “cor” a cada vértice  $v \in V$  recursivamente. Se  $v$  é um vértice sumidouro então  $c(v) = 0$ , senão  $c(v) = 1 + \max\{c(w) \mid v\vec{w} \in \overline{E}\}$ .
  - 3: Devolver a cobertura  $CC(G)$  formada pelos conjuntos de vértices de mesma cor. Cada conjunto de vértices de mesma cor em  $\overline{G}$  é uma clique da cobertura de  $G$ .
- 

O passo 1 pode ser executado em  $O(|E| + |\overline{E}|)$  e o passo 2 em  $O(|V| + |\overline{E}|)$  fazendo uma busca em profundidade em  $\overline{G}$ . Logo, a complexidade total algoritmo é  $O(|E| + |\overline{E}|)$ .

### 4.3.2 Algoritmo aproximativo para DU

Para a classe dos grafos DU, desenvolvemos um algoritmo aproximativo polinomial que usa como subrotina o algoritmo exato descrito na Seção 4.3.1.

---

**Algoritmo 7** CC para grafos DU.

---

**Entrada:** Representação de  $G$

- 1: Dividir o plano em faixas de altura  $\frac{\sqrt{3}}{2}$ ;
  - 2: Associar a cada faixa  $i$  o grafo  $G[V_i]$ , grafo induzido pelo conjunto  $V_i$  de vértices de  $G$  contidos na faixa  $i$ ;
  - 3: Cobrir cada  $G_i$  de maneira ótima usando o algoritmo exato da Seção 4.3.1;
  - 4: A cobertura de  $CC(G)$  será a união das coberturas  $CC(G_i)$ .
- 

O passo 1 divide o plano em  $l$  faixas, onde  $l = \lceil \frac{y}{\sqrt{3}/2} \rceil$  e  $y$  é a maior ordenada dentre todos os centros. Todas as faixas têm largura  $\frac{\sqrt{3}}{2}$ . Seja  $F = \{F_1, F_2, \dots, F_l\}$  o conjunto de faixas resultantes, cada faixa  $F_j$  é definida por dois valores,  $y_i$  e  $y_f$ . Pertencem a  $F_j$  os discos de centro  $c = (x, y)$ ,  $y_i \leq y \leq y_f$ . Para  $F_1$ ,  $y_i = 0$  e  $y_f = \frac{\sqrt{3}}{2}$ . Para as demais faixas, a maior ordenada de uma faixa será a menor da faixa subsequente. No passo 2, para cada faixa  $F_i$ , induzimos um grafo  $G_i$  a partir do conjunto de todos os discos em  $G$  com centro de ordenada  $y_i \leq y \leq y_f$ , onde  $y_i$  e  $y_f$  são o menor e o maior valor de ordenada na faixa  $F_i$ .

O passo 1 pode ser resolvido em  $O(n)$ , basta verificar a maior ordenada entre os centros dos discos de  $G$ . O passo 2 também é  $O(n)$ , pois também pode ser resolvido por verificação das coordenadas dos centros. Pela Seção 4.3.1, o passo 3 é  $O(|V| + |\overline{E}|)$ , pois para cada faixa  $F_i$  temos  $n_i$  vértices e  $m_i$  arestas, onde  $\sum_{1 \leq i \leq l} |n_i| = |V|$  e  $\sum_{1 \leq i \leq l} |m_i| \leq |E|$ . Logo, o algoritmo descrito tem

complexidade  $O(|V| + |\overline{E}|)$ .

Dada uma cobertura mínima qualquer, todos os “vértices” de uma clique estão contidos em no máximo três faixas. Seja  $C(G)$  uma cobertura mínima de  $G$ . Sabemos que cada faixa é coberta de maneira ótima. Ou seja, se os vértices no grafo  $G_i$  induzido a partir da faixa  $F_i$  são cobertos por  $k$  cliques em  $C(G)$ , então nosso algoritmo cobre  $G_i$  com no máximo  $k$  cliques. Como uma clique  $C$  qualquer de  $G$  pode pertencer a no máximo três faixas e algumas arestas não são consideradas no algoritmo, uma clique pode ser particionada e ocorrer em no máximo 3 coberturas de faixa. Logo, a razão de aproximação do algoritmo é três.

### 4.3.3 Algoritmo aproximativo para MU

O Lema 3.1 estabelecia uma restrição quanto à vizinhança dos discos de um grafo DU. Usando a mesma argumentação, é fácil ver que o número de vizinhos de um vértice num grafo de moeda unitária (MU) é no máximo 6. Semelhantemente, a partir do Lema 3.2, vemos que o número de vizinhos do vértice mais à esquerda num grafo MU é no máximo 3. A partir dessa observação, formulamos o algoritmo aproximativo a seguir.

O passo 1 pode ser resolvido em  $O(|V| \log |V|)$  e o passo 2 em  $O(|V| + |E|)$ . A complexidade total do algoritmo é  $O(|V| \log |V| + |E|)$ .

Seja  $p = (x, y)$  o ponto mais à esquerda numa representação de um grafo

---

**Algoritmo 8** CC para grafos MU.

---

**Entrada:** Representação de  $G$

- 1: Ordenar os pontos da representação pela abcissa (da menor à maior).  
Seja  $\pi$  a ordenação construída.
  - 2: **enquanto**  $\pi \neq \emptyset$  **faça**
  - 3:   Seja  $p$  o primeiro ponto na ordenação  $\pi$
  - 4:   Inserir na cobertura a maior clique induzida por  $p$  e sua vizinhança
  - 5:   Retirar de  $\pi$  os pontos já cobertos
  - 6: **fim enquanto**
- 

MU  $G$ . Sabemos que a maior clique num grafo MU tem tamanho 3 e que  $p$  tem no máximo 3 vizinhos. Supondo que a vizinhança de  $p$  tenha tamanho 3, são necessárias pelo menos duas cliques para cobrir  $p$  e sua vizinhança. Ao escolhermos a maior clique na vizinhança de  $p$ , estamos aumentando o número de cliques na cobertura em no máximo uma. Logo, a razão de aproximação do algoritmo é  $\frac{3}{2}$ .

## 4.4 Outras classes

Sabemos que nos grafos grade achar uma cobertura mínima por cliques pode ser resolvido em tempo polinomial, pois os grafos grade são bipartidos. Para resolver esse problema, basta achar um emparelhamento máximo no grafo e adicionar os vértices não cobertos à cobertura.

O problema da cobertura por cliques para os grafos de moeda (planares) é NP-completo [9], e portanto nos grafos ID também é NP-completo.

# Capítulo 5

## Conclusões

Vimos que vários problemas não triviais como conjunto independente, número cromático e cobertura por cliques continuam NP-completos mesmo quando restritos à classe dos grafos de disco unitário. No entanto, o problema da clique máxima num grafo DU pode ser resolvido em tempo polinomial dada a realização do grafo.

Neste trabalho investigamos os problemas do conjunto independente e da cobertura por cliques quando restritos aos grafos moeda e às seguintes subclasses dos grafos DU: grafos grade, grafos de moeda unitária e grafos faixa. Nos grafos grade e faixa, ambos podem ser resolvidos em tempo polinomial, pois essas classes são subclasses dos grafos bipartidos e de comparabilidade, respectivamente. Como vimos no Capítulo 2, o grafos de moeda e planares coincidem. Portanto, os problemas do conjunto independente e da cobertura por cliques são NP-completos nessa classe. A complexidade desses problemas para os grafos MU não era conhecida e foi estabelecida



nos Capítulos 3 e 4. Ambos são problemas NP-completos.

As principais contribuições desse trabalho foram o estabelecimento da complexidade dos problemas do conjunto independente, cobertura por cliques e cobertura por vértices nos grafos MU; dois algoritmos polinomiais aproximativos para o problema da cobertura por cliques, um algoritmo  $\frac{1}{3}$ -aproximativo para os grafos DU e um algoritmo  $\frac{3}{2}$ -aproximativo para os grafos MU.

Propomos como trabalhos futuros investigar os seguintes tópicos:

- Estudar a complexidade desses problemas em outras subclasses dos grafos DU.
- Estudar o problema do número cromático nessas classes e nos grafos ID.
- Aprofundar o estudo dos grafos DU quando hierarquicamente especificados e tentar desenvolver melhores algoritmos para essa especificação.

# Referências Bibliográficas

- [1] APPEL, K., HAKEN, W. “Every planar map is four-colorable”. *Bulletin of the American Mathematical Society*, v. 82, pp. 711–712, 1976.
- [2] BAKER, B.S. “Approximation algorithms for NP-complete problems on planar graphs”. *Journal of the ACM*, v. 41, pp. 153–180, 1994.
- [3] BENTLEY, J.L., OTTMANN, T., WIDMAYER, P. “The complexity of manipulating hierarchically defined set of intervals”. *Advances in Computing Research*, v. 1, pp. 127–158, 1983.
- [4] BRANDSTADT, A., LE, V.B., SPINRAD, J. *Graph Classes: A Survey*. SIAM Monographs on Discrete Mathematics and Applications, 1999.
- [5] BREU, H. *Algorithmic Aspects of Constrained Unit Disk Graphs*. Tese de doutorado, University of British Columbia, 1996.
- [6] BREU, HEINZ, KIRKPATRICK, DAVID G. “Unit disk graph recognition is NP-hard”. *Computational Geometry*, v. 9, pp. 3–24, 1998.

- [7] CHIBA, N., NISHIZEKI, T., SAITO, N. “An approximation algorithm for the maximum independent set problem on planar graph”. *SIAM Journal on Computing*, v. 4, pp. 663–675, 1982.
- [8] CLARK, B.N., COLBOURN, C.J., JOHNSON, D.S. “Unit disk graphs”. *Discrete Mathematics*, v. 86, pp. 165–177, 1990.
- [9] DYER, M.E., FRIEZE, A.M. “Planar 3DM is NP-complete”. *Journal of Algorithms*, v. 7, pp. 174–184, 1986.
- [10] EDMONDS, J., KARP, R.M. “Theoretical improvements in algorithmic efficiency for network flow problems”. *Journal of ACM*, v. 19, pp. 248–264, 1972.
- [11] ERLEBACH, T., JANSEN, K., SEIDEL, E. “Polynomial-time approximation schemes for geometric graphs”. In *In Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms (SODA 2001)*, pp. 671–679, 2001.
- [12] EVEN, S., PNUELI, A., LEMPEL, A. “Permutation graphs and transitive graphs”. *Journal of the ACM*, v. 19, n. 3, pp. 400–410, 1972.
- [13] FEDER, T., GREENE, D.H. “Optimal algorithms for approximate clustering”. In *In Proceedings of the 30th ACM Symposium on Theory of Computing*, pp. 434–444, 1988.

- [14] FOWLER, J., PATERSON, M.S., TANIMOTO, S.L. “Optimal packing and covering in the plane are NP-complete”. *Information Processing Letters*, v. 12, pp. 133–137, 1981.
- [15] GAREY, M.R., JOHNSON, D.S. “The rectilinear Steiner tree problem is NP-complete”. *SIAM Journal of Applied Mathematics*, v. 32, pp. 826–834, 1977.
- [16] GAREY, M.R., JOHNSON, D.S. *COMPUTERS AND INTRACTABILITY: A Guide to the Theory of NP-Completeness*, pp. 191 – 194. W. H. Freeman, 1979.
- [17] GOLUMBIC, M.C. “The complexity of comparability graph recognition and coloring”. *Computing*, v. 18, pp. 199–208, 1977.
- [18] GOLUMBIC, M.C. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, 1980.
- [19] GRÄF, A., STUMPF, M., WEISSENFELS, G. *On coloring unit disk graphs*. Technical report, Musikinformatik und Medientechnik 17/94, Johannes Gutenberg-Universität, Musikwissenschaftliches Institut, Bereich Musikinformatik, Mainz, 1994.
- [20] GRÄF, A., STUMPF, M., WEISSENFELS, G. “On coloring unit disk graphs”. *Algorithmica*, v. 20, pp. 277–293, 1998.
- [21] HALE, W.K. “Frequency assignment: theory and applications”. *Proceedings of the IEEE*, v. 68, pp. 1497–1514, 1980.

- [22] HOCHBAUM, D. “Efficient bounds for the stable set, vertex cover and set packing problems”. *Discrete Applied Mathematics*, v. 6, pp. 243–254, 1983.
- [23] HOCHBAUM, D.S. “Efficient bounds for the stable set, vertex cover, and set packing problems”. *Discrete Applied Mathematics*, v. 6, pp. 243–254, 1983.
- [24] HOCHBAUM, D.S., MAASS, W. “Approximation schemes for covering and packing problems in image processing and vlsi”. *Journal of ACM*, v. 32, pp. 130–138, 1985.
- [25] HOCHBAUM, D.S., MAASS, W. “Fast approximation algorithms for a non-convex covering problem”. *Journal of Algorithms*, v. 8, pp. 305–323, 1987.
- [26] HOPCROF, J. E., TARJAN, R.E. “Efficient planarity testing”. *Journal of ACM*, v. 21, pp. 549–568, 1974.
- [27] HUNT III, H.B., MARATHE, M.V., RADHAKRISHNAN, V., RAVI, S.S., ROSENKRANTZ, D.J., STEARNS, R. E. “NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs”. *Journal of Algorithms*, v. 26, pp. 238–274, 1998.
- [28] ITAI, A., PAPADIMITRIOU, C.H., SZWARCFITER, J.L. “Hamilton paths in grid graphs”. *SIAM Journal on Computing*, v. 11, pp. 676–686, 1982.

- [29] J.L. BENTLEY, D.F. STANAT E E.H. WILLIAM JR. “The complexity of finding fixed-radius near neighbors”. *Information Processing Letters*, v. 6, pp. 209–212, 1977.
- [30] JOHNSON, D.S. “The NP-completeness column: An ongoing guide”. *Journal of Algorithms*, v. 6, pp. 434–451, 1983.
- [31] KAMMERLANDER, K. “C 900 - An advanced mobile radio telephone system with optimum frequency utilization”. *IEEE Journal on Selected Areas in Communications*, v. 2, pp. 589–597, 1984.
- [32] KOEBE, P. “Kontaktprobleme der konformen abbildung”. *Berichte Über die Verhandlungen der Sächsischen Akademie der Wissenschaften, Leipzig, Mathematische-Physische Klasse*, v. 88, pp. 141–164, 1936.
- [33] LI, X., CALINESCU, G., WAN, P. “Distributed construction of planar spanner and routing for ad hoc wireless networks”. In *INFOCOM*, 2002.
- [34] LI, X., WAN, P., WANG, Y. “Power efficient and sparse spanner for wireless ad hoc networks”. In *IEEE International Conference on Computer Communications and Networks (ICCCN01)*, 2001.
- [35] LI, X., WAN, P., WANG, Y., FRIEDER, O. “Sparse power efficient topology for wireless networks”, 2002.
- [36] LICHTENSTEIN, D. “Planar formulae and their uses”. *SIAM Journal on Computing*, v. 11, pp. 329–343, 1982.

- [37] MA, X., LLOYD, E.L. *An incremental algorithm for broadcast scheduling in packet radio networks*. Technical report, Department of Computer and Information Sciences, University of Delaware, 1998.
- [38] MALESIŃSKA, E. *Graph-Theoretical Models for Frequency Assignment Problems*. Tese de doutorado, Technischen Universität Berlin, 1997.
- [39] MALESIŃSKA, E., PISKORZ, S., WEISSENFELS, G. “On the chromatic number of disk graphs”. *Networks*, v. 32, pp. 13–22, 1997.
- [40] MARATHE, M.V., BREU, H., HUNT III, H.B., RAVI, S.S., ROSENKRANTZ, D.J. “Simple heuristics for unit disk graphs”. *Networks*, v. 25, pp. 59–68, 1995.
- [41] MARATHE, M.V., RADHAKRISHNAN, V., HUNT III, H.B., RAVI, S.S. “Hierarchically specified unit disk graphs”. *Theoretical Computer Science*, v. 174, pp. 23–65, 1997.
- [42] MASUYAMA, S., IBARAKI, T., HASEGAWA, T. “The computational complexity of the M-center problems in the plane”. *Transitions of IECE Japan*, v. E64, pp. 57–64, 1981.
- [43] MATSUI, T. “Approximation algorithms for maximum independent set problems and fractional coloring problems on unit disk graphs”. *Discrete and Computational Geometry*, v. 1763, pp. 194–200, 2000.
- [44] MEAD, C., CONWAY, L. *Introduction to VLSI systems*. Addison-Wesley, 1980.

- [45] MEGIDDO, N., SUPOWIT, K.J. “On the complexity of some common geometric location problems”. *SIAM Journal on Computing*, v. 13, pp. 182–196, 1984.
- [46] MOHRING, R.H. “Algorithmic aspects of comparability graphs and interval graphs”, 1985.
- [47] PACH, J., AGARWAL, P.K. *Combinatorial Geometry*. John Wiley & Sons, New York, 1995.
- [48] PAPADIMITRIOU, C.H., YANNAKAKIS, M. “The clique problem for planar graphs”. *Information Processing Letters*, v. 13, pp. 131–133, 1981.
- [49] ROBERTS, F.S. “T-colorings of graphs: recent results and open problems”. *Discrete Mathematics*, v. 93, pp. 229–245, 1991.
- [50] SACHS, H. “Coin graphs, polyhedra, and conformal mapping”. *Discrete Mathematics*, v. 134, pp. 133–138, 1994.
- [51] STOCKMEYER, L. “Planar 3-colorability is NP-complete”. *SIGACT News*, v. 5, pp. 19–25, 1973.
- [52] TOREGAS, C., SWAIN, R., REVELLE, C., BERGMAN, L. “The location of emergency service facilities”. *Operations Research*, v. 19, pp. 1363–1373, 1971.
- [53] TSUKIYAMA, S., IDE, M., ARIYOSHI, H., SHIRAKAWA, I. “A new algorithm for generating all the maximal independent sets”. *SIAM Journal on Computing*, v. 6, pp. 505–517, 1977.



- [54] VALIANT, L. G. “Universality considerations in VLSI circuits”. *IEEE Transactions on Computers*, v. 30, pp. 135–140, 1981.
- [55] WONG, D.W., KUO, Y.S. “A study of two geometric location problems”. *Information Processing Letters*, v. 28, pp. 281–286, 1988.
- [56] Y. YEH, J. WILSON, SCHWARTZ, S.C. “Outage probability in mobile telephony with directive antennas and macrodiversity”. *IEEE Journal on Selected Areas in Communications*, v. 2, pp. 507–511, 1984.