



**UNIVERSIDADE FEDERAL DO TOCANTINS
CÂMPUS UNIVERSITÁRIO DE PALMAS
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

KEDSON ALVES SILVA

UM ESTUDO SOBRE GRAFOS B2-EPG E B2-EPG-HELLY

PALMAS (TO)

2022

KEDSON ALVES SILVA

UM ESTUDO SOBRE GRAFOS B2-EPG E B2-EPG-HELLY

Trabalho de Conclusão de Curso II
apresentado à Universidade Federal do
Tocantins para obtenção do título de Bacharel
em Ciência da Computação, sob a orientação
do(a) Prof.(a) Dr. Tanilson Dias dos Santos.

Orientador: Dr. Tanilson Dias dos Santos

PALMAS (TO)

2022

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da Universidade Federal do Tocantins

S586e Silva, Kedson Alves .
Um estudo sobre grafos B2-EPG e B2-EPG-Helly. / Kedson Alves
Silva. – Palmas, TO, 2022.
55 f.

Monografia Graduação - Universidade Federal do Tocantins –
Câmpus Universitário de Palmas - Curso de Ciências da Computação,
2022.
Orientador: Tanilson Dias dos Santos

1. Grafos. 2. Propriedade Helly. 3. Representação EPG. 4. Teoria
da Computação. I. Título

CDD 004

TODOS OS DIREITOS RESERVADOS – A reprodução total ou parcial, de
qualquer forma ou por qualquer meio deste documento é autorizado desde
que citada a fonte. A violação dos direitos do autor (Lei nº 9.610/98) é crime
estabelecido pelo artigo 184 do Código Penal.

**Elaborado pelo sistema de geração automática de ficha catalográfica
da UFT com os dados fornecidos pelo(a) autor(a).**

KEDSON ALVES SILVA

UM ESTUDO SOBRE GRAFOS B2-EPG E B2-EPG-HELLY

Trabalho de Conclusão de Curso II apresentado à UFT – Universidade Federal do Tocantins – Câmpus Universitário de Palmas, Curso de Ciência da Computação foi avaliado para a obtenção do título de Bacharel e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Data de aprovação: 13 / 7 / 2022

Banca Examinadora:

Prof. Dr. Tanilson Dias dos Santos, UFT

Prof. Dr. Alexandre Tadeu Rossini da Silva, UFT

Prof. Dr. Rogério Azevedo Rocha, UFT

Prof. Dr. Uéverton dos Santos Souza, UFF

*Dedico este trabalho aos meus
pais, pois foi graças ao seus
esforços que hoje posso
concluir o meu curso.*

AGRADECIMENTOS

Em primeiro lugar, a Deus, que fez com que meus objetivos fossem alcançados, durante todos os meus anos de estudo. Agradeço a toda minha família, pois o apoio emocional que me proporcionou foi transformado em força, a qual utilizei para chegar até o fim do curso. Em especial os meus tios que me acolheram em sua casa e não mediram esforços para ajudar quando precisei.

Agradeço ao meu orientador, Tanilson Dias dos Santos, por ter aceitado me acompanhar neste projeto, sem todo seu empenho eu não teria chegado até aqui. Agradeço também a todos aqueles que me apoiaram, motivaram ou ajudaram financeiramente. Sou grato a cada um. Tenham certeza que tudo o que fizeram nunca será esquecido.

RESUMO

A palavra EPG é um acrônimo para Edge-intersection Paths on a Grid, isto é, representa exatamente a classe de grafos de aresta-interseção de caminhos sobre uma grade. Neste trabalho de conclusão de curso iremos explorar principalmente uma sub-classe de grafos EPG, conhecida como B_2 -EPG-Helly (mais especificamente a sua complexidade de reconhecimento). Contudo, também investigamos representações de grafos que não são B_1 -EPG, mas ainda não foram associadas a alguma classe B_k -EPG, além de estudar outras propriedades de caminhos em B_2 -EPG e B_2 -EPG com a propriedade Helly.

Essa pesquisa contém resultados iniciais inéditos sobre a exploração da classe B_2 -EPG, além de propor tópicos interessantes para trabalhos futuros.

Palavra-chave: Complexidade. Grafos. Propriedade Helly. Representação EPG.

ABSTRACT

The word EPG is an acronym for Edge-Intersecting Paths on a Grid, that is, it exactly represents the class of edge-intersection graphs of paths on a grid. In this writing, we started exploring the EPG graph subclass, well known as B_2 -EPG-Helly (more specifically its recognition complexity). However, we have also investigated graph representations that are not B_1 -EPG, but have not yet been associated with any B_k -EPG class, and also study other path properties in B_2 -EPG and B_2 -EPG with the Helly property. This research contains initial unpublished results about an exploration of the class B_2 -EPG, in addition to proposing interesting topics for future work.

Keywords: Complexity. Graphs. Helly Property. EPG representation.

LISTA DE FIGURAS

Figura 1 – Grafo G (3-sun)	12
Figura 2 – Representação B_1 -EPG do grafo 3-sun da Figura 1	17
Figura 3 – Diagrama de Complexidade $\mathcal{P} \neq \mathcal{NP}$	20
Figura 4 – Propriedade Helly	23
Figura 5 – Grafo G_5 de 6 vértices	23
Figura 6 – Representação B_0 -EPG	26
Figura 7 – Representações claw-clique e edge-clique	27
Figura 8 – Representações de um ciclo induzido de tamanho 4 por True pie, False pie e Frame	27
Figura 9 – Representação B_2 -EPG	28
Figura 10 – Interseção de um segmento com um caminho B_2 -EPG	41
Figura 11 – Possíveis caminhos B_2 -EPG (com exatamente 2 dobras)	42
Figura 12 – Variações da z-shape	42
Figura 13 – Variações da c-shape	43
Figura 14 – Grafo $k_{2,q}$ e uma representação B_2 -EPG particular	44
Figura 15 – Grafo k-sun e uma representações B_2 -EPG particular	46
Figura 16 – Grafo k-sun e uma representações B_2 -EPG-Helly particular	47
Figura 17 – Interseção máxima entre dois caminhos B_2 -EPG	49
Figura 18 – Grafo G gerado a partir do Lema 27 usando uma cópia de ϕ	50
Figura 19 – Representação B_2 -EPG do grafo da Figura 18, uma cópia de ϕ , com as seguintes atribuição de ϕ : $1 = 3 = 5 = True$ e $2 = 4 = False$	55

LISTA DE TABELAS

Tabela 1 – Termos e símbolos básicos da Teoria dos Grafos usados neste trabalho.	21
Tabela 2 – Problema de satisfatibilidade	30
Tabela 3 – Problema de 3-satisfatibilidade	31
Tabela 4 – Problema de reconhecimento de grafos B_2 -EPG-Helly	37
Tabela 5 – Problema monótono-nae-3-sat	48

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Objetivos	13
1.2	Organização do Texto	14
2	PRELIMINARES	15
2.1	Conceitos e Definições Iniciais	15
2.2	Tabela de Símbolos	20
3	REFERENCIAL TEÓRICO	22
3.1	A Propriedade Helly	22
3.2	Grafos de Aresta-interseção de Caminhos Sobre uma Grade	23
3.3	Transformações de Tempo Polinomial e Prova de NP-Completeness	28
3.3.1	Exemplos de Problemas NP-Completo	29
4	METODOLOGIA	33
4.1	Metodologia Geral: Técnicas de Demonstração	33
4.2	Metodologia Particular para Grafos EPG	35
4.2.1	Complexidade de Reconhecimento de Grafos B_2 -EPG-Helly	36
4.2.2	Abordagem do Problema	37
5	RESULTADOS DA PESQUISA	39
5.1	Reconhecimento de Grafos B_2 -EPG-Helly	47
6	CONSIDERAÇÕES FINAIS	57
	REFERÊNCIAS	58
A	PROVA DE COMPLEXIDADE DE GRAFOS B_2-EPG (ANEXO)	62

1 INTRODUÇÃO

A **Teoria dos Grafos** é um ramo da matemática e da Ciência da Computação que estuda as relações entre objetos de um determinado conjunto. Com essas relações é possível modelar diversos problemas reais utilizando estruturas de dados chamadas de **grafos**. Neste trabalho de conclusão de curso estudamos representações de grafos através de caminhos sobre uma grade retangular, conhecidas na literatura como representações **EPG (Edge-intersection Paths on a Grid)**. Neste capítulo apresentamos os problemas abordados no trabalho, os objetivos e as motivações para a pesquisa.

Denotamos um grafo não-direcionado por $G = (V, E)$, onde G representa o grafo, $V(G)$ faz referência ao conjunto de **vértices** do grafo G , que deve ser finito e não vazio. E, por último, $E(G)$ representa o conjunto de **arestas** do grafo G , que é composto por pares de elementos tomados de $V(G)$. A título de exemplo, na Figura 1 está ilustrado o grafo $G = 3$ -sun (mais sobre este tipo de grafo na definição 21). Perceba que os conjuntos de vértices e arestas para este grafo, são os seguintes: $V(G) = \{a, b, c, d, e, f\}$; $E(G) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9\}$. Também é interessante notar que cada aresta $e_i \in E(G)$ é definida por um par não-ordenado, e.g. a aresta $e_6 = (a, b)$, a aresta $e_3 = (a, c)$, e a aresta $e_8 = (b, c)$.

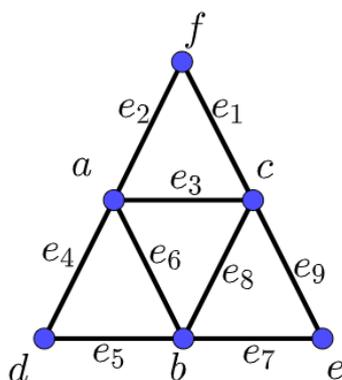


Figura 1 – Grafo G (3-sun)

Neste trabalho abordaremos, principalmente, uma subclasse de grafos de interseção: a classe EPG. Primeiro vamos definir a classe principal. **Grafos de interseção** de uma família de conjuntos são os grafos obtidos quando associamos cada vértice a um conjunto, e a ocorrência de uma interseção entre dois conjuntos existe se e somente se os vértices correspondentes forem adjacentes, ver mais sobre o assunto em Szpilrajn-Marczewski (1945), Kabell (1980) e Pal (2014).

Existem diversos trabalhos na área de Ciência da Computação que manipulam grafos de interseção, como Erdős, Goodman e Pósa (1966), Petit (2002) e Pinto

(2018). Em particular, o estudo de Erdős, Goodman e Pósa (1966) mostrou que todos os grafos podem ser representados por algum modelo de interseção.

A classe de grafos EPG foi definida por Golubic, Lipshteyn e Stern (2009). Essa classe de grafos corresponde exatamente aos grafos de aresta-interseção de caminhos sobre uma grade retangular.

O estudo de grafos de interseção de caminhos sobre uma grade possui alguma motivação prática no desenvolvimento de *layout* de circuitos digitais e *layout* otimização, ver mais em Sinden (1966). Pode ser citado também, como exemplo, o projeto de *layout* de circuitos VSLI, o qual o número de dobras de cada circuito pode aumentar o tamanho do *layout*, dessa forma o custo de fabricação também é aumentado, ver sobre design VSLI em Molitor (1991). Algumas vezes ao desenhar trilhas sobre uma placa *on-board* ou sobre um *microchip* é desejável que os caminhos não se cruzem ou que o número de curvas que cada trilha faz seja o menor possível, estes tipos de aplicações são naturalmente modelados por grafos EPG. Imagine que cada trilha do circuito corresponde a um caminho desenhado sobre a grade e o número de vezes em que cada trilha pode mudar de direção é visto como um problema de otimização correspondente. Ainda, quando modelamos um *layout* com várias camadas, onde os caminhos (fios/trilhas) de uma camada não podem cruzar com os de outra, isso pode ser generalizado como um problema de coloração em grafos.

Um problema interessante, do ponto de vista científico, é o problema de investigação da complexidade de reconhecimento de uma dada classe de grafos, i.e. o problema de decisão que pode ser enunciado da seguinte forma: “Dado um grafo G qualquer, é possível dizer se esse grafo G pertence a uma determinada classe C em tempo polinomial? SIM/NÃO”.

Existem algumas classes de grafos EPG para as quais a complexidade do problema de reconhecimento já foi definida, abordados mais profundamente na Seção 3.2. O problema de reconhecimento de grafos EPG é um problema aberto para B_k -EPG com $k \geq 3$ (PERGEL; RZAŻEWSKI, 2017), quando colocamos a restrição da classe possuir a propriedade Helly, então esse problema está em aberto para B_k -EPG-Helly, com $k \geq 2$, (SANTOS, 2020). Este trabalho se propõe a investigar o problema de reconhecimento de grafos B_2 -EPG-Helly.

1.1 Objetivos

Geral:

Desenvolver ou contribuir para o desenvolvimento de uma prova que demonstre a complexidade de reconhecimento da classe B_2 -EPG-Helly.

Específicos:

- 1 - Explorar representações de grafos B_2 -EPG;
- 2 - Determinar a complexidade do problema de reconhecimento de grafos na classe B_2 -EPG-Helly.

1.2 Organização do Texto

Além deste capítulo de Introdução, o restante deste trabalho contém mais cinco capítulos que estão organizados da seguinte maneira:

- O Capítulo 2 contém as definições iniciais, alguma fundamentação teórica necessária para um bom entendimento deste trabalho e conceitos básicos relacionados à Teoria dos Grafos;
- No Capítulo 3 listamos os trabalhos relacionados e algum referencial teórico com uma breve explicação sobre a propriedade Helly, sobre grafos de interseção e transformações de tempo polinomial;
- No Capítulo 4 apresentamos a metodologia adotada na pesquisa desenvolvida;
- O Capítulo 5 lista todos os resultados alcançados na pesquisa. Neste capítulo o leitor encontrará lemas, definições e teoremas acerca do tema estudado, em particular, este capítulo apresenta entre outros resultados, a prova de NP-Completeness para B_2 -EPG-Helly;
- No Capítulo 6 apresentamos uma breve discussão acerca dos resultados alcançados. Apresentamos também alguns resultados concretos da pesquisa, no formato de papers aceitos para publicação em eventos, e por último deixamos alguma ideia para trabalhos futuros;
- Na sequência aparecem as Referências Bibliográficas e o Anexo A, que contém um link para o paper "*On edge intersection graphs of paths with 2 bends*", de Pergel e Rzażewski (2017), que se trata de uma prova de NP-completeness para o problema de reconhecimento de grafos B_2 -EPG.

2 PRELIMINARES

Nesta seção serão apresentados alguns conceitos iniciais importantes para o entendimento deste trabalho. Além das notações básicas sobre grafos, apresentaremos a propriedade Helly, que é uma propriedade não-trivial que trata de conjuntos mutuamente intersectantes. A seguir o leitor encontrará a seção de definições.

2.1 Conceitos e Definições Iniciais

Dizemos que um **grafo** G é formado pela seguinte estrutura $G = (V, E)$, onde $V(G)$ representa o conjunto de vértices de G e $E(G)$ o conjunto de aresta de G . Na representação geométrica de um grafo, um **vértice** $v_i \in V(G)$ é comumente representado por um ponto ou círculo, e uma **aresta** $e_{i,j} = (v_i, v_j) \in E(G)$, que conecta o vértice $v_i \in V(G)$ ao vértice $v_j \in V(G)$, normalmente é representada por uma linha entre os dois vértices. Quando existe uma aresta $e_{i,j} = (v_i, v_j)$, entre dois vértices v_i e v_j , dizemos que os dois vértices são **adjacentes** entre si. Ainda neste caso, também dizemos que a aresta $e_{i,j} = (v_i, v_j)$ é **incidente** aos vértices v_i e v_j . Podemos denotar a **cardinalidade** dos conjuntos de vértices e arestas de um grafo, respectivamente por $|V(G)| = n$ e $|E(G)| = m$.

O **grau de um vértice** v é o número de arestas incidentes a ele, e é denotado por $d(v)$. Com isso podemos citar também o **grau mínimo** de um grafo descrito como $\delta(G) = \min\{d(v)|v \in V(G)\}$, assim como o **grau máximo** de um grafo denotado por $\Delta(G) = \max\{d(v)|v \in V(G)\}$. A **vizinhança** aberta de um vértice v é representada por $N(v)$ e representa o conjunto de seus vizinhos, já a vizinhança fechada conhecida como $N[v]$, pode ser definida como $N[v] = N(v) \cup \{v\}$, que se trata do conjunto de vizinhos do vértice v incluindo ele mesmo.

Chamamos de **clique** um conjunto de vértices mutuamente adjacentes entre si, i.e. existe uma aresta entre qualquer par de vértices que compõe uma clique. Já um **conjunto independente**, ou ainda **conjunto estável de vértices**, corresponde a um conjunto de vértices não mutuamente adjacentes entre si, i.e. não existe aresta entre qualquer par de vértices no conjunto.

Se existe um grafo $G = (V, E)$ e um grafo $G' = (V', E')$ onde $V' \subseteq V$ e $E' \subseteq E$, dizemos então que G' é um **subgrafo** de G . G' também pode ser considerado **subgrafo induzido** de G por V' , para isso ele deve conter todas as arestas de E que possuem suas extremidades em V' .

Chamamos de **caminho em um grafo** uma sequência finita e alternada de vértices e arestas, em que cada aresta é incidente ao vértice que a sucede e ao que a precede, de forma que não haja repetição de vértices. Se o primeiro vértice coincidir

com o último chamamos essa sequência de **caminho fechado**.

Um **ciclo** (também conhecido de caminho fechado), é um grafo G formado por uma sequência de vértices $v_1, v_2, \dots, v_n, v_1$ que consecutivamente possuem aresta entre si, i.e. o conjunto de arestas é formado por (v_1, v_2) e (v_i, v_{i+1}) , onde $v_i \neq v_j$ para todo $i \neq j$, de forma que $n \geq 3$. O ciclo com n vértices é denotado formalmente como C_n . Dizemos também que um grafo é uma **árvore** quando ele é conexo e não possui ciclos. Um grafo **cordal** trata-se de um grafo onde cada ciclo induzido $C_n, n \geq 4$ possui uma corda. Onde **corda** é uma aresta que liga dois vértices não consecutivos em um ciclo ou caminho.

Chamamos de **grade** o espaço Euclidiano formado por coordenadas inteiras ortogonais, no qual cada par de coordenada de inteiros representa um ponto ou vértice da grade, e dois pontos da grade são adjacentes se estão à uma unidade de distância. A grade será a hospedeira das nossas representações.

EPG é o acrônimo de Edge-intersection Paths on a Grid, em tradução livre, corresponde à classe de grafos que possuem representação sobre uma grade. Em **grafos EPG** denotamos por P_{v_i} o caminho desenhado sobre a grade que corresponde ao vértice v_i de um grafo G . O conceito de **caminho na representação** não se confunde com o conceito de caminho em um grafo, uma vez que vamos denotar o caminho na representação como um conjunto de arestas (e não de vértices), de forma que o caminho é considerado como uma sequência finita de arestas consecutivas $e_1 = (v_1, v_2)$, $e_2 = (v_2, v_3)$, $e_3 = (v_3, v_4)$, \dots , $e_i = (v_i, v_{i+1})$, \dots , $e_m = (v_m, v_{m+1})$, onde $e_i \neq e_j$ para $i \neq j$.

Uma **representação EPG** de um grafo G , denotada por R_G , é um modelo de aresta-interseção de caminhos sobre uma grade. Existe uma relação bijetiva entre o conjunto de vértices e arestas do grafo G e sua representação EPG correspondente. Cada vértice $v_i \in V(G)$ possui um caminho P_{v_i} relacionado, e dois vértices v_i, v_j são adjacentes entre si se e somente se os caminhos correspondentes (P_{v_i} e P_{v_j}) possuem interseção em pelo menos uma aresta.

Uma **dobra** é definida como um par de arestas consecutivas e_1 e e_2 , quando elas possuem direções diferentes na grade. Quando isso acontece e_1 e e_2 são chamadas de arestas de dobra. Se um caminho não possui dobras ele é chamado de **segmento**. Ainda sobre representações EPG, dizemos que uma representação é **B_k -EPG** quando todos os caminhos da representação possuem no máximo k dobras, i.e. k mudanças de direção, onde $k \geq 0$ e é um número inteiro. O parâmetro **bend number** ou **número de dobras** de um grafo G , denotado por $b(G)$, corresponde ao menor inteiro k para o qual G possui uma representação B_k -EPG. Também faz sentido falar no *bend number* de uma classe de grafos $b(\text{classe } C)$, que é o menor k para o qual todos os grafos da classe C possuam representação B_k -EPG. A Figura 2 mostra uma representação B_1 -EPG do grafo 3-sun apresentado Figura 1, esta representação é chamada de B_1 -EPG

porque todos os caminhos apresentados nela possui no máximo uma dobra. Podemos ver representações B_0 -EPG e B_2 -EPG na Figura 7 e Figura 9 respectivamente, da seção 3, na subseção 3.2.

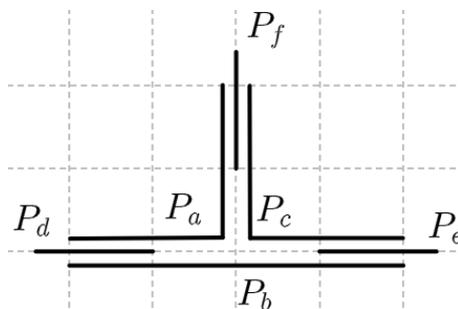


Figura 2 – Representação B_1 -EPG do grafo 3-sun da Figura 1

Dados dois caminhos, P_i e P_j , de uma representação EPG, se esses dois caminhos são aresta-intersectantes, denotamos por **interseção por arestas** $P_i \cap_e P_j$ a operação que retorna o conjunto de arestas da aresta-interseção dos caminhos. Se P_i e P_j se cruzam porém não compartilham arestas sobre a grade, chamamos de **interseção por vértices** $P_i \cap_v P_j$ a operação que retorna o conjunto de vértices que P_i e P_j possuem em comum. Nesse último caso $P_i \cap_e P_j = \emptyset$.

Agora vamos introduzir o conceito de **complexidade**. Na Ciência da Computação o conceito de complexidade faz referência à eficiência de um algoritmo ao executar uma certa tarefa. Dessa forma, a complexidade está ligada ao tempo que o algoritmo leva para ser executado e/ou à memória gasta durante a execução. A principal ferramenta que usamos para analisar a complexidade de um algoritmo é a **análise assintótica**.

A análise assintótica é o estudo da eficiência assintótica de um algoritmo, i.e. na **análise assintótica** consideramos entradas grandes o suficiente de forma que possamos analisar o comportamento de limiar de tempo de execução do problema. Com isso podemos definir a complexidade de **piores caso**, que se aplica quando a função de tempo de execução (ou espaço) do algoritmo assume os máximos valores possíveis. Vale dizer que a análise assintótica pode ser observada de outros pontos de vista, i.e. faz sentido falar também em notação de **médio caso** e **melhor caso**. Para questões práticas deste estudo vamos considerar a análise de pior caso a mais relevante.

Basicamente podemos dizer que a análise assintótica se trata da comparação de funções que representam o tempo de execução de um algoritmo, com isso temos a notação assintótica que se define em três tipos, \mathcal{O} , Ω e θ . Neste trabalho iremos focar na notação Big \mathcal{O} , dessa forma consideremos duas funções $f(n)$ e $g(n)$, dizemos que $f(n)$ é $\mathcal{O}(g(n))$, se existe as constantes positivas c e n_0 , tal que $f(n) \leq cg(n)$, para $n \geq n_0$. O quer dizer que $f(n)$ é limitada superiormente por $g(n)$ para $n \geq n_0$ e aplicando a constante c .

Consideramos como **problema tratável** aquele para o qual é conhecido um algoritmo polinomial capaz de resolvê-lo, i.e. o seu consumo de tempo do pior caso é limitado por uma função polinomial da entrada do problema. Já os problemas para os quais não se conhecem algoritmos de tempo polinomial capaz de resolvê-lo, atribuímos-lhes o nome de **problemas intratáveis**. A questão da (in)tratabilidade de problemas está intimamente relacionada com a questão $\mathcal{P} \stackrel{?}{=} \mathcal{NP}$, ver mais em Garey e Johnson (1979).

Os problemas computacionais podem ser divididos basicamente em três classes: a primeira classe corresponde aos **problemas de decisão**. Nos problemas de decisão o objetivo é decidir sobre uma dada entrada. A resposta ao questionamento de entrada será algo do tipo YES/SIM ou NO/NÃO; a segunda classe corresponde aos **problemas de localização**. Nesses problemas, o objetivo é encontrar uma estrutura na instância de entrada que satisfaça aos requisitos solicitados pela questão apresentada no problema de localização; e a terceira classe de problemas corresponde exatamente à classe de **problemas de otimização**, onde o objetivo é satisfazer algum critério de otimização, e.g. maximização ou minimização, ver mais em Szwarcfiter (2018) e Garey e Johnson (1979).

Sobre as três classes de problemas, citadas anteriormente, é importante saber que para um dado problema P , geralmente, existe um problema de otimização, um problema de localização, e ainda um problema de decisão associado. Geralmente, o problema de decisão é de dificuldade não maior que um problema de localização. Da mesma forma, o problema de localização é de dificuldade não maior que um problema de otimização, ver Szwarcfiter (2018). A título de exemplo, a seguir apresentamos o problema do caixeiro viajante nas três versões: problema de otimização, problema de localização e problema de decisão.

Problema do Caixeiro Viajante (Otimização)	
<i>Entrada:</i>	Um grafo G com pesos nas arestas.
<i>Objetivo:</i>	Encontrar no grafo G um percurso de caixeiro viajante que possua peso mínimo (seja ótimo).

Problema do Caixeiro Viajante (Localização)	
<i>Entrada:</i>	Um grafo G com pesos nas arestas e um inteiro $k > 0$.
<i>Objetivo:</i>	Encontrar em G um percurso de caixeiro viajante de peso $\leq k$.

Agora tomemos como base o problema de decisão do caixeiro viajante definido acima. Dado um grafo completo G onde cada aresta possui um peso $p \geq 0$, um percurso do caixeiro viajante se trata de um percurso (ciclo) hamiltoniano do grafo G .

Problema do Caixeiro Viajante (Decisão)

Entrada: Um grafo G e um inteiro $k > 0$.

Objetivo: Verificar se o grafo G possui um percurso de peso $\leq k$.

Dessa forma, um percurso ótimo para G é o percurso no qual a soma dos pesos das arestas é o menor possível.

Quando falamos de problemas de decisão muitas vezes nos referimos a eles como $\Pi(D, Q)$, onde D se refere ao conjunto de dados de entrada, e Q é a questão à qual estamos nos referindo. Agora assumimos que temos dois problemas de decisão Π e Π' , se Π for transformável em Π' em tempo polinomial então dizemos que $\Pi \propto \Pi'$, i.e. existe um algoritmo A_1 capaz de transformar o problema Π , através de um número polinomial (com relação ao tamanho da entrada) de passos, no problema Π' .

A redução polinomial é interessante porque ela não diz nada, inicialmente, sobre a dificuldade de resolver Π . Porém, perceba que no caso de existência de um algoritmo A_2 capaz de resolver Π' somos capazes de aplicar A_2 para resolver Π . Logo, a dificuldade de resolução de Π não pode ser menor que a dificuldade de resolver Π' . Agora sim, com essa análise, conseguimos definir a relação de dificuldade entre Π e Π' . Na subseção 3.3, deste capítulo, estudaremos mais profundamente transformações de tempo polinomial.

Apresentado o que é um problema de decisão Π , podemos então entender o que são os problemas de complexidade polinomiais e os problemas de complexidade NP-Completo. A classe de problemas \mathcal{P} corresponde exatamente à classe de problemas de decisão para os quais existe uma Máquina de Turing Determinística que resolve Π em tempo polinomial. Já classe de problemas \mathcal{NP} equivale à classe de problemas de decisão em que existe uma Máquina de Turing não Determinística que resolve Π em tempo polinomial.

Devemos lembrar que resolver um problema em uma Máquina de Turing não Determinística equivale à verificar o problema em uma Máquina de Turing Determinística. Este fato se torna muito relevante para definirmos a classe NP-Completo. Dado um problema de decisão Π , a classe NP-Completo é definido como a classe de problemas de decisão em que: a) $\Pi \in \mathcal{NP}$, i.e. Π pode ser verificado por uma Máquina de Turing Determinística; b) Qualquer problema de decisão $\Pi' \in \mathcal{NP}$, satisfaz $\Pi' \propto \Pi$, i.e. Π' pode ser transformado polinomialmente em Π .

Neste ponto do trabalho consideramos que o leitor conheça as definições básicas sobre Máquinas de Turing. Acreditamos que não seja pertinente inserir esses conceitos neste escrito, porém maiores referências sobre Máquinas de Turing podem ser facilmente encontradas, (MENEZES, 1998), (HOPCROFT; MOTWANI; ULLMAN, 2001),(GAREY; JOHNSON, 1979).

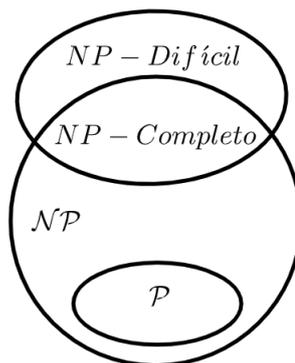


Figura 3 – Diagrama de Complexidade $\mathcal{P} \neq \mathcal{NP}$

A Figura 3 ilustra um possível cenário, onde $\mathcal{P} \neq \mathcal{NP}$. Se por acaso, algum dia, alguém provar que $\mathcal{P} = \mathcal{NP}$, as classes de problemas \mathcal{P} , \mathcal{NP} e $\mathcal{NP-Completo}$, colapsariam em uma só. Como carecemos desta demonstração, para título de estudo, consideramos que $\mathcal{P} \neq \mathcal{NP}$.

2.2 Tabela de Símbolos

A seguir apresentamos uma tabela com os símbolos usados neste trabalho e seus respectivos significados:

Símbolo	Descrição
$G = (V, E)$	Grafo G com o conjunto de vértices $V(G)$ e conjunto de arestas $E(G)$.
$V(G)$	Conjunto de vértices de G .
$E(G)$	Conjunto de arestas de G .
$n(G)$	Número de vértices em G .
$m(G)$	Número de arestas em G .
v_i	vértice v_i .
P_{v_i}	Caminho correspondente ao vértice v_i .
$e = (v_i, v_j)$	Aresta e com pontos finais v_i e v_j .
$d(v)$	Grau de vértice v .
$\delta(G)$	Grau mínimo de um vértice em G .
$\Delta(G)$	Grau máximo de um vértice em G .
$N(v)$	Vizinhança aberta do vértice v .
$N[v]$	Vizinhança fechada do vértice v .
$G[S]$	Subgrafo induzido de G pelo subconjunto de vértices S .
$ S $	Cardinalidade do conjunto S .
C_n	Ciclo induzido com n vértices.
W_n	Grafo roda com n vértices.
$K_{r,s}$	Grafo bipartido completo com partes de tamanho r e s .
K_n	Grafo completo ou clique com n vértices.
B_k -representação	Representação onde cada caminho tem no máximo k dobras.

Tabela 1 – Termos e símbolos básicos da Teoria dos Grafos usados neste trabalho.

3 REFERENCIAL TEÓRICO

Neste capítulo listamos os trabalhos relacionados aos grafos EPG, apresentamos um referencial teórico que explica um pouco sobre a propriedade Helly, sobre grafos de interseção, e por fim apresentamos conceitos de transformações de tempo polinomial e prova de NP-Completeness.

3.1 A Propriedade Helly

Uma propriedade muito famosa da Teoria dos Conjuntos é a **Propriedade Helly**. Essa propriedade de conjuntos intersectantes pode ser definida da seguinte forma: Uma coleção de conjuntos satisfaz a propriedade Helly quando toda subcoleção que é mutuamente intersectante possui pelo menos um elemento em comum.

Dados os seguintes conjuntos: $C_1 = \{1, 2, 3\}$, $C_2 = \{2, 3, 4\}$, $C_3 = \{3, 4, 5\}$, $C_4 = \{3, 4, 5\}$, $C_5 = \{2, 4, 6\}$, tome duas coleções de conjuntos $S_1 = \{C_1, C_2, C_3, C_4\}$ e $S_2 = \{C_1, C_2, C_3, C_5\}$. Como pode ser visto a coleção S_1 é formada por conjuntos onde todos possuem elementos em comum, e ainda mais, um elemento é comum em todos eles ao mesmo tempo, o elemento 3. Dessa forma, podemos afirmar que a coleção S_1 satisfaz a propriedade Helly. De forma semelhante, na coleção S_2 , se tomarmos dois conjuntos quaisquer, eles terão um elemento em comum, mas se verificarmos a interseção de todos os conjuntos simultaneamente, então não haverá um elemento comum a todos os conjuntos.

A propriedade Helly possui aplicação em diversas áreas da ciência, como teoria do código, banco de dados, processamento de imagens, problemas de localização e programação linear, (DOURADO; PROTTI; SZWARCFITER, 2006), (JUNIOR, 2016). Em particular, na Teoria dos Grafos há diversos trabalhos que abordam a propriedade Helly, e.g. Mulder e Schrijver (1979), Lin e Swarcfiter (2007), Dourado et al. (2008) e Safe (2016).

Na Figura 4 podemos observar melhor como funciona essa propriedade. Digamos que cada elipse das imagens representa um conjunto, dessa forma se dois desses conjuntos se intersectam então eles possuem pelo menos um elemento em comum. Na Figura 4 (a) podemos ver que todos os conjuntos se intersectam, mas os elementos compartilhados por eles não são os mesmos, dessa forma não satisfazendo a propriedade Helly. Já na Figura 4 (b) os três conjuntos se intersectam em um mesmo ponto, mostrando assim que os três possuem pelo menos um elemento em comum, o que faz com que eles satisfaçam à propriedade em questão.

Em uma representação EPG de um grafo G , observamos a propriedade Helly com relação às aresta-interseções dos caminhos. Quando uma coleção de caminhos é mutuamente intersectante, se essa coleção de caminhos compartilha pelo menos uma aresta em comum da representação então ela é uma representação Helly. A Fi-

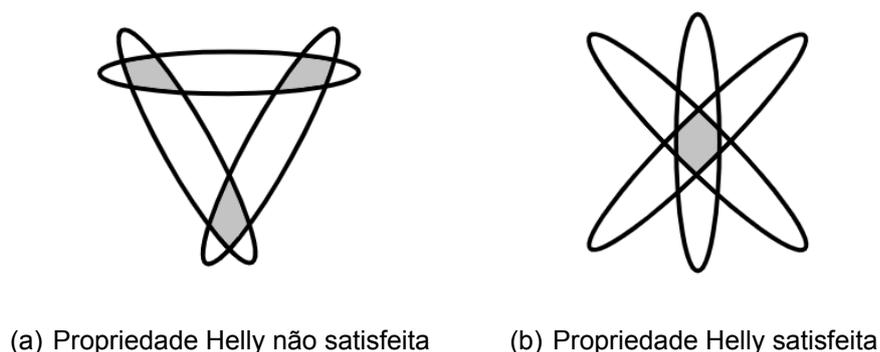


Figura 4 – Propriedade Helly

Figura 5(a) ilustra o grafo G_6 , na Figura 5(b) há uma representação B_1 -EPG-Helly do grafo G_6 e a Figura 5(c) apresenta uma representação B_1 -EPG de G_6 que não é Helly. No grafo G_6 o subgrafo induzido pelo conjunto de vértices $\{a, b, c\}$ forma uma clique, induzida por $G_6[a, b, c]$, e os caminhos correspondentes P_a, P_b e P_c compartilham uma aresta em comum na grade, o que satisfaz à propriedade Helly, portanto a representação é B_1 -EPG-Helly. Enquanto isso, na Figura 5(c) os caminhos P_a, P_b e P_c não compartilham nenhuma aresta da grade simultaneamente, não satisfazendo dessa forma à propriedade Helly, logo essa seria uma representação B_1 -EPG não Helly.

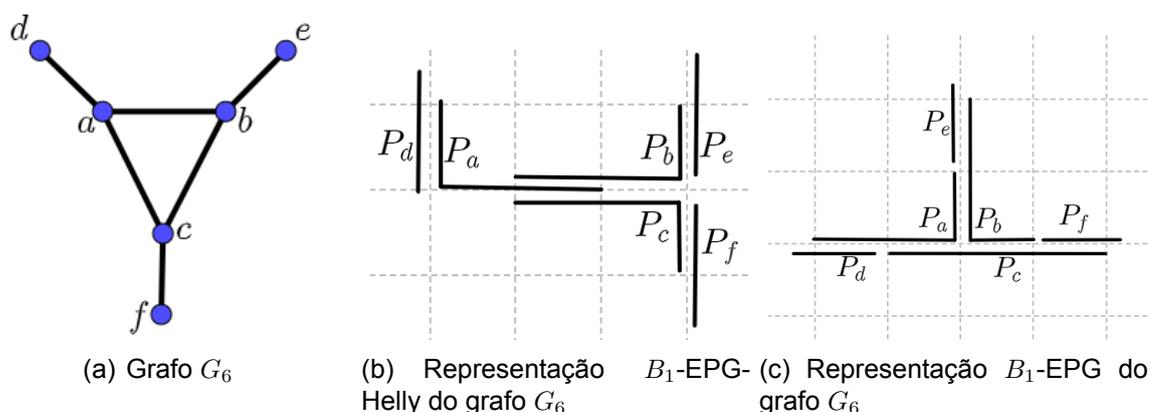


Figura 5 – Grafo G_5 de 6 vértices

3.2 Grafos de Aresta-interseção de Caminhos Sobre uma Grade

Historicamente, as primeiras classes de grafos de interseção de caminhos estudadas na literatura foram tomadas em uma representação cujo hospedeiro era uma árvore. De acordo com a interseção considerada nesses caminhos essas classes poderiam se subdividir em grafos de vértice-interseção de caminhos sobre uma árvore, os grafos VPT; ou, analogamente, grafos de aresta-interseção de caminhos sobre uma árvore, os grafos EPT.

Consideremos P como uma família de caminhos sobre um árvore hospedeira T , em uma representação VPT(P) de um grafo G . Cada vértice do grafo G corresponde a um caminho de VPT(P), e dois vértices são adjacentes (no grafo G) se e somente se, na representação, os caminhos correspondentes compartilham pelo menos um vértice na árvores T . Já uma representação EPT(P), cada membro da família de caminhos também corresponde a um vértice do grafo G sob EPT(P), mas dois vértices são adjacentes se e somente se os caminhos correspondentes em P compartilham pelo menos uma aresta em T .

Essas duas classe de grafos de interseção foram amplamente estudadas na literatura e há abundante material investigando problemas relacionados a elas, (GAVRIL, 1978), (GOLUMBIC; JAMISON, 1985b), (GOLUMBIC; JAMISON, 1985a), (ALCÓN; GUTIERREZ; MAZZOLENI, 2010), (BOYACI et al., 2013). Os grafos de interseção de caminho em uma árvore (EPT) são relacionados aos grafos de interseção de caminho em uma grade (EPG) pois consideram interseções de caminhos, porém sobre hospedeiros distintos (árvore e grade, respectivamente), ver por exemplo Gavril (1974) e Golubic, Lipshteyn e Stern (2004).

No ano de 2009 surgiram as primeiras definições formais de grafos EPG, apresentadas no trabalho de Golubic, Lipshteyn e Stern (2009). EPG é o acrônimo para Edge-intersection Paths on a Grid, que de forma mais direta se trata da representação da classe de grafos de aresta-interseção de caminhos sobre uma grade. Em outras palavras, nessa classe temos a representação de um grafo G em uma grade retangular Q , por meio de caminhos, onde cada vértice $v_i \in V(G)$ será representado como um caminho $P_i \in P$, onde P é o conjunto de caminhos de Q .

Este tipo de representações são caracterizadas pelo seguinte comportamento de interseção: dois vértices $v_i, v_j \in V(G)$ que são adjacentes em G , terão seus respectivos caminhos P_{v_i} e $P_{v_j} \in P$, onde P é o conjunto de caminhos da representação, compartilhando uma ou mais arestas da grade. Da mesma forma se dois caminhos pertencentes a P , sobre a grade Q , e compartilham pelo menos uma aresta, então seus respectivos vértices são adjacentes em G .

As representações EPG de um grafo podem conter mudanças de direção, aqui chamadas de **dobras**. Chamamos de **segmento** um caminho sem dobra. A dobra divide o caminho em segmentos de diferentes direções. Um caminho pode ou não conter dobras. A existência de dobras em uma caminho, às vezes, facilita sua representação. Eventualmente é útil limitar a quantidade de vezes que um caminho pode dobrar. Quando todos os caminhos de uma representação EPG possuem no máximo k dobras, então dizemos que esta é uma representação B_k -EPG. Assim, essa restrição nos permite classificar os grafos EPG em subclasses, e.g. B_0 -EPG, B_1 -EPG, B_2 -EPG, ..., B_k -EPG, para algum $k \geq 0$.

Do ponto de vista científico, o estudo de grafos EPG é um tema novo e bas-

tante ativo. Os seguintes trabalhos atuais atacaram problemas envolvendo grafos EPG, Pergel e Rzażewski (2017), Alcón et al. (2018), Bornstein et al. (2020). A classe dos grafos EPG ainda é bastante explorada por pesquisadores que investigam desde problemas clássicos da Teoria de Grafos como conjunto independente e cômputo de cliques, (BOUGERET et al., 2015), (BOUSQUET; HEINRICH, 2017), até problemas de caracterização, complexidade, delimitação de parâmetros particulares e coloração como Epstein, Golumbic e Morgenstern (2013) e Bonomo, Mazzoleni e Stein (2017). É possível perceber que apesar do estudo na classe dos grafos EPG ser diverso, ainda existe uma vastidão de problemas abertos na literatura. Podemos citar, por exemplo, o caso do problema de reconhecimento de grafos B_k -EPG, aberto para $k \geq 3$, ou ainda o problema de reconhecimento de grafos B_k -EPG-Helly, aberto para $k \geq 2$. A grande maioria dos problemas estudados na literatura de grafos EPG investiga os grafos B_k -EPG para $k \leq 1$ e suas subclasses, e.g. Epstein, Golumbic e Morgenstern (2013), Cameron, Chaplick e Hoàng (2013) e Santos et al. (2021).

A classe de grafos B_2 -EPG foi abordada em poucos trabalhos. Como resultados encontrados na literatura podemos citar os trabalhos de Pergel e Rzażewski (2017), que aborda um estudo da complexidade computacional do problema de reconhecimento da classe. Em tempo, outros trabalhos também pesquisaram a classe B_2 -EPG, com outros propósitos, podemos citar, por exemplo, os trabalhos de Biedl e Stern (2010), Heldt, Knauer e Ueckerdt (2014b) e Francis e Lahiri (2016), que serão abordados em maiores detalhes adiante. Neste trabalho, nossa proposta é investigar o problema de reconhecimento de uma subclasse dos grafos B_2 -EPG.

A seguir listamos problemas relacionados a grafos EPG.

Algumas subclasses de grafos EPG coincidem ou estão propriamente contidas em classes de grafos já conhecidas na literatura. Podemos citar, por exemplo, os grafos de intervalos que são equivalentes à classe B_0 -EPG (GOLUMBIC; LIPSHTEYN; STERN, 2009); os ciclos e as árvores foram classificadas como grafos que possuem representação B_1 -EPG (GOLUMBIC; LIPSHTEYN; STERN, 2013); os grafos outerplanares estão em B_2 -EPG (HELDT; KNAUER; UECKERDT, 2014b); já os planares não foram bem classificados, porém sabe-se que eles estão em B_k -EPG, onde k pertence a $\{3,4\}$ com k não vazio, (HELDT; KNAUER; UECKERDT, 2014b).

Outras classificações relevantes são os grafos de linhas e os grafos planares bipartite que de acordo com Biedl e Stern (2010) estão em B_2 -EPG, assim como os grafos arco-circulares normais e os grafos Halin respectivamente nos trabalhos de Alcón et al. (2018) e Francis e Lahiri (2016).

Sobre o problema de reconhecimento dos grafos B_k -EPG, podemos citar alguns resultados obtidos sobre o assunto, um exemplo é a classe de grafos B_0 -EPG que podem ser reconhecidos em tempo polinomial, já que equivalem aos grafos de intervalo, (LEKKEIKERKER; BOLAND, 1962); a classe B_0 -EPG-Helly também são reconheci-

das em tempo polinomial, pois coincidem com a classe B_0 -EPG (BORNSTEIN et al., 2020); por outro lado, a classe B_1 -EPG é de reconhecimento NP-Completo (HELDT; KNAUER; UECKERDT, 2014a), assim como a classe B_1 -EPG-Helly (BORNSTEIN et al., 2020); por fim, a classe B_2 -EPG também possui o reconhecimento NP-Completo, visto em Pergel e Rzażewski (2017), apesar disso o problema de reconhecimento de B_2 -EPG-Helly permanece em aberto.

As representações B_0 -EPG particularmente são compostas por um conjunto bem restrito de grafos, dado as suas limitações de representação. Uma classe de grafos já identifica ser pertencente a B_0 -EPG é a classe dos grafos de Intervalo (GOLUMBIC; LIPSHTEYN; STERN, 2009), ver exemplo na Figura 6. Isso pode ser visto claramente pelo fato de que um grafo de intervalo pode ser representado por um conjunto de intervalos de segmentos sobre uma reta real, dessa forma podemos representar esse mesmo grafo em uma reta horizontal ou vertical de uma grade retangular.

Uma característica dos grafos de intervalo é que eles são formados basicamente por grafos de acordes (são grafos em que cada ciclo formado por mais de três vértices possui aresta ou arestas que não faz parte do ciclo, mas conecta dois vértices do ciclo, vale dizer que grafos de intervalo são grafos que cada ciclo induzido possui exatamente três vértices), e grafos perfeitos (são grafos onde a ordem do maior clique presente no grafo determina o número cromático de qualquer subgrafo induzido dele). Outro fato, é que eles podem ser reconhecidos em tempo linear, e encontrar a clique máxima e a coloração ótima para esse tipo de grafo é possível em tempo polinomial. Com isso podemos ver que todos grafos caminho podem ser representados em B_0 -EPG, assim como ciclos de tamanho 3.

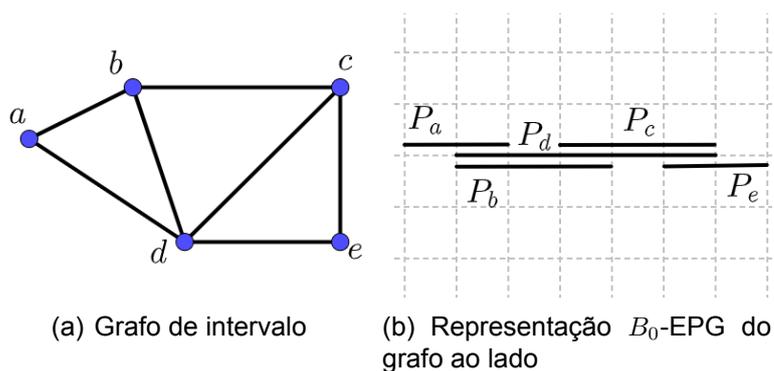


Figura 6 – Representação B_0 -EPG

Existem na literatura algumas classes de grafos cuja pertinência a B_1 -EPG é conhecida. Podemos citar os grafos árvores e os ciclos (GOLUMBIC; LIPSHTEYN; STERN, 2013). No trabalho de Golumbic, Lipshteyn e Stern (2013), foi apresentado um procedimento recursivo que é capaz de produzir uma representação B_1 -EPG de qualquer grafo árvore. Esse procedimento funciona basicamente escolhendo um vértice

tice raiz, que ocuparia todos os pontos da primeira coluna e os da linha inferior, logo em seguida divide-se as linhas restantes igualmente entre os n vizinhos do vértice raiz. E esse passo é repetido para cada novo caminho adicionado na representação. Para mais detalhes, ler Golubic, Lipshteyn e Stern (2013). Nesse mesmo trabalho nos é apresentado outro resultado interessante, o fato de que toda clique de um grafo pode ser representada por uma **edge-clique** ou **claw-clique** na representação B_1 -EPG. Outro resultado útil, presente nesse paper é a afirmava que todo ciclo de tamanho 4 possui representações na forma de **true pie**, **false pie** e **frame**.

Essas cinco formas de representações citadas anteriormente são úteis para construir representações B_1 -EPG. De posse dessas informações sobre as possíveis representações dessas estruturas, então podemos começar a criar uma representação de grafos utilizando-as. Abaixo nas figuras 7 e 8 podemos encontrar exemplos de representações das estruturas citadas.

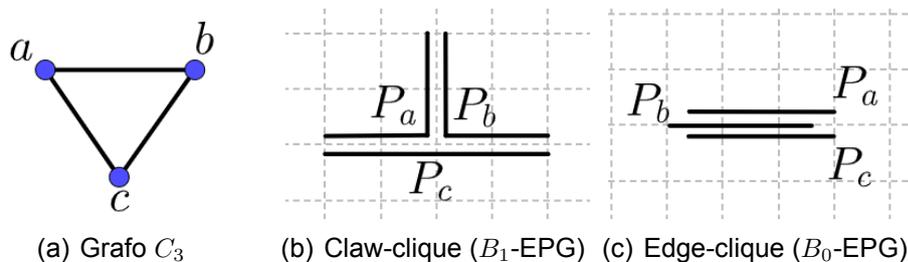


Figura 7 – Representações claw-clique e edge-clique

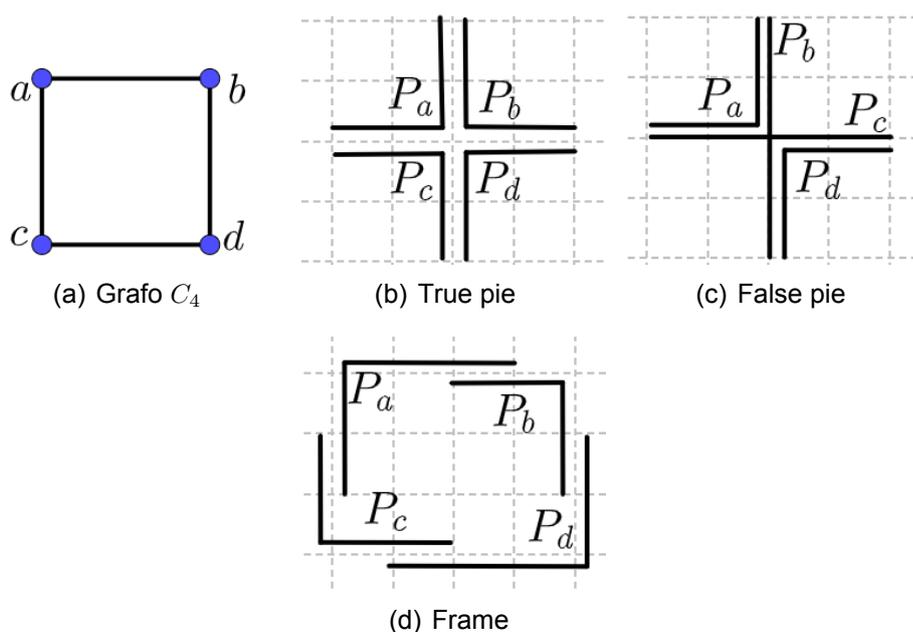


Figura 8 – Representações de um ciclo induzido de tamanho 4 por True pie, False pie e Frame

As representações B_2 -EPG são aquelas onde cada um dos caminhos (da representação) possui no máximo duas dobras, um exemplo deste tipo de representação pode ser vista na Figura 9. Essa subclasse de grafos EPG possui uma grande quantidade de temas que podem ser estudados, visto que possui poucos trabalhos abordando esse tipo de representação. Como um dos trabalhos seminais nessa classe, talvez o mais importante até agora, podemos citar Pergel e Rzażewski (2017). Esse paper nos apresenta a prova de complexidade de reconhecimento dos grafos B_2 -EPG. Ele pode ser encontrado no fim deste trabalho como um anexo.

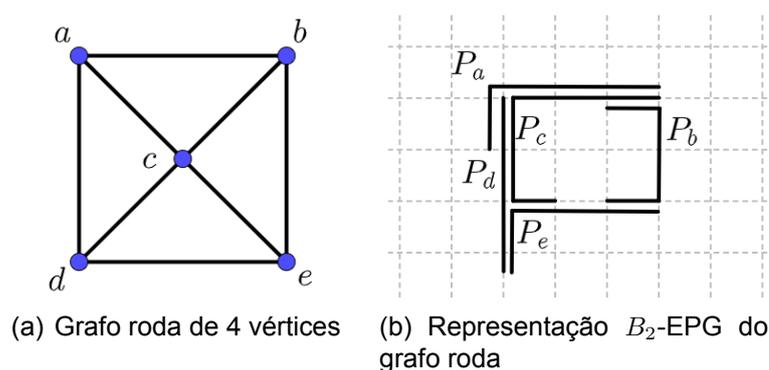


Figura 9 – Representação B_2 -EPG

A produção deste trabalho está sendo fundamentado sobre o trabalho de Pergel e Rzażewski (2017), pois a classe B_2 -EPG-Helly é uma subclasse de B_2 -EPG, (BORNSTEIN et al., 2020). Dessa forma algum resultado já conhecido para B_2 -EPG pode ser útil no roteiro de prova de NP-completude de B_2 -EPG-Helly. Em um primeiro momento pretendemos voltar nossos esforços para estudar a transformação polinomial usada na prova de NP-completude da classe B_2 -EPG. Em seguida, pretendemos responder a seguinte pergunta: A transformação polinomial usada na classe B_2 -EPG também pode ser aplicada na classe B_2 -EPG-Helly?

A seguir vamos estudar mais um pouco sobre reduções polinomiais e prova de dificuldade.

3.3 Transformações de Tempo Polinomial e Prova de NP-Completo

Nesta seção descrevemos resumidamente os principais fundamentos de uma transformação polinomial e a ideia básica para um bom entendimento de como funcionarão os próximos passos deste projeto de graduação. Uma explicação mais detalhada sobre transformação polinomial pode ser encontrada em Szwarcfiter (2018).

Uma transformação de tempo polinomial se dá pela seguinte ideia: dados dois problemas de decisão $\Pi_1(D_1, Q_1)$ e $\Pi_2(D_2, Q_2)$ e um algoritmo A_2 que seja possível resolver o problema Π_2 , temos como objetivo resolver o problema Π_1 com o algoritmo A_2 . Para isso é preciso transformar o problema Π_1 no problema Π_2 e da mesma forma

transformar a solução de Π_2 em uma solução de Π_1 . A ideia principal é resolver o problema Π_1 através do algoritmo A_2 que resolve o problema Π_2 , de forma que todas os passos da transformação de um problema para outro sejam feitos em tempo polinomial.

De forma geral, precisamos transformar uma instância do problema Π_1 em uma instancia do problema Π_2 , para assim então resolver o problema Π_1 com o algoritmo A_2 que resolve Π_2 . Com a resolução do problema precisamos apenas transformar a resposta obtida do algoritmo A_2 em uma resposta de Π_1 . Dessa forma se todas as transformações, de Π_1 para Π_2 e da solução de Π_2 em uma solução para Π_1 ocorrer em tempo polinomial dizemos então que ocorreu uma transformação polinomial e que Π_1 é polinomialmente transformável em Π_2 .

Baseado na definição de Cook (1971), podemos formalizar uma transformação polinomial de um problema de decisão $\Pi_1(D_1, Q_1)$ em um problema de decisão $\Pi_2(D_2, Q_2)$ pela seguinte função $f : D_1 \rightarrow D_2$, que satisfaça o seguinte:

- A função f pode ser computada em tempo polinomial, e
- Dada uma instância $I \in D_1$ do problema Π_1 , $\Pi_1(I)$ possuirá uma resposta SIM apenas se $\Pi_2(f(I))$ também possuir resposta SIM.

A seguir, apresentamos um exemplo de transformação polinomial.

3.3.1 Exemplos de Problemas NP-Completo

Dentro da classe de problemas \mathcal{NP} existem algumas subclasses, e uma delas, a que mais nos interessa no momento é a classe NP-Completo. Essa classe é formada por todos os problemas de decisão Π' , tal que $\Pi' \in \mathcal{NP}$, e Π' pode ser transformado polinomialmente em um problema Π NP-Completo (ou NP-difícil). Isso implica dizer que Π' e Π tem grau de dificuldade equivalente, compartilhado entre todos os os problemas da classe NP-completo, i.e. Π' está entre os problemas mais difíceis da classe \mathcal{NP} . Vamos imaginar agora que um dia alguém consiga provar que um problema da classe NP-Completo pode ser resolvido por um algoritmo de tempo polinomial, então isso implicaria que todos os outros problemas pertencentes a essa classe, inclusive Π' , também poderiam ser resolvidos em tempo polinomial, uma vez que existe uma função de redutibilidade de tempo polinomial entre todos os problemas da Classe NP-completo. Além disso, um algoritmo de tempo polinomial determinístico para resolver algum problema NP-completo seria suficiente para resolver uma grande indagação matemática, a questão $\mathcal{P} \stackrel{?}{=} \mathcal{NP}$.

Os problemas pertencentes a classe NP-Completo são considerados os problemas de maior dificuldade dentro da classe \mathcal{NP} . Como já mencionado, os problemas NP-completos compartilham a propriedade de serem equivalentes entre si, i.e. podem ser transformados polinomialmente entre si, (COOK, 1971).

Agora que sabemos o que é um problema NP-Completo, podemos então apresentar um roteiro de como se prova que um problema de decisão é NP-Completo. Dado um problema Π , para ele ser classificado NP-Completo é necessário cumprir dois requisitos: o primeiro é mostrar que $\Pi \in \mathcal{NP}$, para provar isso, é suficiente, fornecer um certificado de tempo polinomial para Π , em outras palavras isto seria equivalente a dada uma instância I do problema Π e uma solução S para Π , devemos mostrar que somos capazes de verificar deterministicamente em tempo polinomial se S corresponde a uma instância-SIM de Π . Se formos capazes de apresentar tal certificado, então mostramos que $\Pi \in \mathcal{NP}$. Já o segundo requisito pode ser um pouco mais trabalhoso, pois se trata de provar que todo problema de decisão $\Pi' \in \text{NP-Completo}$ pode ser transformado polinomialmente em Π .

Formalmente para um problema Π ser NP-Completo deve:

- (i) $\Pi \in \mathcal{NP}$, e
- (ii) Dado um problema $\Pi' \in \text{NP-Completo}$, então $\Pi' \propto \Pi$.

A seguir apresentaremos um passo a passo de como funciona uma prova de NP-Completo. Para isso, iremos usar o problema de Satisfatibilidade (SAT) de expressões booleanas como alvo da nossa redução, para mostrar que o problema de 3-Satisfatibilidade (3SAT) $\in \text{NP-Completo}$. Mas antes de começar a prova é necessário primeiro entendermos o que são os problemas SAT e 3SAT.

Primeiramente, **SAT** é um problema booleano na **Forma Normal Conjuntiva** (FNC), i.e. corresponde a uma fórmula formada por conjunções de cláusulas disjuntivas, e.g. $(x_1 \vee x_2) \wedge (x_2 \vee x_3 \vee x_4 \vee x_5)$. Este problema é considerado o primeiro problema provado pertencer a classe NP-Completo (COOK, 1971), dessa forma usaremos ele como alvo da redução. A seguir apresentaremos duas tabelas com os problemas de Satisfatibilidade e 3-Satisfatibilidade no “formato padrão”.

SATISFATIBILIDADE (SAT)	
<i>Entrada:</i>	Uma fórmula F no formato FNC.
<i>Objetivo:</i>	F é satisfatível?

Tabela 2 – Problema de satisfatibilidade

Já o 3SAT é um caso particular do SAT, onde cada cláusula da fórmula possui exatamente três variáveis.

Apresentamos agora o resultado de que 3SAT $\in \text{NP-Completo}$, tal resultado foi demonstrado por Karp (1972), mostraremos então uma redução simplificada.

Teorema 1. (Teorema de (KARP, 1972)) 3SAT $\in \text{NP-Completo}$.

3-SATISFATIBILIDADE (3SAT)	
<i>Entrada:</i>	Uma fórmula F no formato FNC, onde cada cláusula possui exatamente 3 variáveis.
<i>Objetivo:</i>	F é satisfatível?

Tabela 3 – Problema de 3-satisfatibilidade

Demonstração. Para mostrar que o problema 3SAT é NP-completo, devemos mostrar que (i) 3SAT pertence a \mathcal{NP} ; e (ii) mostrar que 3SAT é NP-difícil.

- (i) Primeiro vamos efetuar a prova de NP-pertinência, logo devemos mostrar que $3SAT \in \mathcal{NP}$:

Para mostrar que $3SAT \in \mathcal{NP}$ precisamos apenas apresentar um certificado para o SIM para o problema. Esse certificado é composto pela fórmula $F \in 3SAT$ e o conjunto de valores que devem ser atribuídos a cada variável. De posse desses dados podemos então atribuir os valores a cada variável da fórmula F e verificar se a solução se trata realmente de uma atribuição que gera uma resposta SIM para a fórmula.

O processo de efetuar uma atribuição de valor para cada variável da fórmula pode ser feito em tempo polinomial. Da mesma forma, verificar se o conjunto de atribuições anotado satisfaz cada cláusula, também pode ser verificado em tempo polinomial. Por último, verificar se F foi satisfeita, também pode ser feito em tempo polinomial.

Como pode ser notado, para efetuar essa verificação o computador precisa realizar apenas operações simples como atribuições e operações condicionais, o que implica dizer que essa verificação pode ser feita em tempo polinomial, assim $3SAT \in \mathcal{NP}$.

- (ii) Para efetuar a prova de NP-dificuldade, vamos setar como problema alvo da redução polinomial, o problema SAT, já conhecido ser NP-completo pelo teorema de Cook-Levin (COOK, 1971), logo devemos mostrar que $SAT \propto 3SAT$:

Para transformar uma fórmula SAT em uma fórmula 3SAT precisamos substituir cada cláusula da formula SAT que possua um número de variáveis diferente de 3, por um conjunto de cláusulas de exatamente 3 variáveis. Para isso precisamos tratar alguns casos separadamente, supomos então que exista uma instância $F \in SAT$ que possua a cláusula $C \in F$.

- Se $C = (x)$: $|C| = 1$, i.e. possui apenas uma variável, dessa forma precisamos transformar C em uma fórmula $C' \in 3SAT$, onde C' possui a seguinte característica, $C' = (x \vee y \vee z) \wedge (x \vee \neg y \vee z) \wedge (x \vee y \vee \neg z) \wedge (x \vee \neg y \vee \neg z)$. Com essa transformação se C for satisfazível C' também será satisfazível para qualquer atribuição das duas novas variáveis adicionadas y e z , pois a única forma de C e C' ser satisfazível é com $x = 1$ ($x = True$);
- Se $C = (x_1 \vee x_2)$: $|C| = 2$, i.e. possui duas variáveis, quando isso ocorre C irá se transformar em uma fórmula $C' \in 3SAT$, onde C' possui a seguinte característica $C' = (x_1 \vee x_2 \vee y) \wedge (x_1 \vee x_2 \vee \neg y)$, dessa forma como x_1 e x_2 estão em todas as cláusulas da fórmula C' , então C' será satisfazível se C também for, da mesma forma se C' for satisfazível então C também é;
- Se $C = ((x_1 \vee x_2 \vee x_3))$: $|C| = 3$, nada a fazer;
- Se $C = (x_1 \vee x_2 \vee \dots \vee x_k)$: $|C| \geq 4$, i.e. a fórmula possui mais de 3 variáveis, quando isso acontece temos que criar uma fórmula $C' \in 3SAT$ com $k - 3$ novas variáveis, onde cada cláusula de C_i possui exatamente 3 variáveis, para isso seguimos a seguinte regra, $C' = (x_1 \vee x_2 \vee y_1) \wedge (x_3 \vee \neg y_1 \vee y_2) \wedge (x_4 \vee \neg y_2 \vee y_3) \wedge \dots \wedge (x_{k-1} \vee x_k \vee \neg y_{k-3})$. Com essa regra mantemos o valor lógico das variáveis originais de C predominando sobre as novas variáveis adicionadas na fórmula C' . Para que a cláusula seja satisfeita vamos adotar a seguinte regra para a atribuição de valor para y_j : se $x_i = 1$ então $y_j = 1$ para todo $j < i - 1$, os demais y_j devem receber 0. Para entender melhor digamos que $k = 5$, logo $C = (x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5)$ assim $C' = (x_1 \vee x_2 \vee y_1) \wedge (x_3 \vee \neg y_1 \vee y_2) \wedge (x_4 \vee x_5 \vee \neg y_2)$. Com essa transformação C' deverá ser satisfazível se C for satisfazível, para isso pelo menos um x_i deve ser 1 (ou *True*). Dessa forma, qualquer que seja configuração que torne C satisfazível torna C' satisfazível também, da mesma forma se C' é satisfazível então C é satisfazível.

Por (i) mostramos que $3SAT \in \mathcal{NP}$, por (ii) mostramos que $SAT \propto 3SAT$. Como todas as transformações feitas correspondem a processos que podem ser efetuados em tempo polinomial, dessa forma podemos concluir que $3SAT \in \text{NP-Completo}$. \square

4 METODOLOGIA

Neste trabalho adotamos uma metodologia de pesquisa do tipo exploratória e investigativa. Chamamos de pesquisa exploratória aquela realizada sobre um tema (ou questão) pouco abordado, i.e, existem poucos ou nenhum resultado relacionado a ele. Já uma pesquisa investigativa possui como objetivo alcançar resultados capazes de expandir a visão sobre um determinado problema. Neste trabalho buscamos investigar e explorar, de forma geral, a classe de grafos B_2 -EPG e a propriedade Helly relacionada.

4.1 Metodologia Geral: Técnicas de Demonstração

Quando queremos desenvolver ideias e hipóteses criamos afirmações e sugerimos que elas são verdadeiras ou falsas, chamamos estas afirmações de proposições, e tais proposições são do tipo $p \rightarrow q$, sendo p a hipótese e q a tese. Diante disso, a pesquisa iniciou com o estudo de técnicas para determinar a validade das proposições levantadas. Essas alternativas são chamadas de técnicas de demonstração, e tratam de provas matemáticas que demonstram a veracidade de um fato usando argumentos lógicos-matemáticos. As principais técnicas de demonstração são: **prova direta**, **prova por contradição** (também chamada de **prova por redução ao absurdo**), **prova por contraposição** e **prova por indução**.

- **Prova direta:** Este tipo de prova parte do princípio de que a hipótese é verdadeira e a partir dela demonstra-se que a tese também é verdadeira. Para isso podem ser combinadas proposições já existentes e provadas ser verdadeira (i.e. lemas válidos), resultados já conhecidos sobre a hipótese ou características particulares que a hipótese possui, etc. Quando provamos que uma proposição é verdadeira ela se torna um lema ou teorema.

Exemplo de prova:

Proposição 2. *Se n é inteiro ímpar, então n^2 será ímpar.*

Demonstração. Dessa forma:

- se n é ímpar, então podemos assumir que $n = 2k + 1$, onde k é um número inteiro.
- logo: $n^2 = (2k + 1)^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1$.

A operação $2k^2 + 2k$ gera um número inteiro, então podemos dizer que $2k^2 + 2k = k'$, onde k' é um número inteiro, logo $n^2 = 2k' + 1$, portanto n^2 é ímpar. \square

- **Prova por contradição:** Partindo do princípio de que $p \rightarrow q$, sabemos também que $p \rightarrow q \Leftrightarrow (p \wedge \neg q) \rightarrow F$. Dessa forma se assumirmos que se a hipótese acontecer e a tese não acontecer gera um absurdo, concluímos então que necessariamente $p \rightarrow q$.

Exemplo de prova:

Proposição 3. Se n e m são números inteiros ímpares, então $(n + m)$ é um número inteiro par.

Demonstração. Aplicando a lógica da contradição, tomemos nossa hipótese, em seguida efetuamos a negação da tese, isso com a intenção de tentar concluir um absurdo. Dessa forma chegamos no seguinte: Se n e m são números inteiros ímpares e $(n + m)$ gera um inteiro ímpar, então concluímos um absurdo.

- Assumindo que n , m e $(n + m)$ são inteiros ímpares, então $n = 2k + 1$, $m = 2k' + 1$ e $(n + m) = 2k'' + 1$, onde k, k' e k'' são inteiros.
- logo: $n = (n + m) - m = (2k'' + 1) - (2k' + 1) = 2k'' - 2k' = 2(k'' - k')$.

Como estamos trabalhando com números inteiros, a operação $k'' - k'$ gera um número inteiro, então podemos dizer que $2(k'' - k') = 2k'''$, onde k''' é um número inteiro, logo $n = 2k'''$, i.e., n é um inteiro par. Inicialmente dissemos que n era ímpar, mas concluímos que n é par, logo encontramos um absurdo. Com isso se n e m são números inteiros ímpares então necessariamente $(n + m)$ deve ser um número inteiro par. \square

- **Prova por contraposição:** Em Lógica Proposicional sabemos que: $p \rightarrow q \Leftrightarrow \neg q \rightarrow \neg p$. Partindo desse princípio de equivalência, se é possível provar $\neg q \rightarrow \neg p$ então é provado que $p \rightarrow q$.

Exemplo de prova:

Proposição 4. Se $(n + m)$ gera um inteiro par, então n e m são inteiros de paridades iguais.

Demonstração. Aplicando a lógica da contraposição, negamos a tese para concluirmos a negação da hipótese, chegando assim na seguinte proposição: Se n e m são inteiros de paridades diferentes então $(n + m)$ gera um inteiro ímpar.

- Assumindo que n é par e m é ímpar, então $n = 2k$ e $m = 2k' + 1$, onde k e k' são inteiros.
- logo: $(n + m) = 2k + 2k' + 1 = 2(k + k') + 1$.

A operação $k + k'$ gera um número inteiro, então podemos dizer que $k + k' = k''$, onde k'' é um número inteiro, logo $(n + m) = 2k'' + 1$, portanto $(n + m)$ é ímpar, provando assim nossa proposição inicial. \square

- **Prova por indução:** Normalmente esse tipo de prova é aplicada em proposições que se relacionam com números naturais, de forma que se prova um caso base e a partir de uma regra de indução é possível provar os demais casos.

Exemplo de prova:

Proposição 5. *Se n é um número inteiro, então $n^3 - n$ é múltiplo de 3.*

Demonstração. Assumindo um caso base que satisfaz a nossa propriedade, por meio de regra de indução buscamos provar os demais.

- Como caso base, assumimos $n = 1$, pois $1^3 - 1 = 0$ e 0 é múltiplo de 3.
- Como já mostramos que pelo menos um n satisfaz a proposição, devemos mostrar agora que para qualquer n inteiro, $n^3 - n$ é múltiplo de 3. Dessa forma devemos provar que $(n + 1)^3 - (n + 1)$ é múltiplo de 3.
- Logo: $(n + 1)^3 - (n + 1) = n^3 + 3n^2 + 3n + 1 - n - 1 = n^3 - n + 3(n^2 + n)$.

Como podemos observar a primeira parte da soma, i.e. $(n^3 - n)$, é o caso que assumimos se múltiplo de 3 e a segunda parte $3(n^2 + n)$ é claramente múltiplo de 3. Logo podemos concluir que $(n + 1)^3 - (n + 1)$ é múltiplo de 3. \square

4.2 Metodologia Particular para Grafos EPG

Os grafos de aresta-interseção de caminhos sobre uma grade foram definidos em 2009 (GOLUMBIC; LIPSHTEYN; STERN, 2009). Por isso, podemos considerar esse tema como relativamente novo. Dessa forma, existem muitos problemas relacionados à representações EPG que podem ser explorados e muitos outros ainda não resolvidos, em particular o problema que concerne à complexidade de reconhecimento das classes de grafos B_k -EPG para $k \geq 3$.

Nesta pesquisa buscamos investigar e explorar os grafos EPG, com uma atenção particular para o estudo da complexidade de reconhecimento da classe B_2 -EPG-Helly. Essa subclasse de grafos EPG é ainda pouco abordada na literatura, possui poucos resultados relacionados, além disso nem mesmo a complexidade de reconhecimento da classe é conhecida. Dessa forma, esperamos obter com este trabalho resultados significativos sobre a classe em estudo, e ao término desta pesquisa esperamos no mínimo oferecer uma contribuição para o problema de reconhecimento da classe.

Para o desenvolvimento desta pesquisa usamos algumas ferramentas que facilitam o andamento dos estudos. Para desenvolver as imagens disponibilizadas neste trabalho, por exemplo, fizemos uso do Geogebra. O Geogebra é uma aplicação de distribuição gratuita com propósito matemático. Ela é dinâmica, no sentido de permitir interação com os objetos anteriormente definidos, e utiliza geometria e álgebra, possibilitando visualização gráfica de elementos geométricos como pontos (vértices), retas (segmentos e caminhos), e do próprio hospedeiro da nossa representação (a grade), dessa forma possibilitando-nos produzir as imagens das representações EPG, assim como as clássicas representações de grafos.

Outro material de suporte base para nossa pesquisa é o trabalho apresentado na pesquisa de Pergel e Rzażewski (2017). Nesse paper podemos encontrar a prova de NP-completude para o reconhecimento de grafos B_2 -EPG. Esse escrito nos interessa por dois motivos, o primeiro, porque é um paper seminal no estudo da classe B_2 -EPG, e em segundo lugar porque a classe B_2 -EPG é uma superclasse da classe B_2 -EPG-Helly, que abordamos neste trabalho.

Para demonstrar a complexidade de reconhecimento da classe B_2 -EPG-Helly analisamos primeiramente se a prova de complexidade da classe B_2 -EPG (PERGEL; RZAŻEWSKI, 2017), também é aplicável a ela. Verificamos que não, assim nosso principal esforço se volta para atacar o problema de reconhecimento da classe B_2 -EPG-Helly. Algum raciocínio do trabalho base de estudo foi aproveitado para o processo de desenvolvimento da nova prova, sendo necessárias pequenas adaptações. O rumo da pesquisa é o de explorar e procurar conhecer mais a respeito de uma classe que está no limiar do conhecimento sobre grafos de interseção, o suficiente para atestar a relevância e profundidade do assunto.

4.2.1 Complexidade de Reconhecimento de Grafos B_2 -EPG-Helly

No Capítulo 3, subseção 3.3, explicamos como funciona uma redução polinomial e uma prova de NP-Completo. Agora, nesta seção explanaremos o que pretendemos investigar sobre a classe B_2 -EPG-Helly.

Vamos iniciar a discussão pelo que conhecemos. O problema de reconhecimento já é bem definido para algumas classes de grafos, por exemplo os grafos B_0 -EPG já foram mencionados serem equivalentes à classe dos grafos de intervalo. Em particular, a classe B_0 -EPG coincide com a classe B_0 -EPG-Helly, e isso não ocorre para nenhuma outra classe nessa hierarquia, resultado também já citado anteriormente. Em seguida, observemos que também a classe de grafos B_1 -EPG possui complexidade de reconhecimento bem definida, NP-Completo Heldt, Knauer e Ueckerdt (2014a). Mais recentemente, no ano de 2020, o paper de Bornstein et al. (2020) nos apresenta uma demonstração de que a classe B_1 -EPG-Helly também preserva a característica de o problema de reconhecimento ser NP-Completo.

Os fatos mencionados acima servem de alguma forma como motivação deste trabalho. Uma vez que na subclasse B_1 -EPG-Helly, contra-intuitivamente, a dificuldade do reconhecimento se manteve tão difícil quanto a da superclasse B_1 -EPG, então nos perguntamos se o mesmo pode acontecer com a classe B_2 -EPG-Helly. Como já ocorre que reconhecer grafos B_1 -EPG-Helly é difícil, agora o que seria contra-intuitivo, na verdade, seria pensar o contrário para B_2 -EPG-Helly.

De posse dessas informações e com esse raciocínio em mente estamos preparados para formalizar o problema de reconhecimento de grafos B_2 -EPG-Helly, enunciado a seguir.

Reconhecimento de grafos B_2 -EPG-Helly	
<i>Entrada:</i>	Um grafo G .
	Determinar se há um conjunto de caminhos de no máximo duas dobras $P = \{P_1, P_2, \dots, P_n\}$ em uma grade Q tal que:
<i>Objetivo:</i>	<ul style="list-style-type: none"> • $u, v \in V(G)$ são adjacentes em G se e somente se P_u, P_v compartilham uma aresta em Q; e • \mathcal{P} satisfaz a propriedade Helly.

Tabela 4 – Problema de reconhecimento de grafos B_2 -EPG-Helly

4.2.2 Abordagem do Problema

Este trabalho ataca inicialmente a complexidade de reconhecimento da classe de grafos B_2 -EPG-Helly. Baseado nos resultados da literatura conhecidos para o reconhecimento das classes de grafos B_1 -EPG (HELDT; KNAUER; UECKERDT, 2014a), B_2 -EPG (PERGEL; RZAŻEWSKI, 2017) e B_1 -EPG-Helly (BORNSTEIN et al., 2020), todos problemas NP-completos, isso nos serve como uma forte evidência que leva nossa intuição a pensar que o problema de reconhecimento da classe de grafos B_k -EPG-Helly em geral também é NP-completo, mas esse resultado carece de provas e é um resultado em aberto na literatura, ver Bornstein et al. (2020) e Pergel e Rzażewski (2017). Com base no exposto anteriormente, conjecturamos o seguinte resultado para a classe B_2 -EPG-Helly:

Conjectura 6. *O problema de reconhecimento de grafos B_2 -EPG-Helly é NP-Completo.*

Para facilitar nossa prova de NP-Completo vamos usar alguns resultados já existentes, um deles é a prova de NP-Pertinência para B_k -EPG. Essa prova está presente no trabalho de Bornstein et al. (2020). Apesar desse resultado já ser conhecido, não é um resultado trivial, porque envolve utilização de técnicas para determinar

um caminho em tempo polinomial, em um esquema de codificação específico, e também determinar se o conjunto de caminhos da representação satisfaz a propriedade Helly, esse último item é feito através de uma generalização do Teorema de Gilmore, (BERGE; DUCHET, 1975).

Mesmo fazendo uso desses resultados é importante observar que provar a complexidade da classe B_2 -EPG-Helly não é algo trivial e frequentemente requer o estudo de técnicas específicas, *gadgets* particulares para os quais possamos construir representações que satisfaçam as restrições que desejamos. Nesse caso particular, desejamos um *gadget*, i.e. um grafo G construído apropriadamente a partir de um problema de entrada Π de forma que possamos construir uma expressão bicondicional da seguinte forma: (ida) Se o problema Π for satisfatível então o grafo G vai possuir uma representação B_2 -EPG-Helly; e (volta) Dada uma representação, particular, B_2 -EPG-Helly para o grafo G conseguimos recuperar uma instância-SIM do problema Π .

5 RESULTADOS DA PESQUISA

Neste capítulo apresentaremos uma série de resultados obtidos por esta pesquisa. Os resultados tratam de forma geral grafos B_2 -EPG e resultados genéricos sobre relações entre caminhos. O capítulo traz também algumas definições que foram necessárias ser estabelecidas para aprofundamento da pesquisa, algoritmos para representar alguns casos particulares de grafos B_2 -EPG e B_2 -EPG-Helly, e ao final do capítulo apresentamos também uma prova de NP-completude para o problema de reconhecimento de grafos B_2 -EPG-Helly. Os resultados a seguir são fruto de um trabalho teórico-reflexivo profundo sobre grafos EPG.

Lema 7. *Sejam L_1 e L_2 duas linhas da grade paralelas entre si, se um caminho P_1 aresta-intersecta L_1 e L_2 então necessariamente o caminho P_1 possui no mínimo duas dobras.*

Demonstração. Sem perda de generalidade, considere duas linhas distintas quaisquer horizontais (portanto paralelas) da grade, digamos L_1 e L_2 . Se o caminho P_1 , possui um segmento sobre a linha L_1 , então ele deve gastar pelo menos uma dobra para alcançar a linha L_2 . Porém, se P_1 alcança L_2 e não dobrar novamente então P_1 , no máximo, irá vértice-intersectar a Linha L_2 . Portanto para que haja um segmento de P_1 sobre L_2 é necessário que P_1 possua, no mínimo, duas dobras. \square

Corolário 8. *Se o caminho $P_1 \in B_2$ -EPG, onde P_1 possui duas dobras, então P_1 possui exatamente dois segmentos na mesma direção (seja horizontal ou vertical).*

Demonstração. Em uma grade retangular existem apenas duas direções (horizontal e vertical). Se o caminho P_1 possui duas dobras, necessariamente, ele é composto por três segmentos. Logo, pelo princípio da casa dos pombos, necessitamos acomodar esses três segmentos nas duas direções possíveis, i.e. temos mais segmentos que direções para acomodar. Inevitavelmente vamos ter que acomodar dois segmentos em uma das direções. \square

Definição 9. *Seja P_1 um caminho B_2 -EPG com duas dobras. Chamamos o segmento central de P_1 de **corpo**, enquanto os segmentos ligados ao corpo são chamados de **pernas**.*

Corolário 10. *Seja $P_1 \in B_2$ -EPG, onde P_1 possui duas dobras e três segmentos. O segmento do corpo de P_1 possui direção contrária aos segmentos das pernas de P_1 .*

Demonstração. Pelo Corolário 8 sabemos que existem dois segmentos na mesma direção. Suponha por contradição que o corpo e, pelo menos, uma das pernas possua

a mesma direção. Ora, mas se isso ocorrer teremos dois segmentos consecutivos do mesmo caminho na mesma direção. Neste caso, ou os segmentos estão sobre a mesma linha da grade, se estiverem juntos, um ao final do outro, na verdade formam um mesmo segmento. Ou estão sobre a mesma linha e há uma descontinuidade, logo não forma um caminho. Ou então eles estão em linhas da grade distintas, nesse caso pelo Lema 7 o caminho possui mais de duas dobras. Em todos os casos chegamos a uma contradição. \square

Corolário 11. *Seja $P_1 \in B_2$ -EPG, onde P_1 possui duas dobras e três segmentos, se P_1 possui dois segmentos na mesma direção, então esses dois segmentos estão em linhas paralelas da grade de representação.*

Demonstração. Pelo Corolário 10, sabemos que o corpo de P_1 possui direção oposta à direção das pernas, além disso, em cada extremidade do segmento do corpo de P_1 existe uma dobra para os segmentos das pernas. Consideremos P_1 formado pelos segmentos S_1, S_2 e S_3 , com S_2 sendo o corpo, e considerando uma linha qualquer da grade de representação, digamos L_1 , se S_2 está sobre L_1 da coluna C_1 até a coluna C_2 , como S_1 e S_3 devem ter direção oposta a S_2 e compartilham 1 vértice de extremidade com S_2 , então S_1 e S_3 devem estar, necessariamente, um sobre C_1 e o outro sobre C_2 . Como C_1 e C_2 são colunas das extremidades do segmento S_1 , concluímos que S_1 e S_3 são portanto paralelas entre si. \square

Lema 12. *Sejam S_1 e S_2 dois segmentos de caminhos distintos de uma representação EPG, se $S_1 \cap_v S_2 \neq \emptyset$ e $S_1 \cap_e S_2 = \emptyset$, então no máximo $|S_1 \cap_v S_2| = 1$.*

Demonstração. Em uma grade de representação EPG, um segmento de um caminho está disposto unicamente sobre uma linha da grade. Dessa forma digamos que S_1 e S_2 estejam dispostos sobre as linhas da grade L_1 e L_2 respectivamente, onde L_1 e L_2 podem ser linhas de direções distintas, linhas na mesma direção porém paralelas ou podem ser a mesma linha.

- Caso L_1 e L_2 sejam linhas na mesma direção porém paralelas, então $S_1 \cap_v S_2 = \emptyset$;
- Caso L_1 e L_2 sejam linhas de direções distintas, então L_1 e L_2 podem se cruzar exatamente em um ponto, dessa forma no máximo $|S_1 \cap_v S_2| = 1$;
- Caso L_1 e L_2 sejam a mesma linha, para S_1 e S_2 compartilharem mais de um vértice então eles devem ser vértices consecutivos, já que estão sobre a mesma linha. Na grade de representação dois vértices consecutivos formam uma aresta, dessa forma, se S_1 e S_2 compartilharem mais de um vértice consecutivo, então $S_1 \cap_e S_2 \neq \emptyset$, logo no máximo $|S_1 \cap_v S_2| = 1$;

Diante desses fatos concluímos que no máximo $|S_1 \cap_v S_2| = 1$. \square

Lema 13. Dado um caminho $P_1 \in B_2$ -EPG, onde P_1 possui duas dobras, e seja S_2 um segmento da mesma representação. Se $P_1 \cap_v S_2 \neq \emptyset$ e $P_1 \cap_e S_2 = \emptyset$, então no máximo $|P_1 \cap_v S_2| = 2$.

Demonstração. Pelo Corolário 10 sabemos que o corpo de P_1 possui direção contrária aos demais segmentos. Pelo Corolário 11, as pernas de P_1 não estão sobre a mesma linha da grade. Pelo Lema 12, sabemos que dois segmentos que não compartilham aresta podem se vértice-intersectar em, no máximo, um ponto. Por contradição, suponha que $|P_1 \cap_v S_2| > 2$, digamos $|P_1 \cap_v S_2| = 3$, assim deve existir uma configuração em que S_2 vértice-intersecta os três segmentos de P_1 . Pelo Lema 12 sabemos que dois segmentos que não se intersectam em aresta podem ter no máximo um vértice em comum. Assim para que S_2 possua três pontos de interseção com o caminho P_1 , necessariamente S_2 teria que intersectar os três segmentos de P_1 .

- S_2 não pode estar sobre linhas da grade paralelas às pernas de P_1 , já que se disposto dessa forma S_2 poderia vértice-intersectar apenas um dos três segmentos de P_1 ou um vértice de dobra de P_1 , logo $|P_1 \cap_v S_2| = 1$.
- Caso S_2 esteja em uma linha da grade perpendicular às linhas em que as duas pernas de P_1 estão dispostas. Se essa linha for a mesma que o corpo de P_1 está desenhado, então S_2 poderia vértice-intersectar P_1 apenas em uma das extremidade de seu corpo, assim $|P_1 \cap_v S_2| = 1$. Caso S_2 esteja hospedado em uma linha que o permite vértice-intersectar as pernas de P_1 :
 - Se as pernas dobram para lados distintos: nesse caso $|P_1 \cap_v S_2| = 1$;
 - Se as duas pernas dobram para o mesmo lado: com essa configuração $|P_1 \cap_v S_2| = 2$, ver Figura 10.

Logo, chegamos em uma contradição pois em todas as configurações testadas, no máximo, $|P_1 \cap_v S_2| = 2$. □

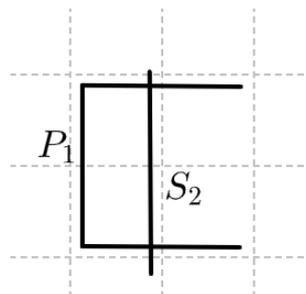


Figura 10 – Interseção de um segmento com um caminho B_2 -EPG

Definição 14. Uma *shape* é uma forma que um caminho ou representação EPG induz sobre uma grade, a menos de rotações e isomorfismos.

Definição 15. Uma *c-shape* é uma forma de um caminho de uma representação B_2 -EPG induz sobre uma grade com formato similar ao da letra “C”.

Definição 16. Uma *z-shape* é uma forma que um caminho de uma representação B_2 -EPG induz sobre uma grade com formato similar ao da letra “Z”.

Corolário 17. O conjunto de caminhos que aresta-intersecta duas linhas paralelas, usando exatamente duas dobras é isomorfo ao conjunto de c-shapes e z-shapes apresentado nas Figuras 11, 12 e 13.

Vale ressaltar que no trabalho de Pergel e Rzażewski (2017) formas semelhantes das estruturas apresentadas nas Figuras 11, 12 e 13 foram utilizadas, mas com nomenclatura distinta das usadas no Corolário 17. Para os fins particulares de nosso trabalho essas nomenclaturas serão mais apropriadas.

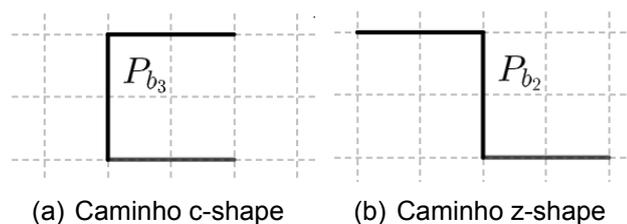


Figura 11 – Possíveis caminhos B_2 -EPG (com exatamente 2 dobras)

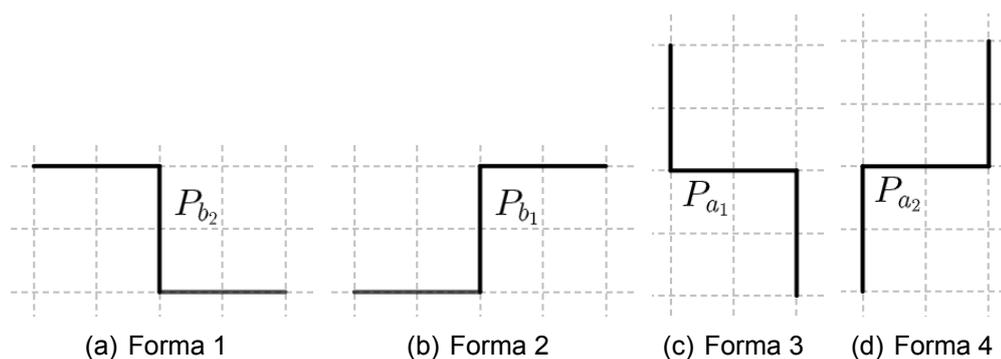


Figura 12 – Variações da z-shape

Definição 18. Um grafo $G = (V, E)$ é considerado bipartido quando $V = V_1 + V_2$ com $V_1 \cap V_2 = \emptyset$, além disso toda aresta de E une um vértice de V_1 à V_2 .

Definição 19. O grafo bipartido completo $G = k_{p,q}$ é um grafo bipartido, onde $G = \{V_1 + V_2, E\}$ com $|V_1| = p$ e $|V_2| = q$, além disso existe $(v_i, v_j) \forall v_i \in V_1, v_j \in V_2$.

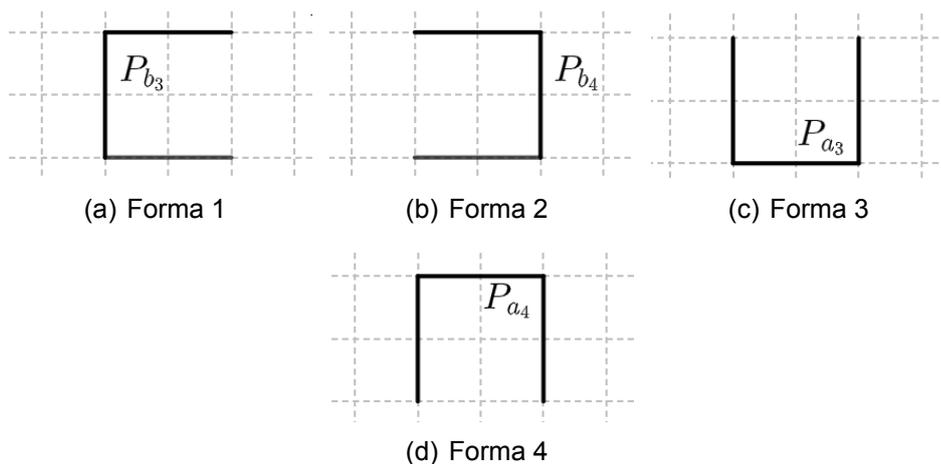


Figura 13 – Variações da c-shape

O grafo $K_{2,q}$ é um grafo bipartido completo. É resultado conhecido na literatura que o grafo $K_{2,q} \notin B_1$ -EPG (ASINOWSKI; SUK, 2009), para $q \geq 5$. A seguir provamos que $K_{2,q} \in B_2$ -EPG para todo $q \geq 0$. Na literatura existem resultados que mostram que todo grafo bipartido $K_{m,n}$, se $n > m^4 + 2m^3 + 5m^2 - 4m$, então $b(K_{m,n}) = 2m - 2$, e para os grafos $K_{3,n}$ em particular, $b(K_{3,n})$ é 1 se $n \leq 2$, 2 se $3 \leq n \leq 10$, 3 se $11 \leq n \leq 39$ e 4 se $n \geq 61$ (HELDT; KNAUER; UECKERDT, 2014a). Logo os grafos $K_{2,n}$ (chamado por nós de $K_{2,q}$), também foram mapeados, mas como conseguimos desenvolver um resultados para esta subclasse, decidimos apresentá-lo. No Lema 20 mostramos um algoritmo que constrói uma representação B_2 -EPG para o grafo $K_{2,q}$.

Lema 20. O grafo $K_{2,q} \in B_2$ -EPG.

Demonstração. Sejam V_1 e V_2 as partes de $K_{2,q}$, onde $|V_1| = 2$ e $|V_2| = q$. Considere $v_1, v_2 \in V_1$, e os demais vértices pertencentes à parte V_2 , ver Figura 14. Selecione duas linhas paralelas horizontais não-coincidentes da grade, digamos h_1 e h_2 , onde h_1 é uma linha acima de h_2 . Considere $r_1, r_2, \dots, r_q, r_{q+1}$ um conjunto de linhas verticais, paralelas, não-coincidentes da grade, tomadas da esquerda para direita, respectivamente. Represente v_1 e v_2 por segmentos, P_{v_1} e P_{v_2} , respectivamente sobre as linhas h_1 e h_2 , de forma que tanto P_{v_1} quanto P_{v_2} possuam segmento com extremidades em r_1 e r_{q+1} . Para representar os demais vértices do grafo, use caminhos *c-shape* de forma que para as dobras superiores, o caminho P_{w_i} possui um segmento vertical sobre r_i entre h_1 e h_2 , que se alonga na linha h_1 e dobra na aresta (r_i, r_{i+1}) da grade. Para as dobras inferiores, o segmento vertical de P_{w_i} deve se alongar na linha h_2 e dobra na aresta (r_i, r_{i+1}) da grade, ver Figura 14. □

Definição 21. Um grafo k -sun S_k , $k \geq 3$ consiste em $2k$ vértices, onde existe um conjunto independente $X = \{x_1, \dots, x_k\}$, uma clique $Y = \{y_1, \dots, y_k\}$ e arestas $E_1 \cup E_2$,

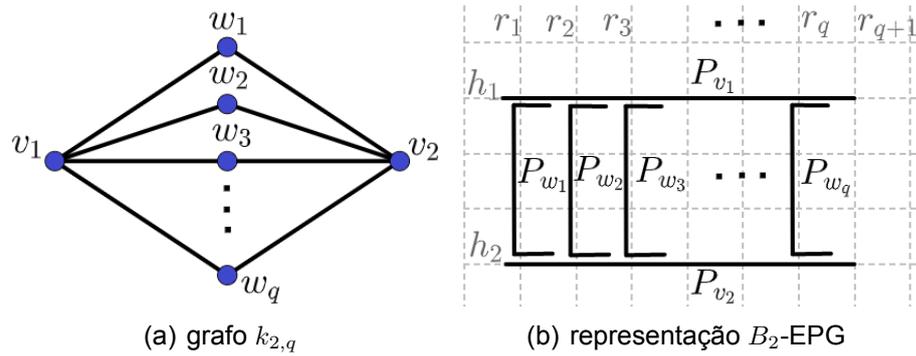


Figura 14 – Grafo $k_{2,q}$ e uma representação B_2 -EPG particular

onde $E_1 = \{x_1y_1, y_1x_2, x_2y_2, y_2x_3, \dots, x_ky_k, y_kx_1\}$ formam o ciclo externo e $E_2 = \{y_iy_j \mid i \neq j\}$ formam a clique interna.

Definição 22. O operador mod é utilizado para retornar o resto da divisão inteira. Quando a divisão é exata, retorna zero. Quando o dividendo é menor que divisor, então retorna o próprio dividendo. Exemplos: $4 \bmod 8 = 4$; $12 \bmod 4 = 0$; $10 \bmod 4 = 2$; $5 \bmod 4 = 1$.

Segundo resultados apresentados por Golubic, Lipshteyn e Stern (2009), o grafo k -sun $\notin B_1$ -EPG, para $k \geq 4$. A seguir vamos provar que k -sun $\in B_2$ -EPG para todo $k \geq 3$. Após o primeiro feedback sobre um trabalho submetido ao congresso ETC da CSBC, fomos surpreendidos com o fato de que o resultado que aparece em nosso Lema 23 já teria sido apresentado, na literatura, no paper de Cela e Gaar (2019), onde é demonstrado um resultado similar ao deste trabalho, porém, com uma abordagem distinta. A diferença na abordagem, inclusive, nos permite concluir o Corolário 24.

Lema 23. O grafo k -sun $\in B_2$ -EPG.

Demonstração. Considere a linha l_0 e a coluna c_0 como linha e coluna centrais da representação, ver Figura 15. Vamos considerar na nossa construção dois casos particulares:

- k par: O caminho P_{y_i} para $i = 1$, neste caso P_{y_i} possui segmento horizontal sobre a linha l_0 entre as colunas c_{-1} e $c_{\frac{k}{2}}$. Na coluna c_{-1} , o caminho P_{y_i} dobra e alcança l_1 , já na coluna $c_{\frac{k}{2}}$, se $k \bmod 4 = 0$, então P_{y_i} dobra e alcança l_{-1} , senão P_{y_i} dobra e alcança l_1 .
- k ímpar: O caminho P_{y_i} para $i = 1$, possui segmento horizontal sobre a linha l_0 entre as colunas c_{-1} e c_0 . Na coluna c_{-1} , o caminho P_{y_i} dobra e alcança l_1 , já na coluna c_0 , P_{y_i} dobra e alcança l_{-1} . O caminho P_{y_i} para $i = k$, possui segmento um horizontal sobre a linha l_0 entre as colunas c_0 e $c_{\frac{k-1}{2}}$. Se $(k + 1) \bmod 4 \neq 0$,

ambas extremidades do caminho P_{y_i} , respectivamente sobre as colunas c_0 e $c_{\frac{k-1}{2}}$ dobram e alcançam l_{-1} , senão o caminho P_{y_i} dobra em c_0 e alcança l_{-1} , e também dobra em $c_{\frac{k-1}{2}}$ e alcança l_1 . O caminho P_{x_i} , para $i = k$, será representado por um segmento vertical sobre a coluna c_0 entre as linhas l_0 e l_{-1} .

Para representar os vértices da clique Y siga o seguinte procedimento. O algoritmo fará representação de todo caminho P_{y_i} , onde i é ímpar. Quando k é par considere $i \neq 1$, e quando k é ímpar considere $i \neq 1$ e $i \neq k$, porque esses caminhos já foram construídos anteriormente pelo algoritmo. Neste caso teremos dois subcasos, caso 1.1, quando $(i + 1) \bmod 4 = 0$, então P_{y_i} possui um segmento horizontal sobre a linha l_0 , entre as colunas $c_{-\frac{i+1}{2}}$ e $c_{\frac{i-1}{2}}$. Na coluna $c_{-\frac{i+1}{2}}$, o caminho P_{y_i} dobra e alcança l_{-1} , já na coluna $c_{\frac{i-1}{2}}$, P_{y_i} dobra e alcança l_1 ; caso 1.2, quando $(i + 1) \bmod 4 \neq 0$, então P_{y_i} possui um segmento horizontal sobre a linha l_0 , entre as colunas $c_{-\frac{i+1}{2}}$ e $c_{\frac{i-1}{2}}$. Na coluna $c_{-\frac{i+1}{2}}$, o caminho P_{y_i} dobra e alcança l_1 , já na coluna $c_{\frac{i-1}{2}}$, P_{y_i} dobra e alcança l_{-1} .

Seja o caminho P_{y_i} , onde i é par e $i \bmod 4 \neq 0$. Neste caso, P_{y_i} possui segmento horizontal sobre a linha l_0 entre as colunas $c_{-\frac{i}{2}}$ e $c_{\frac{i}{2}}$. Ambas extremidades do caminho P_{y_i} dobram, respectivamente sobre as colunas $c_{-\frac{i}{2}}$ e $c_{\frac{i}{2}}$, e alcançam l_1 . Caso caminho P_{y_i} quando i é par e $i \bmod 4 = 0$, então P_{y_i} possui segmento horizontal sobre a linha l_0 entre as colunas $c_{-\frac{i}{2}}$ e $c_{\frac{i}{2}}$. Ambas extremidades do caminho P_{y_i} , respectivamente sobre as colunas $c_{-\frac{i}{2}}$ e $c_{\frac{i}{2}}$ dobram e alcançam l_{-1} .

Os vértices do conjunto independente X são representados por segmentos verticais que ocupam colunas distintas. Considere $i \neq k$ quando k é ímpar. No caso do caminho P_{x_i} , onde i é par e $i \bmod 4 = 0$, P_{x_i} será representado por um segmento sobre a coluna $\frac{i}{2}$ entre as linhas l_0 e l_{-1} . Para P_{x_i} , onde i é par e $i \bmod 4 \neq 0$, P_{x_i} será representado por um segmento sobre a coluna $\frac{i}{2}$ entre as linhas l_0 e l_1 . O caminho P_{x_i} , onde i é ímpar e $(i + 1) \bmod 4 = 0$, P_{x_i} será representado por um segmento sobre a coluna $-\frac{i+1}{2}$ entre as linhas l_0 e l_{-1} . Para P_{x_i} , onde i é ímpar e $(i + 1) \bmod 4 \neq 0$, P_{x_i} será representado por um segmento sobre a coluna $-\frac{i+1}{2}$ entre as linhas l_0 e l_1 . \square

Corolário 24. O grafo k -sun $\in B_2$ -EPG-Helly.

Demonstração. Considere a linha l_0 e a coluna c_0 como linha e coluna centrais da representação. Vamos considerar na nossa construção três casos particulares:

- k par: O caminho P_{y_i} , para $i = k - 1$, possui um segmento horizontal sobre a linha l_0 entre as colunas c_0 e $c_{\lfloor \frac{k-1}{2} \rfloor}$. Na coluna c_0 , o caminho P_{y_i} dobra e se estende até a linha l_1 , já na coluna $c_{\lfloor \frac{k-1}{2} \rfloor}$ o caminho P_{y_i} dobra e se estende até a linha l_{-1} . O caminho P_{y_i} , para $i = k$, possui um segmento vertical sobre a coluna c_0 entre as linhas l_0 e l_1 . Na linha l_0 , P_{y_i} dobra e se estende até a coluna c_1 , já na linha

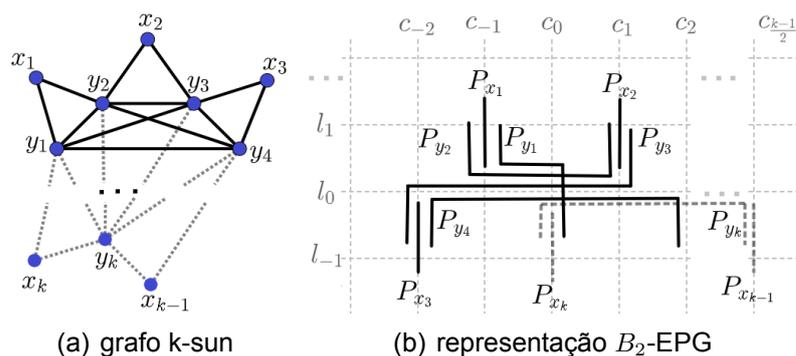


Figura 15 – Grafo k-sun e uma representação B_2 -EPG particular

l_1, P_{y_i} dobra e se estende até a coluna $c_{-\lfloor \frac{k-1}{2} \rfloor}$. O caminho P_{x_i} , para $i = k - 1$ é representado por um segmento vertical sobre a coluna c_0 entre as linhas l_0 e l_1 .

- k ímpar e $k \neq 3$: O caminho P_{y_i} , para $i = k - 1$, possui um segmento horizontal sobre a linha l_0 entre as colunas $c_{-(\lfloor \frac{k-1}{2} \rfloor - 1)}$ e $c_{\lfloor \frac{k-1}{2} \rfloor}$. Na coluna $c_{-(\lfloor \frac{k-1}{2} \rfloor - 1)}$, o caminho P_{y_i} dobra e se estende até a linha l_{-1} . O caminho P_{y_i} , para $i = k$ possui um segmento vertical sobre a coluna c_0 entre as linhas l_0 e l_1 . Na linha l_0 , P_{y_i} dobra e se estende até a coluna $c_{\lfloor \frac{k-1}{2} \rfloor}$, já na linha l_1 , P_{y_i} dobra e se estende até a coluna $c_{-\lfloor \frac{k-1}{2} \rfloor}$. O caminho P_{x_i} , para $i = k - 1$ é representado por um segmento horizontal sobre a linha l_0 entre as colunas $c_{\lfloor \frac{k-1}{2} \rfloor - 1}$ e $c_{\lfloor \frac{k-1}{2} \rfloor}$.
- $k = 3$: O caminho P_{y_i} , para $i = k - 1 = 2$, é representado por um segmento horizontal sobre a linha l_0 entre as colunas c_{-1} e c_2 . O caminho P_{y_i} , para $i = k = 3$ possui um segmento vertical sobre a coluna c_0 entre as linhas l_0 e l_1 . Na linha l_0 , P_{y_i} dobra e se estende até a coluna c_2 , já na linha l_1 , P_{y_i} dobra e se estende até a coluna c_{-1} . O caminho P_{x_i} , para $i = k - 1 = 2$ é representado por um segmento horizontal sobre a linha l_0 entre as colunas c_1 e c_2 .

Para representar os vértices da clique Y siga os seguintes procedimentos. Durante a execução dos passos seguintes caso $k = 3$ ou $k = 4$, considere $-(\lfloor \frac{k-1}{2} \rfloor - 1) = 0$. O caminho P_{y_i} , para $i = 1$, possui um segmento vertical sobre a coluna $c_{-\lfloor \frac{k-1}{2} \rfloor}$ entre as linhas l_0 e l_1 . Na linha l_1 o caminho P_{y_i} dobra e se estende até a coluna $c_{-(\lfloor \frac{k-1}{2} \rfloor - 1)}$, já na linha l_0 , P_{y_i} dobra e se estende até a coluna c_1 . O caminho P_{y_i} , para $i = 2$ (exceto quando $k = 3$, pois já foi definido anteriormente), possui um segmento horizontal sobre a linha l_0 entre as colunas $c_{-\lfloor \frac{k-1}{2} \rfloor}$ e c_1 . Na coluna c_1 o caminho P_{y_i} dobra e se estende até a linha l_{-1} .

Para o demais caminhos P_{y_i} que representam os vértices da clique Y (que excluem $i = 1, i = 2, i = k - 1$ e $i = k$), faça o seguinte: Para P_{y_i} , onde i é ímpar, P_{y_i} possui um segmento horizontal sobre a linha l_0 entre as colunas $c_{-\frac{i-1}{2}}$ e $c_{\frac{i-1}{2}}$, em ambas as extremidades P_{y_i} dobra e se estende até a linha l_{-1} . Para P_{y_i} , onde i é par, P_{y_i} possui

um segmento horizontal sobre a linha l_0 entre as colunas $c_{-(\frac{i}{2}-1)}$ e $c_{\frac{i}{2}}$, em ambas as extremidades P_{y_i} dobra e se estende até a linha l_{-1} .

Os vértices do conjunto X serão segmentos distribuídos através da representação. Para P_{x_i} , onde $i = 1$, P_{x_i} será representado por um segmento sobre a linha l_0 entre as colunas $c_{-(\lfloor \frac{k-1}{2} \rfloor)}$ e $c_{-(\lfloor \frac{k-1}{2} \rfloor - 1)}$. Para P_{x_i} , onde $i = k$, P_{x_i} será representado por um segmento sobre a linha l_1 entre as colunas $c_{-(\lfloor \frac{k-1}{2} \rfloor)}$ e $c_{-(\lfloor \frac{k-1}{2} \rfloor - 1)}$. Para os demais caminhos P_{x_i} (que excluem $i = 1, i = k - 1$ e $i = k$), faça o seguinte: Para P_{x_i} , onde i é par, P_{x_i} será representado por um segmento vertical sobre a coluna $c_{\frac{i}{2}}$ entre as linhas l_0 e l_{-1} . Para P_{x_i} , onde i é ímpar, P_{x_i} será representado por um segmento vertical sobre a coluna $c_{-\frac{i-1}{2}}$ entre as linhas l_0 e l_{-1} . \square

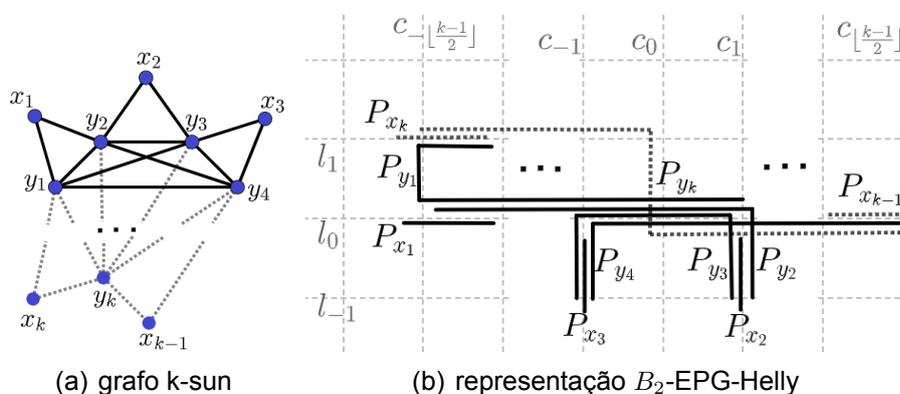


Figura 16 – Grafo k -sun e uma representações B_2 -EPG-Helly particular

5.1 Reconhecimento de Grafos B_2 -EPG-Helly

Nesta seção iremos abordar a complexidade de reconhecimento da classe de grafos B_2 -EPG-Helly. A seguir iniciamos com a prova de NP-pertinência.

Lema 25. B_2 -EPG-Helly $\in \mathcal{NP}$.

Demonstração. Para provar que B_2 -EPG-Helly $\in \mathcal{NP}$ vamos considerar um esquema de codificação apropriado onde cada caminho <https://www.overleaf.com/project/6149e46b24e3d> representado como um conjunto de arestas. Devemos provar que: cada caminho é definido em tempo polinomial; o conjunto de caminhos é definido em tempo polinomial; é possível verificar a bijeção entre a representação R e o grafo G que ela representa em tempo polinomial; é possível verificar se todo caminho da representação possui no máximo duas dobras em tempo polinomial; e, por último, conseguimos verificar que o conjunto de interseção dos caminhos satisfaz à propriedade Helly em tempo polinomial. Esta última é efetuada utilizando uma generalização do teorema de Gilmore

(BERGE; DUCHET, 1975). Uma prova de NP-pertinência para o problema de reconhecimento B_k -EPG-Helly é apresentada por Santos (2020). Portanto B_2 -EPG-Helly $\in \mathcal{NP}$. \square

Provaremos que o problema de reconhecimento de grafos B_2 -EPG-Helly é NP-Completo. Para isso, seguiremos um roteiro da prova de complexidade similar ao utilizado por Pergel e Rzażewski (2017) para reconhecer grafos B_2 -EPG. Setaremos o problema MONÓTONO-NAE-3-SAT (chamado também de PURE-NAE-3-SAT (PERGEL; RZAŻEWSKI, 2017)), definido na Tabela 5, como problema alvo de nossa redução. Monótono-Nae-3-Sat é uma variação do problema de Satisfatibilidade booleana (Sat), já conhecido ser NP-Completo, (COOK, 1971), além disso é equivalente ao problema de 2-coloração de hipergrafos 3-uniformes, (LOVÁSZ, 1973). Esse problema também é equivalente ao problema de divisão de conjuntos (GAREY; JOHNSON, 1979). Sabemos que o problema 3-sat é NP-Completo (KARP, 1972), e o caso particular Monótono-Nae-3-Sat também é NP-Completo, pelo teorema da dicotomia de Schaefer (SCHAEFER, 1978).

MONÓTONO-NAE-3-SAT	
<i>Entrada:</i>	Um conjunto X de variáveis não negadas; Uma coleção de cláusulas C sobre X , tal que cada $c \in C$, $ c = 3$.
<i>Objetivo:</i>	Determinar se existe uma atribuição satisfável para as variáveis de X de forma que cada cláusula possua pelo menos um literal <i>True</i> e um <i>False</i> .

Tabela 5 – Problema monótono-nae-3-sat

Lema 26. *Sejam $P_1, P_2 \in B_2$ -EPG, onde $P_1 \cap_e P_2 = \emptyset$, então $|P_1 \cap_v P_2|$ é no máximo 4.*

Demonstração. Pelo Lema 13, sabemos que um segmento de um caminho B_2 -EPG pode vértice-intersectar um outro caminho B_2 -EPG em no máximo dois pontos distintos. Levando apenas isso em consideração, poderíamos dizer que P_1 e P_2 , cada um com duas dobras e três segmentos, poderiam se vértice-intersectar em no máximo 6 pontos. Pelo Lema 13 podemos ver que para isso acontecer os três segmentos de um caminho, digamos P_2 , deveriam estar na mesma posição, i.e. todos horizontais ou verticais, mas pelo Corolário 8 sabemos que isso não pode acontecer, logo $|P_1 \cap_v P_2| = 6$ é impossível. Supomos agora que $|P_1 \cap_v P_2| = 5$, então pelo Lema 13, sabemos que dois segmentos de cada caminho devem fazer cada um duas vértice-interseção e o terceiro fazer pelo menos uma vértice-interseção. Usando o caminho P_2 como base, sabemos que os dois segmentos de P_2 que fazem duas vértice-interseções cada, devem estar na mesma direção, e como vértice-intersectam os mesmos segmentos então devem estar paralelos entre si. Sabendo disso, o terceiro segmento de P_2 deve estar

no mesmo sentido que dois segmento paralelos de P_1 se encontram e em linhas diferentes da grade para não se aresta-intersectarem, além disso, este terceiro segmento, pelo Corolário 10, deve ser o corpo de P_2 , logo cada uma de suas extremidades vértice-intersecta um dos outros dois segmentos de P_2 . Como as duas pernas do caminho P_2 estão paralelos ao corpo de P_1 , e ambos do mesmos lado deste segmento (as duas do lado direito ou esquerdo do corpo de P_1), então o corpo de P_2 , que vértice-intersecta esses dois segmentos em suas extremidades, não consegue vértice-intersectar o corpo de P_1 , e como está sobre linhas paralelas e distintas das pernas P_1 , então também não pode vértice-intersectar elas sem que uma aresta-interseção aconteça. Logo, o corpo de P_2 , em consequência também o de P_1 , não farão nenhuma interseção. Dessa forma, $|P_1 \cap_v P_2| = 5$ é falso. Para $|P_1 \cap_v P_2| = 4$ ver Figura 17. \square

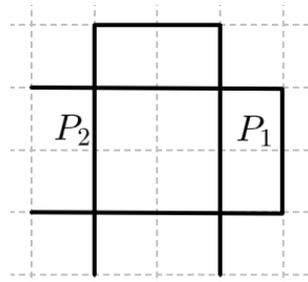


Figura 17 – Interseção máxima entre dois caminhos B_2 -EPG

Lema 27. *Dada uma fórmula $\phi \in \text{MONÓTONO-NAE-3-SAT}$ e ϕ' construída pela repetição de 21 vezes ϕ . Então G é o grafo de síntese de ϕ' .*

Demonstração. Começamos construindo o grafo de síntese. Dada uma fórmula $\phi \in \text{MONÓTONO-NAE-3-SAT}$ construiremos um grafo G . Para iniciar, replique a fórmula ϕ 21 vezes sobre uma cópia distinta do conjunto de variáveis.

Para começarmos a construção de G adicionamos dois vértices especiais, vértices a e b , que chamaremos de *gadget base*. Logo em seguida adicionamos pra cada variável i do conjunto de variáveis de ϕ' , um vértice v_i que é adjacente tanto ao vértice a quanto ao vértice b . Além disso, a cada ocorrência de uma variável i em uma cláusula da fórmula ϕ' , adicionamos ao grafo um novo vértice $O_{i,z}$ que é adjacente a a , b e v_i . Por último, adicionamos quatro novos vértices ao grafo G para cada cláusula $z = (i \vee j \vee k)$: um vértice c_z , onde $N(c_z) = \{o_{i,z}, o_{j,z}, o_{k,z}\}$; um vértice d_z , onde $N(d_z) = \{o_{i,z}, o_{j,z}\}$; um vértice e_z , onde $N(e_z) = \{o_{i,z}, o_{k,z}\}$; e um vértice f_z , onde $N(f_z) = \{o_{j,z}, o_{k,z}\}$. \square

A Figura 18 apresenta um grafo G construído a partir da fórmula $\phi = z_1(1 \vee 2 \vee 3) \wedge z_2(3 \vee 4 \vee 5)$, de acordo com o algoritmo dado no Lema 27. Note que para construir ϕ' é suficiente replicar todo os vértices de G , para cada cópia de ϕ , com exceção do *gadget base*, que é único para todo G .

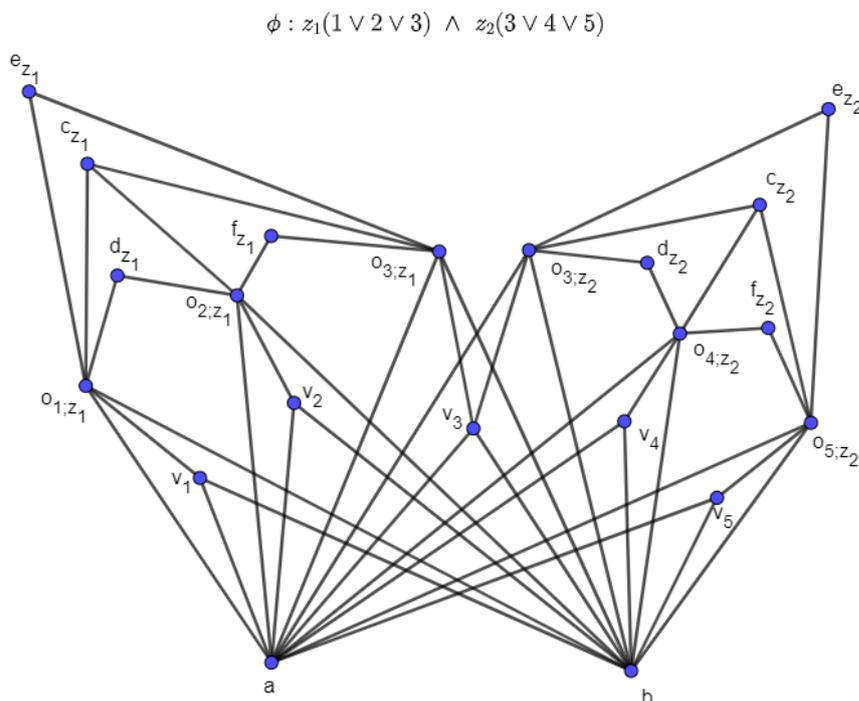


Figura 18 – Grafo G gerado a partir do Lema 27 usando uma cópia de ϕ .

Definição 28. Dado o grafo de síntese G , de uma fórmula ϕ' , dizemos que a cópia (relacionado a cláusula z) está **limpa** quando na representação B_2 -EPG de G as pernas de todo P_{v_i} e $P_{o_{i,z}}$ estão completamente cobertos por P_a e P_b , i.e. não aresta-intersectam arestas relevantes nem de P_a e nem de P_b .

Lema 29. Dada uma fórmula $\phi \in \text{MONÓTONO-NAE-3-SAT}$ e ϕ' construída pela repetição de 21 vezes ϕ , e o grafo de síntese G . Então existe pelo menos uma cópia de ϕ em ϕ' cuja representação está “limpa”.

Demonstração. Nossa intenção é mostrar que em uma representação B_2 -EPG do grafo síntese, no mínimo uma cópia de ϕ vai sofrer uma “restrição adequada”. A existência dessa cópia com uma restrição adequada é a chave para a transformação funcionar.

A adição dos vértices a e b em G possui como objetivo evitar a exposição dos segmentos mais externos dos caminhos P_{v_i} e $P_{o_{i,z}}$ (as pernas do caminho 2-bended), de forma que nenhum vértice-cláusula possa interseccionar esses caminhos em segmentos que não seja o corpo. Se os caminhos P_{v_i} e $P_{o_{i,z}}$ forem representados em uma interseção dos caminhos P_a e P_b ou em uma extremidade deles, essa configuração se torna ineficiente para o que queremos.

Digamos que os caminhos P_a e P_b se intersectam em quatro pontos, o máximo possível, segundo o Lema 26. Nesse caso maximizamos o conjunto formado pela

união dos casos problemáticos (casos onde as pernas do caminho 2-bended ficam livres e/ou vértice-interseções de P_a e P_b). Note que existem 12 arestas relevantes em P_a e P_b e mais 4 pontos de interseção. Guarde essa informação.

Perceba agora que todos os caminhos correspondendo a vértices P_{v_i} e $P_{o_{i,z}}$ das 21 réplicas de ϕ devem intersectar os caminhos P_a e P_b , pois todos esses vértices são adjacentes aos vértices a e b no grafo de síntese de ϕ' . Perceba também que não existem arestas entre as variáveis em ϕ' , analogamente os vértices $o_{i,z}$ também não possuem aresta entre si, portanto os caminhos correspondendo aos vértices que representam variáveis no grafo de síntese devem ser não intersectantes entre si, assim como os caminhos que representam os vértices $o_{i,z}$.

Pensemos agora nos casos problemáticos que citamos inicialmente. Suponha por absurdo que todas as cláusulas em ϕ' recaiam em casos problemáticos, ora mas ϕ' é a repetição 21 vezes de ϕ . Já concluímos que não existem arestas entre as variáveis de ϕ' isso implica que em cada aresta relevante dos caminhos P_a e P_b podemos representar no máximo uma cópia de ϕ , portanto conseguimos colocar no máximo 12 cópias de ϕ presentes nas arestas relevantes de P_a e P_b . Em tempo, perceba que existem quatro pontos de interseção entre P_a e P_b que também estão naqueles casos problemáticos, e em cada interseção de P_a e P_b conseguimos acomodar no máximo duas cópias de ϕ , como temos quatro pontos de interseção, logo podemos acomodar ali no máximo oito cópias de ϕ . Assim temos, no máximo, que 20 cópias de ϕ em ϕ' puderam ser acomodadas em posições que gerariam “situações problema”. Pelo princípio da casa dos pombos, se temos 20 arestas e pontos de interseção que geram situações-problema e temos 21 cópias de ϕ , isso nos garante que pelo menos uma dessas cópias vai ter que ser representada com os caminhos P_{v_i} e $P_{o_{i,z}}$ com pernas totalmente cobertas por P_a e P_b , chamamos esta cópia de “**cópia central**”. \square

Lema 30. *O grafo de síntese G construído a partir de ϕ' está em B_2 -EPG se e somente se ϕ' é satisfável.*

Demonstração. Vamos mostrar que: $Y_\phi \Leftrightarrow Y_{G \in B_2-EPG}$.

Construir uma representação para a cópia central com base no grafo síntese e observando a valoração para ϕ . Iniciemos nossa representação tomando uma representação onde P_a e P_b se intersectam em quatro pontos, como na Figura 17. Agora vamos representar os caminhos correspondendo às variáveis. Para cada variável $v_i \in \phi$, uma nova linha é inserida na grade para hospedar o novo caminho P_{v_i} , então faça:

- Se a variável $v_i = True$: o corpo do caminho P_{v_i} deve ser desenhado de forma vertical; e
 - As interseções $P_{v_i} \cap_e P_a \neq \emptyset$ e $P_{v_i} \cap_e P_b \neq \emptyset$ ocorrem horizontalmente.

- Se a variável $v_i = False$: o corpo do caminho P_{v_i} deve ser desenhado de forma horizontal;
 - As interseções $P_{v_i} \cap_e P_a \neq \emptyset$ e $P_{v_i} \cap_e P_b \neq \emptyset$ ocorrem verticalmente.

Para cada ocorrência de P_{v_i} representar o corpo de $P_{o_{i,z}}$, inserindo uma nova linha na grade ao lado de onde está hospedado o corpo de P_{v_i} , na mesma direção do corpo de P_{v_i} . Perceba que $N(v_i) = N(o_{i,z})$ a menos dos vértices-cláusula. Então $P_{o_{i,z}}$ também deve intersectar P_a e P_b , e também intersectar o próprio caminho P_{v_i} da seguinte forma:

- Onde ocorriam as interseções $P_{v_i} \cap_e P_a$ e $P_{v_i} \cap_e P_b$, agora será adicionada uma nova interseção de forma que haja
 - $P_{v_i} \cap_e P_a \cap_e P_{o_{i,z}} \neq \emptyset$ e $P_{v_i} \cap_e P_b \cap_e P_{o_{i,z}} \neq \emptyset$.

Por último, mas não menos importante, vamos dizer como deve estar organizada a representação dos vértices-cláusula. Para a z -ésima cláusula, vamos representar P_{c_z} , P_{d_z} , P_{e_z} e P_{f_z} , de acordo com as adjacências já citadas anteriormente. Essas interseções vão depender da valoração atribuída às variáveis da z -ésima cláusula $C_z = (1 \vee 2 \vee 3)$, da seguinte forma:

- Se duas das variáveis de C_z forem *True*:
 - Se $1 = 2 = True$ e $3 = False$: Neste caso, o caminho P_{c_z} vai possuir corpo horizontal entre as linhas verticais do corpo de $P_{o_{1,z}}$ e $P_{o_{2,z}}$. O corpo do caminho P_{c_z} está exatamente sobre a linha da grade, de forma a aresta-intersectar horizontalmente o caminho $P_{o_{3,z}}$. Além disso P_{c_z} dobra e aresta-intersecta verticalmente os próprios $P_{o_{1,z}}$ e $P_{o_{2,z}}$.
Os caminhos P_{e_z} e P_{f_z} dobram, respectivamente, nos pontos de interseção $P_{o_{1,z}} \cap_v P_{o_{3,z}}$ e $P_{o_{2,z}} \cap_v P_{o_{3,z}}$. Dessa forma $P_{e_z} \cap_e P_{o_{1,z}} \neq \emptyset$ e $P_{d_z} \cap_e P_{o_{3,z}} \neq \emptyset$. Analogamente $P_{f_z} \cap_e P_{o_{2,z}} \neq \emptyset$ e $P_{e_z} \cap_e P_{o_{3,z}} \neq \emptyset$.
O caminho P_{d_z} possui corpo horizontal entre o corpo dos caminhos $P_{o_{1,z}}$ e $P_{o_{2,z}}$, e dobra em ambas extremidades para aresta-intersectar os caminhos $P_{o_{1,z}}$ e $P_{o_{2,z}}$.
 - Se $1 = 3 = True$ e $2 = False$: Neste caso, o caminho P_{c_z} vai possuir corpo horizontal entre as linhas verticais do corpo de $P_{o_{1,z}}$ e $P_{o_{3,z}}$. O corpo do caminho P_{c_z} está exatamente sobre a linha da grade, de forma a aresta-intersectar horizontalmente o caminho $P_{o_{2,z}}$. Além disso P_{c_z} dobra e aresta-intersecta verticalmente os próprios $P_{o_{1,z}}$ e $P_{o_{3,z}}$.

Os caminhos P_{dz} e P_{fz} dobram, respectivamente, nos pontos de interseção $P_{o1,z} \cap_v P_{o2,z}$ e $P_{o3,z} \cap_v P_{o2,z}$. Dessa forma $P_{dz} \cap_e P_{o1,z} \neq \emptyset$ e $P_{dz} \cap_e P_{o2,z} \neq \emptyset$. Analogamente $P_{fz} \cap_e P_{o3,z} \neq \emptyset$ e $P_{fz} \cap_e P_{o2,z} \neq \emptyset$.

O caminho P_{ez} possui corpo horizontal entre o corpo dos caminhos $P_{o1,z}$ e $P_{o3,z}$, e dobra em ambas extremidades para aresta-intersectar os caminhos $P_{o1,z}$ e $P_{o3,z}$.

- Se $2 = 3 = True$ e $1 = False$: Neste caso, o caminho P_{cz} vai possuir corpo horizontal entre as linhas verticais do corpo de $P_{o2,z}$ e $P_{o3,z}$. O corpo do caminho P_{cz} está exatamente sobre a linha da grade, de forma a aresta-intersectar horizontalmente o caminho $P_{o1,z}$. Além disso P_{cz} dobra e aresta-intersecta verticalmente os próprios $P_{o2,z}$ e $P_{o3,z}$.

Os caminhos P_{dz} e P_{ez} dobram, respectivamente, nos pontos de interseção $P_{o2,z} \cap_v P_{o1,z}$ e $P_{o3,z} \cap_v P_{o1,z}$. Dessa forma $P_{dz} \cap_e P_{o2,z} \neq \emptyset$ e $P_{dz} \cap_e P_{o1,z} \neq \emptyset$. Analogamente $P_{ez} \cap_e P_{o3,z} \neq \emptyset$ e $P_{ez} \cap_e P_{o1,z} \neq \emptyset$.

O caminho P_{fz} possui corpo horizontal entre o corpo dos caminhos $P_{o2,z}$ e $P_{o3,z}$, e dobra em ambas extremidades para aresta-intersectar os caminhos $P_{o2,z}$ e $P_{o3,z}$.

- Se duas das variáveis de C_z forem *False*:
 - Se $1 = 2 = False$ e $3 = True$: Neste caso, o caminho P_{cz} vai possuir corpo vertical entre as linhas horizontais do corpo de $P_{o1,z}$ e $P_{o2,z}$. O corpo do caminho P_{cz} está exatamente sobre a linha da grade, de forma a aresta-intersectar verticalmente o caminho $P_{o3,z}$. Além disso P_{cz} dobra e aresta-intersecta horizontalmente os próprios $P_{o1,z}$ e $P_{o2,z}$.

Os caminhos P_{ez} e P_{fz} dobram, respectivamente, nos pontos de interseção $P_{o1,z} \cap_v P_{o3,z}$ e $P_{o2,z} \cap_v P_{o3,z}$. Dessa forma $P_{ez} \cap_e P_{o1,z} \neq \emptyset$ e $P_{dz} \cap_e P_{o3,z} \neq \emptyset$. Analogamente $P_{fz} \cap_e P_{o2,z} \neq \emptyset$ e $P_{ez} \cap_e P_{o3,z} \neq \emptyset$.

O caminho P_{dz} possui corpo vertical entre o corpo dos caminhos $P_{o1,z}$ e $P_{o2,z}$, e dobra em ambas extremidades para aresta-intersectar os caminhos $P_{o1,z}$ e $P_{o2,z}$.
 - Se $1 = 3 = False$ e $2 = True$: Neste caso, o caminho P_{cz} vai possuir corpo vertical entre as linhas horizontais do corpo de $P_{o1,z}$ e $P_{o3,z}$. O corpo do caminho P_{cz} está exatamente sobre a linha da grade, de forma a aresta-intersectar verticalmente o caminho $P_{o2,z}$. Além disso P_{cz} dobra e aresta-intersecta horizontalmente os próprios $P_{o1,z}$ e $P_{o3,z}$.

Os caminhos P_{dz} e P_{fz} dobram, respectivamente, nos pontos de interseção $P_{o1,z} \cap_v P_{o2,z}$ e $P_{o3,z} \cap_v P_{o2,z}$. Dessa forma $P_{dz} \cap_e P_{o1,z} \neq \emptyset$ e $P_{dz} \cap_e P_{o2,z} \neq \emptyset$. Analogamente $P_{fz} \cap_e P_{o3,z} \neq \emptyset$ e $P_{fz} \cap_e P_{o2,z} \neq \emptyset$.

O caminho P_{e_z} possui corpo vertical entre o corpo dos caminhos $P_{o_{1,z}}$ e $P_{o_{3,z}}$, e dobras em ambas extremidades para aresta-intersectar os caminhos $P_{o_{1,z}}$ e $P_{o_{3,z}}$.

- Se $2 = 3 = False$ e $1 = True$: Neste caso, o caminho P_{c_z} vai possuir corpo vertical entre as linhas horizontais do corpo de $P_{o_{2,z}}$ e $P_{o_{3,z}}$. O corpo do caminho P_{c_z} está exatamente sobre a linha da grade, de forma a aresta-intersectar verticalmente o caminho $P_{o_{1,z}}$. Além disso P_{c_z} dobra e aresta-intersecta horizontalmente os próprios $P_{o_{2,z}}$ e $P_{o_{3,z}}$.

Os caminhos P_{d_z} e P_{e_z} dobram, respectivamente, nos pontos de interseção $P_{o_{2,z}} \cap_v P_{o_{1,z}}$ e $P_{o_{3,z}} \cap_v P_{o_{1,z}}$. Dessa forma $P_{d_z} \cap_e P_{o_{2,z}} \neq \emptyset$ e $P_{d_z} \cap_e P_{o_{1,z}} \neq \emptyset$. Analogamente $P_{e_z} \cap_e P_{o_{3,z}} \neq \emptyset$ e $P_{e_z} \cap_e P_{o_{1,z}} \neq \emptyset$.

O caminho P_{f_z} possui corpo vertical entre o corpo dos caminhos $P_{o_{2,z}}$ e $P_{o_{3,z}}$, e dobras em ambas extremidades para aresta-intersectar os caminhos $P_{o_{2,z}}$ e $P_{o_{3,z}}$.

- Se todas as variáveis tiverem a mesma valoração (indiferente se *True* ou *False*):
 - Neste caso a instância MONÓTONO-NAE-3-SAT é não satisfável, e os caminhos correspondendo aos vértices variáveis estão todos com corpo na mesma direção, portanto o caminho P_{c_z} não pode aresta-intersectar os três caminhos utilizando apenas duas dobras.

Dada uma representação B_2 -EPG do grafo síntese é fácil recuperar a valoração de cada variável $i \in \phi$, basta verificar: se o corpo do caminho correspondente P_{v_i} estiver posicionado verticalmente então $i = True$; e $i = False$, caso contrário.

□

A Figura 19 ilustra uma representação B_2 -EPG para uma cópia de ϕ com base na Figura 18, nela ϕ possui a seguinte valoração das variáveis de ϕ : $1 = 3 = 5 = True$ e $2 = 4 = False$.

Lema 31. $MONÓTONO-NAE-3-SAT \leq_p B_2-EPG-Helly$.

Demonstração. Dada uma fórmula $\phi \in MONÓTONO-NAE-3-SAT$ construímos o grafo síntese de G a partir da replicação da fórmula ϕ 21 vezes sobre uma cópia distinta do conjunto de variáveis. O Lema 27 efetua a construção de G . Perceba que o grafo G é construído em tempo polinomial.

Efetuada a construção do grafo G , explicaremos agora as principais ideias por trás desta redução polinomial.

A adição dos vértices a e b em G possui como objetivo evitar a exposição dos segmentos mais externos (pernas) dos caminhos P_{v_i} e $P_{o_{i,z}}$, de forma que nenhum

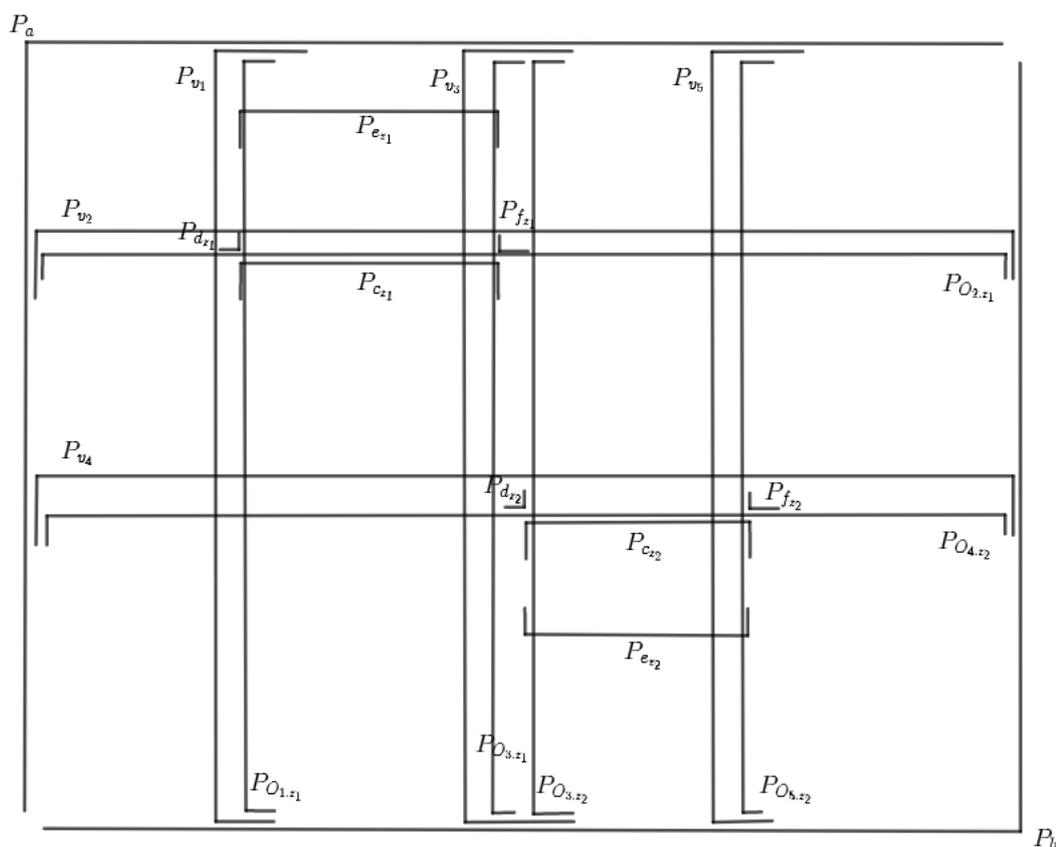


Figura 19 – Representação B_2 -EPG do grafo da Figura 18, uma cópia de ϕ , com as seguintes atribuição de ϕ : $1 = 3 = 5 = True$ e $2 = 4 = False$.

vértice-cláusula possa interseccionar esses caminhos em segmentos que não seja o corpo. Se os caminhos P_{v_i} e $P_{o_{i,z}}$ forem representados em uma interseção dos caminhos P_a e P_b ou em uma extremidade deles, essa configuração se torna ineficiente para o que queremos. O Lema 29 garante que existe uma cópia central que possui pernas totalmente cobertas por P_a e por P_b .

(\Rightarrow) Se ϕ é satisfatível então existe uma representação B_2 -EPG-Helly.

O Lema 30 mostra como construir uma representação B_2 -EPG do grafo G utilizando a valoração para as variáveis de ϕ' . Na verdade, o Lema 30 mostra como construir uma representação somente para cópia de ϕ . Mas ele mostra que quando ϕ não é satisfatível então não é possível representar a cópia central em B_2 -EPG. Por outro lado, quando ϕ é satisfatível podemos então representar ϕ e todas as outras cópias de ϕ em ϕ' utilizando a mesma estratégia. Por último, é fácil notar que a representação construída pelo Lema 30 é uma representação B_2 -EPG porque todo caminho possui no máximo duas dobraduras, e também é Helly porque todas as cliques nessa representação são representadas como edge-cliques. Portanto a representação é B_2 -EPG-Helly.

(\Leftarrow) Se G é B_2 -EPG-Helly então $\phi \in \text{MONÓTONO-NAE-3-SAT}$.

Dada a representação B_2 -EPG-Helly, vamos encontrar a fórmula $\phi \in \text{MONÓTONO-}$

NAE-3-SAT original e a valoração que gerou a representação.

Dada uma representação B_2 -EPG-Helly do grafo síntese, vamos recuperar ϕ . Primeiro, identifique uma cópia central, que deve ser um conjunto de caminhos, representando os vértices v_i e $o_{i,z}$, os quais estão “limpos”, e existem pelo Lema 29. Estes vértices são os únicos que intersectam P_a e P_b . Indiretamente, o caminho P_{c_i} vai indicar quem são exatamente as variáveis que compõem cada cláusula de ϕ . Para recuperar a valoração da variável $i \in \phi$, basta verificar: se o corpo do caminho correspondente P_{v_i} estiver posicionado verticalmente então $i = True$; e $i = False$, caso contrário. \square

Teorema 32. *Reconhecimento de grafos B_2 -EPG-Helly é NP-Completo.*

Demonstração. Pelo Lema 25 e o Lema 31. \square

6 CONSIDERAÇÕES FINAIS

Nesta pesquisa apresentamos resultados genéricos sobre relações de caminhos em representações EPG e particulares para caminhos em B_2 -EPG. Foi preciso efetuar uma série de definições para dar suporte aos lemas desenvolvidos ao desenrolar da pesquisa. Além disso, apresentamos algumas provas de pertinência, por prova de existência e apresentação de representações B_2 -EPG para os grafos bipartidos completos e para a classe dos grafos k -sun. Em suplemento, conseguimos mostrar que todo grafo k -sun $\in B_2$ -EPG-Helly. Por último, tratamos o problema de reconhecimento de grafos B_2 -EPG-Helly. A prova de pertinência a \mathcal{NP} já é dada por Santos (2020). Restando somente apresentar uma prova de dificuldade. A prova de dificuldade foi construída baseada no trabalho de Pergel e Rzażewski (2017), e usa como problema alvo da redução uma variação do problema de Satisfatibilidade booleana, bem conhecido ser NP-completo, Monótono-NAE-3-SAT.

De forma geral, os resultados obtidos com esta pesquisa mostram que os objetivos definidos inicialmente foram totalmente alcançados e geraram alguma contribuição para o conhecimento relacionado a grafos EPG. Em tempo, é importante ressaltar que essa pesquisa é um tipo de pesquisa de base e que os resultados aqui apresentados podem servir de trampolim para desenvolvimento de produtos e aplicações, relacionados ao problema de layout de circuitos como citado na motivação.

Como trabalhos futuros e continuidade desta pesquisa nos perguntamos se seria possível de alguma forma mostrar a NP-completude de toda hierarquia B_k -EPG e B_k -EPG-Helly utilizando um ferramental similar ao truque quantitativo utilizado neste trabalho para prova de NP-completude de B_2 -EPG-Helly. A ideia geral seria utilizar para cada k um grafo G que precisasse de k dobras para ser representado, em seguida restringir todos suas arestas relevantes e pontos de interseção (usando o truque quantitativo). Por último, definir um gadget relacionado a uma fórmula ϕ que só pudesse intersectar o corpo da representação de G sem utilizar arestas-relevantes e nem os pontos de interseção com outros caminhos. Assim, o gadget só poderia ser representado em B_k -EPG ou B_k -EPG-Helly quando ϕ fosse satisfatível. Se essa ideia estiver correta, então conseguiríamos provar a NP-completude de toda hierarquia B_k -EPG. Assim deixamos a seguinte conjectura.

Conjectura 33. *O problema de Reconhecimento de grafos B_k -EPG e B_k -EPG-Helly é NP-completo para $k \geq 1$.*

REFERÊNCIAS

- ALCÓN, L. et al. On the bend number of circular-arc graphs as edge intersection graphs of paths on a grid. **Discrete Applied Mathematics**, Elsevier, v. 234, p. 12–21, 2018.
- ALCÓN, L.; GUTIERREZ, M.; MAZZOLENI, M. P. A necessary condition for ept graphs and a new family of minimal forbidden subgraphs. **Matemática Contemporânea**, v. 39, p. 111–120, 2010.
- ASINOWSKI, A.; SUK, A. Edge intersection graphs of systems of paths on a grid with a bounded number of bends. **Discrete Applied Mathematics**, Elsevier, v. 157, n. 14, p. 3174–3180, 2009.
- BERGE, C.; DUCHET, P. A generalization of gilmore’s theorem. **Recent advances in graph theory**, Acad. Praha, Prague, p. 49–55, 1975.
- BIEDL, T.; STERN, M. On edge-intersection graphs of k-bend paths in grids. **Discrete Mathematics and Theoretical Computer Science**, v. 12, n. 1, p. 1–12, 2010.
- BONOMO, F.; MAZZOLENI, M. P.; STEIN, M. Clique coloring b1-epg graphs. **Discrete Mathematics**, Elsevier, v. 340, n. 5, p. 1008–1011, 2017.
- BORNSTEIN, C. et al. The complexity of Helly-B1 EPG graph recognition. **Discret. Math. Theor. Comput. Sci.**, v. 22, 2020.
- BOUGERET, M. et al. On independent set on B1-EPG graphs. In: SPRINGER. **International Workshop on Approximation and Online Algorithms**. [S.l.], 2015. p. 158–169.
- BOUSQUET, N.; HEINRICH, M. Computing maximum cliques in B2-EPG graphs. In: SPRINGER. **International Workshop on Graph-Theoretic Concepts in Computer Science**. [S.l.], 2017. p. 140–152.
- BOYACI, A. et al. Graphs of edge-intersecting non-splitting paths in a tree: Towards hole representations-part I. **CoRR**, abs/1309.2898, 2013. Disponível em: <<http://arxiv.org/abs/1309.2898>>.
- CAMERON, K.; CHAPLICK, S.; HOÀNG, C. T. Edge intersection graphs of l-shaped paths in grids. **Electronic Notes in Discrete Mathematics**, v. 44, p. 363–369, 2013. ISSN 1571-0653. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1571065313002734>>.
- CELA, E.; GAAR, E. Monotonic representations of outerplanar graphs as edge intersection graphs of paths on a grid. **arXiv preprint arXiv:1908.01981**, 2019.
- COOK, S. A. The complexity of theorem-proving procedures. In: **Proceedings of the third annual ACM symposium on Theory of computing**. [S.l.: s.n.], 1971. p. 151–158.

DOURADO, M. C. et al. Helly property, clique raphs, complementary graph classes, and sandwich problems. **Journal of the Brazilian Computer Society**, SpringerOpen, v. 14, n. 2, p. 45–52, 2008.

DOURADO, M. C.; PROTTI, F.; SZWARCFITER, J. L. Computational aspects of the helly property: a survey. **Journal of the Brazilian Computer Society**, Springer, v. 12, n. 1, p. 7–33, 2006.

EPSTEIN, D.; GOLUMBIC, M. C.; MORGENSTERN, G. Approximation algorithms for b 1-epg graphs. In: SPRINGER. **Workshop on Algorithms and Data Structures**. [S.I.], 2013. p. 328–340.

ERDÖS, P.; GOODMAN, A. W.; PÓSA, L. The representation of a graph by set intersections. **Canadian Journal of Mathematics**, Cambridge University Press, v. 18, p. 106–112, 1966.

FRANCIS, M. C.; LAHIRI, A. VPG and EPG bend-numbers of halin graphs. **Discrete Applied Mathematics**, Elsevier, v. 215, p. 95–105, 2016.

GAREY, M.; JOHNSON, D. **Computers and Intractability: A Guide to the Theory of NP-completeness**. W. H. Freeman, 1979. (Mathematical Sciences Series). ISBN 9780716710448. Disponível em: <<https://books.google.com.br/books?id=fjxGAQAIAAJ>>.

GAVRIL, F. The intersection graphs of subtrees in trees are exactly the chordal graphs. **Journal of Combinatorial Theory, Series B**, Elsevier, v. 16, n. 1, p. 47–56, 1974.

GAVRIL, F. A recognition algorithm for the intersection graphs of paths in trees. **Discrete Mathematics**, Elsevier, v. 23, n. 3, p. 211–227, 1978.

GOLUMBIC, M. C.; JAMISON, R. E. Edge and vertex intersection of paths in a tree. **Discrete Mathematics**, Elsevier, v. 55, n. 2, p. 151–159, 1985.

GOLUMBIC, M. C.; JAMISON, R. E. The edge intersection graphs of paths in a tree. **Journal of Combinatorial Theory, Series B**, Elsevier, v. 38, n. 1, p. 8–22, 1985.

GOLUMBIC, M. C.; LIPSHTEYN, M.; STERN, M. The recognition of k-epg graphs. **Congressus Numerantium**, Citeseer, v. 171, p. 129–139, 2004.

GOLUMBIC, M. C.; LIPSHTEYN, M.; STERN, M. Edge intersection graphs of single bend paths on a grid. **Networks: An International Journal**, Wiley Online Library, v. 54, n. 3, p. 130–138, 2009.

GOLUMBIC, M. C.; LIPSHTEYN, M.; STERN, M. Single bend paths on a grid have strong Helly number 4: errata atque emendationes ad “edge intersection graphs of single bend paths on a grid”. **Networks**, Wiley Online Library, v. 62, n. 2, p. 161–163, 2013.

HELDT, D.; KNAUER, K.; UECKERDT, T. Edge-intersection graphs of grid paths: the bend-number. **Discrete Applied Mathematics**, Elsevier, v. 167, p. 144–162, 2014.

- HELDT, D.; KNAUER, K.; UECKERDT, T. On the bend-number of planar and outerplanar graphs. **Discrete Applied Mathematics**, Elsevier, v. 179, p. 109–119, 2014.
- HOPCROFT, J. E.; MOTWANI, R.; ULLMAN, J. D. Introduction to automata theory, languages, and computation. **Acm Sigact News**, ACM New York, NY, USA, v. 32, n. 1, p. 60–65, 2001.
- JUNIOR, M. T. C. **ONUMERO DE HELLY NA CONVEXIDADE GEODÉTICA EM GRAFOS**. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 2016.
- KABELL, J. A. **Intersection graphs: structure and invariants**. [S.l.]: University of Michigan, 1980.
- KARP, R. M. Reducibility among combinatorial problems. In: **Complexity of computer computations**. [S.l.]: Springer, 1972. p. 85–103.
- LEKKEIKERKER, C.; BOLAND, J. Representation of a finite graph by a set of intervals on the real line. **Fundamenta Mathematicae**, v. 51, n. 1, p. 45–64, 1962. Disponível em: <<http://eudml.org/doc/213681>>.
- LIN, M. C.; SZWARCFITER, J. L. Faster recognition of clique-helly and hereditary clique-helly graphs. **Information Processing Letters**, Elsevier, v. 103, n. 1, p. 40–43, 2007.
- LOVÁSZ, L. **Coverings and coloring of hypergraphs**. [S.l.]: 4th SEICCGTC Proc., 1973. 3–12 p.
- MENEZES, P. B. **Linguagens formais e autômatos**. [S.l.]: Sagra-Dcluzzato, 1998.
- MOLITOR, P. A survey on wiring. **Elektronische Informationsverarbeitung und Kybernetik**, v. 27, n. 1, p. 3–19, 1991.
- MULDER, H. M.; SCHRIJVER, A. Median graphs and helly hypergraphs. **Discrete Mathematics**, Elsevier, v. 25, n. 1, p. 41–50, 1979.
- PAL, M. Intersection graphs: An introduction. **CoRR**, abs/1404.5468, 2014. Disponível em: <<http://arxiv.org/abs/1404.5468>>.
- PERGEL, M.; RZAŻEWSKI, P. On edge intersection graphs of paths with 2 bends. **Discrete Applied Mathematics**, Elsevier, v. 226, p. 106–116, 2017.
- PETITO, P. **Grafos de Interseção em Arestas de Caminhos em uma Árvore**. Tese (Doutorado) — Dissertação de Mestrado, Programa de Engenharia de Sistemas e Computação ..., 2002.
- PINTO, J. W. C. **Grafos Orth [h, s, t]**. Tese (Doutorado) — Tese de Doutorado, UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2018.
- SAFE, M. D. Essential obstacles to Helly circular-arc graphs. **arXiv preprint arXiv:1612.01513**, 2016.

SANTOS, T. et al. On the helly property of some intersection graphs. In: **Anais do XXXIV Concurso de Teses e Dissertações**. Porto Alegre, RS, Brasil: SBC, 2021. p. 19–24. ISSN 2763-8820. Disponível em: <<https://sol.sbc.org.br/index.php/ctd/article/view/15752>>.

SANTOS, T. D. dos. **ON THE HELLY PROPERTY OF SOME INTERSECTION GRAPHS**. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 2020.

SCHAEFER, T. J. The complexity of satisfiability problems. In: **Proceedings of the Tenth Annual ACM Symposium on Theory of Computing**. New York, NY, USA: Association for Computing Machinery, 1978. (STOC '78), p. 216–226. ISBN 9781450374378. Disponível em: <<https://doi.org/10.1145/800133.804350>>.

SINDEN, F. W. Topology of thin film rc circuits. **Bell System Technical Journal**, Wiley Online Library, v. 45, n. 9, p. 1639–1662, 1966.

SZPILRAJN-MARCZEWSKI, E. Sur deux propriétés des classes d'ensembles. **Fundamenta Mathematicae**, v. 33, n. 1, p. 303–307, 1945. Disponível em: <<http://eudml.org/doc/213098>>.

SZWARCFITER, J. **Teoria computacional de grafos: Os Algoritmos**. ELSEVIER, 2018. ISBN 9788535288858. Disponível em: <<https://books.google.com.br/books?id=OTBbDwAAQBAJ>>.

A PROVA DE COMPLEXIDADE DE GRAFOS B_2 -EPG (ANEXO)

Com o seguinte link: <<https://doi.org/10.1016/j.dam.2017.04.023>>, podemos ter acesso ao trabalho realizado por Pergel e Rzażewski (2017), onde foi abordado a complexidade de reconhecimento de grafos B_2 -EPG. Nele pode ser observado uma transformação polinomial, transformação esta, usada por como elemento de estudo para o desenvolvimento desse trabalho de conclusão de curso.