

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
INSTITUTO TERCIO PACITTI DE APLICAÇÕES E PESQUISAS
COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

ALINE RODRIGUES DA SILVA

**O NÚMERO DE HELLY NA CONVEXIDADE
GEODÉTICA: Contribuições Teóricas e Práticas**

RIO DE JANEIRO
2014

ALINE RODRIGUES DA SILVA

**O NÚMERO DE HELLY NA CONVEXIDADE
GEODÉTICA: Contribuições Teóricas e Práticas**

Dissertação de Mestrado submetida ao Corpo Docente do Departamento de Ciência da Computação do Instituto de Matemática, e Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários para obtenção do título de Mestre em Informática.

Orientador: Mitre Costa Dourado

Co-orientador: Guilherme Dias da Fonseca

RIO DE JANEIRO
2014

S586 Silva, Aline Rodrigues da

O NÚMERO DE HELLY NA CONVEXIDADE GEODÉTICA: Contribuições Teóricas e Práticas / Aline Rodrigues da Silva. – 2014.
93 f.: il.

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro, Instituto de Matemática, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Programa de Pós-Graduação em Informática, Rio de Janeiro, 2014.

Orientador: Mitre Costa Dourado. Co-orientador: Guilherme Dias da Fonseca.

1. Número de Helly. 2. Convexidade Geodética. 3. Grafo Bipartido. 4. Grafo Sol. 5. Grafo Roda. 6. Grafo Leque. 7. Grafo $k_{1,t}$ -free. – Teses. I. Dourado, Mitre Costa (Orient.). II. Fonseca, Guilherme Dias da (Co-orient.). III. Universidade Federal do Rio de Janeiro, Instituto de Matemática, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Programa de Pós-Graduação em Informática. IV. Título

CDD

ALINE RODRIGUES DA SILVA

**O NÚMERO DE HELLY NA CONVEXIDADE
GEODÉTICA: Contribuições Teóricas e Práticas**

Dissertação de Mestrado submetida ao Corpo Docente do Departamento de Ciência da Computação do Instituto de Matemática, e Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários para obtenção do título de Mestre em Informática.

Aprovado em: Rio de Janeiro, ____ de _____ de _____.

Prof. Dr. Mitre Costa Dourado (Orientador)

Prof. Dr. Lilian Markenzon

Prof. Dr. Vinícius Gusmão Pereira de Sá

Prof. Dr. Fabiano de Souza Oliveira

RIO DE JANEIRO
2014

Dedico este trabalho à minha mãe.

AGRADECIMENTOS

A Deus, por guiar os meus passos.

À minha mãe, por me apoiar na luta por meus ideais.

Aos meus orientadores, pela dedicação, ensinamento, apoio, confiança e paciência dirigidos a mim.

Aos membros da banca, por terem aceitado o convite para avaliar esta dissertação.

Aos meus professores, pelo conhecimento e inspiração que me proporcionaram ao longo dos anos.

Aos meus amigos, pelo apoio e motivação.

Aos meus colegas de curso, pelas novas amizades, pela boa convivência e pela troca de conhecimentos e experiências.

Ao CNPq e à CAPES, pelo suporte financeiro.

A todos que de alguma forma contribuíram para o desenvolvimento desta dissertação.

RESUMO

SILVA, Aline Rodrigues da. **O NÚMERO DE HELLY NA CONVEXIDADE GEODÉTICA: Contribuições Teóricas e Práticas**. 2014. 93 f. Dissertação (Mestrado em Informática) - Programa de Pós-graduação em Informática, Instituto de Matemática, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2014.

Nesta dissertação, estudamos o número de Helly em grafos na convexidade geodética. Consideramos grafos simples, conexos, não-direcionados e não-ponderados.

Seja $G = (V(G), E(G))$ um grafo. Um conjunto $S \subseteq V(G)$ é *geodesicamente convexo* se para todo par de vértices $u, v \in S$, todos os vértices em todos os caminhos mínimos que ligam u e v também pertencem a S . O *fecho convexo* $H(S)$ de um conjunto $S \subseteq V(G)$ é o menor conjunto convexo contendo S .

O *núcleo* de uma família é a interseção total de seus conjuntos. Uma família F de conjuntos é *k-intersectante* se toda subfamília de F com k conjuntos tem um núcleo não-vazio. Uma família F de conjuntos é *k-Helly* se toda subfamília de F que é *k-intersectante* tem o núcleo não-vazio. O *número de Helly* de uma família F é o menor inteiro k tal que F é *k-Helly*. O número de Helly na convexidade geodética em um grafo G é o número de Helly da família definida pelos conjuntos convexos da convexidade geodética em G .

Nesta dissertação, implementamos um programa que calcula o número de Helly k do grafo G e um certificado que mostra que o grafo G não é $(k - 1)$ -Helly, entre outras propriedades. Encontramos um limite superior para o número de Helly em grafos bipartidos e apresentamos uma família que mostra que este limite é justo. Encontramos limites superiores para o número de Helly em grafos com no máximo 9 vértices. Calculamos o número de Helly para as seguintes classes de grafos: sol, roda e leque. Formulamos algumas proposições relacionando algumas propriedades dos grafos com o seu número de Helly. Por fim, apresentamos algumas estatísticas dos dados processados em nossas implementações.

Palavras-chave: Número de Helly, Convexidade Geodética, Grafo Bipartido, Grafo Sol, Grafo Roda, Grafo Leque, Grafo $k_{1,t}$ -free.

ABSTRACT

SILVA, Aline Rodrigues da. **O NÚMERO DE HELLY NA CONVEXIDADE GEODÉTICA: Contribuições Teóricas e Práticas**. 2014. 93 f. Dissertação (Mestrado em Informática) - Programa de Pós-graduação em Informática, Instituto de Matemática, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2014.

In this thesis, we study the Helly number of graphs on the geodetic convexity. We consider simple, connected, undirected and unweighted graphs.

Let $G = (V(G), E(G))$ be a graph. A set $S \subseteq V(G)$ is *geodesically convex* if for any pair of vertices $u, v \in S$ every vertex in every shortest path joining u and v also belongs to S . The *convex hull* $H(S)$ of a set $S \subseteq V(G)$ is the smallest convex set containing S .

The *core* of a family is the total intersection of its sets. A family of sets is *k-intersecting* if every subfamily with k sets has a non-empty core. A family of sets F is *k-Helly* if every k -intersecting subfamily of F has non-empty core. The *Helly number* of F is the smallest number k such that F is k -Helly. The Helly number of the geodetic convexity on a graph G is the Helly number of the family defined by the convex sets of the geodetic convexity on G .

In this thesis, we have implemented a program that calculates the Helly number k of the graph G and a certificate showing that the graph G is not $(k - 1)$ -Helly, among other properties. We have found an upper bound for the Helly number of bipartite graphs and we present a family that shows that this limit is tight. We have found upper bounds for the Helly number of graphs with at most 9 vertices. We established the Helly number of the following classes of graphs: sun, wheel and fan. We formulate some propositions relating certain properties of graphs with your Helly number. Finally, we present some statistics of the data processed in our implementations.

Keywords: Helly Number, Geodetic Convexity, Bipartite Graph, Sun Graph, Wheel Graph, $K_{1,t}$ -free Graph .

LISTA DE FIGURAS

2.1	Grafo com laço	18
2.2	Grafo K_5	19
2.3	Grafo bipartido	19
2.4	Partições do grafo $K_{3,3}$	20
2.5	Complemento de um grafo	20
2.6	Subgrafos do grafo G	21
2.7	Passeio	22
2.8	Árvore	22
2.9	Conexidade de grafos	23
2.10	Conjunto independente e clique	24
2.11	Grafos isomorfos	24
2.12	Grafo 3-cubo	25
2.13	Grafo tambor	26
2.14	Conjunto convexo	29
2.15	Conjunto não-convexo	30
2.16	Convexidade geodética em grafo	31
2.17	Convexidade monofônica em grafo	32
2.18	Convexidade Steiner em grafo	33
2.19	Convexidade P_3 em grafo	33
2.20	Propriedade de Helly	34
2.21	Hipergrafo	35
2.22	Hipergrafo p-Helly	36
2.23	Grafo com número de Helly maior do que a clique máxima	37
2.24	Grafo de Petersen	39
2.25	Grafos quase completos	39
3.1	Exemplo da Afirmação 3.1.2	42
3.2	Grafo G_4	44
3.3	Fecho convexo dos vértices v_a, v_b	45
3.4	Fecho convexo dos vértices v_a, v_{bc}	45
3.5	Fecho convexo dos vértices v_{ab}, v_{bd}	45
3.6	Fecho convexo dos vértices v_{ab}, v_{cd}	46
3.7	Exemplo da Proposição 3.2.1	48
3.8	Exemplos da Conjectura 3.2.2	49
3.9	Exceção da Proposição 3.2.6	51
3.10	Exemplo para as Proposições 3.2.3, 3.2.4, 3.2.5 e 3.2.6	52

3.11	Grafo 4-leque	52
3.12	Grafo sol S_4	53
3.13	Grafo roda W_6	55
4.1	Arquivo graph6 de grafos conexos com 5 vértices	58
4.2	Arquivo texto parcial de grafos conexos com 5 vértices	59
4.3	Arquivo graph6 de grafos bipartidos com 6 vértices	60
4.4	Arquivo texto parcial de grafos bipartidos com 6 vértices	61
4.5	Arquivo com matriz de adjacências de grafo conexo com 5 vértices	62
4.6	Arquivo de lista de adjacência de grafo conexo com 5 vértices . .	63
4.7	Arquivo <i>gdf</i> de grafo conexo com 5 vértices	64
4.8	Arquivo <i>csv</i> de grafo conexo com 5 vértices	65
4.9	Arquivo texto com dados gerados para um grafo	67
4.10	Arquivo texto com dados gerados para um grafo	68
4.11	Fluxograma da geração das matrizes de adjacências dos grafos .	69
4.12	Fluxograma da separação das matrizes de adjacências dos grafos	70
4.13	Fluxograma de exportação das matrizes de adjacências dos grafos	71
4.14	Fluxograma do processo de experimentação	72

LISTA DE TABELAS

4.1	Quantidade de grafos gerados do Grupo 1	67
4.2	Quantidade de grafos utilizados para os experimentos	73
4.3	Tempo em segundos gasto nos testes dos grafos conexos	74
4.4	Tempo em segundos gasto nos testes dos grafos bipartidos	74
4.5	Porcentagem de grafos em que $h(G) = \omega(G)$	75
4.6	Porcentagem de grafos em que $h(G) > \omega(G)$	76
4.7	Porcentagem de grafos em que $h(G) = \omega(G) + 1$	76
4.8	Porcentagem de grafos em que $h(G) = \omega(G) + 2$	77
4.9	Porcentagem de grafos em que $h(G) = \omega(G) + 3$	77
4.10	Medidas de resumo acerca do número de Helly em grafos conexos	78
4.11	Medidas de resumo acerca do número de Helly em grafos bipartidos	78
4.12	Medidas de resumo acerca do número de Helly em grafos biconexos- bipartidos	78

LISTA DE ALGORITMOS

4.1	Número de Helly em grafos	66
A.1	Grafo k-cubo	88
A.2	Grafo Leque	89
A.3	Grafo Ciclo	89
A.4	Grafo Roda	90
A.5	Grafo Sol	90
A.6	Grafo Tambor	91
A.7	Grafo $K_{1,t}$ -free mínimo	91
A.8	Grafo $K_{1,t}$ -free	92
A.9	Cardinalidade da Clique Máxima	92
A.10	Grafo k-clique	93

SUMÁRIO

1	INTRODUÇÃO	14
2	PRELIMINARES	17
2.1	Conceitos básicos de grafos	17
2.2	Complexidade computacional	25
2.3	Convexidade em geral	29
2.4	Convexidade em grafos	30
2.5	Número de Helly	34
3	CONTRIBUIÇÕES TEÓRICAS	41
3.1	Grafos Bipartidos	41
3.2	Algumas classes	47
4	IMPLEMENTAÇÃO E TESTES	56
4.1	Implementação	57
4.2	Experimentos	65
4.3	Análise dos dados	69
5	CONCLUSÃO	79
	REFERÊNCIAS	81
	APÊNDICE A	86
A.1	Estrutura do Banco de Dados Utilizado	86
A.2	Algoritmos Implementados	87

1 INTRODUÇÃO

Em 1923, Eduard Helly publicou seu célebre teorema, que originou a propriedade Helly. Uma família F de conjuntos tem a *propriedade de Helly* quando toda subfamília de F formada por pares de conjuntos que se interceptam, tem um elemento em comum.

Um conjunto C de um espaço linear real é chamado *convexo* se, e somente se, para todo par de pontos x e $y \in C$, o segmento de linha que liga x e y também pertence a C .

Teorema 1(HELLY, 1923)[14] *Seja K uma família finita de pelo menos $(n + 1)$ conjuntos convexos no espaço finito \mathbb{R}^n . Então, se cada $(n + 1)$ membros de K tem um ponto em comum, há um ponto comum a todos os membros de K .*

Segundo [17], a propriedade de Helly tem aplicações em várias áreas. Em otimização, por exemplo, tem sido aplicada em problemas de localização [19] e programação linear [1]. Em ciência da computação, tem sido utilizada em teoria da semântica [4], codificação [3], biologia computacional [33], banco de dados [20], processamento de imagens [7] e teoria dos grafos [17].

Seja $G = (V(G), E(G))$ um grafo. Um conjunto $S \subseteq V(G)$ é *geodesicamente convexo* se para todo par de vértices $u, v \in S$, todos os vértices em todos os caminhos mínimos que ligam u e v também pertencem a S . O *fecho convexo*, $H(S)$, de um conjunto $S \subseteq V(G)$ é o menor conjunto convexo contendo S .

O *núcleo* de uma família é a interseção total de seus conjuntos. Uma família F de conjuntos é *k -intersectante* se toda subfamília de F com k conjuntos tem um núcleo não-vazio. Uma família F de conjuntos é *k -Helly* se toda subfamília de F que é k -intersectante tem o núcleo não-vazio. O *número de Helly* de uma família F é o menor inteiro k tal que F é k -Helly.

Nesta dissertação, estudamos o número de Helly em grafos na convexidade geodética. O número de Helly na convexidade geodética em um grafo G é o número de Helly da família definida pelos conjuntos convexos da convexidade geodética em G . Na literatura, o número de Helly é amplamente estudado na matemática contínua [14, 26, 34]. Nos últimos anos, tem havido um grande interesse da comunidade acadêmica em pesquisar o número de Helly na matemática discreta, onde tem sido estudado o número de Helly em grafos [2, 15, 18, 32].

Nesta dissertação, consideramos grafos simples, conexos, não-direcionados e não-ponderados. Nossas principais contribuições são descritas a seguir. Implementamos um programa que tem por objetivo calcular as seguintes propriedades em grafos conexos:

- o número de Helly k ;
- um certificado que mostra que o grafo não é $(k - 1)$ -Helly;
- a cardinalidade da clique máxima;
- o conjunto de vértices da clique máxima encontrada;
- o menor inteiro t tal que o grafo é $K_{1,t}$ -free;
- diâmetro do grafo.

Como resultados teóricos, encontramos um limite superior para o número de Helly em grafos bipartidos e apresentamos uma família que mostra que este limite é justo. Encontramos limites superiores para o número de Helly em grafos com no máximo 9 vértices. Calculamos o número de Helly para as seguintes classes de grafos: sol, roda e leque. Formulamos algumas proposições relacionando algumas propriedades dos grafos com o seu número de Helly. Por fim, apresentamos algumas estatísticas dos dados processados em nossas implementações.

Esta dissertação está organizada em capítulos como apresentado a seguir. O Capítulo 1 consiste da Introdução, que é o atual capítulo, onde apresentamos a

definição do nosso problema de estudo, o nosso objetivo, a motivação, uma descrição sucinta de nossas contribuições e a organização dos capítulos. O Capítulo 2 introduz alguns conceitos básicos de grafos, complexidade computacional, convexidade em geral, convexidade em grafos e número de Helly. O Capítulo 3 discursa sobre os resultados teóricos que obtivemos neste estudo. O Capítulo 4 descreve a nossa implementação, os testes realizados e a análise dos dados processados. Por fim, o capítulo 5 consiste da conclusão desta dissertação, que descreve sucintamente as nossas contribuições, cita os artigos aceitos para publicação com resultados deste trabalho e cita alguns problemas em aberto.

2 PRELIMINARES

Neste capítulo, introduzimos alguns conceitos básicos que serão utilizados ao longo deste trabalho. Dividimos o capítulo em cinco seções que abordam, respectivamente, os seguintes temas: teoria de grafos, complexidade de algoritmos, convexidade em geral, convexidade em grafos e número de Helly.

2.1 Conceitos básicos de grafos

Um *grafo* G é um par ordenado $(V(G), E(G))$, onde $V(G)$ é um conjunto finito não-vazio e $E(G)$ é um conjunto de pares não-ordenados de elementos de $V(G)$. Os elementos de $V(G)$ são denominados *vértices* ou *nós* e os de $E(G)$ são denominados *arestas*. Vamos denotar por n a cardinalidade do conjunto de vértices $V(G)$ e por m a cardinalidade do conjunto de arestas $E(G)$.

Cada aresta $e \in E(G)$ será denotada pelo par de vértices $e = (u, v)$. Os vértices u e v são denominados *extremos* de e . A aresta e é dita *incidente* a u e v . Dois vértices u e v são *adjacentes* se existe uma aresta com extremos u e v . Duas arestas que possuem um dos extremos em comum são denominadas *adjacentes*. O grau $d(v)$ de um vértice $v \in V(G)$ é o número de arestas de G incidentes a v . Denotamos por $\delta(G)$ e $\Delta(G)$, respectivamente, o grau mínimo e o grau máximo dos vértices de G .

Uma aresta com extremos idênticos é denominada *laço* e com extremos distintos é denominada *link*. Duas arestas que unem o mesmo par de vértices são denominadas *arestas paralelas*. Um grafo é *simplex* se não possui laços e se não possui arestas paralelas.

Um grafo pode ser visualizado através de sua *representação geométrica*, na qual seus vértices correspondem a pontos distintos no plano e a cada aresta (u, v) é associada uma linha unindo os pontos correspondentes a u e v .

Como exemplo dos conceitos apresentados, observe o grafo G na Figura 2.1.

O grafo G é formado pelo conjunto de vértices $V(G) = \{1, 2, 3, 4, 5\}$ e pelo conjunto de arestas $E(G) = \{(1, 2), (1, 5), (2, 2), (2, 3), (2, 5), (3, 4), (3, 5), (4, 5), \}$. Podemos observar que G tem grau máximo igual a 5 e grau mínimo igual a 2. Além disso, a aresta $e \in G$ é um laço e as demais são *links*.

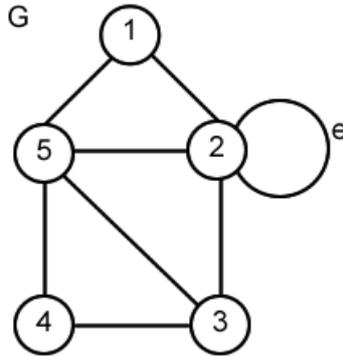


Figura 2.1: Grafo com laço

Um grafo G é denominado um *grafo direcionado* se $E(G)$ é um conjunto de pares ordenados de elementos de $V(G)$. Um grafo G é denominado um *grafo ponderado* se associamos um peso a cada aresta de $E(G)$.

Um grafo simples no qual cada par de vértices distintos é conectado por uma aresta é denominado *grafo completo* e denotado por K_n . Como exemplo, veja o grafo completo K_5 na Figura 2.2. Por outro lado, um grafo que não tem arestas é denominado *grafo vazio*.

Um grafo é *bipartido* se o seu conjunto de vértices pode ser particionado em dois subconjuntos X e Y , tal que cada aresta tem um extremo em X e outro em Y . Assim, uma partição (X, Y) é denominada uma *bipartição* do grafo. Como exemplo, observe o grafo G na Figura 2.3a. Podemos ver as partições X e Y de G na Figura 2.3b.

Um *grafo bipartido completo* é um grafo bipartido simples com bipartição (X, Y) na qual cada vértice de X é adjacente a todos os vértices de Y . Se $|X| = p$ e $|Y| = q$, denotamos este grafo por $K_{p,q}$. Uma *estrela*, denotada por S_k , é o grafo

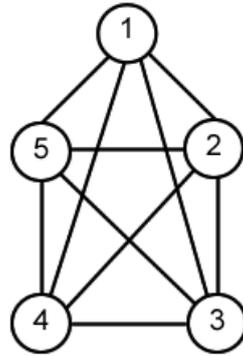
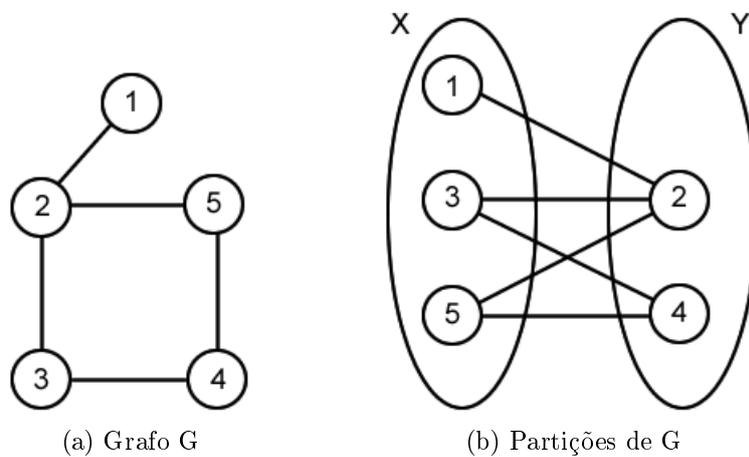
Figura 2.2: Grafo K_5 

Figura 2.3: Grafo bipartido

bipartido completo $K_{1,k}$. Como exemplo de grafo bipartido completo, observe o grafo $K_{3,3}$ na Figura 2.4.

O *complemento* \overline{G} de um grafo simples G é um grafo simples com o mesmo conjunto de vértices de G , tal que dois vértices são adjacentes em \overline{G} se, e somente se, não são adjacentes em G . Como exemplo, observe o grafo G na Figura 2.5a e seu complemento \overline{G} na Figura 2.5b.

Um grafo H é um *subgrafo* de G se $V(H) \subseteq V(G)$ e $E(H) \subseteq E(G)$. Um grafo H é um *subgrafo induzido* de G , denotado por $G[V(H)]$, se $V(H) \subseteq V(G)$ e $E(H)$ é o conjunto de arestas de G que tem ambos os extremos em $V(H)$. Como

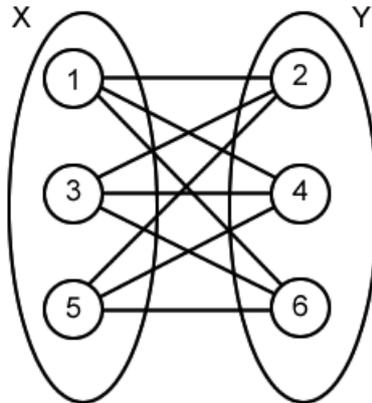


Figura 2.4: Partições do grafo $K_{3,3}$

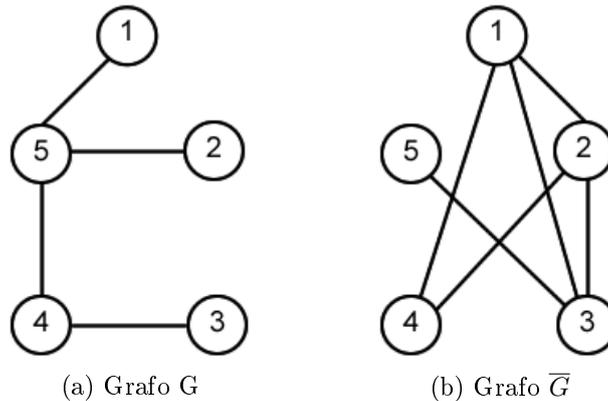


Figura 2.5: Complemento de um grafo

exemplo, observe o grafo G na Figura 2.6a. Na Figura 2.6b, temos um subgrafo de G que não é um subgrafo induzido de G . Já na Figura 2.6c, temos um subgrafo induzido de G .

Um *passeio* em G é uma sequência finita não-nula $W = v_0e_1v_1e_2v_2e_3\dots e_kv_k$ cujos termos são alternadamente vértices e arestas, tal que os extremos de e_i são v_{i-1} e v_i , para $1 \leq i \leq k$. Os vértices v_0 e v_k são, respectivamente, a *origem* e o *término* de W . Os vértices v_1, v_2, \dots, v_{k-1} são os *vértices internos*. O inteiro k é o tamanho de W . Se as arestas de W são distintas, temos uma *trilha*. Se os vértices de W são distintos, temos um *caminho*. Um caminho com origem v_0 e término v_k é

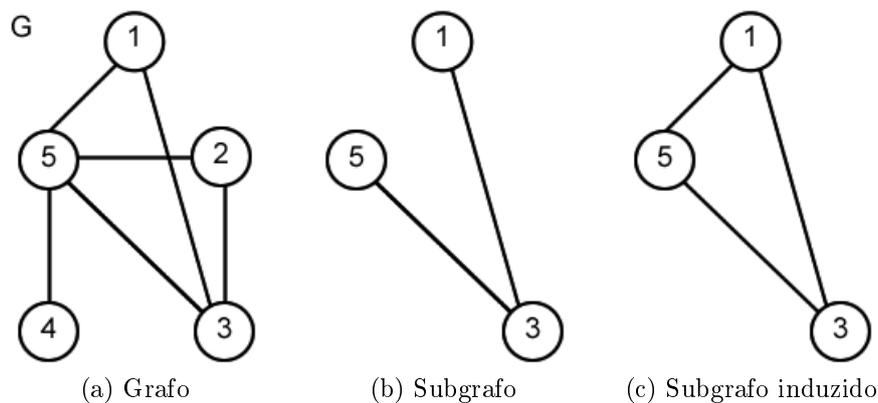


Figura 2.6: Subgrafos do grafo G

denominado um $(v_0 - v_k)$ -caminho. Observe que todo caminho é também uma trilha. Um *ciclo* é um caminho em que v_0 e v_k são adjacentes. Uma *corda* em um caminho W em G é uma aresta $e \in E(G)$ tal que e conecta dois vértices não-consecutivos na sequência de vértices de W . Um caminho sem cordas é denominado um *caminho induzido*. Como exemplos destes conceitos, observe o grafo G na Figura 2.7. Note que:

- O passeio $W = \{1, 3, 5, 4, 2\}$ é um caminho em G e também uma trilha.
- O passeio $W = \{1, 2, 3, 5, 2, 4\}$ não é um caminho, pois repete o vértice 2. No entanto, W é uma trilha em G .
- O passeio $W = \{1, 2, 4, 2, 5, 3\}$ não é uma trilha nem um caminho em G , pois repete aresta $(2, 4)$.
- O passeio $W = \{1, 3, 5, 4, 2, 1\}$ é um ciclo em G .
- O passeio $W = \{1, 3, 5, 4\}$ é um caminho induzido em G .
- O passeio $W = \{1, 2, 3, 5, 4\}$ não é um caminho induzido em G , pois a aresta $(1, 3)$ é uma corda em W .

Um grafo G é *acíclico* se G não possui ciclos. Uma *árvore* é um grafo conexo acíclico. Veja na Figura 2.8 um exemplo de árvore.

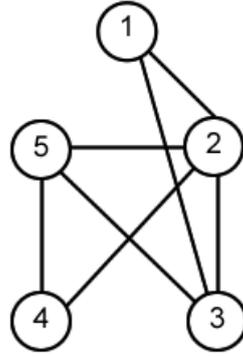


Figura 2.7: Passeio

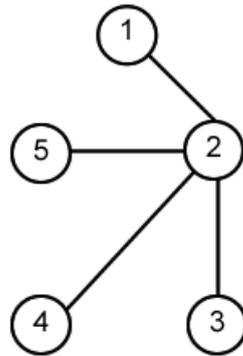


Figura 2.8: Árvore

Sejam u e v dois vértices de G . A *distância* entre u e v , denotada $d(u, v)$, é a quantidade de arestas do menor $(u - v)$ -caminho. O *diâmetro* de G é a distância máxima entre dois vértices de G .

Dois vértices u e v são *conexos* se existe um caminho de u a v . Conexão é uma relação de equivalência no conjunto de vértices $V(G)$. Assim, existe uma partição de $V(G)$ em subconjuntos não-vazios $V_1, V_2, \dots, V_\omega$ tal que dois vértices u e v são conectados se, e somente se, u e v pertencem ao mesmo conjunto V_i . Os subgrafos $G[V_1], G[V_2], \dots, G[V_\omega]$ são denominados *componentes* de G . O grafo G é denominado *conexo* se tem exatamente uma componente, ou seja, se para todo par (u, v) de vértices existe pelo menos um passeio com origem u e término v . Caso

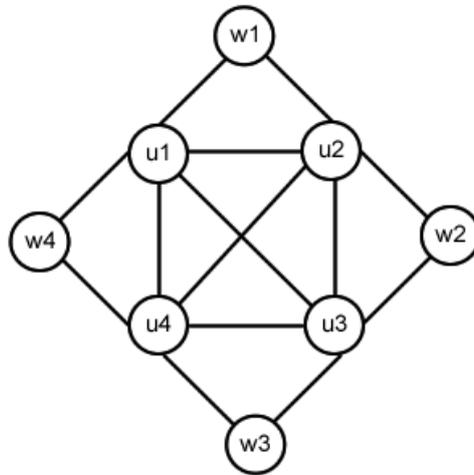


Figura 2.10: Conjunto independente e clique

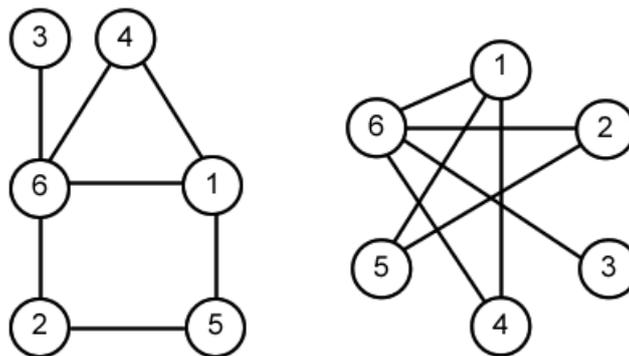


Figura 2.11: Grafos isomorfos

deste grafo.

O *grafo tambor*, denotado por T_k , é um grafo com $2k$ vértices ($k \geq 3$), formado a partir de dois ciclos C'_k e C''_k , cujos conjuntos de vértices são $V(C'_k) = \{u_0, \dots, u_{k-1}\}$ e $V(C''_k) = \{v_0, \dots, v_{k-1}\}$, adicionando uma aresta entre os vértices u_i e v_i . Na Figura 2.13, temos um exemplo deste grafo.

Nesta dissertação, consideramos grafos simples, conexos, não-direcionados e não-ponderados.

Nossas principais referências bibliográficas para esta seção foram [6] e [13].

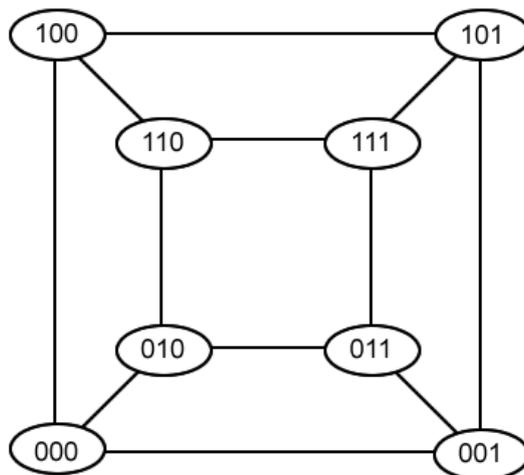


Figura 2.12: Grafo 3-cubo

2.2 Complexidade computacional

Um *algoritmo* é um conjunto de passos ordenados para a resolução de um problema. O algoritmo recebe como *entrada* um conjunto de dados previamente conhecidos, computa os passos necessários e retorna como *saída* a solução ao problema proposto.

Existem dois aspectos primordiais no estudo de algoritmos: a *correção* e a *análise*. A correção consiste em verificar a precisão da técnica empregada através de uma prova matemática. A análise consiste em obter parâmetros para avaliar a eficiência do algoritmo em termos de tempo de execução e memória ocupada.

É possível determinar o tempo de execução de um algoritmo através de métodos empíricos, executando o algoritmo considerando várias entradas, ou através de métodos analíticos, obtendo uma ordem de grandeza do tempo de execução. Estes métodos visam encontrar uma expressão matemática que exprima o comportamento de tempo de execução do algoritmo. Ao contrário do método empírico, o analítico visa medir o tempo de forma independente do computador, da linguagem, dos compiladores e das condições locais de processamento. Para tanto, o método analítico

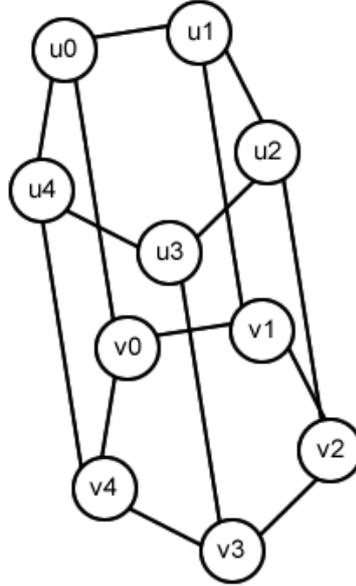


Figura 2.13: Grafo tambor

exprime o tempo em função da entrada do algoritmo. Cada passo do algoritmo consiste na execução de um número fixo de operações cujo tempo de execução é considerado constante. A operação básica de maior frequência é denominada *operação dominante*. A expressão do tempo de execução é obtida desconsiderando constantes aditivas e multiplicativas. Assim, o número de passos pode ser obtido considerando-se apenas o número de execuções da operação dominante.

Podemos definir *complexidade de tempo*, e analogamente *complexidade de espaço*, como a seguir. Seja A um algoritmo e $E = \{E_1, E_2, \dots, E_n\}$ o conjunto de todas as entradas possíveis de A . Dada uma entrada E_i , para $1 \leq i \leq n$, seja t_i o número de passos efetuados por A e p_i a probabilidade de ocorrência da entrada E_i . Podemos definir:

$$\text{complexidade de pior caso} = \max_{E_i \in E} \{t_i\},$$

$$\text{complexidade de melhor caso} = \min_{E_i \in E} \{t_i\},$$

$$\text{complexidade de caso médio} = \sum_{1 \leq i \leq n} p_i t_i.$$

A complexidade de pior caso é a mais utilizada, pois fornece um limite supe-

rior para o número de passos que o algoritmo efetua em qualquer caso. Portanto, quando nos referirmos a *complexidade*, estaremos falando da complexidade de pior caso.

A complexidade de um algoritmo é expressa através de uma notação matemática denominada *notação O, Ω, θ* . A seguir definimos estas notações.

A notação O é utilizada para exprimir limites superiores assintóticos e é definida como a seguir. Sejam f e h funções reais positivas de variável inteira n . Dizemos que f é $O(h)$, quando existir uma constante $c > 0$ e um valor inteiro n_0 , tal que $n > n_0 \Rightarrow f(n) \leq c.h(n)$.

A notação Ω é utilizada para exprimir limites inferiores assintóticos e é definida como a seguir. Sejam f e h funções reais positivas de variável inteira n . Dizemos que f é $\Omega(h)$, quando existir uma constante $c > 0$ e um valor inteiro n_0 , tal que $n > n_0 \Rightarrow f(n) \geq c.h(n)$.

A notação θ é utilizada para exprimir limites superiores justos e é definida como a seguir. Sejam f e h funções reais positivas de variável inteira n . Dizemos que f é $\theta(h)$, quando ambas as condições $f = O(h)$ e $h = O(f)$ forem satisfeitas. Isso significa que as duas funções f e h possuem a mesma ordem de grandeza.

Um algoritmo é denominado *eficiente* quando a sua complexidade é um polinômio no tamanho de sua entrada. Um problema Π é denominado *tratável* se existir um algoritmo eficiente que o resolva. Caso contrário, é denominado *intratável*.

Um *problema algorítmico* é formado por um conjunto de todos os possíveis dados do problema e por uma *questão*. Os dados específicos que constituem uma entrada formam uma *instância do problema*. Há três tipos gerais de problemas algorítmicos, são eles:

- O *problema de decisão* cujo objetivo é decidir a resposta *SIM* ou *NÃO* para uma questão. Vamos denotar por $\Pi = (D, Q)$, um problema de decisão, onde D é o conjunto de dados e Q é a questão do problema. Quando estivermos nos referindo ao problema de decisão, escreveremos o nome do problema em letra maiúscula.

- O *problema de localização* cujo objetivo é localizar uma certa estrutura que atenda a um conjunto de propriedades dadas.
- O *problema de otimização* cujo objetivo é localizar uma certa estrutura que atenda a um conjunto de propriedades dadas satisfazendo critérios de otimização.

Com relação a tratabilidade dos problemas de decisão são definidas a seguir algumas classes de complexidade. A primeira delas é a *Classe P* que corresponde exatamente aos problemas tratáveis.

Um *certificado* para um problema de decisão é uma prova de sua solução. Este certificado pode ser representado por um conjunto de argumentos que quando interpretados podem certificar a resposta *SIM* ou *NÃO* para o problema. Um certificado deve ser validado por um *algoritmo de validação*.

A *Classe NP* de problemas de decisão corresponde aos problemas Π que possuem um certificado para a resposta *SIM*, que possa ser validado por um algoritmo polinomial no tamanho da entrada de Π . Para verificar se Π pertence a *NP* devemos definir um certificado C para a resposta *SIM* e validá-lo através de um algoritmo de validação cuja entrada deve ser C e a entrada de Π . Se o algoritmo de validação for polinomial no tamanho da entrada de Π , concluímos que Π está em *NP*. Caso contrário, não podemos necessariamente concluir que Π não está em *NP*.

A *Classe Co-NP* de problemas de decisão corresponde aos problemas Π que possuem um certificado para a resposta *NÃO*, que possa ser validado por um algoritmo polinomial no tamanho da entrada de Π .

Sejam $\Pi_1 = (D_1, Q_1)$ e $\Pi_2 = (D_2, Q_2)$ dois problemas de decisão. Uma *transformação polinomial* de Π_1 para Π_2 é uma função $f : D_1 \rightarrow D_2$, tal que f pode ser computada em tempo polinomial em I e para cada instância $I \in D_1$, $\Pi_1(I)$ possui resposta *SIM* se, e somente se, $\Pi_2(f(I))$ também o possuir. Neste caso, dizemos que Π_1 se reduz polinomialmente a Π_2 e usamos a notação $\Pi_1 \propto \Pi_2$.

Um problema de decisão Π pertence a *Classe NP-completo* se $\Pi \in NP$ e

todos os outros problemas de decisão em NP se reduzem a ele. No entanto, se apenas a segunda condição for satisfeita, Π pertence a *Classe NP-difícil*.

De modo similar, um problema de decisão Π pertence a *Classe CO-NP-completo* se $\Pi \in CO-NP$ e todos os outros problemas de decisão em $CO-NP$ se reduzem a ele.

Nossas principais referências bibliográficas para esta seção foram [21], [25], [35] e [36].

2.3 Convexidade em geral

Um conjunto C de um espaço linear real é chamado *convexo* se, e somente se, para todo par de pontos x e $y \in C$, o segmento de linha que liga x e y também pertence a C [38].

Como exemplo de conjunto convexo, observe o conjunto A na Figura 2.14a. Podemos ver na Figura 2.14b que os segmentos de linhas que unem os pontos de A também pertencem a A .

Como exemplo de conjunto não-convexo, observe o conjunto B na Figura 2.15a. Podemos ver na Figura 2.15b que o segmento de linha que une os pontos p e q não pertence a B .

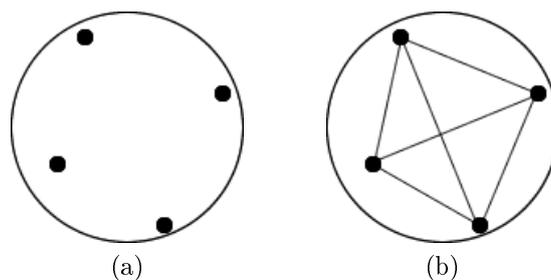


Figura 2.14: Conjunto convexo

Uma convexidade sobre um conjunto não-vazio X é uma família \mathcal{C} de subconjuntos de X que segue os seguintes axiomas [27]:

- $\emptyset, X \in \mathcal{C}$;

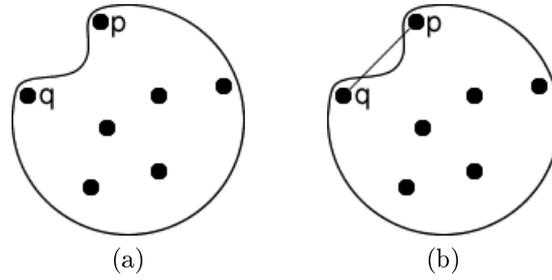


Figura 2.15: Conjunto não-convexo

- interseções arbitrárias de conjuntos convexos são conjuntos convexos;
- toda união de conjuntos convexos aninhados por inclusão é um conjunto convexo.

Uma vez que nesta dissertação estamos trabalhando em uma convexidade sobre um conjunto X finito, o terceiro axioma sempre vale.

O par (X, \mathcal{C}) é chamado de *espaço convexo*. Os elementos de \mathcal{C} são chamados *conjuntos convexos*. O *fecho convexo* de um conjunto $A \subseteq X$ é o menor conjunto convexo contendo A .

2.4 Convexidade em grafos

Discursamos a seguir sobre as seguintes convexidades em grafos: geodética [10, 23], monofônica [24], de Steiner [8, 12] e P_3 [9]. Além dessas, outras convexidades podem ser encontradas na literatura.

Vamos definir a seguir a convexidade geodética. Seja $G = (V(G), E(G))$ um grafo conexo. Sejam u e v dois vértices de G . Um $(u - v)$ -caminho de tamanho $d(u, v)$ é denominado uma $(u - v)$ -geodésica. O *intervalo geodético* $I(u, v)$ consiste de todos os vértices pertencentes a uma $(u - v)$ -geodésica em G . Para um conjunto S de vértices de G , a união de todos os conjuntos $I(u, v)$ para $u, v \in S$ é denotada $I(S)$. Portanto, $x \in I(S)$ se, e somente se, x pertence a alguma $(u - v)$ -geodésica, onde $u, v \in S$. Um conjunto S de vértices de G com $I(S) = V(G)$ é denominado

um *conjunto geodético* de G . A cardinalidade de um conjunto geodético mínimo em G é denominada o *número geodético* de G .

Um conjunto S de vértices em um grafo G é *convexo* se $I(S) = S$. O *fecho convexo* $H(S)$ de um conjunto S de $V(G)$ é o menor conjunto convexo contendo S . Logo, S é um conjunto convexo em G se, e somente se, $H(S) = S$ [11].

Por exemplo, observe a Figura 2.16.

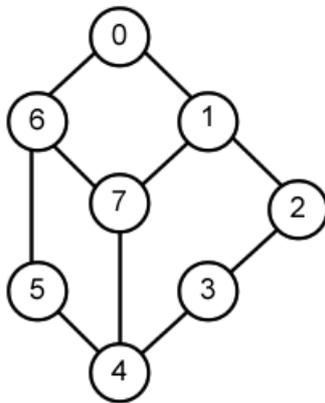


Figura 2.16: Convexidade geodética em grafo

Se escolhermos $S = \{0, 1, 2, 6, 7\}$. Temos que :

- S é convexo, pois $I(S) = S$.
- S não é um conjunto geodético de G , pois $I(S) \neq V(G)$.
- S é um conjunto convexo em G , pois $H(S) = S$.

Se escolhermos $S = \{1, 3, 6\}$. Temos que :

- S não é convexo, pois $I(S) \neq S$.
- S é um conjunto geodético de G , pois $I(S) = V(G)$.
- S não é um conjunto convexo em G , pois $H(S) \neq S$.

A seguir definimos a convexidade monofônica. Seja $G = (V(G), E(G))$ um grafo conexo. Um conjunto $S \subseteq V(G)$ é *m-convexo* se, e somente se, para todo par de vértices $u, v \in S$, todos os vértices em todos os caminhos induzidos que ligam u e v também pertencem a S . A coleção de todos os conjuntos *m-convexos* formam a *convexidade monofônica* em G .

Por exemplo, observe a Figura 2.17. Se escolhermos $S = \{1, 4\}$. Temos que $H(S) = \{1, 2, 4, 5\}$. Portanto, S não é *m-convexo*. No entanto, se escolhermos $S' = \{1, 2, 4, 5\}$. Temos que $H(S') = S'$. Portanto, S' é *m-convexo*.

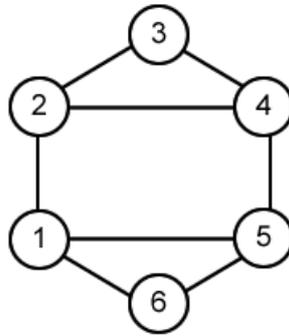


Figura 2.17: Convexidade monofônica em grafo

Definimos agora a convexidade de Steiner. Seja $G = (V(G), E(G))$ um grafo conexo e W um subconjunto de $V(G)$. Um subgrafo conexo T de G com número mínimo de arestas, tal que $W \subseteq V(T)$ é chamado uma *árvore de Steiner* de G . Um subconjunto $S \subseteq V(G)$ é *st-convexo* se, para todo $A \subseteq S$, todos os vértices em toda árvore de *Steiner* de A pertencem a S . A coleção de todos os conjuntos *st-convexos* de $V(G)$ formam a *convexidade de Steiner* em G .

Por exemplo, observe a Figura 2.18. Se escolhermos $S = \{2, 3, 4, 5\}$. Temos que S é *st-convexo*. No entanto, se escolhermos $S' = \{2, 3, 4, 6\}$. Temos S' não é *st-convexo*. Observe o subconjunto $A = \{2, 6\}$, as árvores de Steiner de A são: $T' = \{2, 1, 6\}$ e $T'' = \{2, 5, 6\}$. No entanto, os vértices 1 e 6 não pertencem a S' .

Por fim, definimos a convexidade P_3 . Seja $G = (V(G), E(G))$ um grafo conexo. Um conjunto $C \subseteq V(G)$ é *P_3 -convexo* se para todo caminho uvw em G

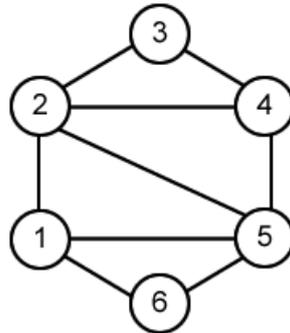


Figura 2.18: Convexidade Steiner em grafo

com $u, w \in C$, o vértice v também pertencer a C . Equivalentemente, o conjunto C é P_3 -convexo se, e somente se, nenhum vértice em $V(G) \setminus C$ tem pelo menos dois vizinhos em C . A coleção de todos os conjuntos P_3 -convexos de $V(G)$ formam a *convexidade P_3* em G .

Por exemplo, observe a Figura 2.19. Se escolhermos $S = \{2, 4, 7\}$. Temos que $H(S) = \{2, 4, 7, 6, 8, 5\}$. Portanto, S não é P_3 -convexo. No entanto, se escolhermos $S' = \{1, 5, 9\}$. Temos que $H(S') = S'$. Portanto, S' é P_3 -convexo.

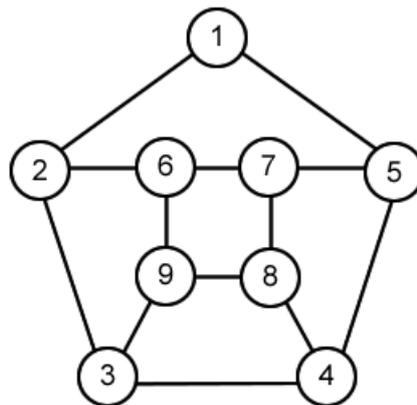


Figura 2.19: Convexidade P_3 em grafo

2.5 Número de Helly

Segundo [17], a propriedade de Helly tem aplicações em várias áreas. Em otimização, por exemplo, tem sido aplicada em problemas de localização [19] e programação linear [1]. Em ciência da computação, tem sido utilizada em teoria da semântica [4], codificação [3], biologia computacional [33], banco de dados [20], processamento de imagens [7] e teoria dos grafos [17].

Uma família F de conjuntos tem a propriedade de Helly quando toda subfamília de F formada por pares de conjuntos que se interceptam, tem um elemento em comum.

Teorema 2.5.1 (HELLY, 1923)[14] *Seja K uma família finita de pelo menos $(n+1)$ conjuntos convexos no espaço finito \mathbb{R}^n . Então, se cada $(n+1)$ membros de K tem um ponto em comum, há um ponto comum a todos os membros de K .*

Como exemplo, observe a Figura 2.20. Na Figura 2.20a, as esferas possuem a propriedade de Helly, pois são 2-intersectantes e possuem um ponto em comum. Já na Figura 2.20b, as esferas não possuem a propriedade de Helly, pois embora sejam 2-intersectantes, elas não possuem um ponto em comum.

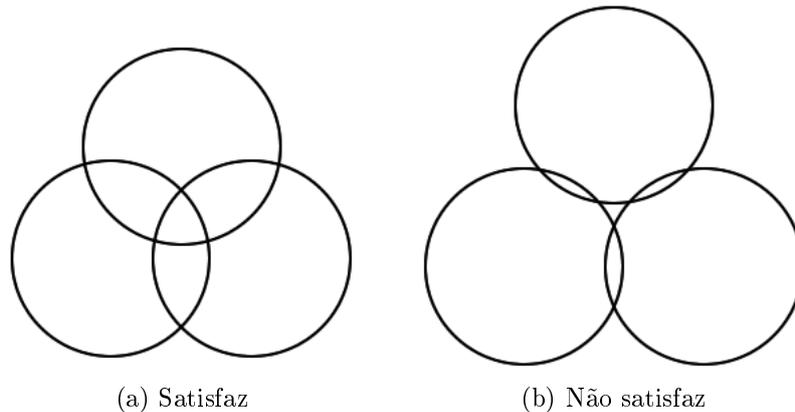


Figura 2.20: Propriedade de Helly

O *núcleo* de uma família é a interseção total de seus conjuntos. Uma família F de conjuntos é *k-intersectante* se toda subfamília de F com k conjuntos tem um

núcleo não-vazio. Uma família F de conjuntos é k -Helly se toda subfamília de F que é k -intersectante tem o núcleo não-vazio. Se uma família F é k -Helly, então F é $(k + 1)$ -Helly, para $k < n$, onde n é o total de subfamílias de F . O *número de Helly* de uma família F é o menor inteiro k tal que F é k -Helly.

Um *hipergrafo* H é um par ordenado $(V(H), E(H))$, onde $V(H) = \{v_1, \dots, v_n\}$ é um conjunto finito não-vazio de vértices e $E(H) = \{E_1, \dots, E_m\}$ é uma família de subconjuntos de $V(H)$. Veja na Figura 2.21, um exemplo de hipergrafo formado pelo conjunto de vértices $V(H) = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ e pelo conjunto de hiperarestas $E(H) = \{\{1, 2, 3\}, \{3, 4, 5\}, \{2, 6, 7\}, \{4, 9\}, \{7, 8, 9\}\}$.

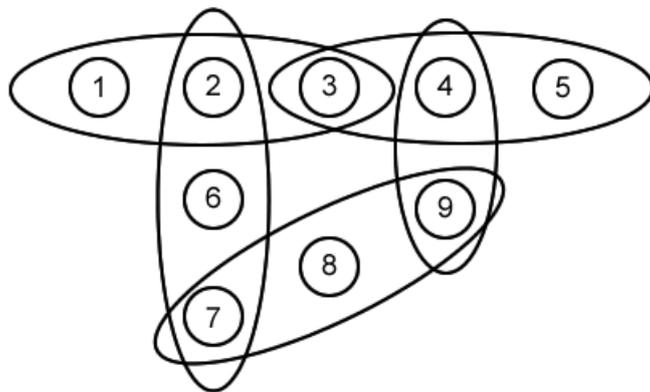


Figura 2.21: Hipergrafo

Um hipergrafo H é p -Helly se todo hipergrafo parcial p -intersectante $H' \subseteq H$ tem interseção não-vazia.

Como exemplo, observe a Figura 2.22. Na Figura 2.22a, o hipergrafo apresentado é 2-Helly, pois todos os seus hipergrafos parciais 2-intersectantes possuem interseção não-vazia. Já na Figura 2.22b, o hipergrafo apresentado não é 2-Helly, pois os seus hipergrafos parciais 2-intersectantes possuem interseção não-vazia.

Um hipergrafo H é r -uniforme se toda hiperaresta de H contém exatamente r vértices. Assim, um grafo é um hipergrafo 2-uniforme.

Vamos considerar a seguinte generalização da propriedade de Helly, apresentada no Teorema 2.5.2, que caracteriza hipergrafos p -Helly.

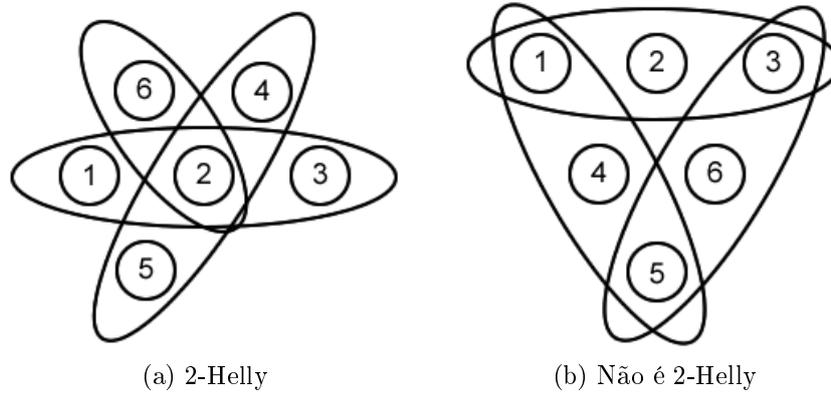


Figura 2.22: Hipergrafo p -Helly

Teorema 2.5.2 (BERGE e DUCHET, 1975)[5] *Um hipergrafo H é p -Helly se, e somente se, para todo $(p+1)$ -subconjunto S de $V(H)$, o hipergrafo parcial formado pelas hiperarestas que contêm pelo menos p elementos de S tem o núcleo não vazio.*

O problema HIPERGRAFO p -HELLY apresentado a seguir é polinomial para p fixo[5]. No entanto, quando p é variável é provado em [16] que o problema torna-se NP-difícil.

Problema: HIPERGRAFO p -HELLY

Instância: Um hipergrafo H e um inteiro p .

Questão: H é p -Helly?

O Teorema 2.5.3 a seguir apresenta uma caracterização para convexidades p -Helly. Lembrando que nesta dissertação, consideramos grafos simples, conexos, não-direcionados e não-ponderados.

Teorema 2.5.3 [38] *Um grafo G é p -Helly se, e somente se, para qualquer conjunto $S \subseteq V(G)$, tal que $|S| = p + 1$,*

$$\bigcap_{v_i \in S} H(S - \{v_i\}) \neq \emptyset.$$

O número de Helly na convexidade \mathcal{C} de um grafo G é o número de Helly da família definida pelos conjuntos convexos da convexidade \mathcal{C} em G .

Na convexidade monofônica, foi provado que o número de Helly é igual à cardinalidade da clique máxima do grafo [24]. Será que vale o mesmo para a convexidade geodética? Vamos ver que nem sempre vale esta igualdade. Observe, por exemplo, o grafo na Figura 2.23. A clique máxima do grafo é igual a 2, porém o seu número de Helly é igual a 4.

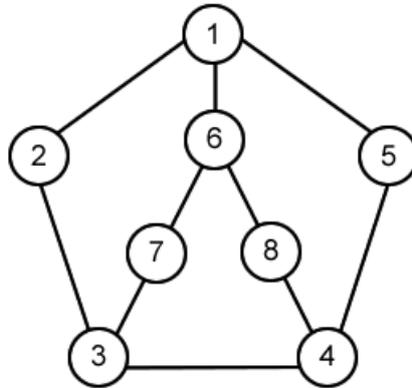


Figura 2.23: Grafo com número de Helly maior do que a clique máxima

O problema GRAFO p -HELLY NA CONVEXIDADE GEODÉTICA apresentado a seguir é polinomial para p fixo [38]. No entanto, quando p é variável, o problema torna-se NP-completo [37].

Problema: GRAFO p -HELLY NA CONVEXIDADE GEODÉTICA

Instância: Um grafo G e um inteiro p .

Questão: G é p -Helly na convexidade geodética?

Devido ao Teorema 2.5.3, vai existir um certificado com k vértices para a resposta *NÃO* ao problema GRAFO p -HELLY NA CONVEXIDADE GEODÉTICA. Dado um grafo G com número de Helly igual a k , um *certificado de que G não é $(k - 1)$ -Helly* seria apresentar um conjunto $S \subseteq V(G)$ com k vértices, tal que

$$\bigcap_{v_i \in S} H(S - \{v_i\}) = \emptyset.$$

Daqui em diante, a menos que digamos o contrário, utilizaremos neste trabalho o termo "número de Helly" para nos referirmos ao número de Helly na convexidade geodética.

Por exemplo, o número de Helly do grafo de *Petersen* é 4. Observe a Figura 2.24. Quando escolhermos o conjunto $S = \{0, 1, 3, 8\}$, temos os seguintes fechos convexos para $S - \{v_i\}$, para qualquer $v_i \in S$:

- $H(S - \{0\}) = \{1, 2, 3, 6, 8\}$
- $H(S - \{1\}) = \{0, 3, 5, 8, 9\}$
- $H(S - \{3\}) = \{0, 1, 5, 6, 8\}$
- $H(S - \{8\}) = \{0, 1, 2, 3, 9\}$

Podemos ver que a interseção dos fechos convexos apresentados é vazia. Assim, concluímos que o grafo de *Petersen* não é 3-Helly, e um certificado disto seria o conjunto $S = \{0, 1, 3, 8\}$. Ao testarmos todos os conjuntos $S \subseteq V(G)$, tal que $|S| = 5$, verificamos que $\bigcap_{v_i \in S} H(S - \{v_i\}) \neq \emptyset$. Então concluímos que o número de Helly do grafo de *Petersen* é 4.

Exibimos a seguir algumas observações simples sobre o número de Helly em grafos. Seja G um grafo conexo com n vértices, $h(G)$ o número de Helly do grafo G e $\omega(G)$ a cardinalidade da clique máxima de G .

Observação 2.5.4 $h(G) \geq \omega(G)$.

Prova Seja S a clique máxima de G , tal que $|S| = k$. Vamos definir $S_i = S - \{v_i\}$, para todo $v_i \in S$. Uma vez que $d(u, v) = 1$, para todo $u, v \in S_i$, temos que $H(S_i) = S_i$. Uma vez que $v_i \in H(S_j)$, para $i \neq j$, mas $v_i \notin H(S_i)$, temos que S é k -intersectante com núcleo vazio. Portanto, G não é $(k - 1)$ -Helly. Logo, $h(G) > \omega(G) - 1$, ou seja, $h(G) \geq \omega(G)$. ■

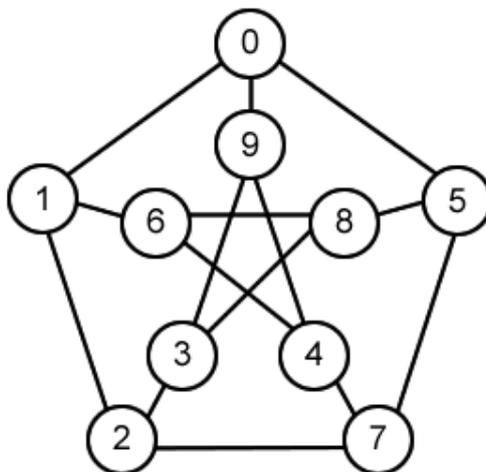


Figura 2.24: Grafo de Petersen

Observação 2.5.5 $h(G) = n$ se, e somente se, G é completo.

Definição Um grafo conexo G é *quase completo* se G possui um subgrafo induzido ao K_{n-1} e um vértice v tal que $d(v) < n - 1$.

Podemos ver na Figura 2.25, todos os grafos conexos quase completos com 5 vértices.

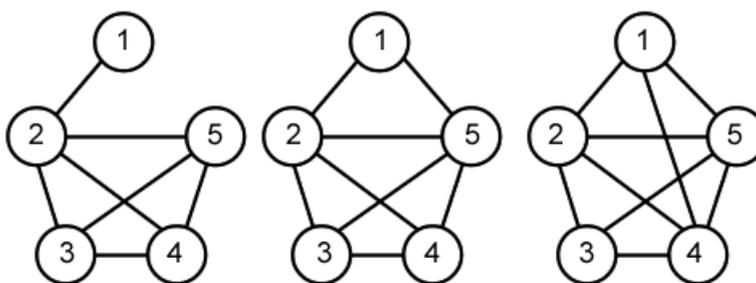


Figura 2.25: Grafos quase completos

Observação 2.5.6 $h(G) = n - 1$ se, e somente se, G é quase completo.

Observação 2.5.7 Se G é uma árvore, então $h(G) = 2$.

Observação 2.5.8 *Seja G um ciclo C_k , para $k \geq 3$. Se $k = 4$, então $h(G) = 2$.
Caso contrário, $h(G) = 3$.*

3 CONTRIBUIÇÕES TEÓRICAS

Neste capítulo, apresentamos nossos resultados teóricos que são sumarizados sucintamente a seguir. Encontramos um limite superior para o número de Helly em grafos bipartidos e apresentamos uma família que mostra que este limite é justo. Encontramos limites superiores para o número de Helly em grafos conexos pequenos. Calculamos o número de Helly para as seguintes classes de grafos: sol, roda e leque. Formulamos algumas proposições relacionando algumas propriedades dos grafos com o seu número de Helly.

3.1 Grafos Bipartidos

Seja G um grafo conexo com n vértices e $h(G)$ o número de Helly do grafo G . Provamos a seguir que se $S \subset V(G)$ é um certificado de que G não é k -Helly, então $G[S]$ é uma união de cliques.

Lema 3.1.1 *Seja $G = (V(G), E(G))$ um grafo. Se $S \subset V(G)$ é um certificado de que G não é k -Helly, então $G[S]$ é uma união de cliques.*

Prova Por absurdo, suponha o contrário, isto é, $G[S]$ não é união de cliques. Então existem dois vértices u e v que pertencem a S tal que $uv \notin E(G)$ e existe um vértice $w \in S$, tal que $w \in N(u) \cap N(v)$. Então $d(u, v) = 2$, logo $w \in I(u, v)$. Portanto, w está no núcleo de $\bigcap_{v_i \in S} H(S - \{v_i\})$, uma contradição. ■

Seja G um grafo bipartido. O resultado a seguir mostra que se $S \subset V(G)$ é um certificado de que G não é k -Helly, então há um vértice exclusivo no intervalo de qualquer par de vértices não-adjacentes de S .

Afirmção 3.1.2 *Seja G um grafo bipartido. Seja $S \subset V(G)$ um certificado de que G não é k -Helly. Sejam v_i e v_j dois vértices de S . Se $(v_i, v_j) \notin E(G)$, então $I(v_i, v_j)$ contém um vértice que não está em nenhum outro intervalo entre os vértices de S .*

Prova Suponha o contrário. Seja $v_i u_1 u_2 \dots u_d v_j$ um caminho mínimo entre v_i e v_j . Então u_d pertence a algum outro intervalo $I(v_p, v_q)$.

Se $\{i, j\} \cap \{p, q\} = \emptyset$, observe que u_d pertence ao fecho convexo de todos os conjuntos $S - \{v_i\}$, para $v_i \in S$, ou seja, u_d está no núcleo desses $k + 1$ conjuntos, o que contradiz a hipótese. Assim, temos $p = i$ e $q \notin \{i, j\}$. Veja na Figura 3.1 como os vértices em questão estão dispostos.

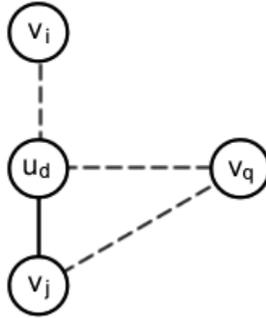


Figura 3.1: Exemplo da Afirmação 3.1.2

Uma vez que $u_d \in I(v_i, v_q)$, temos que u_d está em uma (v_i, v_q) -geodésica. Uma vez que $u_d \notin I(v_j, v_q)$, pois caso contrário estaria no núcleo dos $k + 1$ conjuntos, vale que $u_d \notin N(v_q)$. Além disso, $(v_j, v_q) \notin E(G)$, pois G é bipartido, portanto livre de triângulos.

Sabemos que qualquer caminho $P(v_j, v_q)$, passando por u_d , não é mínimo, ou seja, cada (v_j, v_q) -geodésica não contém u_d , então $d(v_j, v_q) \leq d(u_d, v_q)$. Se valer $d(v_j, v_q) = d(u_d, v_q)$, teremos um ciclo ímpar, uma contradição, pois G é bipartido.

Portanto $d(v_j, v_q) < d(u_d, v_q)$. Então temos o seguinte caminho $v_i \dots u_d v_j \dots v_q$ que é menor ou igual a $d(v_i, v_q)$. Como v_j não está no intervalo de v_i, v_q , existe um caminho entre v_i e v_q menor que $d(v_i, v_q)$, contradição. ■

Estabelecemos a seguir, um limite superior para o número de Helly em grafos bipartidos. Provamos que o número de Helly em grafos bipartidos é menor ou igual a $\left\lfloor \frac{-1 + \sqrt{1 + 8n}}{2} \right\rfloor$.

Teorema 3.1.3 *Seja $G = (V(G), E(G))$ um grafo bipartido com n vértices e $h(G)$ o número de Helly do grafo G . Temos que $h(G) \leq \left\lfloor \frac{-1+\sqrt{1+8n}}{2} \right\rfloor$.*

Prova Vamos definir $k = \left\lfloor \frac{-1+\sqrt{1+8n}}{2} \right\rfloor$. Suponha por contradição que G é um grafo bipartido tal que $h(G) > k$. Então existe um conjunto $S = \{v_1, \dots, v_k\}$ tal que os fechados convexos de $S - \{v_i\}$ são $(k-1)$ -intersectantes com núcleo vazio, para todo $v_i \in S$.

Pela Afirmação 3.1.2, o número de vértices de G é pelo menos o tamanho de S mais o número de pares de S que não são adjacentes em G .

Uma vez que G é livre de K_3 , pelo Lema 3.1.1, $G[S]$ tem no máximo $k/2$ arestas. O que implica que o número de pares de S formados pelos vértices não-adjacentes é pelo menos $\frac{k^2-k}{2} - \frac{k}{2} = \frac{k^2}{2} - k$. Somando este valor a $|S|$, concluímos que $n \geq k^2/2$. Uma vez que $k > \frac{-1+\sqrt{1+8n}}{2} \geq 1$, temos

$$\begin{aligned} n &\geq \frac{k^2}{2} \geq \frac{\left(1 + \left\lfloor \frac{-1+\sqrt{1+8n}}{2} \right\rfloor\right)^2}{2} \\ &= \frac{\left(\left\lfloor \frac{1+\sqrt{1+8n}}{2} \right\rfloor\right)^2}{2} = \frac{\left(\frac{\sqrt{1+8n}}{2}\right)^2}{2} \\ &= \frac{1+8n}{8} \end{aligned}$$

Uma vez que este número é maior do que n , para todo inteiro $n \geq 1$, temos uma contradição. Então o teorema vale. ■

Apresentamos a seguir uma família de grafos que mostra que é justo o limite superior encontrado para o número de Helly em grafos bipartidos.

Defina uma família de grafos $\mathcal{G} = \{G_1, G_2, \dots\}$. Para todo $k \geq 1$, defina $G_k \in \mathcal{G}$ como $V(G_k) = \{v_1, v_2, \dots, v_k\} \cup \{v_{ij} : 1 \leq i < j \leq k\}$, e $E(G_k) = \{v_i v_{ij} \text{ e } v_j v_{ij} : 1 \leq i < j \leq k\}$. O grafo G_k tem $\frac{k(k+1)}{2}$ vértices. Na Figura 3.2, temos um exemplo desta família de grafos.

Provamos a seguir que o número de Helly do grafo G_k é igual a k .

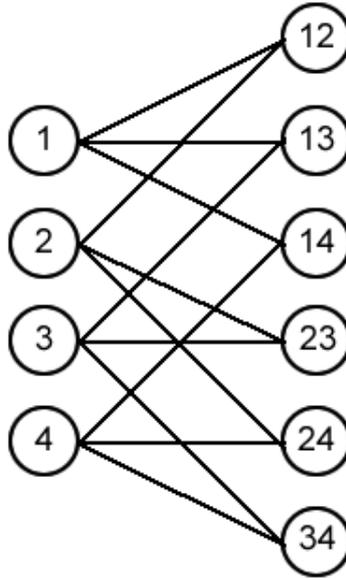


Figura 3.2: Grafo G_4

Proposição 3.1.4 *Seja $G_k = (V(G_k), E(G_k))$ um grafo da família \mathcal{G} e $h(G_k)$ o número de Helly do grafo G_k , tal que G_k tem n vértices. Temos que $h(G_k) = k$.*

Prova Primeiro, vamos provar que G_k não é $(k - 1)$ -Helly. Considere o conjunto $S = \{v_1, \dots, v_k\}$. Sejam u e v dois vértices de S , por definição $(u, v) \notin E(G)$. Seja w um vértice tal que $w \notin S$, $(u, w) \in E(G)$ e $(v, w) \in E(G)$. Então $w \in I(u, v)$ e o caminho $P = \{u, w, v\}$ é o menor (u, v) -caminho. Por definição w é adjacente apenas aos vértices u e v . Então, de modo geral, qualquer par de vértices u e v que pertencem a S não são adjacentes e o $I(u, v)$ contém um vértice que não aparece em nenhum outro intervalo.

Vamos definir $S_i = S - \{v_i\}$. O conjunto S_j contém v_i , para $j \neq i$ e $1 \leq j \leq k$. O que implica que os conjuntos $H(S_i)$ são $(k - 1)$ -intersectantes.

Nas Figuras 3.3, 3.6, 3.5 e 3.4, denotamos o vértice v_i por i e o vértice v_{ij} por ij . Observe os fechos convexos listados a seguir.

- $H(v_a, v_b) = \{v_a, v_b, v_{ab}\}$. Veja a Figura 3.3.

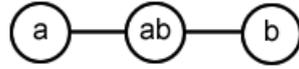


Figura 3.3: Fecho convexo dos vértices v_a, v_b

- $H(v_a, v_{bc}) = \{v_a, v_{bc}, v_b, v_c, v_{ab}, v_{ac}\}$. Veja a Figura 3.4.

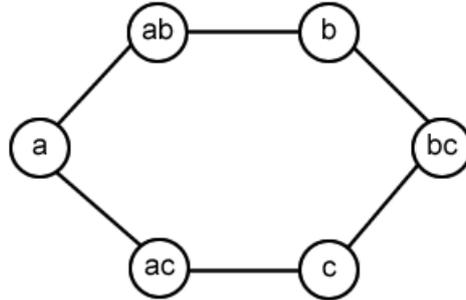


Figura 3.4: Fecho convexo dos vértices v_a, v_{bc}

- $H(v_{ab}, v_{bd}) = \{v_{ab}, v_{bd}, v_b\}$. Veja a Figura 3.5.

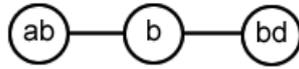


Figura 3.5: Fecho convexo dos vértices v_{ab}, v_{bd}

- $H(v_{ab}, v_{cd}) = \{v_{ab}, v_{cd}, v_a, v_b, v_c, v_d, v_{ac}, v_{ad}, v_{bc}, v_{bd}\}$. Veja a Figura 3.6.

Então $v_a \notin H(S_a)$ e $v_{ab} \notin H(S_a)$.

Além disso, $H(S_i) = S_i \cup \{v_{jk} : v_j, v_k \in S_i\}$. Portanto, o núcleo dos fechos convexos é vazio.

Logo, G_k não é $(k - 1)$ -Helly.

Agora, vamos provar que G_k é k -Helly. Suponha o contrário. Então existe um conjunto S com $k + 1$ vértices, tal que os fechos convexos de $S - \{v_i\}$ são k -intersectantes com núcleo vazio, para todo $v_i \in S$.

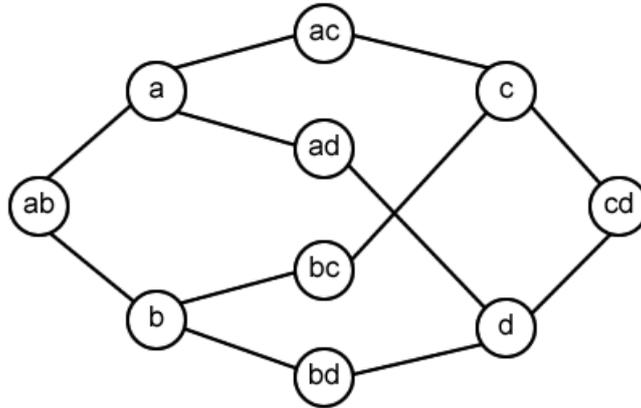


Figura 3.6: Fecho convexo dos vértices v_{ab}, v_{cd}

Sejam X e Y as partições de G_k . Sabemos que $|X| = k$.

Para construir o conjunto S , devemos escolher p vértices de X e q vértices de Y de modo que $p + q = k + 1$.

Vamos chamar de P o conjunto formado por estes p vértices e vamos chamar de Q o conjunto formado por estes q vértices.

Pelo Lema 3.1.1, sabemos que $G[S]$ não tem subgrafo induzido ao P_3 . Então vale o seguinte:

- Se os vértices v_i e v_j pertencem a P , então v_{ij} não pertence a Q .
- Se o vértice v_{ij} pertence a Q , então apenas v_i ou apenas v_j pertence a P , não ambos.

Desta forma, podemos dizer que $\Delta(S) = 1$.

Ao escolhermos o vértice $v_{ij} \in Y$ para pertencer a S , ficamos proibidos de escolher um dos vizinhos de v_{ij} em X para pertencer a S . Então $X \setminus P$ tem pelo menos q vértices.

Como $|X \setminus P| \geq q$ e $|P| = |X| - |X \setminus P|$, temos que $p \leq k - q$.

Como $k = p + q - 1$, temos que $p \leq p + q - 1 - q$, ou seja, $p \leq p - 1$. Chegamos a um absurdo. Logo, o teorema é verdadeiro. ■

Dado um grafo G_k com n vértices, provamos a seguir que $k = \frac{-1+\sqrt{1+8n}}{2}$.

Lema 3.1.5 *Seja $G_k = (V(G_k), E(G_k))$ um grafo da família \mathcal{G} , tal que G_k tem n vértices. Temos que $k = \frac{-1+\sqrt{1+8n}}{2}$.*

Prova Pela definição de G_k , $n = \frac{k(k+1)}{2}$. Então vamos desenvolver a expressão $k = \frac{-1+\sqrt{1+8n}}{2}$,

$$\begin{aligned} \frac{-1 + \sqrt{1 + 8n}}{2} &= \frac{-1 + \sqrt{1 + 8\left(\frac{k^2+k}{2}\right)}}{2} \\ &= \frac{-1 + \sqrt{1 + 4k^2 + 4k}}{2} \\ &= \frac{-1 + \sqrt{(2k + 1)^2}}{2} \\ &= \frac{-1 + 2k + 1}{2} \\ &= k \end{aligned}$$

Logo, o lema é verdadeiro. ■

3.2 Algumas classes

Seja G um grafo conexo com n vértices, $h(G)$ o número de Helly do grafo G e $\omega(G)$ a cardinalidade da clique máxima de G .

Chegamos aos seguintes limites para o número de Helly nos grafos com no máximo 9 vértices:

- Se $2 < n \leq 4$, então $h(G) = \omega(G)$.
- Se $5 \leq n \leq 7$ e $\omega(G) = 2$, então $h(G) \leq \omega(G) + 1$.
- Se $7 \leq n \leq 9$ e $\omega(G) = 3$, então $h(G) \leq \omega(G) + 1$.
- Se $7 < n \leq 9$ e $\omega(G) = 2$, então $h(G) \leq \omega(G) + 2$.

- Se $n = 9$ e $\omega(G) = 4$, então $h(G) \leq \omega(G) + 1$.

Provamos a seguir que todo grafo que possui uma clique máxima de tamanho dois, também possui um subgrafo induzido isomorfo ao $K_{1,\Delta(G)}$.

Proposição 3.2.1 *Seja G um grafo conexo, $\omega(G)$ a cardinalidade da clique máxima de G , $\Delta(G)$ o grau máximo de G e $t(G)$ o menor inteiro t tal que G é livre de $K_{1,t}$. Se $\omega(G) = 2$, então $t(G) = \Delta(G) + 1$, ou seja, G possui um subgrafo induzido isomorfo ao $K_{1,\Delta(G)}$.*

Prova Suponha por absurdo que G não possui um subgrafo induzido isomorfo ao $K_{1,\Delta}$. Seja v um vértice de G que possui grau máximo Δ . Sejam x e y vértices adjacentes que são vizinhos de v . Então v , x e y formam uma clique de tamanho 3, um absurdo. ■

Como exemplo, observe o grafo G na Figura 3.7a. Podemos ver que $\omega(G) = 2$ e $\Delta(G) = 4$. Nas Figuras 3.7b e 3.7c, vemos subgrafos de G que são subgrafos induzidos isomorfos ao $K_{1,3}$.

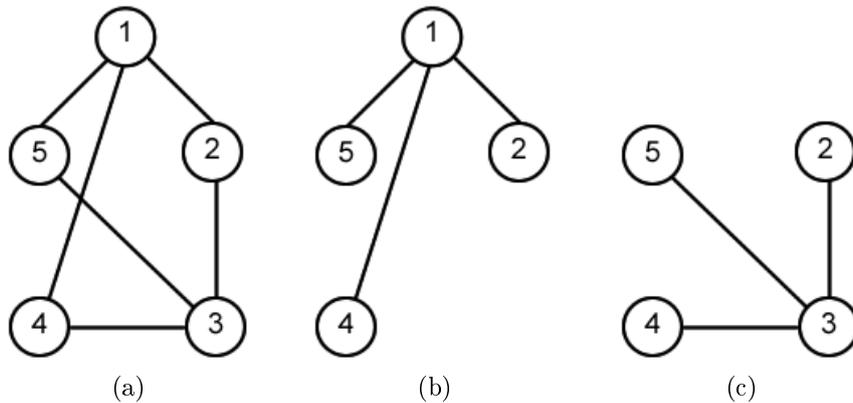


Figura 3.7: Exemplo da Proposição 3.2.1

Conjecturamos a seguir que se um grafo tem a clique máxima maior ou igual a $\lceil n/2 \rceil$, então o seu número de Helly é igual ao tamanho da clique máxima.

Conjectura 3.2.2 *Seja G um grafo conexo, $h(G)$ o número de Helly do grafo G e $\omega(G)$ a cardinalidade da clique máxima de G . Se $\omega(G) \geq \lceil n/2 \rceil$, então $h(G) = \omega(G)$.*

Como exemplos, observe os grafos nas Figuras 3.8a, 3.8b e 3.8c, ambos com 8 vértices, eles têm o número de Helly igual a cardinalidade da clique máxima, respectivamente, 4, 5 e 6.

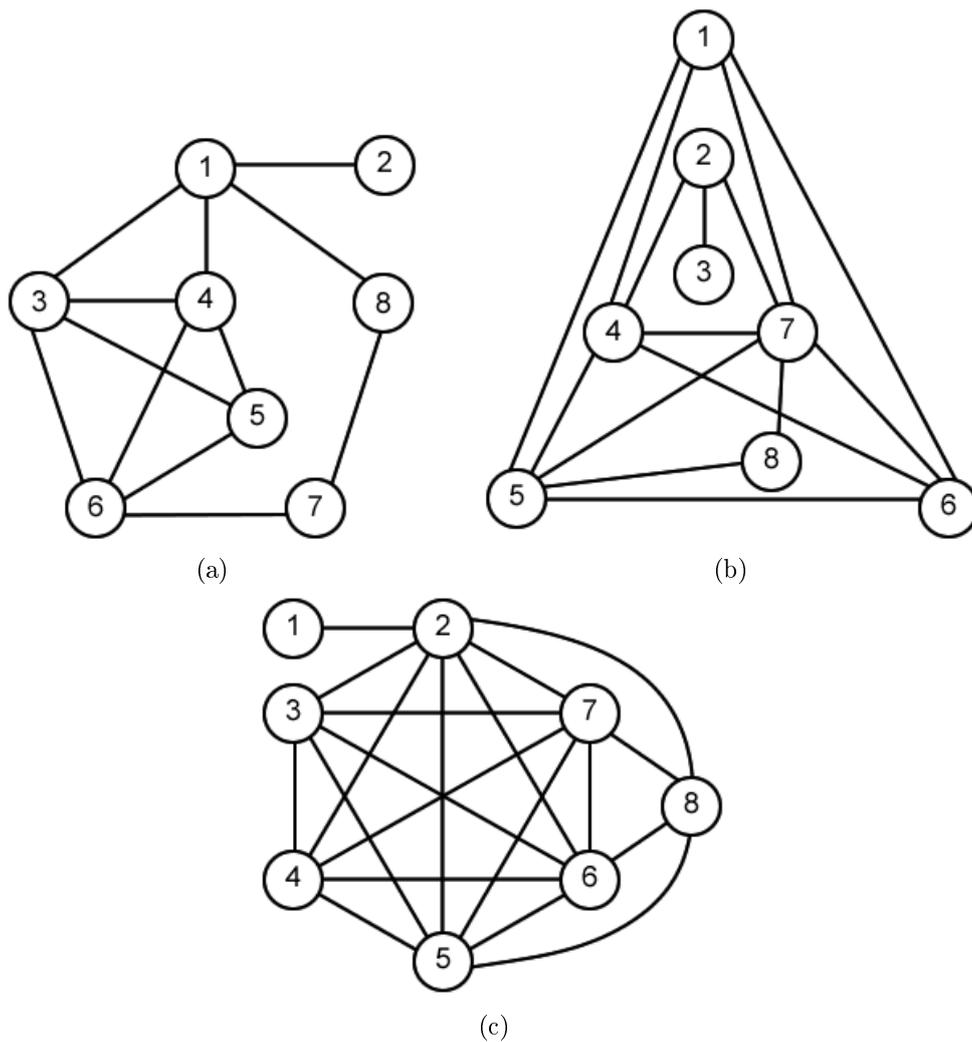


Figura 3.8: Exemplos da Conjectura 3.2.2

Nas Proposições 3.2.3 a 3.2.6 apresentadas a seguir, relacionamos algumas propriedades dos grafos (conexos com $3 \leq n \leq 9$ e bipartidos com $3 \leq n \leq 12$) com

o seu número de Helly.

Proposição 3.2.3 *Seja G um grafo com n vértices, $h(G)$ o número de Helly do grafo G e $\text{diam}(G)$ o diâmetro de G . Se G é conexo com $3 \leq n \leq 9$ ou bipartido com $3 \leq n \leq 12$, então $h(G) \leq n - \text{diam}(G) + 1$.*

Proposição 3.2.4 *Seja G um grafo conexo com n vértices, $h(G)$ o número de Helly do grafo G e $t(G)$ o menor inteiro t tal que G é livre de $K_{1,t}$. Se G é conexo com $3 \leq n \leq 9$ ou bipartido com $3 \leq n \leq 12$, então $h(G) \leq n - t(G) + 2$.*

Proposição 3.2.5 *Seja G um grafo, tal que G é conexo com $3 \leq n \leq 9$ ou bipartido com $3 \leq n \leq 12$. Seja $h(G)$ o número de Helly do grafo G , $\omega(G)$ a cardinalidade da clique máxima de G e $t(G)$ o menor inteiro t tal que G é livre de $K_{1,t}$. Se $h(G) > \omega(G) = t(G)$ então $h(G) = \omega(G) + 1$.*

Proposição 3.2.6 *Seja G um grafo, tal que G é conexo com $3 \leq n \leq 9$ ou bipartido com $3 \leq n \leq 12$. Seja $h(G)$ o número de Helly do grafo G , $\omega(G)$ a cardinalidade da clique máxima de G , $\Delta(G)$ o grau máximo de G e $t(G)$ o menor inteiro t tal que G é livre de $K_{1,t}$. Se $t(G) = \Delta(G)$ e $h(G) > \omega(G)$, então $h(G) = 4$, exceto o grafo da Figura 3.9.*

Observe que o grafo da Figura 3.9 apresenta $t(G) = \Delta(G) = \omega(G) = 4$ e $h(G) = 5$. Um certificado de que este grafo não é 4-Helly seria o seguinte conjunto $S = \{0, 1, 4, 5, 7\}$ e os fechos convexos de $S - \{v_i\}$, para $v_i \in S$, listados a seguir:

- $H(S - \{0\}) = \{1, 2, 4, 5, 6, 7, 8\}$
- $H(S - \{1\}) = \{0, 2, 3, 4, 5, 7, 8\}$
- $H(S - \{4\}) = \{0, 1, 2, 3, 5, 6, 7\}$
- $H(S - \{5\}) = \{0, 1, 3, 4, 6, 7, 8\}$
- $H(S - \{7\}) = \{0, 1, 4, 5, 6, 8\}$

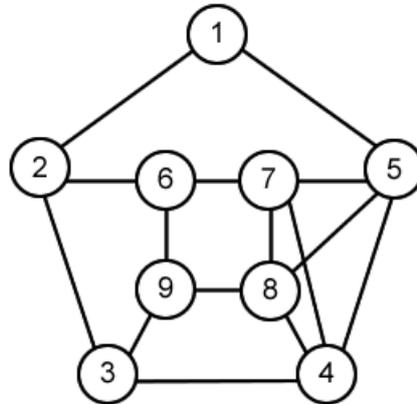


Figura 3.9: Exceção da Proposição 3.2.6

Como exemplo para as Proposições 3.2.3 a 3.2.6, temos o grafo na Figura 3.10. Observe que este grafo, apresenta as seguintes propriedades: $|V(G)| = 8$, $|E(G)| = 11$, $h(G) = 4$, $\omega(G) = 3$, $t(G) = 4$, $diam(G) = 3$, $\Delta(G) = 4$ e $\delta(G) = 2$. Um certificado de que este grafo não é 3-Helly seria o conjunto $S = \{0, 1, 6, 7\}$ e os fechos convexos de $S - \{v_i\}$, para $v_i \in S$, listados a seguir:

- $H(S - \{0\}) = \{1, 3, 5, 6, 7\}$
- $H(S - \{1\}) = \{0, 3, 4, 6, 7\}$
- $H(S - \{6\}) = \{0, 1, 5, 7\}$
- $H(S - \{7\}) = \{0, 1, 4, 5, 6\}$

Discursamos a seguir sobre algumas classes simples de grafos que foram observadas em nossos experimentos e apresentamos conjecturas sobre o número de Helly para as mesmas.

O *grafo leque*, denotado por k -leque, é um grafo formado a partir do caminho P_{k+1} (isto é, um caminho de tamanho k), adicionando um vértice que é adjacente a todos os vértices do caminho. O grafo k -leque tem $n = k + 2$ vértices. Na Figura 3.11, temos um exemplo deste grafo.

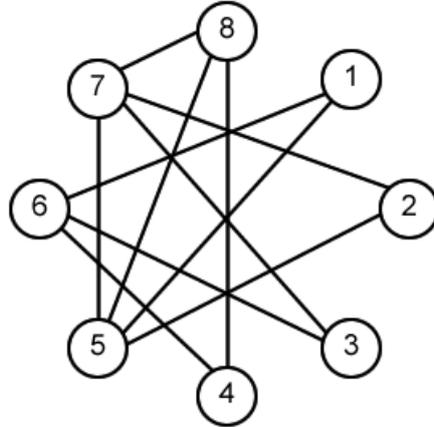


Figura 3.10: Exemplo para as Proposições 3.2.3, 3.2.4, 3.2.5 e 3.2.6

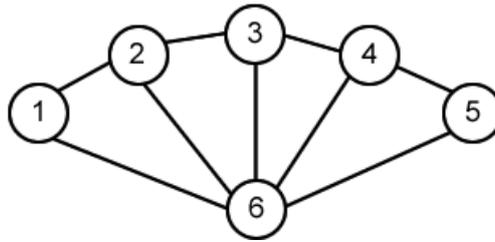


Figura 3.11: Grafo 4-leque

Proposição 3.2.7 *Se G é um grafo leque, então $h(G) = 3$.*

Prova Sabemos pela Observação 2.5.4, que $h(G) \geq 3$. Logo, G não é 2-Helly.

Então, vamos provar que G é 3-Helly. Seja G um grafo leque formado a partir do caminho P , adicionando um vértice u que é adjacente a todos os vértices de P . Como u é universal, sabemos que u pertence a todo $I(x, y)$, tal que $(x, y) \notin E(G)$.

Seja S um subconjunto de $V(G)$, tal que $|S| = 4$. Como todo vértice v_i de $V(P)$ tem no máximo 2 vizinhos, ao escolhermos os vértices para formar o conjunto S , sempre haverá pelo menos dois vértices em S que não são vizinhos, exceto se G for o K_3 .

Agora, consideramos dois casos.

Caso 1: $u \in S$. Então haverá um par de vértices não adjacentes em S , o que obrigará u a pertencer ao intervalo deste par.

Caso 2: $u \notin S$. Então haverão dois pares de vértices não adjacentes em S , então pelo menos um desses pares vai estar em $S - v_i$, para todo $v_i \in S$, obrigando u a pertencer ao intervalo deste par.

Logo, $u \in H(S - \{v_i\})$, para $v_i \in S$ e todo $S \subseteq V(G)$. Portanto, todo $S \subseteq V(G)$ é 3-intersectante com núcleo não-vazio, tal que $|S| = 4$. Logo, G é 3-Helly. ■

O *grafo sol*, denotado por S_k , é um grafo com $2k$ vértices ($k \geq 3$) cujo conjunto de vértices pode ser particionado em $W = \{w_1, \dots, w_k\}$ e $U = \{u_1, \dots, u_k\}$, tal que W forma um conjunto independente, U forma um ciclo e u_i é adjacente a w_j se, e somente se, $i = j$ ou $i = j + 1 \pmod{k}$. Na Figura 3.12, temos um exemplo deste grafo.

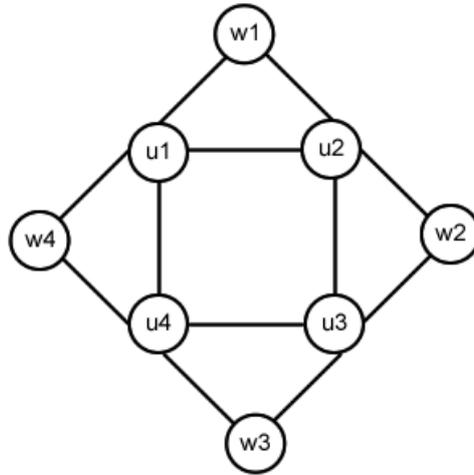


Figura 3.12: Grafo sol S_4

Proposição 3.2.8 *Se G é um grafo sol, então $h(G) = 3$.*

Prova Suponha por contradição que $h > 3$. Então existe um conjunto $S = \{v_1, v_2, v_3, v_4\}$ que é um certificado de que G não é 3-Helly.

Denote que $S_a = S \setminus \{v_a\}$, para $1 \leq i \leq 4$, que v_2 é u_i ou w_i , para algum i e que v_3 é u_j ou w_j , para algum j . Vamos mostrar que u_i ou u_{i+1} está no núcleo, gerando uma contradição.

Sem perda de generalidade, podemos considerar a ordem $\alpha = u_1 w_1 u_2 w_2 \dots u_n w_n$ dos vértices de G e que v_i aparece antes de v_j em α se $i < j$. Além disso, podemos assumir que o caminho mais curto de v_1 a v_3 contém u_i .

Observe que: $H(S_4)$ contém u_i e u_{i+1} , porque o caminho mais curto de v_1 a v_3 contém u_i .

Agora, consideramos dois casos.

Caso 1: o caminho mais curto de v_2 a v_4 contém u_j . $H(S_1)$ e $H(S_3)$ contêm u_{i+1} , porque o caminho mais curto de v_2 a v_4 contém u_j . $H(S_2)$ contém u_{i+1} , porque o caminho mais curto de v_1 a v_3 contém u_i . Então, u_{i+1} está no núcleo.

Caso 2: o caminho mais curto de v_2 a v_4 não contém u_j . $H(S_1)$ e $H(S_3)$ contêm u_i , porque o caminho mais curto de v_2 a v_4 não contém u_j . $H(S_2)$ contém u_i , porque o caminho mais curto de v_1 a v_3 contém u_i . Então, u_i está no núcleo, uma contradição. ■

O *grafo roda*, denotado por W_k , é um grafo formado a partir do ciclo C_k , adicionando um vértice e o tornando adjacente a todos os vértices do ciclo. O grafo W_k tem $n = k + 1$ vértices. Na Figura 3.13, temos um exemplo deste grafo.

Proposição 3.2.9 *Se G é um grafo roda, então $h(G) = 3$.*

Prova Sabemos pela Observação 2.5.4, que $h(G) \geq 3$. Logo, G não é 2-Helly.

Então, vamos provar que G é 3-Helly. Seja G um grafo roda formado a partir do ciclo P , adicionando um vértice u que é adjacente a todos os vértices de P . Como u é universal, sabemos que u pertence a todo $I(x, y)$, tal que $(x, y) \notin E(G)$.

Seja S um subconjunto de $V(G)$, tal que $|S| = 4$. Como todo vértice v_i de $V(P)$ tem exatamente 2 vizinhos, ao escolhermos os vértices para formar o conjunto S , sempre haverá pelo menos dois vértices em S que não são vizinhos.

Agora, consideramos dois casos.

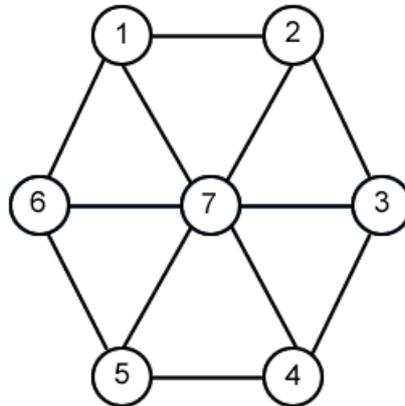


Figura 3.13: Grafo roda W_6

Caso 1: $u \in S$. Então haverá um par de vértices não adjacentes em S , o que obrigará u a pertencer ao intervalo deste par.

Caso 2: $u \notin S$. Então haverão dois pares de vértices não adjacentes em S , então pelo menos um desses pares vai estar em $S - v_i$, para todo $v_i \in S$, obrigando u a pertencer ao intervalo deste par.

Logo, $u \in H(S - \{v_i\})$, para $v_i \in S$ e todo $S \subseteq V(G)$. Portanto, todo $S \subseteq V(G)$ é 3-intersectante com núcleo não-vazio, tal que $|S| = 4$. Logo, G é 3-Helly. ■

4 IMPLEMENTAÇÃO E TESTES

Neste capítulo, iremos discursar sobre os experimentos realizados para investigar o número de Helly em grafos, sobre as implementações realizadas para o processo de experimentação e sobre a análise dos dados obtidos.

Discursamos sobre o método utilizado para a obtenção dos dados, sobre a preparação dos dados para o processo de experimentação e sobre os três programas que implementamos para realizar a experimentação. Os grafos analisados foram gerados por dois programas distintos. Um destes programas nós implementamos e o outro programa foi baixado da web, chama-se *nauty and Traces*. Os dados gerados por este último necessitaram ser tratados para tornarem-se entradas válidas para os nossos experimentos. O primeiro programa que implementamos tem por objetivo auxiliar a formatação dos arquivos dos grafos analisados. O segundo programa tem por objetivo produzir algumas das classes de grafos analisadas. O terceiro programa tem por objetivo principal encontrar o número de Helly k do grafo e um certificado que mostra que o grafo não é $(k - 1)$ -Helly, além de outras propriedades do grafo.

No processo de experimentação, analisamos todos os grafos conexos com no máximo 9 vértices, uma fração dos grafos conexos com 10 vértices; todos os grafos bipartidos com até 12 vértices; todos os grafos biconexos-bipartidos com até 12 vértices; os grafos k -cubo com até 64 vértices; os grafos tambor e sol com até 20 vértices; os grafos roda e leque com até 15 vértices; entre outras classes. Ao todo, analisamos 1.540.392 grafos.

Os testes foram realizados num computador com um processador Intel Core 2 Duo de 1.66GHz e memória de 2GB. O sistema operacional utilizado foi o Microsoft Windows XP. As implementações foram desenvolvidas na linguagem de programação Java (jdk 7). Os dados foram armazenados no banco de dados *Firebird 2.5*.

4.1 Implementação

Utilizamos para os nossos experimentos grafos conexos e não-isomorfos. Grande parcela desses grafos foi produzida por um programa do conjunto de programas *nauty and Traces*¹ que baixamos pronto da web. Este programa foi desenvolvido na linguagem de programação C, sua base teórica pode ser encontrada em [28, 29, 31]. Uma outra parcela dos grafos foi gerada por um programa que nós desenvolvemos. Mais adiante, iremos descrevê-lo.

Vamos descrever a seguir o processo de geração dos grafos através de *nauty and Traces*. Para gerar grafos não-isomorfos utilizamos o programa *geng* de *nauty and Traces*. Definido o número de vértices dos grafos que desejamos gerar, os grafos são gerados e salvos no mesmo arquivo. Os arquivos de saída podem ser de dois formatos: *graph6* e *sparse6*. Utilizamos o primeiro formato, que a seguir foi exportado para arquivo texto através do programa *showg* de *nauty and Traces*. O conteúdo deste arquivo é o nome de cada grafo, seu número de vértices e sua matriz de adjacências. Exibimos a seguir dois exemplos do processo de geração dos grafos através de *nauty and Traces*.

Exemplo 1: para gerar os grafos conexos não-isomorfos com 5 vértices e salvá-los em um arquivo com nome "conexo5" no formato *graph6*, utilizamos a seguinte linha de comando: "*geng 5 -c conexo5.g6*". Este comando gera 21 grafos. O conteúdo do arquivo "conexo5.g6" pode ser visualizado na Figura 4.1. A codificação de cada linha corresponde a um grafo.

Para transformar estas codificações em matrizes de adjacências e salvá-las em um arquivo texto com nome "conexo5", utilizamos a seguinte linha de comando: "*showg -a grafos5.g6 grafos5.txt*". Cada codificação é transformada em uma matriz de adjacências. Podemos visualizar parte do conteúdo do arquivo "conexo5.txt" na Figura 4.2.

Exemplo 2: para gerar os grafos bipartidos não-isomorfos com 6 vértices e

¹Mais informações sobre *nauty and Traces* podem ser obtidas em <http://pallini.di.uniroma1.it/index.html>



Figura 4.1: Arquivo graph6 de grafos conexos com 5 vértices

salvá-los em um arquivo com nome "bipartido6" no formato *graph6*, utilizamos a seguinte linha de comando: `"geng 6 -b bipartido6.g6"`. Este comando gera 17 grafos. O conteúdo do arquivo "bipartido6.g6" pode ser visualizado na Figura 4.3. A codificação de cada linha corresponde a um grafo.

Para transformar estas codificações em matrizes de adjacências e salvá-las em um arquivo texto com nome "bipartido6", utilizamos a seguinte linha de comando: `"showg -a bipartido6.g6 bipartido6.txt"`. Cada codificação é transformada em uma matriz de adjacências. Podemos visualizar parte do conteúdo do arquivo "bipartido6.txt" na Figura 4.4.

Para que pudéssemos utilizar estes grafos em nossos experimentos precisamos que cada grafo estivesse em um arquivo único. Mais adiante, iremos discursar sobre o programa que implementamos para tratar estes grafos para torná-los entradas válidas para os nossos experimentos.

Para os nossos experimentos, desenvolvemos três programas utilizando a lin-

```

conexo5 - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda

Graph 1, order 5.
00001
00001
00001
00001
11110

Graph 2, order 5.
00011
00001
00001
10000
11100

Graph 3, order 5.
00011
00001
00001
10001
11110

```

Figura 4.2: Arquivo texto parcial de grafos conexos com 5 vértices

guagem de programação Java. O código fonte destes programas pode ser encontrado no seguinte endereço da web <http://www.alinerodrigues.eti.br/thesis>. A seguir descrevemos cada um destes programas.

O primeiro programa, denominado *Programa 1*, tem os seguintes objetivos:

- Separar as matrizes de adjacências contidas no arquivo texto gerado por *nauty and Traces* e salvá-las em arquivos texto distintos.
- Exportar as matrizes de adjacências separadas para listas de adjacência salvando em arquivo texto. Este é o formato utilizado no nosso programa que calcula propriedades dos grafos.
- Exportar as matrizes de adjacências separadas para os formatos *gdf* e *csv* para possibilitar a visualização dos grafos no *software Gephi*.

Exibimos a seguir um exemplo de utilização do Programa 1 para separar as matrizes de adjacências dos grafos gerados por *nauty and Traces*.

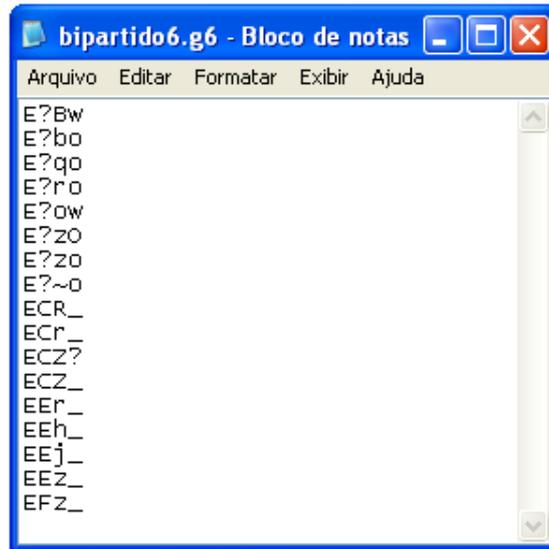


Figura 4.3: Arquivo graph6 de grafos bipartidos com 6 vértices

Exemplo: O Programa 1 chama a função "SepararMatrizes" que tem como entrada o nome do arquivo com as matrizes de adjacências e o número de vértices dos grafos. A saída da função é um conjunto de arquivos tal que cada arquivo contém uma matriz de adjacência de um grafo. Passando como entrada o arquivo "conexo5.txt" e o inteiro 5 referente ao número de vértices dos grafos, podemos ver na Figura 4.5 o conteúdo de um dos arquivos gerados pela função "SepararMatrizes".

Exibimos a seguir um exemplo de utilização do Programa 1 para exportar as matrizes de adjacências para listas de adjacências.

Exemplo: Ao exportarmos o grafo do arquivo texto mostrado na Figura 4.5 para o formato de listas de adjacências, temos como resultado o arquivo da Figura 4.6. Na primeira linha do arquivo é escrito o número de vértices do grafo, na última linha é escrito um delimitador de fim de arquivo e nas outras linhas é escrita a lista de adjacências do grafo.

Exibimos a seguir exemplos de utilização do Programa 1 para exportar as matrizes de adjacências para os formatos *gdf* e *csv*, respectivamente.

Exemplo 1: Ao exportarmos o grafo do arquivo texto mostrado na Figura 4.5

```

bipartido6 - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda

|
Graph 1, order 6.
000001
000001
000001
000001
000001
000001
111110

Graph 2, order 6.
000011
000001
000001
000001
000001
100000
111100

Graph 3, order 6.
000011
000010
000001

```

Figura 4.4: Arquivo texto parcial de grafos bipartidos com 6 vértices

para o formato *gdf* do Gephi, temos como resultado o arquivo da Figura 4.7.

Exemplo 2: Ao exportarmos o grafo do arquivo texto mostrado na Figura 4.5 para o formato *csv* do Gephi, temos como resultado o arquivo da Figura 4.8.

O segundo programa que desenvolvemos, denominado *Programa 2*, tem por objetivo produzir grafos das seguintes classes: k -cubo, tambor, sol, roda, leque e grafos conexos com $\omega(G) \geq \lceil n/2 \rceil$, para $11 \leq n \leq 15$. Este programa gera em arquivos texto as matrizes de adjacências destes grafos. Cada arquivo contém uma única matriz. Os algoritmos utilizados para gerar estas matrizes foram desenvolvidos por nós e podem ser encontrados no Apêndice desta dissertação.

O terceiro programa que desenvolvemos, denominado *Programa 3*, tem por objetivo calcular as seguintes propriedades em grafos conexos: o número de Helly k , um certificado que mostra que o grafo não é $(k - 1)$ -Helly, a cardinalidade da clique máxima, o conjunto de vértices da clique máxima encontrada, o menor inteiro t tal que o grafo é $K_{1,t}$ -free e o diâmetro do grafo. Além disso, o *Programa 3* retorna



Figura 4.5: Arquivo com matriz de adjacências de grafo conexo com 5 vértices

o grau máximo, o grau mínimo e o número de arestas do grafo.

Para construir a nossa implementação, utilizamos como embasamento teórico o conhecimento abordado nas Seções 2.4 e 2.5. O Algoritmo 4.1 apresenta os principais passos utilizados para calcular o número de Helly em grafos na nossa implementação.

Seja $G = (V(G), E(G))$ um grafo com n vértices e $\omega(G)$ a cardinalidade da clique máxima de G . No passo 1 do algoritmo, temos uma chamada a uma função que encontra a cardinalidade da clique máxima do grafo. No passo 8, a variável l recebe o total de possíveis subconjuntos $S \subseteq V(G)$, tal que $|S| = r$. No passo 10, verificamos através de algumas funções se é vazia a interseção dos fechos convexos dos conjuntos $S - \{v_i\}$, para todo $v_i \in S$.

Apresentamos a seguir um exemplo de como é formado o texto do certificado que o nosso programa produz.

Exemplo: Dado um grafo G com número de Helly igual a k , o texto do certificado gerado consiste de um conjunto de vértices $S \subseteq V(G)$ tal que $|S| = k$ e $\bigcap_{v_i \in S} H(S - \{v_i\}) = \emptyset$, seguido por dois pontos(":"), e os fechos convexos dos conjuntos

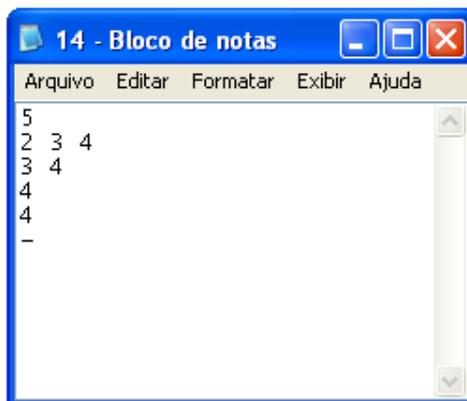


Figura 4.6: Arquivo de lista de adjacência de grafo conexo com 5 vértices

$S - v_i$, separados por vírgula(", "). Por exemplo, o certificado "0 1 2 : 1 2 3, 0 2, 0 1 3" representa o conjunto $S = \{0, 1, 2\}$ e os fechos convexos $H(S - v_0) = \{1, 2, 3\}$, $H(S - v_1) = \{0, 2\}$ e $H(S - v_2) = \{0, 1, 3\}$.

Os algoritmos utilizados para encontrar a cardinalidade da clique máxima e o menor inteiro t tal que o grafo é $K_{1,t}$ -free foram desenvolvidos por nós. Estes algoritmos podem ser encontrados no Apêndice desta dissertação.

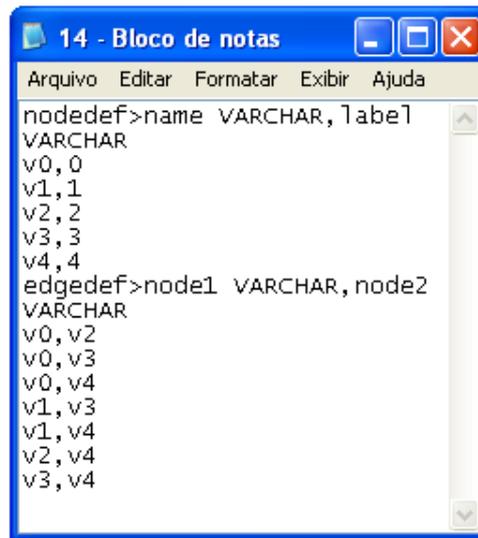
Os dados obtidos do processamento deste programa são salvos em um arquivo, podendo ser em arquivo texto ou arquivo SQL. Apresentamos a seguir exemplos do conteúdo destes arquivos.

Na Figura 4.9 podemos ver o conteúdo salvo no arquivo texto referente aos dados gerados para o grafo contido no arquivo "437283.txt".

Na Figura 4.10 podemos ver o conteúdo salvo no arquivo SQL referente aos dados gerados para o grafo contido no arquivo "12.txt".

Os dados que obtivemos no processamento deste programa foram armazenados em um banco de dados em *Firebird* para facilitar os diversos tipos de consultas que podemos fazer. A estrutura do nosso banco de dados pode ser encontrada no Apêndice desta dissertação.

Devemos ressaltar que no Programa 3, utilizamos um pacote externo de classes para calcular o fecho convexo dos grafos. Este pacote é proveniente do programa



```

14 - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
nodedef>name VARCHAR, label
VARCHAR
v0, 0
v1, 1
v2, 2
v3, 3
v4, 4
edgedef>node1 VARCHAR, node2
VARCHAR
v0, v2
v0, v3
v0, v4
v1, v3
v1, v4
v2, v4
v3, v4

```

Figura 4.7: Arquivo *gdf* de grafo conexo com 5 vértices

desenvolvido por [30] como parte de seu Trabalho de Conclusão de Curso da graduação.

Exibimos a seguir os fluxogramas dos processos de geração, separação e formatação dos grafos, além do processo de experimentação. Como exemplo, considere a geração dos grafos conexos não-isoformos com 3 vértices.

Na Figura 4.11, podemos ver o fluxograma do processo de geração dos arquivos das matrizes de adjacências dos grafos.

Na Figura 4.12, podemos ver o fluxograma do processo de separação dos arquivos das matrizes de adjacências dos grafos.

Na Figura 4.13, podemos ver o fluxograma do processo de exportação dos arquivos das matrizes de adjacências dos grafos.

Na Figura 4.14, podemos ver o fluxograma principal de atividades realizadas no processo de experimentação.

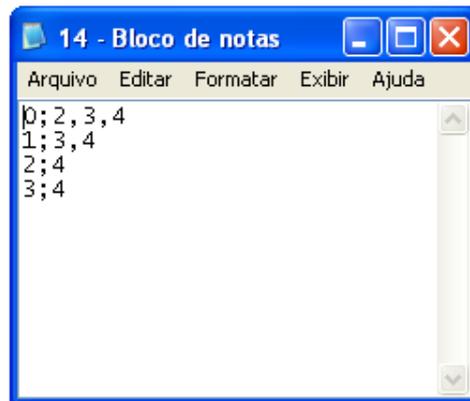


Figura 4.8: Arquivo *csv* de grafo conexo com 5 vértices

4.2 Experimentos

Os experimentos foram realizados num computador com um processador Intel Core 2 Duo de 1.66GHz e memória de 2GB. O sistema operacional utilizado foi o Microsoft Windows XP.

Geramos todos os grafos conexos para $3 \leq n \leq 10$, todos os grafos bipartidos para $3 \leq n \leq 12$ e todos os grafos biconexos-bipartidos para $4 \leq n \leq 12$.

Testamos todos os grafo bipartidos e todos os biconexos-bipartidos. Por restrições computacionais, não testamos todos os 11.716.571 de grafos conexos com 10 vértices, por se tratar de um número elevado de arquivos a serem computados. Porém, testamos todos os grafos conexos para $3 \leq n \leq 9$ e 945.131 de grafos conexos com 10 vértices, escolhidos da seguinte forma:

- todos os grafos $k_{1,3} - free$, um total de 26.389 grafos;
- todos os grafos com $\omega(G) = 2$, um total de 3.685 grafos;
- 472.931 grafos com $\omega(G) = 3$;
- 442.126 grafos escolhidos aleatoriamente.

Além disso, testamos as seguintes classes de grafos:

Algorithm 4.1 Número de Helly em grafos

Require: um grafo G
Ensure: o número de Helly h de G

```

1:  $r \leftarrow \omega(G)$ 
2: if  $r = n$  then
3:    $h \leftarrow r$ 
4: else
5:    $r \leftarrow r + 1$ 
6:   while  $r \leq n$  do
7:      $k \leftarrow 0$ 
8:      $l \leftarrow C(n, r)$ 
9:     while  $k < l$  do
10:      if  $\bigcap_{v_i \in S} H(S - \{v_i\}) = \emptyset$  then
11:         $r \leftarrow r + 1$ 
12:         $k \leftarrow l$ 
13:      end if
14:       $k \leftarrow k + 1$ 
15:    end while
16:    if  $k = l$  then
17:       $h \leftarrow r - 1$ 
18:       $r \leftarrow n + 1$ 
19:    end if
20:  end while
21: end if
22: return  $h$ 

```

- 5 grafos k -cubo, para $4 \leq n \leq 64$;
- 8 grafos tambor, para $6 \leq n \leq 20$;
- 8 grafos sol, para $6 \leq n \leq 20$;
- 12 grafos roda, para $4 \leq n \leq 15$;
- 12 grafos leque, para $4 \leq n \leq 15$;
- 2.500 grafos biconexos, $C_4 - free$ e com $\Delta \leq 4$, sendo 500 grafos para cada n , para $11 \leq n \leq 15$. Chamaremos esta classe de *Grupo 1*. A Tabela 4.1

```

437283_090714_135257 - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
arquivo           = C:\Dados\437283.txt
número de vértices = 10
número de arestas = 15
número de helly   = 5
clique            = 3
k1,t-free mínimo = 4
grau máximo       = 4
grau mínimo       = 2
diâmetro          = 3
certificado       = 0 6 7 8 9:1 2 3 4 6 7 8 9,0 3 4 5 7 8 9,0
2 4 5 6 8 9,0 1 2 5 6 7 9,0 1 3 5 6 7 8
conjunto clique   = 0 6 8

```

Figura 4.9: Arquivo texto com dados gerados para um grafo

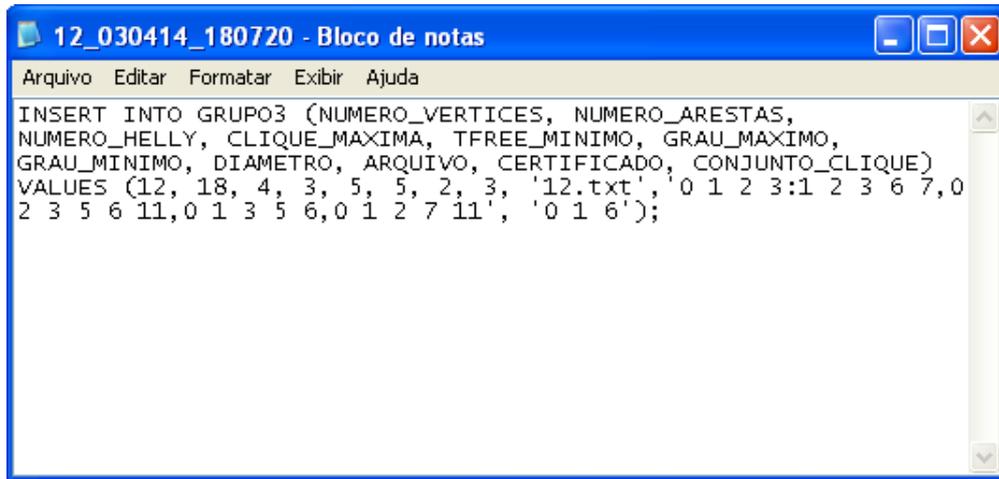
apresenta o total de grafos gerados desta classe dos quais foram escolhidos aleatoriamente os 2.500 testados.

- 2.500 grafos conexos com $\omega(G) \geq \lceil n/2 \rceil$, sendo 500 grafos para cada n , para $11 \leq n \leq 15$. Estes grafos foram gerados a partir da inclusão de vértices e arestas em grafos com 10 vértices que tinham a propriedade desejada. Chamaremos esta classe de *Grupo 2*.

Número de Vértices	Grupo 1
11	1.887
12	12.674
13	98.152
14	859.505
15	8.364.535
Totais	9.336.753

Tabela 4.1: Quantidade de grafos gerados do Grupo 1

Na Tabela 4.2, apresentamos a quantidade de grafos conexos, de grafos bi-



```

INSERT INTO GRUPO3 (NUMERO_VERTICES, NUMERO_arestas,
NUMERO_HELLY, CLIQUE_MAXIMA, TFREE_MINIMO, GRAU_MAXIMO,
GRAU_MINIMO, DIAMETRO, ARQUIVO, CERTIFICADO, CONJUNTO_CLIQUÉ)
VALUES (12, 18, 4, 3, 5, 5, 2, 3, '12.txt', '0 1 2 3:1 2 3 6 7,0
2 3 5 6 11,0 1 3 5 6,0 1 2 7 11', '0 1 6');

```

Figura 4.10: Arquivo texto com dados gerados para um grafo

partidos, de grafos biconexos-bipartidos, de grafos do Grupo 1 e de grafos do Grupo 2 testados.

Nas Tabelas 4.3 e 4.4, apresentamos a estimativa de tempo gasto no processo de testes dos grafos conexos e dos grafos bipartidos, respectivamente. Expressamos o tempo em segundos de minutos. Nestas tabelas, temos as seguintes informações:

- na primeira coluna, o número de vértices dos grafos;
- na segunda coluna, o total de grafos processados;
- na terceira coluna, o tempo referente a geração dos grafos em arquivo *graph6*, através de *nauty and Traces*;
- na quarta coluna, o tempo referente a exportação, através de *nauty and Traces*, dos grafos do arquivo *graph6* para arquivo texto contendo as matrizes de adjacências;
- na quinta coluna, o tempo referente a separação, através do Programa 1, das matrizes de adjacências geradas por *nauty and Traces*;

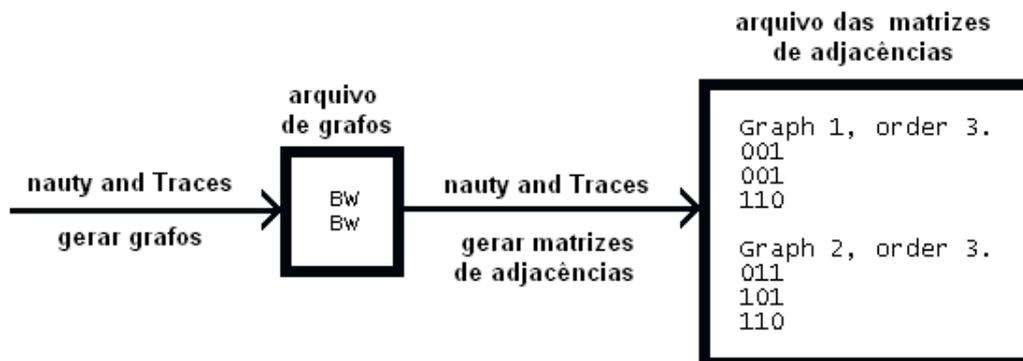


Figura 4.11: Fluxograma da geração das matrizes de adjacências dos grafos

- na sexta coluna, o tempo referente a exportação das matrizes de adjacências para listas de adjacências, através do Programa 1;
- na sétima coluna, o tempo referente ao processo de calcular as propriedades dos grafos, através do Programa 3.

O tempo total gasto para processar os grafos conexos foi de cerca de 1.277.441 segundos ou 335 horas. Já o tempo total gasto para processar os grafos bipartidos foi de cerca de 244.627 segundos ou 68 horas. Ao todo, utilizamos cerca de 423 horas para processar esses grafos. Não contabilizamos o tempo gasto com as outras classes, pois em relação a essas contabilizadas, são bem pequenas, e portanto gastamos pouco tempo as processando.

4.3 Análise dos dados

Nesta seção, apresentamos algumas estatísticas dos dados processados em nossas implementações.

Na Tabela 4.5, exibimos a porcentagem de grafos em que $h(G) = \omega(G)$. Observando esta tabela podemos chegar às seguintes conclusões:

- os grafos conexos com no máximo 4 vértices têm o número de Helly igual a cardinalidade da clique máxima;

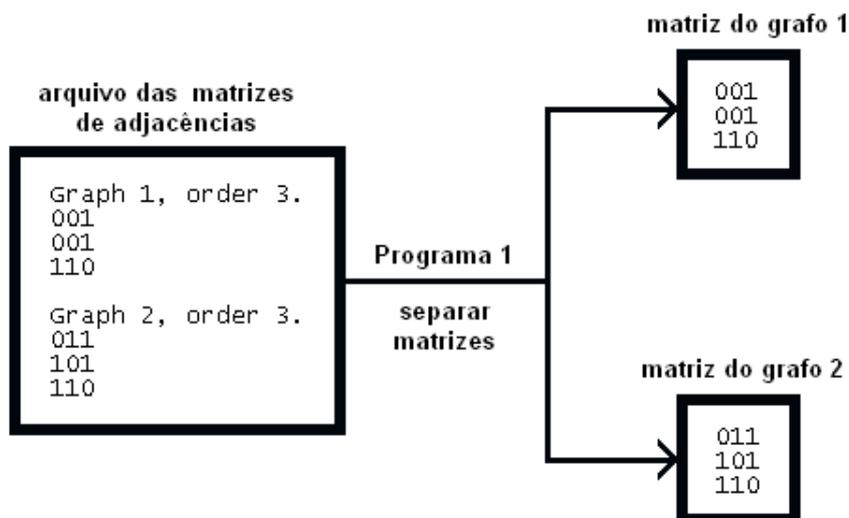


Figura 4.12: Fluxograma da separação das matrizes de adjacências dos grafos

- os grafos bipartidos com no máximo 5 vértices têm o número de Helly igual a cardinalidade da clique máxima;
- mais de 95% dos grafos conexos com no mínimo 5 vértices e no máximo 9 vértices têm o número de Helly igual a cardinalidade da clique máxima;

Na Tabela 4.6, exibimos a porcentagem de grafos em que $h(G) > \omega(G)$. Observando esta tabela podemos chegar às seguintes conclusões:

- 0,75% dos grafos conexos com no máximo 9 vértices têm o número de Helly maior do que a cardinalidade da clique máxima.
- 46,47% dos grafos bipartidos com no máximo 12 vértices têm o número de Helly maior do que a cardinalidade da clique máxima.
- 56,53% dos grafos biconexos-bipartidos com no máximo 12 vértices têm o número de Helly maior do que a cardinalidade da clique máxima.
- 61% dos grafos do Grupo 1 têm o número de Helly maior do que a cardinalidade da clique máxima.

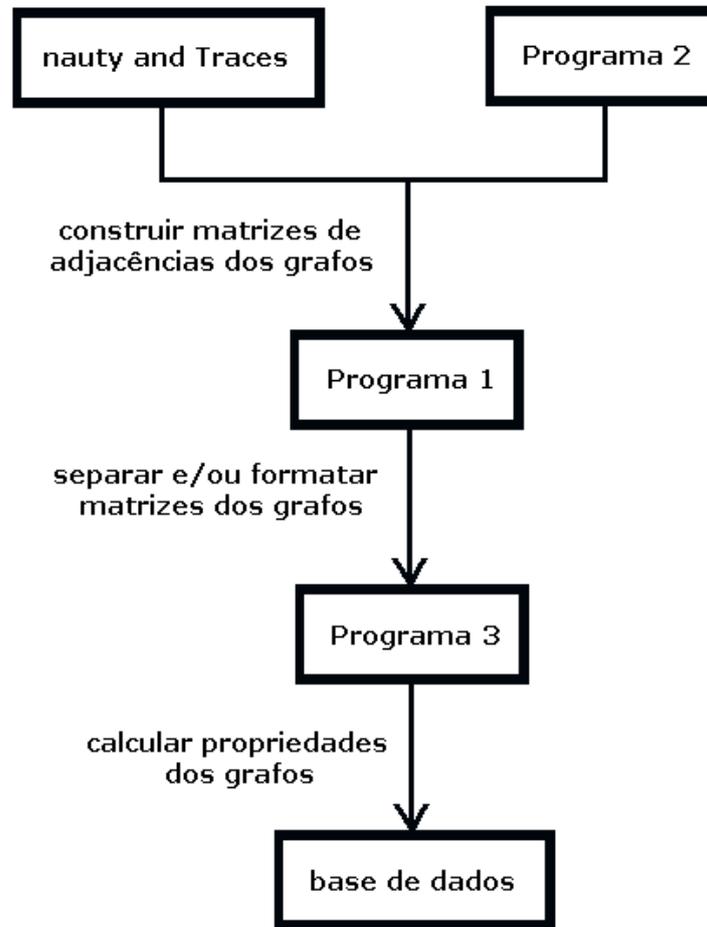


Figura 4.14: Fluxograma do processo de experimentação

- 0% dos grafos conexos com no máximo 9 vértices têm o número de Helly igual a cardinalidade da clique máxima mais duas unidades.
- 0,03% dos grafos bipartidos com no máximo 12 vértices têm o número de Helly igual a cardinalidade da clique máxima mais duas unidades.
- 0,04% dos grafos biconexos-bipartidos com no máximo 12 vértices têm o número de Helly igual a cardinalidade da clique máxima mais duas unidades.
- 17,80% dos grafos do Grupo 1 têm o número de Helly igual a cardinalidade

Número de Vértices	Conexos	Bipartidos	Biconexos-bipartidos	Grupo 1	Grupo 2
3	2	1	0	-	-
4	6	2	1	-	-
5	21	5	1	-	-
6	112	17	5	-	-
7	853	44	8	-	-
8	11.117	182	42	-	-
9	261.080	730	146	-	-
10	945.131	4.032	956	-	-
11	-	25.598	6.643	500	500
12	-	212.780	65.921	500	500
13	-	-	-	500	500
14	-	-	-	500	500
15	-	-	-	500	500
Totais	1.218.324	243.391	73.723	2.500	2.500

Tabela 4.2: Quantidade de grafos utilizados para os experimentos

da clique máxima mais duas unidades.

Na Tabela 4.9, exibimos a porcentagem de grafos em que $h(G) = \omega(G) + 3$. Observando esta tabela podemos chegar às seguintes conclusões:

- 0% dos grafos conexos, dos grafos bipartidos e dos grafos biconexos-bipartidos estudados têm o número de Helly igual a cardinalidade da clique máxima mais três unidades.
- 0,2% dos grafos do Grupo 1 têm o número de Helly igual a cardinalidade da clique máxima mais três unidades.

Na Tabela 4.10, apresentamos alguns dados estatísticos acerca do número de Helly dos grafos conexos com no máximo 9 vértices. Observando esta tabela podemos chegar às seguintes conclusões para estes grafos:

- o número de Helly mais frequente é 3;

Número de Vértices	Total de Grafos	Programa Geng	Programa Showg	Separar Matrizes	Exportar Matrizes	Calcular Propriedades
3	2	1	1	1	1	1
4	6	1	1	1	1	1
5	21	1	1	1	1	1
6	112	1	1	1	1	11
7	853	1	1	1	1	25
8	11.117	1	1	14	24	7.426
9	261.080	1	2	368	572	207.681
10	945.131	40	40	1.777	2.070	1.057.365
Totais	1.218.322	47	48	2.164	2.671	1.272.511

Tabela 4.3: Tempo em segundos gasto nos testes dos grafos conexos

Número de Vértices	Total de Grafos	Programa Geng	Programa Showg	Separar Matrizes	Exportar Matrizes	Calcular Propriedades
8	182	1	1	1	1	54
9	730	1	1	1	1	309
10	4.032	1	1	8	8	2.383
11	25.598	1	2	54	52	20.155
12	212.780	2	8	457	432	220.693
Totais	243.322	6	13	521	494	243.593

Tabela 4.4: Tempo em segundos gasto nos testes dos grafos bipartidos

- a média do número de Helly está no intervalo de 2,50 a 3,75;
- o número de Helly que ocupa a posição central nos dados ordenados está no intervalo de 2,5 a 4,0.

Na Tabela 4.11, apresentamos alguns dados estatísticos acerca do número de Helly dos grafos bipartidos com no máximo 12 vértices. Observando esta tabela podemos chegar às seguintes conclusões para estes grafos:

- o número de Helly mais frequente é 2;
- a média do número de Helly está no intervalo de 2,00 a 2,47;
- o número de Helly que ocupa a posição central nos dados ordenados é 2.

Número de Vértices	Conexos	Bipartidos	Biconexos-bipartidos	Grupo 1
3	100,00	100,00	-	-
4	100,00	100,00	100,00	-
5	95,24	100,00	100,00	-
6	97,32	94,12	80,00	-
7	97,66	93,18	75,00	-
8	98,69	86,81	71,43	-
9	99,28	80,96	64,38	-
10	-	72,15	57,32	-
11	-	62,66	49,44	63,00
12	-	53,59	42,61	50,60
13	-	-	-	36,60
14	-	-	-	29,20
15	-	-	-	15,60
Totais	99,25	54,97	43,48	39,00

Tabela 4.5: Porcentagem de grafos em que $h(G) = \omega(G)$

Na Tabela 4.12, apresentamos alguns dados estatísticos acerca do número de Helly dos grafos biconexos-bipartidos com no máximo 12 vértices. Observando esta tabela podemos chegar às seguintes conclusões para estes grafos:

- o número de Helly mais frequente é 2;
- a média do número de Helly está no intervalo de 2,00 a 2,57;
- o número de Helly que ocupa a posição central nos dados ordenados está no intervalo de 2 a 3.

Número de Vértices	Conexos	Bipartidos	Biconexos-bipartidos	Grupo 1
3	0,00	0,00	-	-
4	0,00	0,00	0,00	-
5	4,76	0,00	0,00	-
6	2,68	5,88	20,00	-
7	2,34	6,82	25,00	-
8	1,31	13,19	28,57	-
9	0,72	19,04	35,62	-
10	-	27,85	42,68	-
11	-	37,34	50,56	37,00
12	-	48,06	57,40	49,40
13	-	-	-	63,40
14	-	-	-	70,80
15	-	-	-	84,40
Totais	0,75	46,47	56,53	61,00

Tabela 4.6: Porcentagem de grafos em que $h(G) > \omega(G)$

Número de Vértices	Conexos	Bipartidos	Biconexos-bipartidos	Grupo 1
3	0,00	0,00	-	-
4	0,00	0,00	0,00	-
5	4,76	0,00	0,00	-
6	2,68	5,88	20,00	-
7	2,34	6,82	25,00	-
8	1,30	13,19	28,57	-
9	0,71	19,04	35,62	-
10	-	27,83	42,57	-
11	-	37,32	50,52	33,40
12	-	48,04	57,36	41,20
13	-	-	-	46,20
14	-	-	-	44,40
15	-	-	-	49,80
Totais	0,74	46,45	56,49	43,00

Tabela 4.7: Porcentagem de grafos em que $h(G) = \omega(G) + 1$

Número de Vértices	Conexos	Bipartidos	Biconexos-bipartidos	Grupo 1
3	0,00	0,00	-	-
4	0,00	0,00	0,00	-
5	0,00	0,00	0,00	-
6	0,00	0,00	0,00	-
7	0,00	0,00	0,00	-
8	0,01	0,00	0,00	-
9	0,00	0,00	0,00	-
10	-	0,02	0,10	-
11	-	0,02	0,05	3,60
12	-	0,03	0,04	8,20
13	-	-	-	17,00
14	-	-	-	26,20
15	-	-	-	34,00
Totais	0,00	0,03	0,04	17,80

Tabela 4.8: Porcentagem de grafos em que $h(G) = \omega(G) + 2$

Número de Vértices	Conexos	Bipartidos	Biconexos-bipartidos	Grupo 1
3	0,00	0,00	-	-
4	0,00	0,00	0,00	-
5	0,00	0,00	0,00	-
6	0,00	0,00	0,00	-
7	0,00	0,00	0,00	-
8	0,00	0,00	0,00	-
9	0,00	0,00	0,00	-
10	-	0,00	0,00	-
11	-	0,00	0,00	0,00
12	-	0,00	0,00	0,00
13	-	-	-	0,20
14	-	-	-	0,20
15	-	-	-	0,60
Totais	0,00	0,00	0,00	0,20

Tabela 4.9: Porcentagem de grafos em que $h(G) = \omega(G) + 3$

Número de Vértices	Moda	Mediana	Média	Desvio Mediano	Desvio Médio	Variância	Desvio Padrão
3	2 e 3	2,5	2,50	0,50	0,50	0,25	0,50
4	2	2,5	2,67	0,67	0,67	0,56	0,75
5	3	3,0	3,00	0,48	0,48	0,57	0,76
6	3	3,0	3,18	0,46	0,55	0,56	0,75
7	3	3,0	3,40	0,50	0,60	0,53	0,73
8	3	3,0	3,57	0,60	0,60	0,46	0,68
9	4	4,0	3,75	0,48	0,55	0,44	0,66

Tabela 4.10: Medidas de resumo acerca do número de Helly em grafos conexos

Número de Vértices	Moda	Mediana	Média	Desvio Mediano	Desvio Médio	Variância	Desvio Padrão
3	2	2	2,00	0,00	0,00	0,00	0,00
4	2	2	2,00	0,00	0,00	0,00	0,00
5	2	2	2,00	0,00	0,00	0,00	0,00
6	2	2	2,06	0,06	0,11	0,06	0,24
7	2	2	2,07	0,07	0,13	0,06	0,25
8	2	2	2,13	0,13	0,23	0,11	0,34
9	2	2	2,19	0,19	0,31	0,15	0,39
10	2	2	2,28	0,28	0,40	0,20	0,45
11	2	2	2,37	0,37	0,47	0,23	0,48
12	2	2	2,47	0,47	0,50	0,25	0,50

Tabela 4.11: Medidas de resumo acerca do número de Helly em grafos bipartidos

Número de Vértices	Moda	Mediana	Média	Desvio Mediano	Desvio Médio	Variância	Desvio Padrão
4	2	2	2,00	0,00	0,00	0,00	0,00
5	2	2	2,00	0,00	0,00	0,00	0,00
6	2	2	2,20	0,20	0,32	0,16	0,40
7	2	2	2,25	0,25	0,38	0,19	0,43
8	2	2	2,29	0,29	0,41	0,20	0,45
9	2	2	2,36	0,36	0,46	0,23	0,48
10	2	2	2,43	0,43	0,49	0,25	0,50
11	3	3	2,51	0,49	0,50	0,25	0,50
12	3	3	2,57	0,43	0,49	0,25	0,50

Tabela 4.12: Medidas de resumo acerca do número de Helly em grafos biconexos-bipartidos

5 CONCLUSÃO

Nesta dissertação, estudamos o número de Helly em grafos na convexidade geodética. Nossas principais contribuições são descritas a seguir.

Implementamos um programa que tem por objetivo calcular as seguintes propriedades em grafos: o número de Helly k , um certificado que mostra que o grafo não é $(k - 1)$ -Helly, a cardinalidade da clique máxima, o conjunto de vértices da clique máxima encontrada, o menor inteiro t tal que o grafo é $K_{1,t}$ -free e o diâmetro do grafo.

Encontramos um limite superior para o número de Helly em grafos bipartidos e apresentamos uma família que mostra que este limite é justo. Encontramos limites superiores para o número de Helly em grafos com no máximo 9 vértices. Calculamos o número de Helly para as seguintes classes de grafos: sol, roda e leque. Formulamos algumas conjecturas relacionando algumas propriedades dos grafos com o seu número de Helly. Por fim, apresentamos algumas estatísticas dos dados processados em nossas implementações.

No processo de experimentação, analisamos todos os grafos conexos com no máximo 9 vértices, uma fração dos grafos conexos com 10 vértices; todos os grafos bipartidos com até 12 vértices; todos os grafos biconexos-bipartidos com até 12 vértices; os grafos k -cubo com até 64 vértices; os grafos tambor e sol com até 20 vértices; os grafos roda e leque com até 15 vértices; entre outras classes. Ao todo, analisamos 1.540.392 grafos.

Esta dissertação resultou em dois artigos aceitos para publicação, os listamos a seguir:

- "O Número de Helly em Grafos na Convexidade Geodésica", no XVII Latin-Iberian-American Conference on Operations Research (CLAIO).
- "A tight upper bound for the Helly number of the geodetic convexity on bi-

partite graphs", no Latin American Workshop on Cliques in Graphs 2014 (LAWCG).

O número Helly na convexidade geodética tem problemas em aberto para muitas classes de grafos. Por exemplo, encontrar o número de Helly para grafos bipartidos, $K_{1,t}$ -free ou k -regular.

As implementações desenvolvidas nesta dissertação podem servir de auxílio ao trabalho de outros pesquisadores. Para tanto, o código fonte dos programas pode ser encontrado para *download* no seguinte endereço da web:

<http://www.alinerodrigues.eti.br/thesis>.

REFERÊNCIAS

- [1] AMENTA, N. Helly theorems and generalized linear programming. In: ANNUAL SYMPOSIUM ON COMPUTATIONAL GEOMETRY, 9. 1993, San Diego, CA. **Proceedings ...** New York: ACM, 1993. p.63-72.
- [2] APPOLONIO, N. ; CARAMIA, M. Recognizing Helly Edge-Path-Tree graphs and their clique graphs. **Discrete Applied Mathematics**, Amsterdam, v. 159, n. 11, p. 1166-1175, 2011.
- [3] BARG, A. ; COHEN, G. ; ENCHEVA, S. A hypergraph approach to the identifying parent property: the case of multiple parents. **SIAM Journal on Discrete Mathematics**, Philadelphia, v.14, n. 3, p. 423-431, 2001.
- [4] BENKE, M. Efficient type reconstruction in the presence of inheritance. In: BORZYSZKOWSKI, A. ; SOKOLOWSKI, S. (Eds). **Mathematical Foundations of Computer Science 1993**. 18th International Symposium MFCS'93 Gdansk Poland Aug. 30-Sept. 3, 1993 Proceedings. Berlin Heidelberg: Springer, 1993. p. 272-280. (Lecture Notes in Computer Science, v.711).
- [5] BERGE, C. ; DUCHET, P. Generalization of Gilmore's theorem. In: FIEDLER, M. (Ed). **Recent advances in graph stheory**. Prague: Acad. Praha, 1975. p. 49-55.
- [6] BONDY, J. A. ; MURTY, U. S. R. **Graph theory with applications**. New York: Springer/NorthHolland, 2008.
- [7] BRETTO, A. et al. Combinatorics and image processing. **Grafical Models and Image Processing**, Duluth, v. 59, n. 5, p. 265-277, Sept. 1997.
- [8] C  CERES, J. ; M  RQUEZ, A. ; PUERTAS, M. L. Steiner distance and conve-

- xity in graphs. **European Journal of Combinatorics**, London, v. 29, n. 3, p. 726-736, Apr. 2008.
- [9] CENTENO, C. C. et al. Convex partitions of graphs induced by paths of order three. **Discrete Mathematics and Theoretical Computer Science**, Providence, v. 12, n. 12, p. 175-184, 2010.
- [10] CHARTRAND, G. ; WALL, C. E. ; ZHANG, P. The convexity number of a graph. **Graphs and Combinatorics**, Tokyo, v. 18, n. 2, p. 209-217, May 2002.
- [11] CHARTRAND, G. ; ZHANG, P. H-Convex graphs. **Mathematica Bohemica**, Praga, v. 126, n. 1, p. 209-220, 2001.
- [12] _____. The Steiner number of a graph. **Discrete Mathematics**, Amsterdam, v. 242, n. 1-3, p. 41-54, Jan. 2002.
- [13] CORMEN, T. H. et al. **Introduction to algorithms**. Cambridge: MIT Press, 2002.
- [14] DANZER, L. ; GRÜNBAUM, B. ; KLEE, V. Helly's theorem and its relatives. In: **Convexity**. Proceedings of the Symposia in Pure Mathematics, Seattle: American Mathematical Society, 1963. v.7, p.101-180.
- [15] DOURADO, M. C. ; PETITO, P. ; TEIXEIRA, R. B. Helly property and Sandwich graphs. **Electronic Notes in Discrete Mathematics**, [S.l.], v. 22, p.497-500, 2005.
- [16] DOURADO, M. C. ; PROTTI, F. ; SZWARCFITER, J. L. Complexity aspects of generalized Helly hypergraphs. **Information Processing Letters**, Amsterdam, v.99, n. 1, p. 13-18, Jul. 2006.
- [17] _____. Complexity aspects of the Helly property: graphs and hypergraphs. **The Electronic Journal of Combinatorics**, [S.l.], v. 17, Jun. 2009.

- [18] _____. On the strong p -Helly property. **Discrete Applied Mathematics**, Amsterdam, v. 156, n. 7, p. 1053-1057, Apr. 2008.
- [19] DRAGAN, F. F. ; PRISAKAR, K. F. ; CHEPOI, V. D. Location problems in graphs and the Helly property. **Diskretnaya Matematika**, Moscow, v. 4, n. 4, p. 67-73, 1992.
- [20] FAGIN, R. Acyclic database schemes of various degrees: a painless introduction. In: AUSIELLO, G. ; PROTASI, M. (Eds.). **CAAP'83**. Trees in algebra and programming 8th Colloquium Láquila, March 9-11 1983. Proceedings. Berlin Heidelberg: Springer, 1983. p.65-89. (Lecture Notes in Computer Science, v.159).
- [21] GAREY, M. R. ; JOHNSON, D. S. **Computers and intractability: a guide to the theory of NP-completeness**. New York: W. H. Freeman & Co., 1990.
- [22] GERA, R. On the dominator colorings in bipartite graphs. In: ITNG '07 – INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY, 4., 2007, Las Vegas. **Proceedings ...** Los Alamitos: IEEE, 2007. p.947-952.
- [23] JAMISON, R. E. A perspective on abstract convexity: classifying alignments by varieties. In: KAY, D.C. ; BREEN, M. (Eds.). **Convexity and related combinatorial geometry**. New York: Marcel Decker, 1982. p.113-150.
- [24] JAMISON, R. E.; NOWAKOWSKI, R. A Helly theorem for convexity in graphs. **Discrete Mathematics**, Amsterdam, v. 51, n. 1, p. 35-39, 1984.
- [25] KLEINBERG, J. ; TARDOS, E. **Algorithm design**. New York: Addison-Wesley, 2005.
- [26] KULPA, W. Intersection properties of Helly families. **Topology and its Applications**, Amsterdam, v.116, n. 2, p. 227-233, Nov. 2001.

- [27] LEVI, F. W. On Helly's theorem and the axioms of convexity. **Journal of Indian Math. Society**, New Dehli, v. 15, p. 65-76, 1951.
- [28] MCKAY, B. D. Practical graph isomorphism. **Congressus Numerantium**, Winnipeg, v. 30, p. 45-87, 1981.
- [29] MCKAY, B. D. ; PIPERNO, A. Practical graph isomorphism II. **Journal of Symbolic Computation**, London, v. 60, p. 94-112, Jan. 2014.
- [30] OLIVEIRA, R. A. **Uma implementação do algoritmo para encontrar o fecho convexo em grafo**. 2007. Trabalho de Conclusão de Curso (Graduação em Matemática Aplicada Computacional) – Universidade Federal Rural do Rio de Janeiro, Rio de Janeiro, 2007.
- [31] PIPERNO, A. Search space contraction in canonical labeling of graphs. **Computing Research Repository**, [S.l.], 2008.
- [32] POLAT, N. A Helly theorem for geodesic convexity in strongly dismantlable graphs. **Discrete Mathematics**, Amsterdam, v. 140, n. 1-3, p. 119-127, Jun. 1995.
- [33] PRZYTYCKA, T. M. et al. Graph theoretical insights into evolution of multi-domain proteins. In: MIYAMO, S. et al. (Eds.). **Research in computational molecular biology**. 9Th Annual International Conference RECOMB 2005, Cambridge, MA, May 14-18, 2005 Proceedings. Berlin Heidelberg: Springer, 2005. p. 311-325. (Lecture Notes in Computer Science, v.3500).
- [34] SIERKSMA, G. Caratheodory and Helly-numbers of convex product structures. **Pacific Journal of Mathematics**, Carmel Valley, v. 61, n. 1, p. 275-282, 1975.
- [35] SZWARCFITER, J. L. **Grafos e algoritmos computacionais**. Rio de Janeiro: Campus, 1984.

- [36] SZWARCFITER, J. L. ; MARKENZON, L. **Estruturas de dados e seus algoritmos**. Rio de Janeiro: LTC, 2010.
- [37] TELES, M. **Sobre o número de Helly em convexidades**. 2014. Tese (Doutorado em Sistemas e Computação) – COPPE Sistema, Universidade Federal do Rio de Janeiro, Rio de Janeiro. Em andamento.
- [38] VAN DE VEL, M. L. J. **Theory of convex structures**. Amsterdam: North Holland, 1993. (North-Holland Mathematical Library, v.50).

APÊNDICE A

A.1 Estrutura do Banco de Dados Utilizado

Nesta seção, apresentamos o *script* SQL necessário para a criação das tabelas do nosso banco de dados. Temos as seguintes tabelas: Conexos, Bipartidos, Biconexos_Bipartidos, Grupo1 e Grupo2, que possuem a estrutura da tabela Resultado apresentada adiante.

```
Create Table Resultado(  
Id Integer Not Null,  
Numero_Vertices Integer,  
Numero_Arestas Integer,  
Numero_Helly Integer,  
Clique_Maxima Integer,  
Tfree_Minimo Integer,  
Grau_Maximo Integer,  
Grau_Minimo Integer,  
Diametro Integer,  
Arquivo Dm_Nome_Arquivo,  
Conjunto_Clique Dm_Conjunto_Clique,  
Certificado Dm_Certificado);  
  
Create Domain Dm_Nome_Arquivo As Varchar(300);  
  
Create Domain Dm_Conjunto_Clique As Varchar(100);  
  
Create Domain Dm_Certificado As Varchar(1000);
```

```
Alter Table Resultado Add Constraint Pk_Resultado Primary Key (Id);
```

```
Create Generator Gen_Resultado_Id;
```

```
Create Trigger Resultado_Bi For Resultado Active Before Insert Position 0 As
Begin
If (New.Id Is Null) Then
New.Id = Gen_Id(Gen_Resultado_Id,1);
End
```

A.2 Algoritmos Implementados

Nesta seção, apresentamos alguns dos algoritmos que construímos em nossas implementações.

Apresentamos a seguir alguns dos algoritmos implementados no Programa 2.

O Algoritmo A.1 apresenta os principais passos utilizados para construir a matriz de adjacências de um grafo k -cubo. Nos passos 9 e 10, chamamos a função *IntToBinStr* que tem o objetivo de converter um número inteiro para binário e a seguir para *string* com k posições.

O Algoritmo A.2 apresenta os passos utilizados para construir a matriz de adjacências de um grafo leque.

O Algoritmo A.3 apresenta os passos utilizados para construir a matriz de adjacências de grafo ciclo C_k . Este algoritmo é útil na construção dos grafos sol e roda.

O Algoritmo A.4 apresenta os passos utilizados para construir a matriz de adjacências de um grafo roda.

O Algoritmo A.5 apresenta os passos utilizados para construir a matriz de adjacências de um grafo sol.

Algorithm A.1 Grafo k -cubo

Ensure: um inteiro k

Require: a matriz de adjacências $M_{n,n}$ do grafo k -cubo

```

1:  $n \leftarrow 2^k$ ;
2:  $dif \leftarrow 0$ ;
3:  $bin1 \leftarrow ""$ ;
4:  $bin2 \leftarrow ""$ ;
5: for  $i$  de 0 até  $n - 1$  do
6:   for  $j$  de 0 até  $n - 1$  do
7:     if  $j > i$  then
8:       for  $p$  de 0 até  $k - 1$  do
9:          $bin1 \leftarrow IntToBinStr(i, k)$  ;
10:         $bin2 \leftarrow IntToBinStr(j, k)$  ;
11:        if  $p$ -ésimo bit de  $bin1 \neq p$ -ésimo bit de  $bin2$  then
12:           $dif \leftarrow dif + 1$ ;
13:        end if
14:      end for
15:      if  $dif \leq 1$  then
16:         $M[i][j] \leftarrow 1$ ;
17:         $M[j][i] \leftarrow 1$ ;
18:      end if
19:       $dif \leftarrow 0$ ;
20:    end if
21:  end for
22: end for
23: return  $M$ 

```

O Algoritmo A.6 apresenta os passos utilizados para construir a matriz de adjacências de um grafo tambor.

Apresentamos a seguir alguns dos algoritmos implementados no Programa 3.

Seja $G = (V(G), E(G))$ um grafo e $\Delta(G)$ o grau máximo de G .

O Algoritmo A.7 apresenta os principais passos utilizados para calcular o menor inteiro t tal que G é $K_{1,t}$ -free. No passo 4 do algoritmo, verificamos se G é $k_{1,t}$ -free, para isso construímos o algoritmo A.8.

O algoritmo A.8 conclui que G é $K_{1,t}$ -free, se no passo 4 a condição sempre retornar *false*. E conclui que G não é $K_{1,t}$ -free, se no passo 4 a condição retornar

Algorithm A.2 Grafo Leque

Ensure: um inteiro k

Require: a matriz de adjacências $M_{n,n}$ do grafo k -leque

```

1:  $n \leftarrow k + 1$ ;
2:  $j \leftarrow 0$ ;
3: for  $i$  de 0 até  $k - 2$  do % desenhar o caminho  $P_k$ 
4:    $j \leftarrow i + 1$ ;
5:    $M[i][j] \leftarrow 1$ ;
6:    $M[j][i] \leftarrow 1$ ;
7: end for
8: for  $i$  de 0 até  $k - 1$  do % tornar universal, o último vértice
9:    $M[i][k] \leftarrow 1$ ;
10:   $M[k][i] \leftarrow 1$ ;
11: end for
12: return  $M$ 

```

Algorithm A.3 Grafo Ciclo

Ensure: uma matriz $M_{n,n}$ e um inteiro k

Require: a matriz de adjacências M do grafo ciclo

```

1: for  $i$  de 0 até  $k - 1$  do
2:    $j \leftarrow (i + 1) \bmod k$ ;
3:    $M[i][j] \leftarrow 1$ ;
4:    $M[j][i] \leftarrow 1$ ;
5: end for
6: return  $M$ 

```

true ao menos uma vez.

O Algoritmo A.9 apresenta os principais passos utilizados para calcular a cardinalidade da clique máxima do grafo G . No passo 4 do algoritmo, verificamos se G tem uma clique de tamanho k , para isso construímos o algoritmo A.10.

O algoritmo A.10 conclui que G tem uma clique de tamanho k , se no passo 5 a condição retornar *true* ao menos uma vez. E conclui que G não tem uma clique de tamanho k , se no passo 5 a condição sempre retornar *false*.

Algorithm A.4 Grafo Roda

Ensure: um inteiro k **Require:** a matriz de adjacências $M_{n,n}$ do grafo roda

```
1:  $n \leftarrow k + 1$ ;  
2:  $j \leftarrow 0$ ;  
3:  $M \leftarrow$  matriz de adjacências do grafo  $C_k$ ;  
4: for  $i$  de 0 até  $k - 1$  do % tornar universal, o último vértice  
5:    $M[i][k] \leftarrow 1$ ;  
6:    $M[k][i] \leftarrow 1$ ;  
7: end for  
8: return  $M$ 
```

Algorithm A.5 Grafo Sol

Ensure: um inteiro k **Require:** a matriz de adjacências $M_{n,n}$ do grafo sol

```
1:  $n \leftarrow 2k$ ;  
2:  $j \leftarrow 0$ ;  
3:  $M \leftarrow$  matriz de adjacências do grafo  $C_k$ ;  
4: for  $i$  de  $k$  até  $n - 1$  do % desenhar raios  
5:    $j \leftarrow i \bmod k$ ;  
6:    $M[i][j] \leftarrow 1$ ;  
7:    $M[j][i] \leftarrow 1$ ;  
8:    $j \leftarrow (i + 1) \bmod k$ ;  
9:    $M[i][j] \leftarrow 1$ ;  
10:   $M[j][i] \leftarrow 1$ ;  
11: end for  
12: return  $M$ 
```

Algorithm A.6 Grafo Tambor

```

1:  $n \leftarrow 2 * k$ ;
2:  $j \leftarrow 0$ ;
   % topo
3: for  $i$  de 0 até  $k - 1$  do
4:    $j \leftarrow (i + 1) \bmod k$ ;
5:    $M[i][j] \leftarrow 1$ ;
6:    $M[j][i] \leftarrow 1$ ;
7: end for
   % base
8:  $M[n - 1][k] \leftarrow 1$ ;
9:  $M[k][n - 1] \leftarrow 1$ ;
10: for  $i$  de  $k$  até  $n - 2$  do
11:    $j \leftarrow i + 1$ ;
12:    $M[i][j] \leftarrow 1$ ;
13:    $M[j][i] \leftarrow 1$ ;
14: end for
   % laterais
15: for  $i$  de 0 até  $k - 1$  do
16:    $j \leftarrow i + k$ ;
17:    $M[i][j] \leftarrow 1$ ;
18:    $M[j][i] \leftarrow 1$ ;
19: end for
20: return  $M$ 

```

Algorithm A.7 Grafo $K_{1,t}$ -free mínimo

Require: um grafo G

Ensure: o menor inteiro t tal que G é $K_{1,t}$ -free

```

1:  $t \leftarrow \Delta(G) + 1$ ;
2:  $ti \leftarrow 2$ ;
3: while  $ti \leq \Delta(G)$  do
4:   if  $G$  é  $k_{1,ti}$ -free then
5:      $t \leftarrow ti$ ;
6:      $ti \leftarrow \Delta(G)$ ;
7:   end if
8:    $ti \leftarrow ti + 1$ ;
9: end while
10: return  $t$ 

```

Algorithm A.8 Grafo $K_{1,t}$ -free

Require: um grafo G e um inteiro t **Ensure:** se G é $K_{1,t}$ -free, então retorna *true*, caso contrário, retorna *false*

```
1: verifica  $\leftarrow$  true;  
2: for cada  $v \in V(G)$  do  
3:   if  $d(v) \geq t$  then  
4:     if  $N(v)$  tem algum conjunto independente de tamanho  $t$  then  
5:       verifica  $\leftarrow$  false;  
6:     pare;  
7:   end if  
8: end if  
9: end for  
10: return verifica
```

Algorithm A.9 Cardinalidade da Clique Máxima

Require: um grafo G **Ensure:** a cardinalidade ω da clique máxima de G

```
1:  $\omega \leftarrow 2$ ;  
2:  $k \leftarrow 3$ ;  
3: while  $k \leq \Delta(G) + 1$  do  
4:   if  $G$  tem uma clique de tamanho  $k$  then  
5:      $\omega \leftarrow k$ ;  
6:   else  
7:      $k \leftarrow \Delta(G) + 1$ ;  
8:   end if  
9:    $k \leftarrow k + 1$ ;  
10: end while  
11: return  $\omega$ 
```

Algorithm A.10 Grafo k-clique

Require: um grafo G e um inteiro k

Ensure: se G tem uma clique de tamanho k , então retorna *true*, caso contrário, retorna *false*

```
1: verifica  $\leftarrow$  false;
2: for cada  $v \in V(G)$  do
3:   if  $d(v) \geq k - 1$  then
4:     for cada  $S \subseteq N(v)$ , tal que  $|S| = k - 1$  do
5:       if todo par  $(x, y)$  é conexo, para  $x, y \in S$  then
6:         verifica  $\leftarrow$  true;
7:       pare;
8:     end if
9:   end for
10: end if
11: end for
12: return verifica
```
