



Universidade Federal do Ceará

Centro de Ciências

Departamento de Computação

Mestrado e Doutorado em Ciência da Computação

Resultados no Tempo Máximo e no Número de Envoltória nas Convexidades P_3 e Geodésica

Thiago Braga Marcilon

Fortaleza, Ceará

Janeiro, 2017



Universidade Federal do Ceará

Centro de Ciências

Departamento de Computação

Mestrado e Doutorado em Ciência da Computação

Resultados no Tempo Máximo e no Número de Envoltória nas Convexidades P_3 e Geodésica

Autor

Thiago Braga Marcilon

Orientador

Prof. Dr. Rudini Menezes Sampaio

Tese apresentada ao programa Mestrado e Doutorado em Ciência da Computação da Universidade Federal do Ceará como parte dos requisitos necessários à obtenção do título de Doutor em Ciência da Computação.

Fortaleza, Ceará

Janeiro, 2017

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- M266r Marcilon, Thiago Braga.
Resultados no tempo máximo e no número de envoltória nas convexidades p3 e geodésica / Thiago Braga Marcilon. – 2017.
188 f. : il. color.
- Tese (doutorado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2017.
Orientação: Prof. Dr. Rudini Menezes Sampaio.
1. Convexidade em grafos. 2. Número de envoltória. 3. Tempo máximo de infecção. 4. Complexidade clássica. 5. Complexidade parametrizada. I. Título.

CDD 005

“Pois dele, por ele e para ele são todas as coisas. A ele seja a glória para sempre. Amém.”
(Bíblia Sagrada, Romanos 11:36)

AGRADECIMENTOS

Primeiramente, eu agradeço a Deus pela sua providência e cuidado comigo durante toda a minha vida. Especialmente nesses últimos quatro anos e meio, experimentei na pele o ditado “Deus sabe o que faz” e, por isso, sou grato a Ele.

Agradeço à minha esposa, Paloma, pelo suporte, paciência e palavras de encorajamento durante o doutorado, sobretudo nas horas mais difíceis. Agradeço também a toda a minha família por sempre ter deixado claro para mim que ajuda não tardaria caso eu precisasse e, de fato, nunca tardou. Agradeço especialmente aos meus pais e a minha irmã Lilian, pela preocupação e cuidado comigo não só durante o doutorado, mas durante toda a minha vida.

Gostaria de agradecer ao professor Rudini pela orientação e paciência, sobretudo com a minha escrita. Agradeço ainda à banca por aceitar ler a minha tese em apenas um mês até a defesa. Também queria agradecer aos professores Júlio Araújo, Ignasi Sau Valls e Cláudia Sales pelas valiosas dicas que fizeram desta tese uma tese de fato. Agradeço também a todos os meus professores, sejam os que encontrei na graduação, no mestrado ou ainda no doutorado, que, sem dúvida, me ajudaram a me tornar a pessoa que sou academicamente falando, sobretudo os professores do grupo de pesquisa ParGO.

Gostaria ainda de agradecer ao Max e à Nane pela amizade duradoura que temos e que muito me ajuda a enfrentar as adversidades da vida ainda que eles talvez não saibam disso. Agradeço também ao pessoal do Lab. 2: o Nicolas, Rafael, Renan e vários outros pelas conversas e comédias no laboratório e nos congressos que fomos juntos.

*“O homem sábio tem olhos que enxergam,
mas o tolo anda nas trevas; todavia, percebi
que ambos têm o mesmo destino.”*

(Bíblia Sagrada, Eclesiastes 2:14)

RESUMO

Convexidade em grafos é um conceito inspirado na convexidade da geometria euclidiana e vem sendo bastante explorado nas últimas décadas. Muitas aplicações reais como a difusão de uma infecção ou de informação em uma rede social podem ser modeladas usando alguma convexidade em grafos baseada em caminhos do grafo. Nesta tese, nós abordamos dois problemas computacionais relacionados à convexidade em grafos: NÚMERO DE ENVOLTÓRIA e TEMPO MÁXIMO DE INFECÇÃO. Fixada uma convexidade C , o problema NÚMERO DE ENVOLTÓRIA consiste em, dados um grafo G e inteiro k , determinar se o número de envoltória de G na convexidade C é no máximo k . O problema TEMPO MÁXIMO DE INFECÇÃO consiste em, dados um grafo G e inteiro k , determinar se o tempo máximo de infecção de G na convexidade C é no mínimo k . Apresentamos algoritmos polinomiais para o problema TEMPO MÁXIMO DE INFECÇÃO na convexidade P_3 em algumas classes de grafos e para k fixo. Apresentamos também algoritmos tratáveis por parâmetro fixo na combinação dos parâmetros grau máximo do grafo e k e na combinação dos parâmetros largura em árvore do grafo e k . Também apresentamos resultados de **NP**-completude para o problema TEMPO MÁXIMO DE INFECÇÃO na convexidade P_3 em algumas classes de grafos e para algumas restrições da entrada k . Apresentamos ainda a sua **W[1]**-dificuldade para o parâmetro largura em árvore. Finalmente, mostramos que o problema NÚMERO DE ENVOLTÓRIA na convexidade P_3 é **W[1]**-difícil no parâmetro k em grafos livres de K_3 com grau máximo seis. Mostramos também que o problema NÚMERO DE ENVOLTÓRIA na convexidade geodésica é **W[1]**-difícil no parâmetro k em grafos com diâmetro dois e na combinação dos parâmetros largura em árvore e k . Apresentamos ainda um algoritmo XP no parâmetro largura em árvore que calcula o número de envoltória de um grafo na convexidade geodésica.

Palavras-chave: Convexidade em Grafos. Número de Envoltória. Tempo Máximo de Infecção. Complexidade Clássica. Complexidade Parametrizada.

ABSTRACT

Graph convexity is a concept inspired by the convexity notions in the euclidian geometry and has been receiving a lot of attention in the last few decades. Several real world applications like spread of an infection or information in a social network can be modeled using some graph convexity based on paths in the graph. In this thesis, we consider two computational problems relating to graph convexity: HULL NUMBER and MAXIMUM INFECTION TIME. Fixed a convexity C , the HULL NUMBER problem consists in, given a graph G and an integer k , determine whether the hull number of G in the convexity C is at most k . The MAXIMUM INFECTION TIME problem consists in, given a graph G and an integer k , determine whether the hull number of G in the convexity C is at least k . We present some polynomial algorithms for the MAXIMUM INFECTION TIME problem in the P_3 convexity for some graph classes and when we fix the input k . We also present fixed parameter tractable algorithms when we combine the parameters maximum degree of the graph and the input k and when we combine the parameters treewidth of the graph and the input k . Additionally, we present **NP**-completeness results for the MAXIMUM INFECTION TIME problem in the P_3 convexity for some graph classes and when we restrict the input k . We also prove that the problem is **W[1]**-hard when parameterized by the treewidth of the graph. Finally, we show that the HULL NUMBER problem in the P_3 convexity is **W[1]**-hard when parameterized by k even in K_3 -free graphs with maximum degree six. We also show that the HULL NUMBER problem in the geodesic convexity is **W[1]**-hard when parameterized by k even in graphs with diameter two and when we combine the parameters treewidth of the graph and k . Furthermore, we present a XP algorithm parameterized by the treewidth of the graph that computes the hull number of a graph in the geodesic convexity.

Keywords: Convexity in Graphs. Hull Number. Maximum Infection Time. Classical Complexity. Parameterized Complexity.

LISTA DE FIGURAS

Figura 1	– A representação gráfica de um grafo.	27
Figura 2	– Operação produto cartesiano entre dois grafos.	28
Figura 3	– Árvore estritamente binária à esquerda e binária completa à direita sendo os nós r as suas raízes.	31
Figura 4	– Um grafo G à esquerda e uma decomposição em árvore de G à direita.	33
Figura 5	– Exemplo de circuito booleano com entrelaçamento 2 e entradas e_1, \dots, e_5	36
Figura 6	– Um grafo G tal que $hn_g(G) = 2$ e $hn_{P_3}(G) = 3$	39
Figura 7	– Um grafo G tal que $t_g(G) = 1$ e $t_{P_3}(G) = 2$	39
Figura 8	– Tempos de infecção dos conjuntos $T_0 \cup N_{\geq 3}(u) \cup \{v, s\}$ à esquerda, e $T_0 \cup N_{\geq 3}(u) \cup \{v, s, q\}$ à direita.	45
Figura 9	– O conjunto $\{t_1, t_2, t_3, t_4\}$ está em \mathcal{T}_0^u , com $t_1 \in V(H_{x,1})$, $t_2 \in V(H_{x,2})$, $t_3 \in V(H_{z,1})$ e $t_4 \in V(H_{y,1})$	47
Figura 10	– Vértices da componente C_i antes e depois da união de $\{y_i\}$ à S_{i-1} com seus tempos de infecção.	49
Figura 11	– Conjuntos P_u^u , Q_u^u , P_v^u e Q_v^u	51
Figura 12	– Vértices da componente C_i antes e depois da adição de y_i a S_{i-1} com seus tempos de infecção.	54
Figura 13	– Exemplo de grafo grade sólido.	55
Figura 14	– Um grafo com $\Delta = 3$ infectado pelo conjunto S' à esquerda e pelo conjunto S à direita com os tempos de infecção dos vértices.	57
Figura 15	– Um grafo grade sólido com grau máximo três.	58
Figura 16	– O grafo resultante da transformação do grafo da Figura 15.	59
Figura 17	– <i>Gadget</i> da cláusula C_i com os tempos de infecção de seus vértices.	70
Figura 18	– <i>Gadget</i> bipartido adicionado para cada cláusula C_i com os tempos de infecção de seus vértices.	71
Figura 19	– Grafo grade com $\Delta = 3$	73
Figura 20	– Bloco 4x4 antes e depois da adição dos vizinhos dos vértices auxiliares.	74
Figura 21	– Grafo grade resultante da aplicação da redução ao grafo da Figura 19.	75
Figura 22	– <i>Gadget</i> de cada cláusula C_i com os tempos de infecção dos vértices para o conjunto $\{u_{i,3}\} \cup$ todos os vértices pendentes.	77
Figura 23	– Exemplo de árvore $(\Delta - 2)$ -ária cheia adicionada a G , para $\Delta = 4$ e $y = 6$	78
Figura 24	– Uma aresta com peso à direita e a sua estrutura real à esquerda.	80

Figura 25 – Exemplo de <i>gadget</i> Q^x	80
Figura 26 – <i>Gadget</i> que colocamos entre <i>gadgets</i> de cor sempre que não houver aresta entre os vértices v_i e v_j	81
Figura 27 – Grafo G	81
Figura 28 – Primeira parte do grafo G' resultante da aplicação da redução no grafo da Figura 27.	82
Figura 29 – Ligações dos vértices em L	83
Figura 30 – Segunda parte do grafo G' resultante da aplicação da redução no grafo da Figura 27.	83
Figura 31 – BAEB com biraiz $\{r, s\}$ e três bifolhas $\{l_1, l_2\}$, $\{l_3, l_4\}$ e $\{l_5, l_6\}$	87
Figura 32 – <i>Gadget de cor</i> parcial para a cor x e $d_1^x = 3$	89
Figura 33 – <i>Gadget</i> de aresta das cores x e x' com arestas entre os vértices v_1^x e $v_3^{x'}$, v_1^x e $v_4^{x'}$, e v_2^x e $v_3^{x'}$	90
Figura 34 – Instância do problema REVESTIMENTO DE GRADE com $k = 3$ e $n = 5$	92
Figura 35 – Caminhos $P_{1,1}$ e $R_{1,1}$ para $S_{1,1} = \{(2, 4), (1, 3), (5, 4), (2, 2)\}$	93
Figura 36 – Vértices u , z e z' e árvore $T_{5,4}^1$. A subárvore $T_{5,4}^{1,e}$ está destacada em negrito. Note que $l_{5,4}^{1,w}$ e $z_{5,4}^{1,1}$ não estão em $V(T_{5,4}^{1,e})$	94
Figura 37 – <i>Gadget</i> de aresta $E_{1,2}$, onde $S_{1,1} = \{(2, 4), (1, 3), (5, 4)\}$ e $S_{1,2} = \{(5, 4), (7, 3)\}$	95
Figura 38 – <i>Gadgets</i> de conjunto $G_{1,1}$ e $G_{1,2}$ e <i>gadget</i> de aresta $E_{1,2}$ sem os vértices z e z' . As arestas entre u e os vértices em $L \cup \{a_1^2, b_1^2\}$ foram omitidas.	96

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Aplicações para convexidade	16
1.2	Estado da arte e contribuições	17
1.3	Estrutura da tese	23
2	PRELIMINARES	25
2.1	Definições e notações básicas	25
2.1.1	Conjuntos, funções e operadores lógicos	25
2.1.2	Grafos	26
2.2	Largura em Árvore	32
2.3	Teoria da Complexidade Parametrizada	34
2.4	Hipótese do Tempo Exponencial	36
2.5	Convexidade	37
3	ALGORITMOS PARA O TEMPO MÁXIMO DE INFECCÃO NA CONVEXIDADE P_3	41
3.1	Algoritmo polinomial para $k = n - c$ onde c é fixo	42
3.2	Discussão sobre os resultados polinomiais para k fixo	43
3.3	Algoritmo polinomial para $k = 3$ em grafos bipartidos	44
3.4	Algoritmo polinomial para $k = 3$ em grafos quaisquer	46
3.5	Algoritmo polinomial para $k = 4$ em grafos bipartidos	50
3.6	Algoritmo polinomial em grafos grade sólidos com grau máximo três	55
3.7	Algoritmo FPT no parâmetro $\Delta + k$	60
3.8	Resultados parametrizados na largura em árvore	61
3.8.1	Aplicação do Teorema de Courcelle	61
3.8.2	Algoritmos parametrizados	63
4	RESULTADOS DE COMPLEXIDADE PARA TEMPO MÁXIMO DE INFECCÃO NA CONVEXIDADE P_3	68
4.1	NP-Completeness para $k = n - n^\epsilon + 4$	69
4.2	NP-Completeness para $k \geq 5$ em grafos bipartidos	71
4.3	NP-Completeness em grafos grade com grau máximo três	72
4.4	NP-Completeness para $k = \Theta(\log n)$ em grafos com grau máximo limitado	76

4.5	W[1]-dificuldade no parâmetro largura em árvore	79
5	RESULTADOS PARAMETRIZADOS PARA O NÚMERO DE ENVOL- TÓRIA	86
5.1	W[1]-dificuldade no parâmetro k nas convexidades P_3 e Geodésica . .	87
5.2	W[1]-dificuldade no parâmetro $tw(G) + k$ na convexidade Geodésica .	92
5.3	Consequências da Hipótese do Tempo Exponencial	98
5.4	Algoritmo XP no parâmetro $tw(G)$ na convexidade Geodésica	99
6	CONCLUSÃO	105
6.1	Questões em aberto	106
6.2	Trabalhos futuros	108
	REFERÊNCIAS BIBLIOGRÁFICAS	109
	APÊNDICES	119
	APÊNDICE A – THE MAXIMUM TIME OF 2-NEIGHBOR BOOTS- TRAP PERCOLATION: COMPLEXITY RESULTS .	120
	APÊNDICE B – PERCOLATION TIME PROBLEM IN GRID GRAPHS AND SOME PARAMETERIZED RESULTS	140
	APÊNDICE C – ON THE PARAMETERIZED COMPLEXITY OF THE HULL NUMBER PROBLEM	165

1 INTRODUÇÃO

Nesta tese, nós apresentamos contribuições na área de convexidade em grafos. O conceito de convexidade provém da noção de conjuntos convexos no espaço euclidiano. No espaço euclidiano, um conjunto convexo é um conjunto C no qual todos os pontos no segmento de reta entre quaisquer dois pontos de C estão em C e a envoltória de um conjunto C é o menor conjunto convexo que contém C . Esses e outros conceitos podem ser estendidos para outros espaços além do euclidiano. Sobre convexidade no espaço euclidiano, nós referimos o leitor ao livro de Rockafellar [110].

Uma *convexidade* finita [72, 84, 114] em um conjunto V é uma família finita de subconjuntos \mathcal{C} de V que satisfaz as seguintes condições: $\emptyset, V \in \mathcal{C}$ e, se $C_1, C_2 \in \mathcal{C}$, então $C_1 \cap C_2 \in \mathcal{C}$. Os subconjuntos em \mathcal{C} são chamados de *conjuntos convexos*. A *envoltória* $H(C)$ de um conjunto C é o menor conjunto convexo que contém C . Um conjunto $C \subseteq V$ é um *conjunto de envoltória* se $H(C) = V$.

Uma convexidade \mathcal{C} em V é uma *convexidade de intervalo* se existe uma função, chamada de *função de intervalo*, $I : \binom{V}{2} \rightarrow 2^V$ tal que $C \in \mathcal{C}$ se e somente se $I(\{x, y\}) \subseteq C$ para todo par de elementos distintos x e y de C . Assim, é comum definir uma convexidade de intervalo apenas pela sua função de intervalo já que, uma vez que a função de intervalo está definida, todos os conjuntos convexos daquela convexidade também estão.

Uma convexidade em grafos consiste em uma convexidade cujo conjunto universo é o conjunto de vértices de um grafo. As convexidades em grafos mais estudadas são convexidades de intervalo em que a sua função de intervalo retorna os vértices em todos os caminhos entre os vértices do argumento com alguma propriedade.

Nesta tese, trabalhamos com convexidades de intervalo em grafos. Os primeiros estudos de convexidade em grafos são da década de 70 com os trabalhos de Pfaltz [101], Erdős [66] e Moon [92]. Alguns exemplos de convexidades de intervalo em grafos são a convexidade geodésica [68], a convexidade P_3 [66], a convexidade P_3^* [8] e a convexidade monofônica [64]. Na *convexidade geodésica*, $I(\{x, y\})$ retorna todos os vértices em um caminho mínimo entre x e y . Na *convexidade P_3* , $I(\{x, y\})$ retorna todos os vértices em um caminho com três vértices entre x e y . Na *convexidade P_3^** , $I(\{x, y\})$ retorna todos os vértices em um caminho induzido com três vértices entre x e y . Na *convexidade monofônica*, $I(\{x, y\})$ retorna todos os vértices em um caminho induzido entre x e y .

Vários parâmetros em grafos relacionados à convexidade têm sido explorados na literatura como o número de Radon [56, 104], o número de Carathéodory [17, 33, 45], o número de envoltória [36, 42, 67], o número de convexidade [39, 40], o número de intervalo [55, 73] e o tempo máximo de infecção [19, 20, 103]. Os parâmetros número de envoltória e o tempo máximo de infecção são particularmente relevantes para esta tese.

O *número de envoltória*, do inglês *hull number*, na convexidade \mathcal{C} de um grafo G , denotado por $hn_{\mathcal{C}}(G)$, é o tamanho do menor conjunto de envoltória de G na convexidade \mathcal{C} . Ele vem sendo bastante estudado desde 1985, ano em que foi definido por Everett e Seidman [67], na ocasião, na convexidade geodésica.

O *tempo máximo de infecção*, do inglês *maximum infection time*, na convexidade de intervalo \mathcal{C} de um grafo $G = (V, E)$, denotado por $t_{\mathcal{C}}(G)$, é o maior t para o qual existe um subconjunto de vértices V' tal que $I^{t-1}[V'] \subset I^t[V'] = V$, onde I é a função de intervalo de \mathcal{C} , $I[V'] = \bigcup_{x,y \in V'} I(\{x, y\})$ e $I^k[\cdot]$ é a composição da função I k vezes. O parâmetro tempo máximo de infecção se originou de uma pergunta proposta por Bollobás e foi estudado primeiramente por Przykucki [103], porém sob um modelo de processo de nome *2-neighbour bootstrap percolation*, o qual equivale ao processo de criação da envoltória de um conjunto na convexidade P_3 através das sucessivas aplicações da sua função de intervalo. O seu nome vem da similaridade do *2-neighbour bootstrap percolation* a um processo de infecção de vértices. Quando a convexidade estiver implícita, podemos omitir o subscrito das notações $t_{\mathcal{C}}(G)$ e $hn_{\mathcal{C}}(G)$.

Fixada uma convexidade \mathcal{C} , o problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE \mathcal{C} consiste em, dados um grafo G e um inteiro k , determinar se $hn_{\mathcal{C}}(G) \leq k$ e o problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE \mathcal{C} consiste em, dados um grafo G e um inteiro k , determinar se $t_{\mathcal{C}}(G) \geq k$. Nesta tese, exploramos os problemas computacionais NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE GEODÉSICA, NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE P_3 e TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 .

Os três problemas trabalhados nesta tese são conhecidamente **NP**-completos [18, 42, 53]. Um problema de decisão é **NP**-completo se ele é **NP** e todos os problemas em **NP** se reduzem a ele através de uma redução polinomial. Assim, temos que um problema **NP**-completo possui um algoritmo polinomial se e somente se **P** = **NP**, porém, até hoje, não se conhece nenhum problema **NP**-completo que possua um algoritmo polinomial.

Portanto, como os três problemas são **NP**-completos, nós tentamos contornar a dificuldade inerente a eles de duas formas. Na primeira forma, tentamos determinar a dificuldade de cada um dos três problemas quando restringimos a sua entrada, ou seja,

queremos determinar se eles permanecem **NP**-completos ou se passam a possuir algoritmos polinomiais quando restringimos o grafo de entrada ou o inteiro k .

Na segunda forma, estamos interessados em obter algoritmos cuja complexidade de tempo seja polinomial no tamanho da entrada e superpolinomial apenas no valor de algum parâmetro da entrada, por exemplo, um algoritmo cuja complexidade seja $O(2^d \cdot n^3)$, onde d é o diâmetro do grafo e n o número de vértices do grafo, pois, assim, teríamos um algoritmo eficiente quando o valor do parâmetro é baixo, apesar de ainda ser um algoritmo superpolinomial.

Para isso, nós investigamos a complexidade das versões parametrizadas dos três problemas acima. Na teoria da complexidade parametrizada [51, 62, 70], um *problema parametrizado* é uma dupla (P, k) , onde P é um problema de decisão e k , chamado de parâmetro do problema, é uma função nas instâncias de P que retorna um inteiro não-negativo. Na verdade, um problema pode ser parametrizado por mais de um parâmetro. Nesse caso, um problema parametrizado na combinação de p parâmetros seria uma $(p + 1)$ -upla (P, k_1, \dots, k_p) , onde k_1, \dots, k_p são funções nas instâncias de P que retornam um inteiro não-negativo. Assim, quando nos referimos ao problema P parametrizado por $k_1 + k_2 + \dots + k_p$, estamos nos referindo ao problema parametrizado (P, k_1, \dots, k_p) . As definições a seguir que dizem respeito a teoria da complexidade parametrizada serão feitas usando apenas um parâmetro, porém elas podem ser facilmente extrapoladas para dois ou mais parâmetros.

Assim, nós queremos determinar se os problemas parametrizados que nós trabalhamos possuem um algoritmo tratável por parâmetro fixo ou dar indícios que tal algoritmo não existe. Um algoritmo *tratável por parâmetro fixo*, do inglês *fixed parameter tractable* (FPT), é um algoritmo cujo tempo de execução é $f(k(I)) \cdot n^{O(1)}$, onde I é a instância do problema passada ao algoritmo, n é o tamanho da instância I e f é uma função computável qualquer que não depende de n .

Um algoritmo FPT se torna polinomial se restringirmos o problema a instâncias I tais que $k(I) \leq c$ para alguma constante c . Assim, achar algoritmos tratáveis por parâmetro fixo é uma forma de mitigar a dificuldade de se encontrar algoritmos polinomiais para problemas **NP**-completos, pois, nas instâncias que possuem o valor do parâmetro pequeno, o algoritmo executa em um tempo aceitável dado que o tempo do algoritmo, na prática, se torna polinomial no tamanho da entrada.

Uma *redução parametrizada* de um problema parametrizado (P_1, k_1) para outro (P_2, k_2) é uma dupla (f, g) tal que, para cada instância I de P_1 , (a) $f(I)$ é uma instância SIM de P_2 se e somente se I é uma instância SIM de P_1 , (b) f é computável por um algoritmo FPT com

relação ao parâmetro k_1 e (c) g é uma função computável tal que $k_2(f(I)) \leq g(k_1(I))$ para toda instância I de P_1 . Assim, se A e B são problemas parametrizados tais que A possui uma redução parametrizada para B e B é um problema em **FPT**, então A também é **FPT**.

Assim como na teoria da complexidade clássica onde temos a conjectura bastante aceita de que $\mathbf{P} \neq \mathbf{NP}$, na teoria da complexidade parametrizada temos a conjectura, também geralmente aceita, de que $\mathbf{FPT} \neq \mathbf{W}[1]$, onde **FPT** é a classe de problemas parametrizados que possuem algoritmos FPT e $\mathbf{W}[1]$ é a classe de problemas parametrizados que possuem uma redução parametrizada para o problema SATISFABILIDADE DE CIRCUITOS BOOLEANOS restritos a circuitos com entrelaçamento 1 parametrizado pelo número de nós de entrada que recebem o valor 1, onde o entrelaçamento de um circuito booleano é o máximo número de portas com grau de entrada pelo menos três em um caminho entre um nó de entrada e o nó de saída.

Dessa forma, como $\mathbf{W}[1]$ -difícil é o conjunto de problemas parametrizados para os quais todos os problemas em $\mathbf{W}[1]$ possuem uma redução parametrizada, se mostramos que um problema parametrizado é $\mathbf{W}[1]$ -difícil, estamos mostrando que ele não possui um algoritmo tratável por parâmetro fixo a não ser que $\mathbf{FPT} = \mathbf{W}[1]$. Assim, uma vez que provamos que um determinado problema é $\mathbf{W}[1]$ -difícil, estamos dando fortes indícios que ele não possui um algoritmo FPT.

No entanto, mesmo aceitando a conjectura de que $\mathbf{FPT} \neq \mathbf{W}[1]$, o fato de um problema ser $\mathbf{W}[1]$ -difícil não exclui a possibilidade de haver um algoritmo que o decide em tempo $f(k(I)) \cdot n^{g(k(I))}$, onde f e g são funções computáveis quaisquer. Tais algoritmos são polinomiais se restringirmos o problema a instâncias I tais que $k(I) \leq c$ para qualquer constante c , porém o expoente de n pode depender do valor $k(I)$. Tais algoritmos são chamados XP e a classe dos problemas parametrizados que possuem algoritmos XP é a classe **XP**. Assim, quando provamos que um problema é **XP**, nossa próxima linha de ação é tentar mostrar que ele é **FPT** ou $\mathbf{W}[1]$ -difícil.

Outra forma que usamos para obter limites inferiores para o tempo de algoritmos que decidem os problemas parametrizados trabalhados é através da *Hipótese do Tempo Exponencial* [75, 76], do inglês *Exponential Time Hypothesis* (ETH). Introduzida por Impagliazzo e Paturi em 2001, a ETH declara que existe uma constante $\delta_3 > 0$ tal que, para qualquer polinômio $p(\cdot)$ e constante $c < \delta_3$, o problema 3-SAT não pode ser decidido em tempo $O(2^{cN} p(N))$, onde N é o número de variáveis na instância do problema.

Uma das consequências da ETH é que, para qualquer função computável f , o problema CLIQUE [43] não pode ser decidido em tempo $f(k) \cdot n^{o(k)}$ para qualquer função computável f ,

onde n é o número de vértices do grafo, pois isso implicaria em um algoritmo que resolve 3-SAT em tempo subexponencial o que contradiz a ETH. Assim, nós usamos a ETH para mostrar limites inferiores específicos para os problemas que trabalhamos, mostrando que os nossos algoritmos XP e FPT são assintoticamente ótimos em relação a função do parâmetro no expoente de n .

Um dos parâmetros em grafos que utilizamos amplamente nesta tese é o parâmetro largura em árvore. A largura em árvore é um parâmetro em grafos que foi introduzido com esse nome em 1984 por Robertson e Seymour [107] e mede o quão um grafo se parece com uma árvore. Desde a sua criação, o parâmetro largura em árvore vem sendo amplamente usado na derivação de resultados estruturais [108, 109, 112] e computacionais [10, 27, 48, 95, 115]. Vários problemas NP-completos em grafos possuem algoritmos FPT ou XP no problema parametrizado pela largura em árvore do grafo de entrada como os problemas COBERTURA DE VÉRTICES [44] e 3-COLORAÇÃO [10].

Portanto, dada a sua importância na área de teoria dos grafos e construção de algoritmos, nós utilizamos o parâmetro largura em árvore para derivar algoritmos parametrizados e resultados de $\mathbf{W}[1]$ -dificuldade nesta tese. A técnica que usamos nesta tese para conseguir algoritmos parametrizados pela largura em árvore é uma técnica de programação dinâmica bastante utilizada [10, 25, 31, 47, 77].

1.1 Aplicações para convexidade

O conceito de convexidade em grafos possui aplicações em diversas áreas. Por exemplo, podemos citar aplicações em redes neurais orgânicas [5, 65], difusão de opinião [63, 79, 102], movimento de fluidos [2, 29], sistemas distribuídos [80, 82, 93], estratégias de marketing [52] e difusão de epidemias [90, 91, 105].

Em [63], Dreyer e Roberts consideram alguns modelos para difusão de uma infecção ou opinião em redes sociais. A rede social é modelada através de um grafo em que indivíduos são representados por vértices em um grafo e as arestas representam pessoas que possuem uma conexão, seja ela de proximidade ou de influência nos casos da difusão de infecção ou opinião respectivamente. Em particular, eles dão mais ênfase ao modelo *k-limite irreversível*, do inglês *irreversible k-threshold*, em que o processo se dá em passos, onde os vértices estão inicialmente no estado 0 ou 1, um vértice nunca pode mudar do estado 1 para o estado 0 e um vértice no estado 0 muda para o estado 1 em um dado passo se ele possui k vizinhos no estado 1 no passo anterior.

Os resultados principais de [63] concernem o que eles chamam de *conjuntos de k-*

conversão, do inglês *k-conversion sets*, que nada mais são do que conjuntos de vértices inicialmente no estado 1 que eventualmente fazem com que todos os vértices do grafo mudem para o estado 1. Esses conjuntos são importantes porque modelam o conjunto de pessoas que inicialmente precisam se convencer de algo para que, então, elas possam convencer a rede inteira.

Perceba que o modelo 2-limite irreversível é exatamente o processo de criação da envoltória de um conjunto através de sucessivas aplicações da função de intervalo na convexidade P_3 e os conjuntos de 2-conversão são, na realidade, conjuntos de envoltória na convexidade P_3 . De fato, alguns dos resultados são sobre o número de envoltória de grafos na convexidade P_3 .

Em um exemplo mais prático [65], porém um pouco mais distante da convexidade P_3 , é estudado o comportamento de ativação de neurônios em redes neurais criadas *in vitro*. Nesse artigo, dentre outros resultados, eles estudaram diversas características de uma rede neural criada *in vitro* como a velocidade de propagação, transmissão sináptica e criação da informação. Para isso, é crítico se saber como acontece a propagação dos impulsos elétricos na rede neural. Em uma das topologias estudadas, que é uma grade de duas dimensões, a propagação é modelada por um modelo de *bond-percolation* da seguinte forma: um grupo de neurônios são inicialmente excitados e um neurônio excitado envia um impulso elétrico para seus vizinhos debaixo, à esquerda e à direita com uma probabilidade p , excitando-os se for bem sucedido.

1.2 Estado da arte e contribuições

Ambos os parâmetros número de envoltória e tempo máximo de infecção nas convexidades geodésica e P_3 são bastante explorados na literatura. Nesta seção, apresentamos um breve e, portanto, não exaustivo, apanhado de resultados computacionais e de limites recentes relevantes sobre ambos os problemas em ambas as convexidades em grafos simples e não-direcionados, dando mais ênfase aos resultados computacionais. Adicionalmente, apresentamos as nossas contribuições para o quadro geral dos problemas NÚMERO DE ENVOLTÓRIA nas convexidades P_3 e geodésica e TEMPO MÁXIMO DE INFECÇÃO na convexidade P_3 .

É importante ressaltar que o processo de criação da envoltória de um conjunto na convexidade P_3 através das sucessivas aplicações da função de intervalo vem sendo estudada sob diversos nomes diferentes na literatura como *irreversible 2-threshold process* [63], *2-neighbour bootstrap percolation* [3], *IRR₂-conversion sets* [35] e *activation process* [22]. Muitos trabalhos como em [13–16, 28, 74] se dedicam ao cálculo da probabilidade de um conjunto de vértices escolhidos aleatoriamente em alguma distribuição ser um conjunto de envoltória de

um grafo na convexidade P_3 .

Adicionalmente, há também muitos resultados relacionados ao parâmetro número de envoltória em ambas as convexidades para grafos direcionados e hipergrafos como [41, 66, 68, 86, 92, 98, 101, 116].

Para uma leitura mais abrangente sobre convexidades, nós referimos o leitor para o livro de Van De Vel [114] sobre convexidades abstratas e para o livro de Pelayo [99] sobre convexidades em grafos, mais especificamente sobre a convexidade geodésica.

Doravante, neste capítulo, sempre que não houver ambiguidade, sejam n o número de vértices, m o número de arestas, Δ o grau máximo, que é o máximo número de vizinhos de um vértice, e tw a largura em árvore do grafo em questão. Seja o diâmetro de um grafo G o tamanho do maior caminho dentre todos os menores caminhos entre dois vértices de G . Seja a cintura de um grafo G como sendo o tamanho do menor ciclo de G .

Defina ainda grafos livres de $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k$, ou ainda $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k\}$ -livres, como sendo o conjunto de grafos que, para todo $1 \leq i \leq k$, não possuem nenhum subgrafo induzido no conjunto de grafos \mathcal{C}_i . Dizemos que um grafo é $(q, q - 4)$ se qualquer subconjunto de no máximo q vértices induz no máximo $q - 4$ P_4 's distintos. Esses grafos foram definidos por Babel em [11]. Seja $q(G)$ o menor q para o qual G é um grafo $(q, q - 4)$.

Sobre o tempo máximo de infecção na convexidade geodésica:

- em 2013, Araújo, Sampaio e Szwarcfiter [8], dentre outros resultados, provaram que TEMPO MÁXIMO DE INFECÇÃO é **NP**-completo para $k \geq 5$ fixo;
- em 2016, Benevides *et al.* [21] mostraram que o problema TEMPO MÁXIMO DE INFECÇÃO é polinomial quando $k = 1$, porém mostraram que o problema é **NP**-completo para qualquer $k \geq 2$ fixo mesmo em grafos bipartidos, melhorando o resultado em [8]. Eles também apresentaram um algoritmo que computa o tempo máximo de infecção de grafos distância-hereditários que executa em tempo $O(n^3m)$, onde grafos distância-hereditários são grafos nos quais as distâncias entre vértices de qualquer subgrafo induzido conexo são as mesmas do grafo original.

Sobre o tempo máximo de infecção na convexidade P_3 :

- em 2012, Przykucki [103] determinou o exato valor do tempo máximo de infecção de um hipercubo com n dimensões que é $\lfloor n^2/3 \rfloor$;

- em 2013, Benevides e Przykucki [19] determinaram o valor do maior tempo de infecção em grades $n \times n$ considerando-se apenas os conjuntos de envoltória minimais. Para $n \geq 4$ esse valor é $(5n^2 - 2n)/8$;
- em 2015, Benevides e Przykucki [20] determinaram uma equação recursiva, que pode ser computada em tempo polinomial, que determina o tempo máximo de infecção em grades $[n]^2$. Adicionalmente, eles determinaram que o tempo máximo de infecção em grades $[n]^2$ é $13n^2/18 + O(n)$;
- também em 2015, Benevides *et al.* [18] provaram que TEMPO MÁXIMO DE INFECÇÃO é **NP**-completo para $k \geq 4$ fixo, para $k \geq 7$ fixo em grafos bipartidos e para grafos planares. Adicionalmente, eles provaram que o problema é polinomial para árvores, grafos cordais e quando $k = 2$.

Existem muito mais trabalhos que estudam o número de envoltória do que trabalhos que tratam sobre o tempo máximo de infecção, o que pode ser explicado pelo tempo que eles são estudados. Sobre o número de envoltória na convexidade geodésica, temos:

- em 2009, Dourado *et al.* [53] provaram que NÚMERO DE ENVOLTÓRIA é **NP**-completo, porém apresentaram algoritmos polinomiais para os casos em que o grafo de entrada é um cografo, um grafo *split* e um grafo de intervalo unitário;
- em 2010, Dourado *et al.* [54] provaram vários limites superiores para o número de envoltória de um grafo. Para G sendo um grafo conexo livre de K_3 com diâmetro d , eles provaram que, se G tem grau mínimo pelo menos 3, $hn(G) \leq (n - d + 3)/3$ e, se G é cúbico, $hn(G) \leq 2(n - d + 5)/7$. Também provaram que $hn(G) \leq 2 + (n - d - 1)/\lceil \frac{g-1}{2} \rceil$, onde g é a cintura de G , para qualquer grafo G conexo, com cintura pelo menos cinco e grau mínimo pelo menos dois;
- em 2013, Araújo *et al.* [6] melhoraram o resultado de Dourado *et al.* [53] e provaram que NÚMERO DE ENVOLTÓRIA é **NP**-completo mesmo para grafos bipartidos. Também mostraram que o problema é polinomial em cactos e complementos de grafos bipartidos e tratável por parâmetro fixo quando parametrizado por $q(G)$. Eles também provaram que, se s é o número de vértices de grau um e $q = \max(1, s)$, $hn(G) \leq q + \lceil 3(n - q)/5 \rceil$ para grafos gerais, $hn(G) \leq q + \lceil (n - q)/2 \rceil$ para grafos regulares ou livres de K_3 , e $hn(G) \leq q + \lceil (n - q)/3 \rceil$ para grafos com cintura pelo menos 6;
- em 2015, Coelho, Dourado e Sampaio [46] provaram, dentre outros resultados, que o NÚMERO DE ENVOLTÓRIA é **APX**-difícil;

- em 2016, Kanté e Nourine [78] apresentaram, dentre outros resultados, um algoritmo que computa o número de envoltória em grafos cordais em tempo $O(n^3)$ e em grafos distância-hereditários em tempo $O(n + m)$;
- no mesmo ano, foi apresentado por Araújo *et al.* [7], um algoritmo polinomial que resolve NÚMERO DE ENVOLTÓRIA em grafos $\{P_5, K_3\}$ -livres e um algoritmo tratável por parâmetro fixo para o problema parametrizado pela diversidade de vizinhança, do inglês *neighborhood diversity*, parâmetro introduzido em [83];
- ainda em 2016, Albenque e Knauer [4] provaram que o problema NÚMERO DE ENVOLTÓRIA é **NP**-completo mesmo em cubos parciais, onde um cubo parcial é um subgrafo de um hipercubo que preserva as distâncias do hipercubo original e um hipercubo de dimensão d é o produto cartesiano de d P_2 s;
- também em 2016, Dourado, Penso e Rautenbach [57] provaram que o problema NÚMERO DE ENVOLTÓRIA é **NP**-completo em grafos livres de P_9 , porém polinomial em grafos $\{K_{1,3}, P_5\}$ -livres e em grafos livres de C_3 tais que cada seis vértices induzem no máximo um P_5 . Adicionalmente, eles provaram que o problema é polinomial em grafos $\{C_3, C_4, \dots, C_{k-2}, P_k\}$ -livres para todo $k \geq 5$ inteiro, estendendo o resultado de Araújo *et al.* [7] a respeito de grafos $\{P_5, K_3\}$ -livres.

Sobre o número de envoltória na convexidade P_3 , temos:

- em 2009, foi provado por Chen [42] que o NÚMERO DE ENVOLTÓRIA é **NP**-completo mesmo em grafos bipartido com $\Delta \geq 8$ fixo. Ele também provou que o problema é difícil de aproximar numa razão de $O(2^{\log^{1-\varepsilon} n})$, para qualquer $\varepsilon > 0$, a não ser que $\mathbf{NP} \subseteq \mathbf{DTIME}(n^{\text{polylog}(n)})$;
- em 2011, Centeno *et al.* [35] mostraram, independentemente de Chen em [42], que o problema NÚMERO DE ENVOLTÓRIA é **NP**-completo. Eles ainda apresentaram algoritmos polinomiais para árvores e grafos cordais;
- em 2011, Ben-Zwi *et al.* [22], dentre outros resultados, apresentaram um algoritmo tratável por parâmetro fixo para o problema NÚMERO DE ENVOLTÓRIA parametrizado na largura em árvore do grafo de entrada que executa em $O(3^{tw+1}tw^{tw+2}n)$;
- em 2013, Centeno *et al.* [36] caracterizam os grafos que satisfazem $hn_{P_3}(G) = in_{P_3}(G)$, onde $in_{P_3}(G)$ é o tamanho do menor conjunto S tal que $I[S] = V(G)$. Eles também construíram uma classe de grafos que satisfazem a igualdade e apresentaram um algoritmo polinomial que reconhece os subgrafos livres de K_3 que satisfazem a igualdade;

- também em 2013, Nichterlein *et al.* [94] mostraram que NÚMERO DE ENVOLTÓRIA é $\mathbf{W}[2]$ -difícil quando parametrizado pela entrada k mesmo em grafos bipartidos de diâmetro quatro. Eles também apresentaram algoritmos tratáveis por parâmetro fixo para um problema mais geral, o problema TARGET SET SELECTION, nos parâmetros número de cobertura de vértices e número de conjunto de arestas de retroalimentação (do inglês *feedback edge set number*), dentre outros;
- em 2015, Coelho, Dourado e Sampaio [46] mostraram, dentre outros resultados, que NÚMERO DE ENVOLTÓRIA é \mathbf{APX} -difícil mesmo em grafos bipartidos;
- também 2015, Campos *et al.* [32] apresentaram um algoritmo tratável por parâmetro fixo para o problema NÚMERO DE ENVOLTÓRIA parametrizado por $q(G)$;
- ainda em 2015, Penso *et al.* [100], dentre outros resultados, mostraram que NÚMERO DE ENVOLTÓRIA é \mathbf{NP} -completo mesmo em grafos planares com $\Delta = 3$. Eles também apresentaram algoritmos polinomiais para o problema quando o grafo é cúbico, ou o número de conjunto de vértices de retroalimentação (do inglês *feedback vertex set number*) do grafo é limitado, ou ainda o grau mínimo do grafo é pelo menos n/c , onde c é um constante.

Em [18], Benevides *et al.* deixaram algumas questões em aberto. Parte delas estão relacionadas ao problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 quando fixamos o tempo k . Eles deixaram em aberto o problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 quando $k = 3$ e, para grafos bipartidos, quando o tempo k é fixo em algum valor em $\{3, 4, 5, 6\}$. Nesta tese, nós preenchemos essas lacunas. Nós apresentamos um algoritmo que determina se $t_{P_3}(G) \geq 3$ para qualquer G em tempo $O(mn^5)$. Quando G é bipartido, nós apresentamos um algoritmo que determina se $t_{P_3}(G) \geq 3$ em tempo $O(mn^3)$ e se $t_{P_3}(G) \geq 4$ em tempo $O(m^2n^9)$. Nós também provamos que o problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 em grafos bipartidos para qualquer $k \geq 5$ fixo é \mathbf{NP} -completo.

O conjunto dessas contribuições foi submetido e aceito no congresso *40th International Workshop on Graph-Theoretic Concepts in Computer Science* (WG-2014) em 2014 cujos anais foram publicados no *Lecture Notes in Computer Science* [87]. Também nós o submetemos para o periódico *Theoretical Computer Science*.

A outra questão deixada em aberto em [18] foi sobre o problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 em grafos planares bipartidos. Nós também respondemos essa questão. Nós provamos que o problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE

P_3 é **NP**-completo mesmo em grafos grade, que são subgrafos induzidos de grades, de grau máximo três. Nós também apresentamos um algoritmo polinomial que determina $t(G)$ quando G é um grafo grade sólido de grau máximo três. Grafos grade sólidos são grafos grade sem “buracos” ou seja, são grafos grade tais que, se colocarmos os seus vértices no plano cartesiano de forma que cada aresta tenha tamanho unitário, a área de cada uma das suas faces internas é um.

Adicionalmente, provamos que TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 é **NP**-completo quando $k = n - n^\varepsilon + 4$ para qualquer $0 < \varepsilon \leq 1$ fixo e em grafos com $\Delta \geq 4$ fixo e $k = \Theta(\log n)$. Por outro lado, apresentamos um algoritmo polinomial que decide se $t_{P_3}(G) \geq n - c$ para qualquer constante c em tempo $O(m \cdot n^{c+1})$.

Na área de complexidade parametrizada, nós apresentamos um algoritmo tratável por parâmetro fixo que decide o problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 parametrizado por $\Delta + k$ em $O(2^{\Delta^k} \cdot k \cdot \Delta \cdot n^2)$ para qualquer $\Delta \geq 4$ fixo e em $O(4^k \cdot n)$ para $\Delta = 3$ e mostramos que esse problema não pode ser decidido em tempo $O(2^{\Delta^{\varepsilon k}} \cdot p(k, \Delta, n))$ para qualquer polinômio p e constante $\varepsilon < 1$ a não ser que a ETH seja falsa. Também apresentamos um algoritmo que computa $t_{P_3}(G)$ em $O(50^{tw} \cdot tw^3 \cdot n^{tw+2})$, o que mostra que o problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 parametrizado por tw é **XP**. Tal algoritmo pode facilmente ser modificado para decidir se $t_{P_3}(G) \geq k$ em tempo $O((50k)^{tw+1} \cdot tw^3 \cdot n)$, o que prova que o problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 parametrizado por $tw + k$ é **FPT**. Por outro lado, mostramos que TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 parametrizado por tw é **W[1]**-difícil e, a não ser que a ETH seja falsa, mostramos que TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 não pode ser decidido em tempo $f(tw) \cdot n^{o(tw)}$ para qualquer função computável f o que mostra que nosso algoritmo é assintoticamente ótimo no que diz respeito ao expoente de n , pois ele decide o problema em $O(f(tw) \cdot n^{tw+2})$.

O conjunto desses resultados foi submetido e aceito no congresso *41st International Workshop on Graph-Theoretic Concepts in Computer Science* (WG-2015) em 2015 cujos anais foram publicados no *Lecture Notes in Computer Science* [88]. Também o submetemos para o periódico *Discrete Applied Mathematics*.

Em [7], Araújo *et al.* propuseram a seguinte questão: existe um algoritmo FPT que decide se $hn_g(G) \leq k$ onde o parâmetro é k ? Em resposta a essa pergunta, nós provamos que NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE GEODÉSICA é **W[1]**-difícil quando parametrizado por $tw + k$ e quando parametrizado por k mesmo em grafos com diâmetro dois. Como subproduto dessas provas, nós mostramos que, a não ser que a ETH seja falsa, para qualquer função computável f , NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE GEODÉSICA

não pode ser decidido em tempo $f(k) \cdot n^{o(k)}$ mesmo em grafos de diâmetro dois (o que mostra que o algoritmo que enumera todos os subconjuntos de tamanho k e os testa para saber se são conjuntos de envoltória é ótimo, pois executa em tempo $O(m \cdot n^{k+1})$) e nem decidido em tempo $f(tw, k) \cdot n^{o(tw) + o(\sqrt{k})}$. Adicionalmente, apresentamos um algoritmo que computa $hn_g(G)$ em tempo $O((tw + 1)^{tw+6} \cdot n^{2tw+5})$ que, de acordo com o nosso resultado anterior, é assintoticamente ótimo no que diz respeito ao expoente de n , pois decide NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE GEODÉSICA em tempo $O(f(tw) \cdot n^{2tw+o(k)})$.

Mostramos ainda que o problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE P_3 parametrizado no tamanho da solução k também é $W[1]$ -difícil mesmo em grafos livres de K_3 de grau máximo seis. Como subproduto dessa prova, nós mostramos que, a não ser que a ETH seja falsa, para qualquer função computável f , o NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE P_3 não pode ser decidido em tempo $f(k) \cdot n^{o(k)}$, mesmo em grafos livres de K_3 de grau máximo 6, o que mostra, assim como no caso desse problema na convexidade geodésica, que o algoritmo que enumera todos os subconjunto de tamanho k e os testa para saber se são conjuntos de envoltória é assintoticamente ótimo no que diz respeito ao expoente de n , pois executa em tempo $O(m \cdot n^{k+1})$. Em 2011, um algoritmo FPT para o problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE P_3 parametrizado por tw já foi apresentado por Ben-Zwi *et al.* [22].

Essas contribuições ainda estão em fase de elaboração e constituirão um trabalho conjunto entre Júlio Araújo, Mamadou Kanté, Thiago Marcilon e Rudini Sampaio.

1.3 Estrutura da tese

Para que o leitor tenha uma leitura mais suave, nós omitimos os detalhes técnicos das provas nos próximos capítulos, apresentando apenas as ideias principais das demonstrações. As demonstrações completas podem ser encontradas nos apêndices.

No Capítulo 2, nós introduzimos os conceitos relevantes e notações para esta tese: grafos, largura em árvore de grafos, complexidade parametrizada e convexidade. Apresentamos também as notações aqui usadas para conjuntos e funções.

No Capítulo 3, nós apresentamos nossos resultados algorítmicos para o problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 . Apresentamos algoritmos polinomiais para tempo $k = n - c$, onde c é uma constante, para tempo $k = 3$ em grafos gerais, tempo $k = 3, 4$ em grafos bipartidos e em grafos grade sólidos de grau máximo três. Adicionalmente, apresentamos algoritmos FPT quando o problema é parametrizado nos parâmetros $tw + k$ e $\Delta + k$. Também apresentamos uma fórmula monádica de segunda ordem para o problema

cujo tamanho depende apenas de k .

O Capítulo 4 contém nossos resultados de **NP**-completude e **W[1]**-dificuldade para o problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 . Neste capítulo, apresentamos a sua **NP**-completude para grafos bipartidos e $k \geq 5$ fixo, para grafos grade de grau máximo três, para $k = n - n^\varepsilon + 4$, onde $0 < \varepsilon \leq 1$ é fixo, e para grafos de grau máximo fixo pelo menos quatro e $k = \Theta(\log n)$. Também provamos que ele é **W[1]**-difícil no parâmetro tw ;

No Capítulo 5, estão os nossos resultados parametrizados para o problema NÚMERO DE ENVOLTÓRIA nas convexidades P_3 e geodésica. Nós mostramos que ele é **W[1]**-difícil quando o parâmetro é o tempo k na convexidade P_3 em grafos livres de K_3 e com grau máximo seis e na convexidade geodésica em grafos com diâmetro dois. Adicionalmente, nós provamos que ele é **W[1]**-difícil quando o parâmetro é $tw + k$ na convexidade geodésica. Apresentamos também um algoritmo XP para o problema na convexidade geodésica parametrizado em $tw + k$.

Finalmente, no Capítulo 6, nós fazemos algumas considerações finais, indicamos questões em aberto e, por fim, apontamos possíveis direções para trabalhos futuros.

Após as referências bibliográficas, apresentamos os apêndices que, como dito anteriormente, contêm as provas completas dos resultados apresentados nos capítulos.

O Apêndice A contém as provas completas dos resultados em [87]. Tais resultados estão relacionados ao problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 e são sua **NP**-completude para grafos bipartidos e $k \geq 5$ fixo e os algoritmos polinomiais para $k = 3$ fixo em grafos gerais e $k = 3, 4$ fixo em grafos bipartidos.

No Apêndice B, apresentamos as provas completas dos resultados em [88]. Os resultados apresentados nesse apêndice também estão relacionados ao problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 . Eles são a sua **NP**-completude para grafos grade de grau máximo três, para $k = n - n^\varepsilon + 4$, onde $0 < \varepsilon \leq 1$ é fixo, e para grafos de grau máximo fixo pelo menos quatro e $k = \Theta(\log n)$; a sua **W[1]**-dificuldade no parâmetro tw ; os algoritmos polinomiais para grafos grade sólidos de grau máximo três, para $k = n - c$, onde c é constante; e os algoritmos FPT nos parâmetros $\Delta + k$ e $tw + k$.

Finalmente, no Apêndice C, as provas completas dos resultados contidos no capítulo 5 são apresentadas.

2 PRELIMINARES

Neste capítulo, nós apresentamos as definições e notações utilizadas nos capítulos subsequentes. Também introduzimos brevemente alguns assuntos pertinentes a esta tese. Não é o nosso objetivo esgotar os assuntos expostos neste capítulo, mas oferecer uma base para que o leitor possa prosseguir a leitura dos próximos capítulos sem maiores dificuldades.

2.1 Definições e notações básicas

Nesta seção, nós apresentamos as definições básicas de grafos e notações de conjuntos, funções e operadores lógicos que serão utilizadas nos próximos capítulos.

2.1.1 Conjuntos, funções e operadores lógicos

Seja C um conjunto qualquer. Denote por $\binom{C}{k}$ o conjunto de todos os subconjuntos de C com k elementos. Seja 2^C o conjunto das partes de C . Uma k -upla é uma lista ordenada com k elementos. Seja $C_1 \times C_2 \times \dots \times C_k$ o conjunto de todas as k -uplas onde o i -ésimo elemento da k -upla pertence a C_i . Seja ainda C^k o conjunto de todas as k -uplas formadas apenas com elementos de C . Denote por $|C|$ a cardinalidade do conjunto C , ou seja, o número de elementos em C .

Sejam \cup e \cap os operadores usuais de união e interseção entre dois conjuntos. Sejam $\bigcup_{1 \leq i \leq k} C_i = C_1 \cup C_2 \cup \dots \cup C_k$ e $\bigcap_{1 \leq i \leq k} C_i = C_1 \cap C_2 \cap \dots \cap C_k$. Seja ainda $C_1 \setminus C_2$ o conjunto de elementos de C_1 que não pertencem a C_2 .

Represente por $C_1 \subseteq C_2$ o fato de C_1 estar contido em C_2 e por $C_1 \supseteq C_2$ o fato de C_1 conter C_2 . De forma similar, denote por $C_1 \subset C_2$ o fato de C_1 estar contido em C_2 e ser diferente de C_2 e $C_1 \supset C_2$ o fato de C_1 conter C_2 e ser diferente de C_2 . Adicionalmente, denote por $C_1 \not\subseteq C_2$ o fato de C_1 não estar contido em C_2 e por $C_1 \not\supseteq C_2$ o fato de C_1 não conter C_2 . Denote por $v \in C$ o fato de v pertencer a C e por $v \notin C$ o fato de v não pertencer a C .

Se C é um conjunto de números, sejam $\min C$ o menor valor em C e $\max C$ o maior valor em C . Sejam \emptyset o conjunto vazio, $[n] = \{1, 2, \dots, n\}$ e $[n]_0 = \{0, 1, 2, \dots, n\}$. Seja ainda $\{v \in C \mid P(v)\}$ o conjunto de todos os elementos v em C que satisfazem uma determinada proposição P .

Sejam \wedge , \vee e \neg os operadores lógicos proposicionais e, ou e não respectivamente. Sejam também \forall e \exists os operadores lógicos de primeira ordem para todo e existe.

Seja $f : C_1 \rightarrow C_2$ uma função dos elementos de C_1 nos elementos de C_2 . Dizemos que C_1 é o *domínio* de f e é denotado por $Dom(f)$ e que C_2 é o *contradomínio* de f e é denotado por $Con(f)$. Adicionalmente, dizemos que o conjunto $\{v \in C_2 \mid \exists w \in C_1 : f(w) = v\}$ é a *imagem* de f e a representamos por $Im(f)$.

Se f é uma função cuja imagem é um conjunto numérico e C um subconjunto do domínio de f , sejam $\min_{v \in C} f(v)$ o menor valor no conjunto $\{x \in Im(f) \mid \exists v \in C : f(v) = x\}$ e $\max_{v \in C} f(v)$ o maior valor no conjunto $\{x \in Im(f) \mid \exists v \in C : f(v) = x\}$.

Dizemos que f é uma função *injetora* se para quaisquer elementos $v, w \in Dom(f)$, $f(v) \neq f(w)$. Dizemos que f é uma função *sobrejetora* se $Con(f) = Im(f)$. Se f é uma função injetora e sobrejetora, então f é *bijetora*. Dizemos que uma função f é uma bijeção de C_1 em C_2 se f é bijetora, $Dom(f) = C_1$ e $Con(f) = C_2$. Se f é uma função bijetora, então a função inversa de f , denotada por f^{-1} , é uma bijeção de $Con(f)$ em $Dom(f)$ que satisfaz $f^{-1}(f(v)) = v$ para todo $v \in Dom(f)$. Uma função f é uma extensão de uma função g se $Dom(g) \subseteq Dom(f)$ e, para todo $x \in Dom(g)$, $g(x) = f(x)$. Dizemos que uma função f é g restrita a C se $Dom(f) = C \subseteq Dom(g)$ e, para todo $x \in C$, $f(x) = g(x)$.

Seja f uma função. Defina $O(f(n))$ como sendo o conjunto de todas as funções g sobre o mesmo domínio de f tais que existem constantes positivas c e n_0 que satisfazem a inequação $0 \leq g(n) \leq c \cdot f(n)$ para todo $n \geq n_0$. Similarmente, defina $\Theta(f(n))$ como sendo o conjunto de todas as funções g tais que existem constantes positivas c_1, c_2 e n_0 que satisfazem a inequação $0 \leq c_1 \cdot f(n) \leq g(n) \leq c_2 \cdot f(n)$ para todo $n \geq n_0$. Defina $o(f(n))$ como sendo o conjunto de todas as funções g tais que, para toda constante positiva c , existe uma constante positiva n_0 que satisfaz a inequação $0 \leq g(n) < c \cdot f(n)$ para todo $n \geq n_0$.

Dizemos que uma função $g(n)$ é $O(f(n))$ se $g(n) \in O(f(n))$. O mesmo vale para as notações Θ e o . Frequentemente, também usamos uma dessas notações onde deveria haver uma função quando queremos apenas informar que ali há uma função que pertence ao conjunto da notação ali posta. Nós fazemos isso quando saber a função exata não for importante naquele momento. Por exemplo, se escrevermos $f(n) = 2^{O(n^2)}$, estamos querendo informar ao leitor que essa função é $f(n) = 2^{g(n)}$ para alguma função $g(n) \in O(n^2)$.

2.1.2 Grafos

As definições desta subseção foram, em sua maioria, retiradas do livro *Introduction to Graph Theory* cujo autor é West [117]. Um *grafo simples* G é um par ordenado (V, E) ,

onde V é um conjunto e $E \subseteq \binom{V}{2}$. Dizemos que V é o *conjunto de vértices* de G e E o *conjunto de arestas* de G . Se temos $G = (V, E)$, defina $V(G) = V$ e $E(G) = E$. É comum representar um grafo através de uma gravura como na Figura 1, a qual ilustra um grafo simples tal que o seu conjunto de vértices é $\{v_1, v_2, v_3, v_4, v_5\}$ e o de arestas é $\{\{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_4\}, \{v_3, v_4\}, \{v_3, v_5\}, \{v_4, v_5\}\}$.

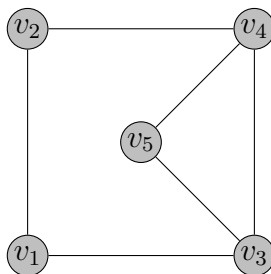


Figura 1 – A representação gráfica de um grafo.

Daqui para frente, todo grafo que definirmos é um grafo simples.

Seja v um vértice de um grafo G . A *vizinhança* de v em G , denotada por $N_G(v)$, é o conjunto de vértices w em $V(G)$ tais que $\{v, w\} \in E(G)$. Se $w \in N_G(v)$, dizemos que w é adjacente a v em G ou que w e v são vizinhos em G . A *vizinhança fechada* de v , denotada por $N_G[v]$, é o conjunto $N_G(v) \cup \{v\}$. O *grau* de um vértice v em um grafo G é a cardinalidade de $N_G(v)$ e denotamos por $d_G(v)$. Sejam ainda o *grau máximo* e *grau mínimo* de um grafo G , denotados por $\Delta(G)$ e $\delta(G)$ respectivamente, o maior e menor grau dentre todos os vértices. Quando o grafo G está implícito, o subscrito nas notações $N_G(v)$ e $d_G(v)$ pode ser omitido. Também podemos omitir o parâmetro G nas notações $\Delta(G)$ e $\delta(G)$. Dois vértices u e v em um grafo G são gêmeos falsos (verdadeiros) se $N_G(v) = N_G(u)$ ($N_G[v] = N_G[u]$).

Um *conjunto independente* ou estável em um grafo G é um subconjunto V' dos vértices de G tal que, para quaisquer dois vértices v e w de V' , $\{v, w\} \notin E(G)$. Uma *clique* em um grafo G é um subconjunto V' dos vértices de G tal que, para quaisquer dois vértices v e w de V' , $\{v, w\} \in E(G)$.

Se v é um vértice de grau 1, dizemos que v é um *vértice pendente*. Se $e = \{v, w\}$ é uma aresta de G , dizemos que v e w são as extremidades de e . Uma *subdivisão de aresta* $e = \{v, w\}$ de $G = (V, E)$ é uma operação que transforma G em um grafo $G' = (V', E')$ de forma que $V' = V \cup \{x\}$, onde x é um novo vértice, e $E' = ((E \setminus \{e\}) \cup \{\{v, x\}, \{x, w\}\})$. Para qualquer $k \geq 1$, a subdivisão k vezes de uma aresta $e = \{v, w\}$ de $G = (V, E)$ é uma operação que transforma G em um grafo $G' = (V', E')$ de forma que $V' = V \cup S$, onde $S = \{s_1, s_2, \dots, s_k\}$ é um conjunto com k novos vértices, e $E' = (E \setminus \{e\}) \cup \{\{v, s_1\}, \{s_1, s_2\}, \{s_2, s_3\}, \dots, \{s_{k-1}, s_k\}, \{s_k, w\}\}$.

Sejam $G = (V, E)$ e $G' = (V', E')$ dois grafos. Dizemos que G' é *subgrafo* de G , ou que G' está contido em G , se $V' \subseteq V$ e $E' \subseteq E$. Dizemos que G' é *subgrafo induzido* de G se $V' \subseteq V$ e $E' = \{\{v, w\} \in E \mid v, w \in V'\}$. Dizemos que $G = G'$ se $V(G) = V(G')$ e $E(G) = E(G')$.

Dizemos ainda que $G = (V, E)$ e $G' = (V', E')$ são *isomorfos* se existe uma bijeção f de V em V' tal que $\{v, w\} \in E$ se e somente se $\{f(v), f(w)\} \in E'$. Uma classe de isomorfismo de grafos é um conjunto de grafos isomorfos dois a dois.

Ao longo desta tese, mostramos várias definições e resultados estruturais para grafos em que o rótulo dos vértices não é importante, apenas as arestas entre eles. Nesses casos, quando não especificarmos explicitamente o conjunto de vértices do grafo, tais definições e resultados valem para qualquer grafo na mesma classe de isomorfismo do grafo em questão.

Sejam $G = (V, E)$ e $G' = (V', E')$ grafos. A operação *produto cartesiano* entre G e G' , denotada por $G \square G'$, produz o grafo $H = (V \times V', F)$, em que dois vértices $\{v, v'\}$ e $\{w, w'\}$ de H são vizinhos se e somente se ou $v = w$ e $\{v', w'\} \in E(G')$ ou $v' = w'$ e $\{v, w\} \in E(G)$. A Figura 2 apresenta um exemplo dessa operação.

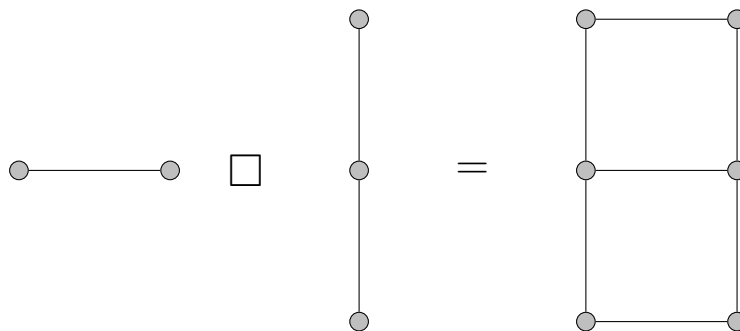


Figura 2 – Operação produto cartesiano entre dois grafos.

Sejam $G = (V, E)$ um grafo e $V' \subseteq V$. O subgrafo de G induzido por V' é o subgrafo induzido G' de G tal que $V(G') = V'$. Denotamos por $G[V']$ o subgrafo de G induzido por V' . O grafo $G' = G - V'$ é o subgrafo induzido de G pelos vértices $V \setminus V'$. Se v é um vértice de G , seja $G - v = G - \{v\}$.

O conjunto dos *grafos caminho de tamanho k* , para $k \geq 0$, denotado por P_{k+1} , consiste no conjunto de grafos $P = (V, E)$ com $k+1$ vértices os quais podem ser colocados em sequência de forma que um vértice aparece apenas uma vez e dois vértices em P são adjacentes se e somente se eles aparecem em posições consecutivas na sequência. Pelo fato de haver só uma classe de isomorfismo em P_k , dizemos que um grafo é um P_k quando queremos dizer que ele está em P_k . Um caminho P em (ou de) G entre os vértices $u, v \in V(G)$ é um subgrafo de G que é um grafo caminho e u e v são as extremidades de P . Dizemos que P é um caminho

induzido em (ou de) G entre os vértices $u, v \in V(G)$ se P é um caminho de G entre u e v e é um subgrafo induzido de G .

Similarmente, o conjunto dos *grafos ciclo de tamanho k* , para $k \geq 3$, denotado por C_k , consiste no conjunto de grafos C com k vértices os quais podem ser colocados em sequência de forma que um vértice aparece apenas uma vez na sequência e dois vértices em C são adjacentes se e somente se eles aparecem em posições consecutivas na sequência ou são o primeiro e último da sequência. Pelo fato de haver só uma classe de isomorfismo em C_k , dizemos que um grafo é um C_k quando queremos dizer que ele está em C_k . Um ciclo C em (ou de) G é um subgrafo de G que é um grafo ciclo. Dizemos que C é um ciclo induzido em (ou de) G se C é um ciclo de G e é um subgrafo induzido de G . Adicionalmente, um ciclo de G é ímpar se ele possui um número ímpar de vértices, caso contrário, ele é par.

Se P é um caminho, então as extremidades de P são os seus vértices que não possuem grau 2 e $Ext(P)$ denota o conjunto das extremidades de P . Todos os outros vértices são vértices internos de P e o seu conjunto é denotado por $Int(P)$.

Dizemos que um grafo G é *conexo* se existe um caminho em G entre quaisquer par de vértices. Caso contrário, dizemos que G é desconexo. A *distância* entre dois vértices u e v de um grafo G , denotada por $d_G(u, v)$, é o tamanho do caminho de menor tamanho entre eles. Quando o grafo G está implícito, podemos omitir o subscrito da notação $d_G(u, v)$. Defina o *diâmetro* de um grafo G como o tamanho do maior caminho dentre todos os menores caminhos entre dois vértices de G . Uma *componente conexa* de um grafo G é um subgrafo maximal conexo de G , ou seja, um subgrafo conexo G' de G que não está contido em nenhum outro subgrafo conexo de G . Assim, temos que um grafo G é conexo se e somente se ele possui apenas uma componente conexa, que é ele próprio.

Sejam $N_{G,k}(v)$ o conjunto de vértices que estão à distância k de v , $N_{G,\geq k}(v)$ o conjunto de vértices que estão à distância pelo menos k de v e $N_{G,\leq k}(v)$ o conjunto de vértices que estão à distância no máximo k de v . De forma similar, defina $N_{G,>k}(v)$ e $N_{G,<k}(v)$. Novamente, quando o grafo G está implícito, podemos omitir o subscrito G das notações $N_{G,k}(v)$, $N_{G,\geq k}(v)$ e $N_{G,\leq k}(v)$. Note que $N_{G,1}(v) = N_G(v)$.

Sejam $G = (V, E)$ um grafo conexo, $V' \subseteq V$ e $a, b \in V \setminus V'$. Temos que V' é um (a, b) -*separador* de G se a e b pertencem a componentes conexas distintas do grafo $G - V'$. Nesse caso, dizemos que V' separa a de b em G . Dizemos ainda que V' é um *separador* de G se $G - V'$ é desconexo. Dizemos também que $v \in V$ é um *vértice de corte*, ou *articulação*, ou ainda um *separador*, se o conjunto $\{v\}$ é um separador.

Classes de Grafos

Dizemos que um grafo é um *caminho* (respectivamente *ciclo*), se existe algum n tal que ele seja um P_n (C_n).

Dizemos que um grafo G é *bipartido* se os vértices de G podem ser particionados em dois conjunto independentes X e Y . Uma caracterização importante de grafos bipartidos é a seguinte: um grafo é bipartido se e somente se ele não possui ciclos ímpares. Seja G um grafo bipartido com partições X e Y , onde $n = |X|$ e $m = |Y|$. Dizemos que G é um $K_{n,m}$ se e somente se o seu conjunto de arestas é o $\{\{u, v\} \mid u \in X \wedge v \in Y\}$.

Uma curva é uma imagem de um mapeamento contínuo de $[0, 1]$ para \mathbb{R}^2 . Uma u, v -curva poligonal é uma curva composta por um número finito de segmentos de linha que começa no ponto u e termina no ponto v . O desenho de um grafo G é uma função f injetora em $V(G) \cup E(G)$ que associa cada vértice a um ponto no plano e associa cada aresta $\{u, v\}$ a uma $f(u), f(v)$ -curva poligonal. Note que um grafo possui infinitos desenhos. Para quaisquer duas arestas e e e' de G , um ponto em $f(e) \cap f(e')$ que não é uma extremidade de ambas as curvas poligonais é chamado de cruzamento. Dizemos que um grafo G é *planar* se G possui um desenho sem cruzamentos. As faces de um grafo planar em um desenho são as regiões maximais do plano que não possuem pontos de nenhuma curva poligonal usada no desenho do grafo.

O conjunto de *grafos completos* de tamanho n , para $n \geq 1$, denotado por K_n , consiste no conjunto de grafos com n vértices onde cada vértice é adjacente a todos os outros do grafo. Assim, dizemos que um grafo é *completo* se ele é um K_n para algum $n \geq 1$.

Um grafo é uma *árvore* se ele é conexo e não possui ciclos, em outras palavras, se G é conexo e acíclico. Um grafo é uma *floresta* se ele é acíclico. Os vértices de um grafo também podem ser chamados de nós, porém só usamos o termo nó para nos referirmos a um vértice de uma árvore. Os vértices pendentos de uma árvore são chamados de folhas e todos os outros vértices são os seus vértices internos. A classe das árvores está contida na interseção da classe dos grafos bipartidos com a dos grafos planares.

Uma *árvore enraizada* é uma árvore com um nó especial denominado *raiz*. A profundidade de um nó de uma árvore enraizada é a distância entre esse nó e a raiz. O *nível* k de uma árvore enraizada é o conjunto de vértices de profundidade k . A *altura* (ou *profundidade*) de uma árvore enraizada T , denotada por $altura(T)$, é a maior profundidade dentre todos os nós da árvore.

Os *antepassados*, ou ancestrais, de um nó em uma árvore enraizada são todos os vértices no caminho entre ele e a raiz da árvore, incluindo a raiz e o próprio vértice. Se um

nó x é um antepassado de um nó y , então o nó y é um *descendente* de x . O *pai* de um nó em uma árvore enraizada é o seu antepassado mais próximo, ou seja, o seu antepassado que também é seu vizinho. Se um nó x é pai de um nó y , então o nó y é *filho* de x . Se dois nós em uma árvore enraizada têm o mesmo pai, dizemos que eles são *irmãos*. Note que a raiz é o único nó em uma árvore enraizada que não possui pai e as *folhas* são os únicos nós que não possuem filhos. Os nós *internos* são todos os nós que não são folhas. A *subárvore* enraizada em um nó v de uma árvore enraizada G é o subgrafo induzido de G pelo conjunto de vértices formado por v e todos os seus descendentes.

Dizemos que uma árvore é uma *árvore binária* se ela é uma árvore enraizada e todo nó é pai de no máximo dois filhos. As definições de árvore binária cheia, estrita, completa e perfeita são apresentadas de formas diferentes na literatura. Definimos então uma *árvore estritamente binária* como sendo uma árvore binária cujos todos os nós internos possuem dois filhos. Uma árvore é uma *árvore binária completa* se todos os seus nós internos possuem dois filhos e todas as folhas estão no mesmo nível na árvore.

Note que toda árvore binária completa é uma árvore estritamente binária, mas o contrário nem sempre é verdade. Na Figura 3, temos um exemplo de árvore estritamente binária que não é completa à esquerda e uma árvore binária completa à direita.

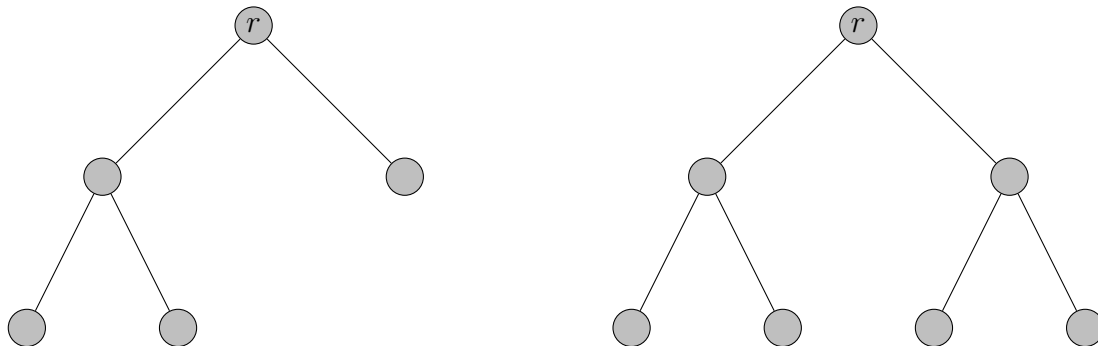


Figura 3 – Árvore estritamente binária à esquerda e binária completa à direita sendo os nós r as suas raízes.

Da mesma forma, definimos uma *árvore k -ária* como sendo uma árvore enraizada onde cada nó possui no máximo k filhos. Analogamente à *árvore binária completa* e *árvore estritamente binária*, definimos *árvore k -ária completa* e *árvore estritamente k -ária*. Note então que uma árvore binária nada mais é do que uma árvore 2-ária.

Uma classe de grafos importante nesta tese é a classe das grades. Uma *grade* $n \times m$, classe denotada por $G_{n,m}$, é o conjunto dos grafos resultantes do produto cartesiano entre um P_n e um P_m . Um vértice é uma *extremidade* de uma grade se ele possui apenas dois vizinhos

na grade. Se um vértice não é uma extremidade, ele é um *vértice interno* da grade. Na Figura 2, temos uma grade $G_{2,3}$ como resultado do produto cartesiano entre um P_2 e um P_3 .

Um grafo é uma *escada* L_n de tamanho n , para algum $n \geq 1$, se ele é um $G_{2,n}$. A classe das grades consiste no conjunto de grades $n \times m$ para todo $n, m \geq 1$.

Uma *grade parcial* é um subgrafo de uma grade. Um *grafo grade* é um subgrafo induzido de uma grade. Perceba que a classe dos grafos grade parciais também está contida na interseção dos grafos bipartidos com os grafos planares. Um grafo grade é *sólido* se ele não possui “buracos”, ou seja, se colocarmos seus vértices em um plano cartesiano de forma que suas arestas tenham tamanho unitário, as suas faces internas terão área unitária. Se G é uma grade parcial, a operação de multiplicação de sua escala por um inteiro k consiste em subdividir todas as suas arestas $k - 1$ vezes.

Para algum $k > 0$, sejam $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k$ classes de grafos. A classe de grafos *livres de* $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k$, ou ainda $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k\}$ -livres, é o conjunto de todos os grafos que não possuem nenhum subgrafo induzido no conjunto $\mathcal{C}_1 \cup \mathcal{C}_2 \cup \dots \cup \mathcal{C}_k$.

2.2 Largura em Árvore

Nesta seção, nós apresentamos os conceitos de largura em árvore, decomposição em árvore e algumas das suas propriedades.

O parâmetro largura em árvore foi introduzido em 1973 por Bertelè e Brioschi [23], ele foi redescoberto por Halin [71] em 1976, porém ele só foi apresentado com esse nome em 1984 por Robertson e Seymour [107] e ele mede o quão parecido um grafo é de uma árvore.

Desde a sua criação, o parâmetro largura em árvore vem sendo amplamente usado na derivação de resultados estruturais [108, 109, 112] e computacionais [10, 27, 48, 95, 115]. Vários problemas **NP**-completos em grafos possuem algoritmos FPT ou XP no problema parametrizado pela largura em árvore do grafo de entrada como os problemas COBERTURA DE VÉRTICES [44] e 3-COLORAÇÃO [10].

Antes de definir formalmente o conceito de largura em árvore, precisamos definir o conceito de decomposição em árvore. Uma *decomposição em árvore* de um grafo G é uma dupla $D = (\mathcal{T}, \mathcal{B})$ tal que $\mathcal{T} = (T, F)$ é uma árvore em que T é o seu conjunto de nós e F é o seu conjunto de arestas e $\mathcal{B} = \{B_j : j \in T\}$ é uma família de subconjuntos B_j de $V(G)$, chamadas de sacolas, indexados pelos nós de T sobre os quais as seguintes condições se aplicam:

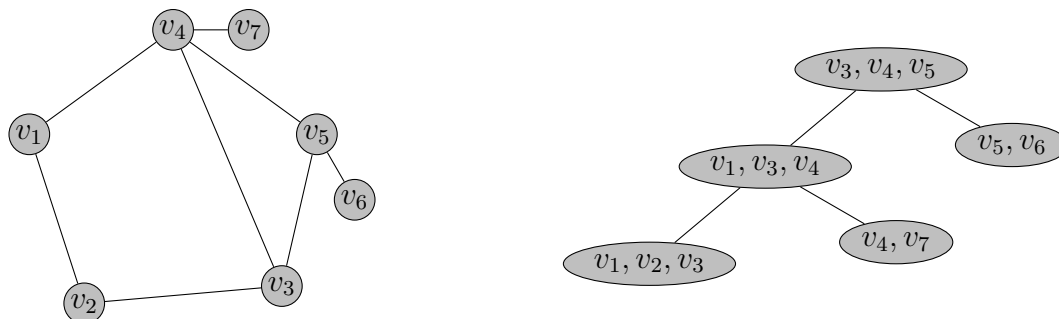


Figura 4 – Um grafo G à esquerda e uma decomposição em árvore de G à direita.

1. $\bigcup_{j \in T} B_j = V(G)$;
2. para todo $\{u, v\} \in E(G)$, existe um $j \in T$ tal que $v, u \in B_j$ e
3. para todo $v \in V(G)$, o subgrafo de \mathcal{T} induzido pelos conjunto de nós $\{j \in T \mid v \in B_j\}$ é uma árvore.

A Figura 4 exemplifica o conceito de decomposição em árvore. A largura de uma decomposição em árvore D é $w(D) = \max_{j \in T} |B_j| - 1$. A *largura em árvore* de um grafo G é $tw(G) = \min_D w(D)$, onde D itera sobre todas as decomposições em árvore de G .

Um parâmetro conceitualmente bastante similar à largura em árvore é a *largura em caminho* [106], denotada por $pw(G)$. Calculamos a largura em caminho da mesma forma que a largura em árvore, $pw(G) = \min_D w(D)$, exceto que $D = (\mathcal{P}, \mathcal{B})$ itera sobre todas as decomposições em caminho, as quais nada mais são do que decomposições iguais as decomposições em árvore exceto pelo fato de que \mathcal{P} deve ser um caminho em vez de uma árvore. Naturalmente, toda decomposição em caminho é também uma decomposição em árvore e, assim, temos que $tw(G) \leq pw(G)$ para todo grafo G .

Uma fato importante sobre as sacolas de uma decomposição em árvores é que elas são separadores do grafo [81]. Assim, se B_j é uma sacola de uma decomposição em árvore $(\mathcal{T}, \mathcal{B})$ de G e C_1^j, C_2^j, \dots são as componentes conexas da floresta $\mathcal{T} - j$, então B_j separa os vértices em $\bigcup_{k \in V(C_i^k)} B_k \setminus B_j$ dos vértices em $\bigcup_{k \in V(C_{i'}^k)} B_k \setminus B_j$ para quaisquer $i \neq i'$. Outro fato importante é que $m \leq w \cdot n - \binom{w+1}{2}$, onde $m = |E(G)|, n = |V(G)|$ e $w = tw(G)$.

Dados um grafo G e inteiro k , o problema de determinar se $tw(G) \leq k$ é **NP-completo** [9]. No entanto, se considerarmos um k fixo, existem algoritmos lineares que determinam se $tw(G) \leq k$ e, caso a largura em árvore de G seja no máximo k , fornecem uma decomposição em árvore de G de largura k [26].

Existe um tipo especial de decomposição em árvore que facilita o seu uso em algoritmos. Uma decomposição em árvore $D = (\mathcal{T}, \mathcal{B})$ é boa se \mathcal{T} é uma árvore enraizada em um nó r e cada nó j de \mathcal{T} se enquadra em um dos quatro seguintes tipos:

1. nó folha, se j é uma folha de \mathcal{T} ;
2. nó introduz, se j possui exatamente um filho j' , $B_{j'} \subset B_j$, e $|B_j| = |B_{j'}| + 1$;
3. nó esquece, se j possui exatamente um filho j' , $B_{j'} \supset B_j$, e $|B_j| = |B_{j'}| - 1$;
4. nó junção, se j possui exatamente dois filhos j_1 e j_2 , e $B_j = B_{j_1} = B_{j_2}$.

Sabe-se que um grafo G possui uma decomposição em árvore de largura k se e somente se G possui uma boa decomposição em árvore de largura k , e, dada uma decomposição em árvore com largura k de um grafo G , onde $n = |V(G)|$, podemos achar uma boa decomposição em árvore $(\mathcal{T}, \mathcal{B})$ de G de largura k tal que \mathcal{T} tem $O(k \cdot n)$ nós em tempo $O(k^2 \cdot n)$ [81].

2.3 Teoria da Complexidade Parametrizada

Nesta seção, nós introduzimos a teoria da complexidade parametrizada e as suas classes de problemas que utilizamos nesta tese. Também apresentamos algumas conjecturas sobre essas classes e suas consequências. Introduzimos ainda a Hipótese do Tempo Exponencial e o seu impacto sobre as classes apresentadas. Sobre a teoria da complexidade parametrizada, nós referimos o leitor aos livros de Downey e Fellows [62], Flum e Grohe [70] e Cygan *et al.* [51].

A teoria da complexidade parametrizada foi introduzida por Downey e Fellows na sequência de artigos [1, 58–61]. Segundo Flum e Grohe [70], um problema parametrizado é definido como sendo um par (P, k) , onde P é um problema de decisão, e k é uma função nas instâncias de P que retorna um inteiro não-negativo. Na verdade, um problema pode ser parametrizado por mais de um parâmetro. Nesse caso, um problema parametrizado na combinação de p parâmetros seria uma $p + 1$ -upla (P, k_1, \dots, k_p) , onde k_1, \dots, k_p são funções nas instâncias de P que retornam um inteiro não-negativo. Assim, quando nos referimos ao problema P parametrizado por $k_1 + k_2 + \dots + k_p$ ou ainda ao problema $P(k_1, \dots, k_p)$, estamos nos referindo ao problema parametrizado (P, k_1, \dots, k_p) . Doravante, nesta seção, as definições serão feitas usando apenas um parâmetro, porém elas podem ser facilmente estendidas para dois ou mais parâmetros.

Adicionalmente, se um problema de decisão P possui instâncias do tipo (I, k) , onde k é um inteiro, quando nos referimos ao problema P parametrizado no parâmetro natural, estamos nos referindo ao problema parametrizado (P, κ) , onde $\kappa((I, k)) = k$.

Dizemos que um algoritmo é *tratável por parâmetro fixo* (FPT) em relação a um parâmetro $k = k(I)$ sobre suas instâncias I se seu tempo de execução é $f(k(I)) \cdot |I|^c$, onde c é uma constante e f é uma função computável e $|I|$ é o tamanho da instância [62]. De forma semelhante, dizemos que um algoritmo é XP em relação a um parâmetro $k = k(I)$ sobre suas instâncias I se seu tempo de execução é $f(k(I)) \cdot |I|^{g(k(I))}$, onde f e g são funções computáveis que só dependem de k e $|I|$ é o tamanho da instância.

Assim, dizemos que um problema parametrizado (P, k) é **FPT** se existe um algoritmo FPT no parâmetro k que decide P . A classe dos problemas tratáveis por parâmetro fixo (**FPT**) consiste no conjunto de problemas parametrizados **FPT**. Similarmente, dizemos que um problema parametrizado (P, k) é **XP** se existir um algoritmo XP no parâmetro k que decide P . Assim, a classe dos problemas **XP** consiste no conjunto de problemas parametrizados **XP**.

Vários problemas **NP**-completos possuem versões parametrizadas que estão em **FPT** ou **XP**, como o problema COBERTURA DE VÉRTICES parametrizado no tamanho da solução k , o qual pode ser decidido em tempo $O(kn + 1.2738^k)$ [44], e o problema CLIQUE parametrizado no tamanho da solução k , o qual pode ser decidido em tempo $O(k^2 n^k)$, onde n é o número de vértices do grafo em ambos os casos. Assim, apesar desses problemas provavelmente não possuírem um algoritmo polinomial, ainda assim eles podem ser decididos de forma rápida quando os parâmetros das instâncias não são grandes.

Uma redução parametrizada de um problema parametrizado (P_1, k_1) para outro (P_2, k_2) é uma dupla (f, g) tal que, para cada instância I de P_1 , (a) $f(I)$ é uma instância SIM de P_2 se e somente se I é uma instância SIM de P_1 , (b) f é computável por um algoritmo FPT com relação ao parâmetro k_1 e (c) g é uma função computável tal que $k_2(f(I)) \leq g(k_1(I))$ para toda instância I de P_1 .

Assim, temos que, se existe uma redução parametrizada de um problema (P_1, k_1) para outro (P_2, k_2) e (P_2, k_2) é **FPT**, então (P_1, k_1) também é **FPT**. Além disso, se (P_2, k_2) é **XP**, então (P_1, k_1) também é **XP**.

Para definirmos a hierarquia de problemas parametrizados W , precisamos definir alguns conceitos antes. O problema SATISFABILIDADE DE CIRCUITOS BOOLEANOS consiste em, dado um circuito booleano e um inteiro k , determinar se existe uma atribuição de valores 0 ou 1 aos seus nós de entrada de forma que k nós de entrada recebam o valor 1 e o circuito seja satisfeito, ou seja, o nó de saída tenha valor 1. O entrelaçamento de um circuito booleano é o máximo número de vértices rotulados com operadores lógicos com grau de entrada pelo menos três em um caminho entre um nó de entrada e o nó de saída. A Figura 5 apresenta um exemplo de circuito booleano com entrelaçamento 2.

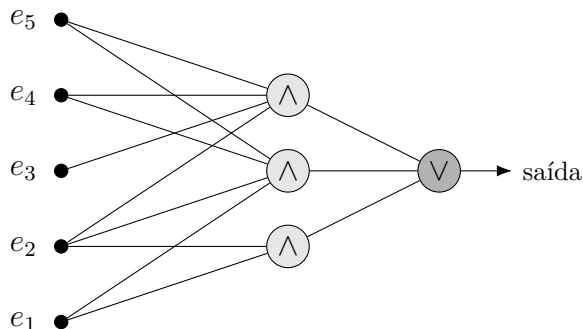


Figura 5 – Exemplo de circuito booleano com entrelaçamento 2 e entradas e_1, \dots, e_5 .

A classe $\mathbf{W}[t]$, para todo natural $t \geq 1$, consiste no conjunto de problemas parametrizados que possuem uma redução parametrizada para o problema SATISFABILIDADE DE CIRCUITOS BOOLEANOS restritos a circuitos com entrelaçamento t parametrizado pelo número de entradas k que recebem o valor 1.

A classe $\mathbf{W}[t]$ -difícil consiste no conjunto de problemas para os quais existe uma redução parametrizada a partir de todos os problemas em $\mathbf{W}[t]$. A classe de problemas $\mathbf{W}[t]$ -completo é a interseção das classes $\mathbf{W}[t]$ e $\mathbf{W}[t]$ -difícil.

Dois exemplos de problemas $\mathbf{W}[1]$ -difíceis são CLIQUE e CLIQUE MULTICOLORIDA parametrizados no tamanho da solução k e REVESTIMENTO DE GRADE parametrizado pela entrada k . Dois exemplos de problemas $\mathbf{W}[2]$ -difíceis são CONJUNTO DOMINANTE e COBERTURA DE CONJUNTOS parametrizados pelo tamanho da solução k . As provas de dificuldade acima mencionadas podem ser encontradas em [51].

Temos que $\mathbf{FPT} \subseteq \mathbf{W}[1] \subseteq \mathbf{W}[2] \subseteq \dots \subseteq \mathbf{XP}$ e que $\mathbf{FPT} \subset \mathbf{XP}$ [70]. No entanto, não se sabe se $\mathbf{FPT} = \mathbf{W}[1]$ ou não. Apesar disso, a crença de que $\mathbf{FPT} \neq \mathbf{W}[1]$ é tão difundida que provar que um problema é $\mathbf{W}[1]$ -difícil é considerado um forte indicativo de que o mesmo não é \mathbf{FPT} . De fato, temos que $\mathbf{FPT} \neq \mathbf{W}[1] \implies \mathbf{P} \neq \mathbf{NP}$.

2.4 Hipótese do Tempo Exponencial

Em 2001, Impagliazzo e Paturi [75] introduziram a Hipótese do Tempo Exponencial, do inglês *Exponential Time Hypothesis* (ETH). A ETH declara que existe uma constante $\delta_3 > 0$ tal que, para qualquer polinômio $p(\cdot)$ e constante $c < \delta_3$, o problema 3-SAT não pode ser decidido em tempo $O(2^{cN} p(N))$, onde N é o número de variáveis na instância de 3-SAT. Ainda em 2001, Impagliazzo, Paturi e Zane [76] mostraram, através do Lema da Esparsificação, que a ETH implica que existe uma constante $\delta_3 > 0$ tal que, para qualquer

polinômio $p(\cdot)$ e constante $c < \delta_3$, o problema 3-SAT não pode ser decidido em tempo $O(2^{c(N+M)}p(N, M))$, onde N é o número de variáveis e M o de cláusulas na instância de 3-SAT.

Assim, se assumirmos a ETH, podemos tirar várias conclusões acerca de limites inferiores para os tempos de algoritmos que decidem problemas **NP**-difíceis, **W[1]**-difíceis e até mesmo **FPT**. Nesse caso, os limites seguem diretamente das provas de **NP**-dificuldade e **W[1]**-dificuldade dos problemas. De fato, temos que $\text{ETH} \implies \text{FPT} \neq \text{W[1]} \implies \text{P} \neq \text{NP}$.

Algumas consequências da ETH para problemas clássicos são as de que os problemas COBERTURA DE VÉRTICES, CONJUNTO DOMINANTE, 3-COLORAÇÃO e CICLO HAMILTONIANO não podem ser decididos em tempo $2^{o(n+m)} \cdot p(n, m)$ [51] para qualquer função polinomial $p(\cdot)$, onde n é o número de vértices e m o número de arestas do grafo de entrada. Tais consequências são derivadas direto da prova de **NP**-completude dos problemas.

Algumas consequências da ETH a respeito da teoria da complexidade parametrizada são que, para qualquer função polinomial $p(\cdot)$, os problemas COBERTURA DE VÉRTICES e CONJUNTO DE VÉRTICES DE RETROALIMENTAÇÃO não podem ser decididos em tempo $2^{o(k)} \cdot p(n)$ [51], onde n é o número de vértices. Adicionalmente, para qualquer função computável f , os problemas CLIQUE [43] e CLIQUE MULTICOLORIDA [85] não podem ser decididos em tempo $f(k) \cdot n^{o(k)}$, onde n é o número de vértices do grafo de entrada, e o problema REVESTIMENTO DE GRADE [89] não pode ser decidido em tempo $f(k) \cdot n^{o(k)}$, onde n é uma das entradas do problema.

2.5 Convexidade

Um *espaço de convexidade finito* [72, 84, 114] é um par (V, \mathcal{C}) que consiste em um conjunto finito V e um conjunto \mathcal{C} de subconjuntos de V satisfazendo as seguintes condições: $\emptyset, V \in \mathcal{C}$ e, se $C_1, C_2 \in \mathcal{C}$, então $C_1 \cap C_2 \in \mathcal{C}$. Se (V, \mathcal{C}) é um espaço de convexidade finito, podemos dizer que \mathcal{C} é uma *convexidade finita* em V .

Seja \mathcal{C} uma convexidade em V . Os membros de \mathcal{C} são chamados *conjuntos \mathcal{C} -convexos*, ou, quando a convexidade estiver implícita, conjuntos *convexos*, e a *envoltória*, ou fecho, de um conjunto S é o conjunto convexo mínimo $H(S) \in \mathcal{C}$ que contém S . Um *conjunto de envoltória* é um conjunto cuja envoltória é V . Um subconjunto S de V é chamado de *coconvexo* se $V \setminus S$ é um conjunto convexo. Assim, se S é um conjunto coconvexo não vazio, temos que todo conjunto de envoltória deve conter pelo menos um elemento de S .

Por exemplo, o conjunto $\mathcal{C} = \{\emptyset, \{2\}, \{5\}, \{1, 2\}, \{2, 3\}, \{1, 2, 3, 4, 5\}\}$ é uma convexi-

dade em $V = \{1, 2, 3, 4, 5\}$. Os conjuntos $\{1, 2\}$ e $\{5\}$ são convexos, mas os conjuntos $\{1\}$ e $\{1, 2, 3, 4\}$ não são. Temos ainda que $H(\{1\}) = \{1, 2\}$ e, portanto, $\{1\}$ não é um conjunto de envoltória. No entanto, como $H(\{4\}) = \{1, 2, 3, 4, 5\}$, temos que $\{4\}$ é um conjunto de envoltória.

Uma convexidade \mathcal{C} em V é uma *convexidade de intervalo* [30], se existe uma função $I : \binom{V}{2} \rightarrow 2^V$, chamada *função de intervalo*, tal que um subconjunto C de V pertence a \mathcal{C} se e somente se $I(\{x, y\}) \subseteq C$ para todo par de elementos distintos x e y de C . Abusando da notação, vamos definir $I[S] = S \cup \bigcup_{x, y \in S} I(\{x, y\})$. Em uma convexidade de intervalo, a envoltória de um conjunto S pode ser computada aplicando-se exaustivamente a função de intervalo correspondente até obtermos um conjunto convexo. Defina ainda $I^0[S] = S$, $I^k[S] = I[I^{k-1}[S]]$ e seja $I_h[S] = I^\infty[S]$ a envoltória $H(S)$ de S .

As convexidades de intervalo em grafos mais estudadas são aquelas em que V é o conjunto de vértices do grafo e $I(\{x, y\})$ é um conjunto de caminhos entre x e y que satisfazem alguma propriedade particular. Por exemplo, temos a *convexidade P_3* [35, 66], a convexidade P_3^* [8], a *convexidade geodésica* [34, 68] e a *convexidade monofônica* [64, 68].

Na convexidade monofônica, $I(\{x, y\})$ é o conjunto de todos os vértices que estão em um caminho induzido entre x e y . Na convexidade P_3^* , $I(\{x, y\})$ é o conjunto de todos os vértices que estão em um caminho P_3 induzido entre x e y . Na convexidade geodésica, $I(\{x, y\})$ é o conjunto de todos os vértices que estão em um caminho mínimo entre x e y . Na convexidade P_3 , $I(\{x, y\})$ é o conjunto de todos os vértices que estão em um caminho P_3 entre x e y , ou seja, todos os vértices adjacentes a ambos os vértices x e y . Nesta tese, focamos em problemas nas convexidades P_3 e geodésica.

Vários parâmetros em grafos relacionados à convexidade têm sido explorados na literatura. O número de Radon [56, 104], número de Carathéodory [17, 33, 45], número de envoltória [36, 67], o número de convexidade [39, 40], o número de intervalo [55, 73] e tempo máximo de infecção [19–21] são alguns dos parâmetros de convexidade estudados. O número de envoltória e o tempo máximo de infecção são parâmetros particularmente relevantes para esta tese.

O *número de envoltória* de um grafo na convexidade \mathcal{C} , denotado por $hn_{\mathcal{C}}(G)$, é a cardinalidade do conjunto de envoltória de menor cardinalidade daquela convexidade.

No grafo da Figura 6, um conjunto de envoltória mínimo na convexidade P_3 é $\{v_1, v_3, v_5\}$, enquanto um conjunto de envoltória mínimo na convexidade geodésica é $\{v_1, v_4\}$. Por isso, temos que, sendo G o grafo da Figura 6, $hn_g(G) = 2$ e $hn_{P_3}(G) = 3$.

Seja \mathcal{C} uma convexidade de intervalo com função de intervalo $I[\cdot]$. Podemos ver as

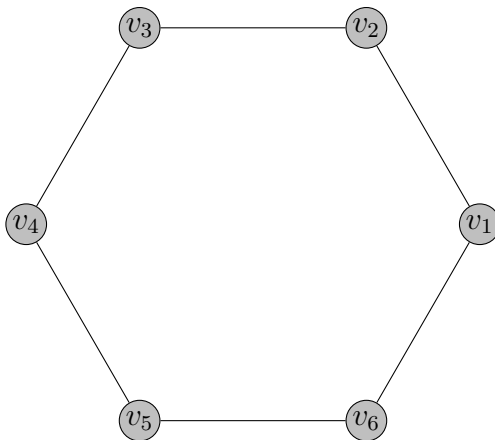


Figura 6 – Um grafo G tal que $hn_g(G) = 2$ e $hn_{P_3}(G) = 3$.

sucessivas aplicações de $I[\cdot]$ em um conjunto S para formar a envoltória de S como um processo de infecção, onde o conjunto inicial dos vértices infectados é S , um vértice infectado nunca fica saudável novamente e um vértice v é infectado em tempo k se e somente se $v \in I^k[S] \setminus I^{k-1}[S]$. Portanto, daqui para frente, quando dissermos que v é infectado em tempo k por S , queremos dizer que $v \in I^k[S] \setminus I^{k-1}[S]$. Dizemos também que um grafo G é infectado por S em tempo k se e somente se S é um conjunto de envoltória tal que $I^{k-1}[S] \subset I^k[S] = V(G)$. O processo de formação da envoltória de um conjunto na convexidade P_3 também tem outros nomes como *2-neighbor bootstrap percolation* [37] ou *irreversible 2-threshold process* [63].

Se S é um conjunto de envoltória em uma convexidade \mathcal{C} , denote por $t_{\mathcal{C}}(G, S)$ o tempo em que S infecta G . Se S é um conjunto, não obrigatoriamente de envoltória, denote por $t_{\mathcal{C}}(G, S, v)$ o tempo em que S infecta v , se S infecta v , ou ∞ caso contrário. Quando a convexidade ou o grafo estiverem implícitos, tanto o subscrito quando o grafo G podem ser omitidos das notações $t_{\mathcal{C}}(G, S)$ e $t_{\mathcal{C}}(G, S, v)$.

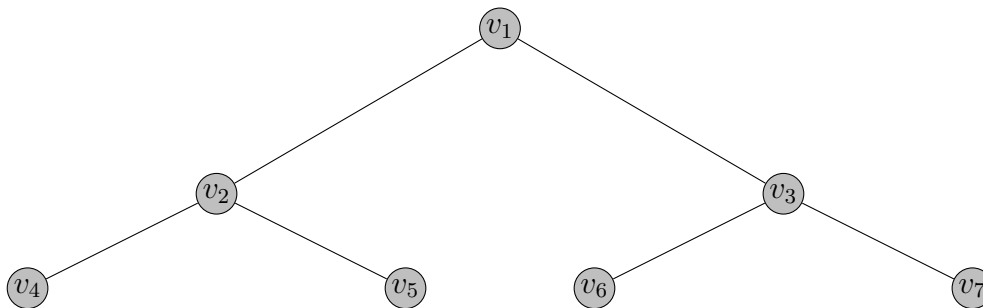


Figura 7 – Um grafo G tal que $t_g(G) = 1$ e $t_{P_3}(G) = 2$.

Da relação entre a formação de envoltórias e um processo de infecção é que vem o nome do parâmetro *tempo máximo de infecção* de um grafo na convexidade de intervalo

\mathcal{C} , denotado por $t_{\mathcal{C}}(G)$. Ele consiste no maior valor k para o qual existe um conjunto de envoltória S tal que $t_{\mathcal{C}}(G, S) = k$. Quando a convexidade estiver implícita, o subscrito pode ser omitido da notação $t_{\mathcal{C}}(G)$.

Por exemplo, no grafo da Figura 7, o conjunto de envoltória que infecta G no maior tempo, tanto na convexidade P_3 quanto na convexidade geodésica, é o conjunto $\{v_4, v_5, v_6, v_7\}$. No entanto, temos que $t_{P_3}(G, \{v_4, v_5, v_6, v_7\}) = 2$, enquanto $t_g(G, \{v_4, v_5, v_6, v_7\}) = 1$. Por isso, temos que, sendo G o grafo da Figura 7, $t_g(G) = 1$ e $t_{P_3}(G) = 2$.

3 ALGORITMOS PARA O TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3

Neste capítulo, serão apresentados os nossos resultados algorítmicos para o problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 . O tempo máximo de infecção de um grafo G na convexidade \mathcal{C} , denotado por $t_{\mathcal{C}}(G)$, é o maior inteiro k para o qual existe um conjunto de envoltória que infecta o grafo G em tempo k . Assim, o problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE \mathcal{C} é o problema de decisão definido da seguinte forma:

TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE \mathcal{C}

Entrada: Um grafo G e um inteiro k .

Saída: Sim, se $t_{\mathcal{C}}(G) \geq k$. Não, caso contrário.

Em outras palavras, o problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE \mathcal{C} consiste em determinar, dados um grafo G e um inteiro k , se existe um conjunto de envoltória na convexidade \mathcal{C} que infecta G em tempo pelo menos k .

Como vamos trabalhar apenas a convexidade P_3 neste capítulo, daqui em diante, nós não mencionamos explicitamente a convexidade a qual estamos nos referindo. Também vamos omitir o subscrito das notação P_3 das notações que envolvem o nome da convexidade. Adicionalmente, neste capítulo, nós abreviamos o nome do problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 para TEMPO MÁXIMO DE INFECÇÃO. Além disso, neste capítulo, todos os grafos definidos são conexos, pois, para calcular o tempo máximo de infecção de um grafo, basta calcular o máximo dos tempos máximos de infecção das suas componentes conexas.

Na Seção 3.1, nós apresentamos um algoritmo que decide o problema TEMPO MÁXIMO DE INFECÇÃO em tempo $O(m \cdot n^{c+1})$ para $k = n - c$, onde c é um inteiro constante. Ele resulta diretamente de observações simples quando $k = n - c$.

Na Seção 3.2, nós discutimos brevemente os resultados algorítmicos para k fixo que o seguem. Após isso, apresentamos caracterizações estruturais do tempo máximo de infecção em grafos bipartidos para $k = 3$, em grafos gerais para $k = 3$ e em grafos bipartidos para $k = 4$ nas Seções 3.3, 3.4 e 3.5 respectivamente. Em cada um desses casos, os algoritmos polinomiais são derivados a partir das caracterizações e têm complexidade de tempo $O(m \cdot n^3)$, $O(m \cdot n^5)$

e $O(m^2 \cdot n^9)$ respectivamente. Esses resultados foram motivados pelo trabalho de Benevides *et al.* [18].

Após isso, na Seção 3.6, nós apresentamos uma caracterização do tempo máximo de um grafo de grau máximo três baseada nos tamanhos dos seus caminhos induzidos e, nos baseando nessa caracterização, um algoritmo polinomial que calcula o tempo máximo de infecção em grafos grade sólidos de grau máximo três em tempo $O(n^2)$. Esses resultados, por sua vez, também foram motivados pelo trabalho de Benevides *et al.* [18], e foram descobertos quando buscávamos resultados em grafos planares bipartidos.

As duas últimas seções contêm resultados parametrizados para o problema TEMPO MÁXIMO DE INFECÇÃO. Na Seção 3.7, nós apresentamos um algoritmo FPT para o problema TEMPO MÁXIMO DE INFECÇÃO parametrizado por $\Delta + k$ que executa em tempo $O(2^{\Delta^k} \cdot k \cdot \Delta \cdot n^2)$, onde $\Delta = \Delta(G)$, $n = |V(G)|$ e G é o grafo de entrada.

Finalmente, na Seção 3.8, nós utilizamos o Teorema de Courcelle [48] para mostrar que o problema parametrizado por $tw(G) + k$ é **FPT**. Adicionalmente, nós apresentamos um algoritmo XP no parâmetro $tw(G)$ que computa $t(G)$ em tempo $O(50^w \cdot w^3 \cdot n^{w+2})$ que pode ser facilmente modificado para se tornar um algoritmo FPT no parâmetro $tw(G) + k$ que decide se $t(G) \geq k$ em tempo $O((50k)^{w+1} \cdot w^3 \cdot n)$, onde $w = tw(G)$ e $n = |V(G)|$.

Neste capítulo, nós apresentamos o enunciado de cada lema e teorema e, quando se tratar de um lema ou teorema importante, as ideias centrais da demonstração. As demonstrações completas dos resultados para $k = 3, 4$ em grafos bipartidos e $k = 3$ em grafos gerais se encontram no Apêndice A. As demonstrações completas de todos os outros resultados deste capítulo se encontram no Apêndice B.

3.1 Algoritmo polinomial para $k = n - c$ onde c é fixo

Nesta seção, nós provamos um resultado simples cuja demonstração é bastante curta. Nós mostramos que existe um algoritmo polinomial para o problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 quando $k = n - c$ e c é um inteiro constante. No entanto, antes de apresentá-lo, nós vamos demonstrar um lema que será bastante útil nesse e no próximo capítulo.

Lema 3.1. *Seja um grafo G conexo e um subconjunto de vértices S de G . É possível determinar $I^k[S]$ em tempo $O(k \cdot m)$.*

Demonstração. Observe que $I[S]$ pode ser computado em tempo $O(m)$, pois G é conexo

e, para cada vértice v que não está em S , nós analisamos a sua vizinhança para checar se possui pelo menos dois vizinhos em S . Portanto, aplicando a função de intervalo k vezes, nós obtemos o tempo $O(k \cdot m)$. \square

Esse lema implica que, dados um grafo G , um subconjunto de vértices S de G e um vértice u de G , podemos determinar $t(G, S)$ e $t(G, S, u)$ em tempo $O(m \cdot t(G, S))$ e $O(m \cdot t(G, S, u))$ respectivamente, onde $t(G, S)$ e $t(G, S, u)$ é o tempo em que o conjunto S infecta o grafo G e o vértice u respectivamente.

Teorema 3.2. *Para qualquer constante inteira c , o problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 para $k = n - c$ pode ser decidido em tempo $O(m \cdot n^{c+1})$.*

Demonstração. Qualquer conjunto de envoltória S que infecta G em tempo $n - c$ deve ter tamanho no máximo c . Isso acontece porque deve haver pelo menos um vértice infectado em cada tempo $1, 2, \dots, n - c$.

Como existem $O(n^c)$ subconjuntos de vértices de tamanho c e levamos tempo $O(n \cdot m)$ para saber se um dado conjunto infecta o grafo inteiro e em que tempo o faz, nós podemos decidir se $t(G) \geq n - c$ em tempo $O(m \cdot n^{c+1})$. \square

3.2 Discussão sobre os resultados polinomiais para k fixo

Em [18], Benevides *et al.* apresentou um algoritmo polinomial para o TEMPO MÁXIMO DE INFECÇÃO quando $k = 2$ e mostrou que o problema era **NP**-completo para $k \geq 4$ fixo em grafos gerais e para $k \geq 7$ fixo em grafos bipartidos. Nas próximas seções, nós mostramos algoritmos polinomiais para resolver o problema TEMPO MÁXIMO DE INFECÇÃO para $k = 3$ em grafos quaisquer e para $k = 3, 4$ em grafos bipartidos.

Em cada um desses casos, os algoritmos são uma consequência direta de caracterizações estruturais que podem ser computadas em tempo polinomial. As três caracterizações são similares: um grafo G tem tempo máximo de infecção pelo menos k se e somente se existe um vértice $u \in V(G)$, um conjunto de vértices F , onde $|F| = O(1)$, e um conjunto de vértices T_0 , o qual deve satisfazer algumas restrições específicas em cada caso, tais que o conjunto $F \cup T_0 \cup N_{\geq k}(u)$ infecta algum vértice v em tempo k . Em cada uma das demonstrações, como nós apresentamos um algoritmo que, dado o conjunto $F \cup T_0 \cup N_{\geq k}(u)$ que infecta algum vértice v em tempo k , fornece um conjunto de envoltória que infecta o grafo em tempo pelo menos k , uma consequência da prova das caracterizações estruturais é que podemos, na

verdade, computar um conjunto de envoltória que infecta o grafo em tempo pelo menos k se ele existir.

Como $|F| = O(1)$, a dificuldade principal de mostrar que o algoritmo oriundo de tais caracterizações é polinomial reside em mostrar que o número de conjuntos T_0 que precisamos testar é polinomial no tamanho da entrada. Apesar disso, a complexidade de tempo dos algoritmos polinomiais derivados também dependem diretamente do tamanho de F . Nos três casos, a prova de que a condição é suficiente não leva em conta o tamanho de F diretamente, porém a prova de que a condição é necessária se torna bem mais difícil à medida que tentamos diminuir o tamanho de F . Para $|F| = 2^k$, a prova de que a condição é necessária é bastante simples, porém é possível provar que F pode ter tamanho menor em todos os três casos.

Doravante, a instância do problema TEMPO MÁXIMO DE INFECÇÃO que trabalhamos é a instância (G, k) , onde $n = |V(G)|$ e $m = |E(G)|$. Os seguintes lemas são úteis e implicitamente utilizados nas próximas seções.

Lema 3.3. *Sejam G um grafo conexo e S um conjunto de envoltória. Para todo $0 \leq t \leq t(G, S)$, existe um conjunto de envoltória S' tal que $t(G, S', u) = \max(t(G, S, u) - t, 0)$ para todo vértice $u \in V(G)$.*

Com Lema 3.3, podemos concluir que existe um conjunto de envoltória que infecta o grafo em tempo t para todo $0 \leq t \leq t(G)$.

Lema 3.4. *Sejam S um conjunto de vértices de um grafo G e w um vértice de G . Se $t(G, S, w) \geq t$, então $t(G, S, w) \geq t(G, S \cup \{z\}, w) \geq t$ para qualquer $z \in N_{\geq t}(w)$.*

Uma consequência importante e bastante usada do lema anterior é que, se $t(G, S, w) \geq t$, então $t(G, S, w) \geq t(G, S \cup N_{\geq t}(w), w) \geq t$. Isso implica que, se um conjunto S infecta um vértice v em tempo t , então $S \cup N_{\geq t}(v)$ também infecta v em tempo t .

Lema 3.5. *Seja G um grafo e sejam S e S' conjuntos tais que $S \subseteq S' \subseteq V(G)$. Então $t(G, S, u) \geq t(G, S', u)$ para qualquer vértice u .*

3.3 Algoritmo polinomial para $k = 3$ em grafos bipartidos

Nesta seção, nós mostramos um algoritmo que decide TEMPO MÁXIMO DE INFECÇÃO para $k = 3$ em grafos bipartidos em tempo $O(m \cdot n^3)$. Para isso, primeiramente, nós vamos apresentar um importante resultado estrutural.

Lema 3.6. *Sejam G um grafo conexo bipartido e T_0 o conjunto de vértices de grau 1 de G . Então, temos que $t(G) \geq 3$ se e somente se existem vértices u, v e s tais que $v \in N(u)$, $s \in N_2(u)$ e o conjunto $T_0 \cup N_{\geq 3}(u) \cup \{v, s\}$ infecta u em tempo três.*

Esboço da demonstração. Suponha que $t(G) \geq 3$. Então existe um conjunto de envoltória S' e um vértice u tal que S' infecta u em tempo três. Temos que $T_0 \subseteq S'$. Também, temos que $S = S' \cup N_{\geq 3}(u)$ infecta u em tempo três. Se S contém um vértice em $N(u)$, seja v tal vértice. Senão, seja v um vizinho de u infectado no menor tempo por S . Note que todos os vértices em $N(u) \setminus \{v\}$ são infectados em tempo pelo menos dois por S . Como G é bipartido, a distância de v para qualquer outro vértice em $N(u)$ é de pelo menos dois. Assim, temos que todos os vértices em $N(u) \setminus \{v\}$ são infectados em tempo pelo menos dois por $S \cup \{v\}$. Portanto, temos que u é infectado por $S \cup \{v\}$ em tempo três, o que implica que u é infectado por $S' = S \cup \{v\} \cup N_{\geq 3}(u)$ também em tempo três. Note que $T_0 \subseteq S'$ e que nenhum vizinho de u , exceto v , está em S' .

Analisando os vértices em $N(u) \setminus \{v\}$, é possível concluir que existe um vértice $s \in N_2(u)$ tal que, se retirarmos todos os vértices em $N_2(u) \setminus T_0$ de S' , exceto s , o conjunto resultante continua infectando u em tempo três, mesmo que ele não seja mais um conjunto de envoltória. Em outras palavras, temos que existe um vértice $s \in N_2(u)$ tal que o conjunto $T_0 \cup N_{\geq 3}(u) \cup \{v, s\}$ infecta u em tempo três.

Agora, suponha que existem vértices $u, v \in N(u)$ e $s \in N_2(u)$ tais que $S_0 = T_0 \cup N_{\geq 3}(u) \cup \{v, s\}$ infecta u em tempo três. Nós vamos mostrar um algoritmo que, dado S_0 , constrói um conjunto de envoltória S que infecta G em tempo pelo menos três.

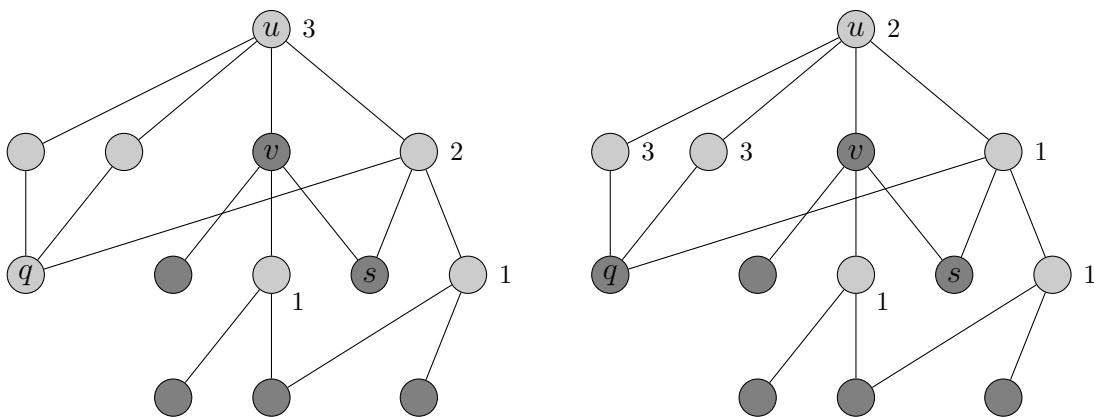


Figura 8 – Tempos de infecção dos conjuntos $T_0 \cup N_{\geq 3}(u) \cup \{v, s\}$ à esquerda, e $T_0 \cup N_{\geq 3}(u) \cup \{v, s, q\}$ à direita.

Inicialmente, faça $S = S_0$. Se S_0 for um conjunto de envoltória, como S_0 infecta u em tempo três, o algoritmo para e S infecta G em tempo pelo menos 3. Senão, em cada passo,

o algoritmo adiciona um vértice a S de forma que, após adicionado, S infecta u em tempo pelo menos dois e passa a infectar um vértice em $N[u]$ em tempo pelo menos três. Seja S_i o conjunto S no i -ésimo passo, para $i \geq 1$. Se S_{i-1} não é um conjunto de envoltória, sejam q qualquer vértice em $N_2(u)$ e $S_i = S_{i-1} \cup \{q\}$. Temos que S_i infecta u em tempo pelo menos dois, já que q está a distância 2 de u e u é infectado por S_{i-1} em tempo pelo menos dois. Também pode-se provar que todos os vértices $w \in N(u) \cap N(q)$ que não são infectados por S_{i-1} são infectados por S_i em tempo pelo menos 3. A Figura 8 ilustra esse argumento.

Eventualmente, nós teremos que S_i é um conjunto de envoltória e, como sempre garantimos que existe um vértice em $N[u]$ que é infectado em tempo pelo menos três por S em cada passo, temos então que $t(G) \geq 3$. \square

Com esse resultado estrutural, podemos derivar um algoritmo para o TEMPO MÁXIMO DE INFECÇÃO para $k = 3$.

Teorema 3.7. TEMPO MÁXIMO DE INFECÇÃO para $k = 3$ pode ser decidido em tempo $O(m \cdot n^3)$ em grafos bipartidos.

Demonstração. Considerando G conexo, primeiro o algoritmo computa o conjunto T_0 em $O(n)$. Então, o algoritmo seleciona, em cada passo, um vértice $u \in V(G)$ e computa os conjuntos $N(u)$, $N_2(u)$ e $N_{\geq 3}(u)$ em $O(m)$. Depois, o algoritmo seleciona um vértice v em $N(u)$ e um vértice s em $N_2(u)$ e, em seguida, verifica se o conjunto $T_0 \cup N_{\geq 3}(u) \cup \{v, s\}$ infecta u em tempo três em $O(m)$. Se existir uma tripla (u, v, s) tal que o conjunto $T_0 \cup N_{\geq 3}(u) \cup \{v, s\}$ infecta u em tempo três, o algoritmo responde SIM, caso contrário, ele responde NÃO. A corretude do algoritmo segue diretamente do Lema 3.6 e a complexidade de tempo do algoritmo é $O(m \cdot n^3)$. \square

3.4 Algoritmo polinomial para $k = 3$ em grafos quaisquer

Nesta seção, nós mostramos um algoritmo que decide TEMPO MÁXIMO DE INFECÇÃO para $k = 3$ em tempo $O(m \cdot n^5)$ para quaisquer grafos.

A definição a seguir é bastante técnica, porém representa um simples fato: se v é um separador e alguma componente conexa de $G - v$ contém apenas vértices adjacentes à v , então qualquer conjunto de envoltória deve conter pelo menos um vértice dessa componente.

Definição 3.8. Para qualquer $u \in V(G)$, seja \mathcal{T}_0^u a família de subconjuntos de $V(G)$ tal que um subconjunto T_0 de $V(G)$ está em \mathcal{T}_0^u se e somente se para todo separador v de G e toda componente conexa $H_{v,i}$ de $G - v$ tais que $u \notin V(H_{v,i})$ e $V(H_{v,i}) \subseteq N(v)$, T_0 contém

exatamente um vértice de $H_{v,i}$ e, para todo vértice $w \in T_0$, existe um separador v e uma componente conexa $H_{v,i}$ de $G - v$ tais que $w \in V(H_{v,i})$, $u \notin V(H_{v,i})$ e $V(H_{v,i}) \subseteq N(v)$.

A Figura 9 ilustra a Definição 3.8. Na Figura 9, os vértices x, y e z são separadores do grafo tais que $t_1 \in V(H_{x,1})$, $t_2 \in V(H_{x,2})$, $t_3 \in V(H_{z,1})$ e $t_4 \in V(H_{y,1})$. Também temos que $u \notin V(H_{x,1}) \subseteq N(x)$, $u \notin V(H_{x,2}) \subseteq N(x)$, $u \notin V(H_{y,1}) \subseteq N(y)$ e $u \notin V(H_{z,1}) \subseteq N(z)$. Portanto, temos que $\{t_1, t_2, t_3, t_4\} \in \mathcal{T}_0^u$.

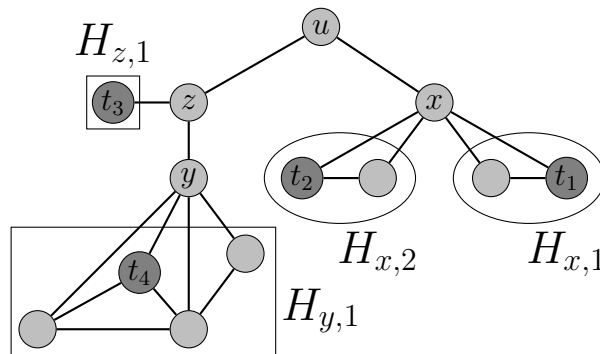


Figura 9 – O conjunto $\{t_1, t_2, t_3, t_4\}$ está em \mathcal{T}_0^u , com $t_1 \in V(H_{x,1})$, $t_2 \in V(H_{x,2})$, $t_3 \in V(H_{z,1})$ e $t_4 \in V(H_{y,1})$.

Antes de apresentar o resultado principal desta seção, novamente, nós vamos apresentar um importante resultado estrutural.

Lema 3.9. *Seja G um grafo conexo. Temos que $t(G) \geq 3$ se e somente se existe um vértice u , um conjunto $T_0 \in \mathcal{T}_0^u$ e um conjunto de vértices F tais que $|F| \leq 4$ e $T_0 \cup N_{\geq 3}(u) \cup F$ infecta u em tempo três.*

Esboço da demonstração. Suponha que $t(G) \geq 3$. Então, existe um conjunto de envoltória S' e um vértice u tal que S' infecta u em tempo três. Como S' é um conjunto de envoltória, pode-se provar que existe um conjunto $T_0 \in \mathcal{T}_0^u$ tal que $T_0 \subseteq S'$. Também, temos que o conjunto $S = S' \cup N_{\geq 3}(u)$ é um conjunto de envoltória que infecta u em tempo três. Como S infecta u em tempo três, então existe um conjunto $F \subseteq S$ tal que $|F| \leq 2^3 = 8$ e F infecta u em tempo três, pois, para infectar cada vértice, necessitamos apenas de dois vizinhos infectados. Portanto, temos que o conjunto $F \cup T_0 \cup N_{\geq 3}(u)$ infecta u em tempo três. Na realidade, com algum esforço, é possível diminuir o tamanho de F na afirmação anterior. Pode-se provar que existe um conjunto F , não obrigatoriamente subconjunto de S , tal que $|F| \leq 4$ e o conjunto $F \cup T_0 \cup N_{\geq 3}(u)$ infecta u em tempo três.

Agora, suponha que existe um vértice u , um conjunto $T_0 \in \mathcal{T}_0^u$ e um conjunto F tal que $|F| \leq 4$ e $T_0 \cup N_{\geq 3}(u) \cup F$ infecta u em tempo três. Nós vamos mostrar um algoritmo

que, dado $S_0 = T_0 \cup N_{\geq 3}(u) \cup F$, constrói um conjunto de envoltória S que infecta G em tempo pelo menos três.

Inicialmente, seja $S = S_0$. Se S_0 for um conjunto de envoltória, como S_0 infecta u em tempo três, o algoritmo para. Senão, seja S_i o conjunto S no i -ésimo passo, para $i \geq 1$. Em cada passo $i \geq 1$, o algoritmo adiciona um vértice a S_{i-1} para obter S_i de forma que S_i infecta algum vértice u_i em tempo pelo menos três, onde $u_0 = u$. Se S_i não é um conjunto de envoltória, seja Y_i o conjunto não-vazio de vértices de G não infectados por S_i .

Primeiro, suponha que existe um vértice $y_i \in Y_{i-1} \cap N_2(u_{i-1})$ que não possui nenhum vizinho infectado por S_{i-1} em tempo pelo menos dois, ou seja, cada vizinho ou está em Y_{i-1} ou é infectado por S_{i-1} em tempo no máximo um. Seja $S'_i = S_{i-1} \cup \{y_i\}$. Temos que u_{i-1} , que é infectado por S_{i-1} em tempo pelo menos três, tem no máximo um vizinho infectado por S_{i-1} em tempo no máximo um e, pela nossa escolha do y_i , u_{i-1} não é adjacente à y_i .

É possível provar que todo vizinho de u_{i-1} infectado por S_{i-1} em tempo pelo menos dois é infectado por S'_i em tempo pelo menos dois. Todo vizinho z de u_{i-1} que não é infectado por S_{i-1} e que é infectado por S'_i é infectado por S'_i em tempo pelo menos dois, pois, caso contrário, ou $z = y_i$, o que é uma contradição porque y_i não é adjacente à u , ou z é adjacente à y_i e a outro vértice em S_{i-1} , o que também é uma contradição, pois z seria vizinho de u_{i-1} e de um vértice em S_{i-1} , porém ele não é infectado por S_{i-1} . Então, temos que S'_i infecta u_{i-1} em tempo pelo menos três e, assim, fazemos $u_i = u_{i-1}$, o que implica que o conjunto $S_i = S'_i \cup N_{\geq 3}(u_i)$ infecta u_i em tempo pelo menos três.

Agora, suponha que todo vértice em $Y_{i-1} \cap N_2(u_{i-1})$ possui exatamente um vizinho infectado por S_{i-1} , o qual é infectado em tempo pelo menos dois. Sejam y_i um vértice em $Y_{i-1} \cap N_2(u_{i-1})$, C_i a componente conexa do subgrafo $G[Y_{i-1}]$ que contém y_i e z_i o único vizinho de y_i que é infectado por S_{i-1} , o qual é infectado em tempo pelo menos dois.

Suponha, por contradição, que todo vértice de C_i é adjacente à z_i . Nesse caso, temos que z_i é um separador e $V(C_i) \subseteq N(z_i)$. Porém, isso implica que há um vértice de C_i em T_0 , o que é um absurdo. Essa é a razão pela qual definimos a família T_0^u da forma que ela é definida, para que existisse um vértice em C_i não adjacente a z_i .

Portanto, temos que existe um vértice y'_i de C_i que não é adjacente à z_i . Seja z'_i o seu vizinho infectado por S_{i-1} . Seja $S_i = S_{i-1} \cup \{y_i\} \cup N_{\geq 3}(y'_i)$. Como todos os vértices de C_i são adjacentes a um vértice infectado por S_{i-1} , não é difícil ver que S_i infecta todos os vértices de C_i . Também, temos que S_i infecta y'_i em tempo pelo menos 3. Isso acontece porque, como y_i está a distância pelo menos dois de z'_i , S_i infecta z'_i e todos os vértices em C_i cujo vizinho em Y_{i-1} é z_i em tempo pelo menos dois. Esse argumento é ilustrado na Figura 10.

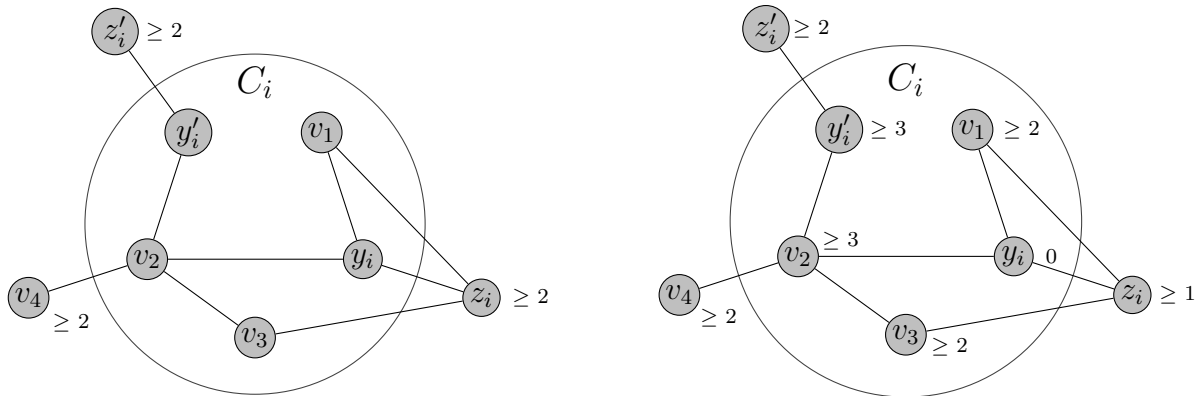


Figura 10 – Vértices da componente C_i antes e depois da união de $\{y_i\}$ à S_{i-1} com seus tempos de infecção.

Adicionalmente, a adição dos vértices em $N_{\geq 3}(y'_i)$ não diminui o tempo de infecção de y'_i para menos de 3. Assim, basta fazer $u_i = y'_i$ para o próximo passo, se houver.

Quando S_i , para algum i , infectar todos os vértices de $N_2(u_i)$, ou seja, quando $Y_i \cap N_2(u_i) = \emptyset$, é possível provar que S_i é um conjunto de envoltória. Adicionalmente, como S_i infecta u_i em tempo pelo menos três, então $t(G) \geq 3$. \square

Antes de apresentar o teorema principal da seção, como a família \mathcal{T}_0^u pode não ter tamanho polinomial, precisamos diminuir o espaço de busca em \mathcal{T}_0^u . Para resolver esse problema, provamos o seguinte lema.

Lema 3.10. *Seja $u \in V(G)$. Se existe um conjunto $T_0 \in \mathcal{T}_0^u$ e um subconjunto de vértices F tais que $|F| \leq 4$ e $T_0 \cup N_{\geq 3}(u) \cup F$ infecta u em tempo três, então, para todo conjunto $T'_0 \in \mathcal{T}_0^u$, existe um subconjunto de vértices F' tais que $|F'| \leq 4$ e $T'_0 \cup N_{\geq 3}(u) \cup F'$ infecta u em tempo três.*

Sem o Lema 3.10, o nosso algoritmo precisaria verificar todos os conjuntos de \mathcal{T}_0^u para cada vértice u no pior caso. Como uma família \mathcal{T}_0^u pode ter tamanho $\Theta(n^n)$, o nosso algoritmo não seria polinomial. No entanto, com o Lema 3.10, é o bastante selecionarmos apenas um conjunto qualquer de \mathcal{T}_0^u . Assim, temos o seguinte teorema.

Teorema 3.11. TEMPO MÁXIMO DE INFECÇÃO para $k = 3$ pode ser decidido em tempo $O(m \cdot n^5)$.

Demonstração. Considerando G conexo, primeiro o algoritmo computa todos os separadores v de G e as componentes conexas de $G - v$, o que pode ser feito em $O(n \cdot m)$. Então, ele seleciona um vértice $u \in V(G)$ e computa um conjunto T_0 qualquer da família \mathcal{T}_0^u . Note

que, pelo Lema 3.10, o algoritmo pode selecionar qualquer conjunto de \mathcal{T}_0^u o que, dado os separadores de G , pode ser feito em $O(n)$. Depois, o algoritmo seleciona um conjunto F de no máximo quatro vértices e verifica se $T_0 \cup N_{\geq 3}(u) \cup F$ infecta u em tempo três em $O(m)$.

Se existir uma tripla (u, T_0, F) tal que o conjunto $T_0 \cup N_{\geq 3}(u) \cup F$ infecta u em tempo três, o algoritmo responde SIM. Caso contrário, responde NÃO. A corretude do algoritmo segue diretamente dos lemas 3.9 e 3.10 e a complexidade de tempo do algoritmo é $O(m \cdot n^5)$. \square

Note que não mostramos que o tamanho de F é apertado, ou seja, é possível que seja necessário menos do que quatro vértices em F na caracterização do lema 3.9. De fato, nós conjecturamos o seguinte.

Conjectura 3.12. *Se $t(G) \geq 3$ então existe um vértice u , um conjunto $T_0 \in \mathcal{T}_0^u$ e um conjunto de vértices F tais que $|F| \leq 3$ e $T_0 \cup N_{\geq 3}(u) \cup F$ infecta u em tempo três.*

Com isso, o Lema 3.10 mudaria de acordo, já que a sua demonstração não depende diretamente do tamanho de F , ela apenas mantém o tamanho de F . Assim, se a Conjectura 3.12 fosse verdadeira, nós teríamos um algoritmo que decide se $t(G) \geq 3$ em tempo $O(m \cdot n^4)$.

3.5 Algoritmo polinomial para $k = 4$ em grafos bipartidos

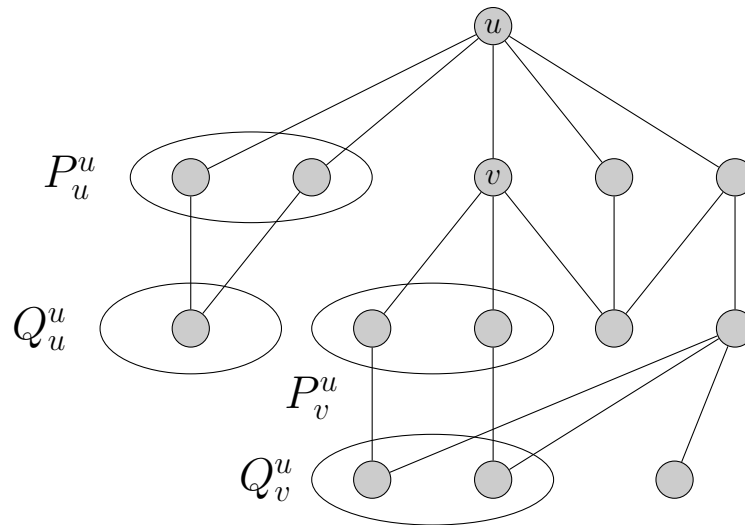
Nesta seção, mostramos um algoritmo que decide TEMPO MÁXIMO DE INFECÇÃO em grafos bipartidos para $k = 4$ em tempo $O(m^2 \cdot n^9)$.

Primeiro, vamos apresentar algumas definições importantes.

Definição 3.13. Sejam u, v vértices de G . Se $v \in N_{\geq 2}(u)$, defina $M_v^u = P_v^u = Q_v^u = \emptyset$. Se $v \in N[u]$, seja M_v^u o conjunto de todos os vértices x, y tais que $x \neq u$, $y \neq u, v$, x, y é um caminho e x e y têm grau dois em G . Adicionalmente, defina $P_v^u = M_v^u \cap N(v)$ e $Q_v^u = M_v^u \setminus P_v^u$.

Note que, se $v \in N(u)$, então $P_v^u = M_v^u \cap N_2(u)$, e $Q_v^u = M_v^u \cap N_2(u)$. Também temos que, para todo $v \in N[u]$, como no máximo dois vértices em P_v^u podem ter o mesmo vizinho em Q_v^u , nós temos que $|P_v^u|/2 \leq |Q_v^u| \leq |P_v^u|$. A Figura 11 apresenta um exemplo de conjuntos definidos na Definição 3.13.

Definição 3.14. Sejam $u \in V(G)$, $v \in N(u)$ e $k_v = |P_v^u|$. Sejam ainda $P_v^u = \{x_{v,1}, \dots, x_{v,k_v}\}$ e $y_{v,i}$ o vizinho de $x_{v,i}$ em Q_v^u , para todo $i \in [k_v]$. Para um $u \in V(G)$ fixo:


 Figura 11 – Conjuntos P_u^u , Q_u^u , P_v^u e Q_v^u .

- Defina T'_0 como sendo o conjunto de todos os vértices em $M_v^u \cap N_2(u)$, para todo $v \in N[u]$, e todos os vértices de grau um;
- Defina T_0^v como sendo o conjunto de todos os vértices em $(M_{v'}^u \setminus M_v^u) \cap N_2(u)$ para todo $v' \in N[u] \setminus \{v\}$, todos os vértices em Q_v^u e todos os vértices de grau um;
- Defina $T_0^{v,i}$ como sendo o conjunto de todos os vértices em $(M_{v'}^u \setminus M_v^u) \cap N_2(u)$ para todo $v' \in N[u] \setminus \{v\}$, todos os vértices em $(Q_v^u \setminus \{y_{v,i}\}) \cup \{x_{v,i}\}$ e todos os vértices de grau um.

Finalmente, seja Γ_0^u a família constituída pelo conjunto T'_0 , para todo $v \in N(u)$, pelo conjunto T_0^v e, para todo $v \in N(u)$ e todo $i \in [k_v]$, pelo conjunto $T_0^{v,i}$.

Note que a família Γ_0^u tem tamanho polinomial. Note também que podem haver inteiros $1 \leq i \neq j \leq k_v$ tais que $y_{v,i} = y_{v,j}$. De fato, temos o seguinte lema.

Lema 3.15. *Dado um vértice u , $|\Gamma_0^u| = O(m)$ e a família Γ_0^u pode ser computada em tempo $O(m)$.*

Note ainda que, para qualquer conjunto de envoltória S , $\{x_{v,i}, y_{v,i}\} \cap S \neq \emptyset$, pois $\{x_{v,i}, y_{v,i}\}$ é um conjunto coconvexo.

Lema 3.16. *Seja G um grafo conexo bipartido. Então $t(G) \geq 4$ se e somente se existe um vértice u , um conjunto $T_0 \in \Gamma_0^u$ e um conjunto de vértices F tais que $|F| \leq 8$ e $T_0 \cup F \cup N_{\geq 4}(u)$ infecta algum vértice em tempo quatro.*

Esboço da demonstração. Suponha que $t(G) \geq 4$. Então existe um conjunto de envoltória Z e um vértice u tais que Z infecta u em tempo quatro. Isso implica que o conjunto $S'' = Z \cup N_{\geq 4}(u)$ também infecta u em tempo quatro. Primeiro, vamos mostrar que existe um conjunto de envoltória $S \supseteq S''$ que infecta algum vértice x em tempo quatro e contém algum conjunto $T_0 \in \Gamma_0^u$. Vamos dividir essa parte da prova em dois casos.

O primeiro caso ocorre quando existe um vértice $v \in N(u)$ tal que S'' infecta v em tempo pelo menos três e ou $|P_v^u| = 1$ e o único vértice x em P_v^u não está em S'' ou $|P_v^u| \geq 2$. Se $|P_v^u| \geq 2$, como u é infectado em tempo maior que um por S'' , então existe algum vértice $x = x_{v,i}$ em P_v^u infectado por S'' em tempo $t(S'', v) + 1$, ou seja, pelo menos em tempo quatro, já que o seu vizinho $y_{v,i}$, nesse caso, precisa estar em S'' . Se $|P_v^u| = 1$, então o único vértice $x = x_{v,i}$ em P_v^u é infectado em tempo $t(S'', v) + 1$, ou seja, pelo menos em tempo quatro por S'' , já que o seu vizinho $y_{v,i}$, nesse caso, precisa estar em S'' .

Pelo Lema 3.3, existe um conjunto de envoltória $S' \supseteq S''$ que infecta um vértice $x \in P_v^u$ em tempo exatamente quatro e o vértice v em tempo exatamente três.

Seja

$$S = S' \cup (N_2(u) \cap \bigcup_{v' \in N[u] - \{v\}} (M_{v'}^u \setminus M_v^u)).$$

É possível mostrar que S é um conjunto de envoltória que infecta v em tempo três e x em tempo quatro. Também temos que ou não existem vértices em $S \cap P_v^u$, o que implica que $Q_v^u \subseteq S$, ou existe exatamente um vértice $x_{v,i}$ em $S \cap P_v^u$, o que implica que $(Q_v^u \setminus \{y_{v,i}\}) \cup \{x_{v,i}\} \subseteq S$. Portanto, pela Definição 3.14, temos que existe um conjunto $T_0 \in \Gamma_0^u$ tal que $T_0 \subseteq S$.

O segundo caso ocorre quando, para todo vértice $v \in N(u)$, ou $P_v^u = \emptyset$, ou S'' infecta v em tempo no máximo dois, ou $|P_v^u| = 1$ e o único vértice x em P_v^u está em S'' . Seja $S = S'' \cup (N_2(u) \cap \bigcup_{v \in N[u]} M_v^u)$ e faça $x = u$. Assim, para cada $w \in S \setminus S''$, ou $w \in P_v^u$, ou $w \in Q_v^u$. Após uma análise desses dois casos, pode-se concluir que quando adicionamos w à S'' para construir S , temos que o conjunto resultante infecta os vizinhos de u , exceto talvez por um deles, em tempo pelo menos três, o que implica que ele infecta u em tempo quatro.

Adicionalmente, temos que, pela Definição 3.14, existe um conjunto $T_0 \in \Gamma_0^u$ tal que $T_0 \subseteq S$, que é o conjunto T_0' .

Assim, como S infecta x em tempo quatro e, para ser infectado, um vértice precisa de dois vizinhos infectados, então existe um conjunto $F \subseteq S$ que infecta x em tempo quatro tal que $|F| \leq 2^4 = 16$. Assim, como $F \subseteq F \cup T_0 \cup N_{\geq 4}(u) \subseteq S$, temos que o conjunto $F \cup T_0 \cup N_{\geq 4}(u)$ infecta x em tempo quatro. Na realidade, com algum esforço, é possível provar que existe um vértice u , um conjunto $T_0 \in \Gamma_0^u$ e um conjunto F , com $|F| \leq 8$, tal que

$F \cup T_0 \cup N_{\geq 4}(u)$ infecta x em tempo quatro.

Agora suponha que existe um vértice u , um conjunto $T_0 \in \Gamma_0^u$ e um conjunto de vértices F , com $|F| \leq 8$, tal que $T_0 \cup F \cup N_{\geq 4}(u)$ infecta algum vértice x em tempo quatro. Nós vamos mostrar um algoritmo que, dado $S_0 = T_0 \cup F \cup N_{\geq 4}(u)$, constrói um conjunto de envoltória que infecta G em tempo pelo menos quatro.

Inicialmente, faça $S = S_0$. Se S_0 for um conjunto de envoltória, como S_0 infecta x em tempo quatro, o algoritmo para. Senão, seja S_i o conjunto S no i -ésimo passo, para $i \geq 1$. Em cada passo $i \geq 1$, o algoritmo adiciona um vértice a S_{i-1} para obter S_i de forma que S_i infecta algum vértice em tempo pelo menos quatro. Seja Y_i o conjunto de vértices não infectados por S_i .

Primeiro, suponha que existe um vértice $y_i \in Y_{i-1} \cap N_2(x)$ que não possui vizinhos infectados por S_{i-1} em tempo pelo menos dois. Faça $S_i = S_{i-1} \cup \{y_i\}$. Temos que x tem no máximo um vizinho infectado por S_{i-1} em tempo zero ou um e, pela nossa escolha do y_i , temos que x não é adjacente a y_i . É possível provar, baseando-se fortemente no fato de G ser bipartido, que para todo vizinho z de x , se $t(S_{i-1}, z) \geq 3$, então $t(S_i, z) \geq 3$. Como x tem no máximo um vizinho infectado em tempo no máximo dois, isso mostra que S_i infecta x em tempo quatro.

Se todos os vértices do conjunto $Y_{i-1} \cap N_2(x)$ possuem um vizinho infectado por S_{i-1} em tempo pelo menos dois, suponha que exista um vértice $y_i \in Y_i \cap N_3(x)$ que não possui vizinhos infectados por S_{i-1} em tempo pelo menos dois. Faça $S_i = S_{i-1} \cup \{y_i\}$. Como todos os vértices do conjunto $Y_{i-1} \cap N_2(x)$ possuem pelo menos um vizinho infectado por S_{i-1} em tempo pelo menos dois, é possível provar que, todo vértice $v \in N(x)$ tal que $t(S_{i-1}, v) \geq 3$ é tal que, para todo vizinho z de v , se $t(S_{i-1}, z) \geq 2$, então $t(S_i, z) \geq 2$. Com isso, para todo vértice $v \in N(x)$ tal que $t(S_{i-1}, v) \geq 3$, nós temos que $t(S_i, v) \geq 3$. Isso implica que $t(S_i, x) = 4$.

Quando todos os vértices dos conjuntos $Y_{i-1} \cap N_2(x)$ e $Y_{i-1} \cap N_3(x)$ tiverem um vizinho infectado em tempo pelo menos dois por S_{i-1} , seja C_i qualquer componente conexa de $G[Y_{i-1}]$. Assim, temos que cada vértice de C_i tem exatamente um vizinho fora de C_i e esse vizinho é infectado em tempo pelo menos dois por S_{i-1} . A componente C_i tem pelo menos três vértices porque, caso contrário, haveria um vértice de C_i em T_0 , o que é uma contradição, pois todos os vértices de C_i estão em Y_{i-1} . Portanto, como o grafo é bipartido, existem dois vértices y_i e y'_i a distância dois um do outro.

Temos que o conjunto $S_{i-1} \cup \{y_i\}$ infecta todos os vértices de C_i . Adicionalmente, é possível provar que o conjunto $S_{i-1} \cup \{y_i\}$ infecta todos os vizinhos de y_i em C_i em tempo

pelo menos três e o vizinho de y'_i fora de C_i em tempo pelo menos dois. Assim, temos que o conjunto $S_{i-1} \cup \{y_i\}$ infecta y'_i em tempo pelo menos quatro. Esse argumento é ilustrado pela Figura 12.

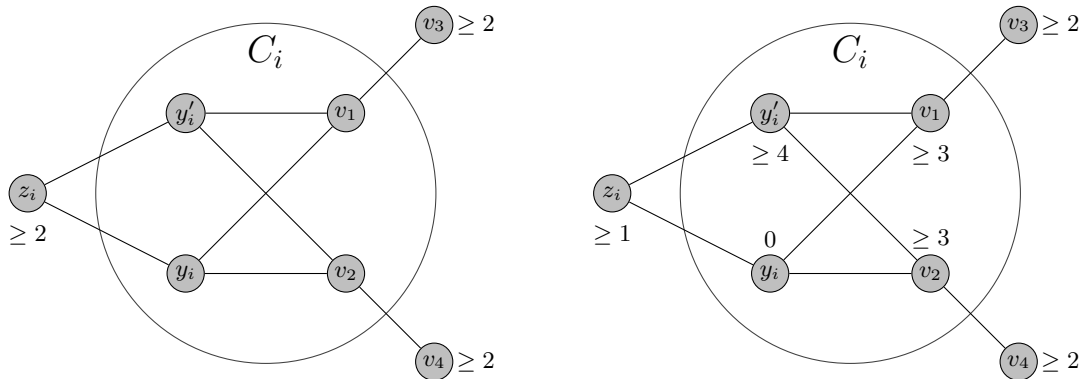


Figura 12 – Vértices da componente C_i antes e depois da adição de y_i a S_{i-1} com seus tempos de infecção.

Temos que em toda componente conexa do grafo $G[Y_{i-1}]$, existe pelo menos um vértice em $N_2(x)$ e um vértice em $N(x) \cup N_3(x)$. Se y'_i estiver em $N_2(x)$ (respec. em $N(x) \cup N_3(x)$), seja $S_i = S_{i-1} \cup \{y_i\} \cup ((Y_i \setminus V(C_i)) \cap N_2(x))$ (respec. $S_i = S_{i-1} \cup \{y_i\} \cup ((Y_i \setminus V(C_i)) \cap (N(x) \cup N_3(x)))$). Assim, temos que S_i infecta todos as componentes conexas de $G[Y_{i-1}]$, o que implica que S_i é um conjunto de envoltória. Também é possível provar que os vértices adicionados em componentes conexas diferentes de C_i estão a distância pelo menos três de todos os vizinhos de y'_i em C_i . Portanto, como todos os vizinhos de y'_i em C_i são infectados em tempo pelo menos três por $S_{i-1} \cup \{y_i\}$ e y'_i só possui um vizinho fora de C_i , temos que S_i infecta y'_i em tempo pelo menos quatro. \square

Pelo Lema 3.15, há apenas $O(m)$ conjuntos em cada família Γ_0^u e eles podem ser todos computados em tempo $O(m)$. Portanto, o nosso algoritmo pode testar todos eles em tempo polinomial. Assim, temos o seguinte teorema.

Teorema 3.17. TEMPO MÁXIMO DE INFECÇÃO para $k = 4$ pode ser decidido em tempo $O(m^2 \cdot n^9)$ em grafos bipartidos.

Demonstração. Considerando G conexo, primeiro o algoritmo seleciona um vértice $u \in V(G)$. Então, computa a família Γ_0^u em $O(m)$. Depois disso, ele seleciona um conjunto $T_0 \in \Gamma_0^u$ e um conjunto F de no máximo oito vértices. Então, o algoritmo verifica se o conjunto $T_0 \cup N_{\geq 4}(u) \cup F$ infecta algum vértice em tempo quatro em $O(m)$. Se existir uma tripla (u, T_0, F) tal que o conjunto $T_0 \cup N_{\geq 4}(u) \cup F$ infecta algum vértice em tempo quatro, responde SIM. Caso contrário, responde NÃO. A corretude do algoritmo segue diretamente do Lema

3.16 e a complexidade de tempo do algoritmo é $O(m^2 \cdot n^9)$, pois há $O(m)$ conjuntos em Γ_0^u . \square

Novamente, note que não mostramos que o tamanho de F é apertado, ou seja, é possível que seja necessário menos do que oito vértices em F na caracterização do lema 3.9. De fato, nós conjecturamos o seguinte.

Conjectura 3.18. *Se $t(G) \geq 4$ então existe um vértice u , um conjunto $T_0 \in \Gamma_0^u$ e um conjunto de vértices F tais que $|F| \leq 5$ e $T_0 \cup F \cup N_{\geq 4}(u)$ infecta algum vértice em tempo quatro.*

Com isso, se a Conjectura 3.18 fosse verdadeira, teríamos um algoritmo para decidir se $t(G) \geq 4$ em tempo $O(m^2 \cdot n^6)$, quando G é bipartido.

3.6 Algoritmo polinomial em grafos grade sólidos com grau máximo três

Nesta seção, nós apresentamos uma importante caracterização dos grafos de grau máximo três que possuem tempo máximo de infecção pelo menos k através do tamanho de seus caminhos induzidos. Baseando-nos nessa caracterização, nós construímos um algoritmo que decide o problema TEMPO MÁXIMO DE INFECÇÃO em tempo polinomial para grafos grade sólidos de grau máximo três. De fato, podemos usar essa caracterização para calcular $t(G)$ de qualquer grafo de grau máximo três do qual possamos achar os caminhos máximos induzidos de G com algumas restrições de grau definidas na caracterização.

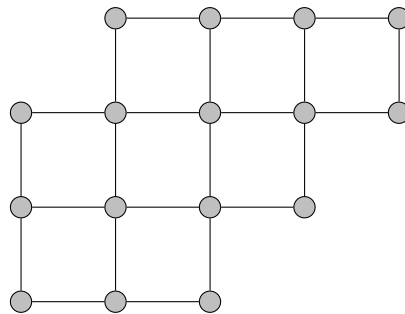


Figura 13 – Exemplo de grafo grade sólido.

Um grafo grade sólido é um grafo grade em que, se ele for desenhado em um plano cartesiano de forma que cada vértice está em uma coordenada inteira e cada aresta tenha

tamanho unitário, a área de cada face interna é unitária. Um exemplo de grafo grade pode ser encontrado na Figura 13.

Seja S um conjunto de envoltória de um grafo G . Um caminho $P = v_0, v_1, \dots, v_n$ em G é um S -caminho de infecção se, para todo $0 \leq i \leq n - 1$, $t(G, S, v_i) < t(G, S, v_{i+1})$. Note que, se $t(G, S, v) = k$, então certamente existe um S -caminho de infecção $v_0, v_1, \dots, v_k = v$, onde $t(S, v_i) = i$ para cada $0 \leq i \leq k$.

O próximo lema é uma importante caracterização dos grafos de grau máximo três que possuem tempo máximo de infecção pelo menos k .

Lema 3.19. *Sejam G um grafo tal que $\Delta(G) = 3$ e $k \geq 1$ um inteiro. Temos que, $t(G) \geq k$ se e somente se G possui um caminho induzido P tal que, ou todos os vértices de P têm grau três e P tem tamanho pelo menos $2k - 2$, ou todos os vértices de P têm grau três, exceto uma de suas extremidades a qual possui grau dois, e P tem tamanho pelo menos $k - 1$.*

Esboço da demonstração. Primeiro, suponha que $t(G) \geq k$. Sejam S o conjunto de envoltória que infecta G em tempo pelo menos k , $t = t(G, S)$ e v um vértice de G tal que $t(G, S, v) = t$. Temos que v tem grau pelo menos dois, pois, caso contrário, ele estaria em S . Vamos dividir a prova em dois casos.

O primeiro ocorre quando $d(v) = 2$. Nesse caso, defina $P = v_1, \dots, v_{t-1}, v_t = v$ como sendo um S -caminho de infecção onde cada v_i é infectado em tempo i por S . Todos os vértices em $\{v_1, v_2, \dots, v_{t-2}, v_{t-1}\}$ possuem grau três porque cada vértice v_i , para $1 \leq i \leq t - 1$, possui um vizinho em P que é infectado depois por S e um vizinho em P infectado antes e, assim, precisa de mais um vizinho infectado antes dele para ser infectado em tempo i por S . Note que, como isso vale para cada v_i em $\{v_1, v_2, \dots, v_{t-2}, v_{t-1}\}$, esse terceiro vizinho não pode estar em P . Assim, temos que P é um caminho induzido de tamanho $t - 1 \geq k - 1$ tal que todos os vértices internos têm grau três exceto v que possui grau dois.

O segundo caso ocorre quando $d(v) = 3$. De forma semelhante ao caso anterior, é possível provar que existe dois S -caminhos de infecção induzidos e disjuntos partindo de v , cada um com tamanho $t - 1$ e todos os seus vértices têm grau três. Assim, temos um caminho induzido de tamanho $2t - 2 \geq 2k - 2$ tal que todos os seus vértices têm grau três.

Agora suponha que G possui um caminho induzido P tal que, ou todos os vértices de P têm grau três e P tem tamanho pelo menos $2k - 2$, ou todos os vértices de P têm grau três, exceto uma de suas extremidades a qual possui grau dois, e P tem tamanho pelo menos $k - 1$.

Primeiro, suponha que G possui um caminho induzido $P = v_1, v_2, \dots, v_t$ tal que $t \geq k$

e todos os seus vértices possuem grau três, exceto v_t o qual possui grau dois. Assim, cada vértice v_i , para $1 \leq i \leq t - 1$, possui exatamente um vizinho fora de P . Seja S' o conjunto dos vizinhos fora de P de todos os vértices v_i . É possível provar que S' infecta cada v_i em tempo i .

Agora, suponha que G possui um caminho induzido $P = v_1, v_2, \dots, v_{2t-1}$ tal que $t \geq k$ e todos os seus vértices possuem grau três. De forma semelhante ao caso anterior, temos que, se S' é o conjunto dos vizinhos fora de P de todos os vértices v_i , é possível provar que todo vértice a distância d de v_t em P é infectado por S' em tempo $t - d$, ou seja, S' infecta cada vértice v_i em tempo $t - |t - i|$.

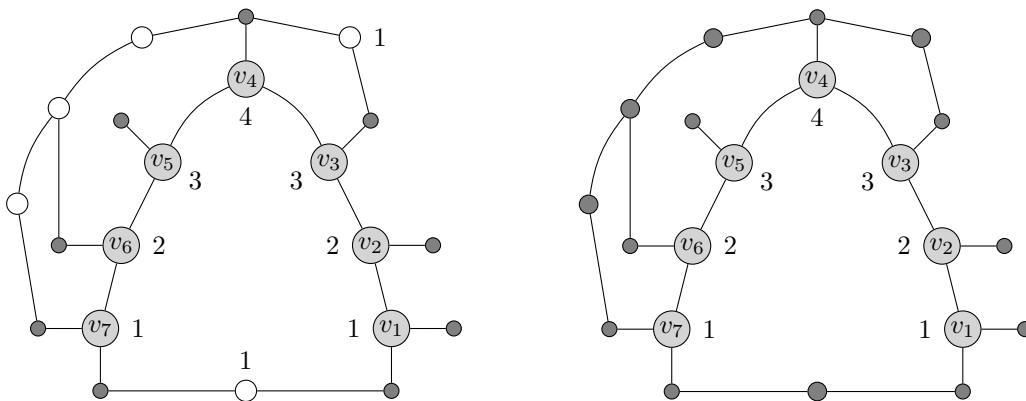


Figura 14 – Um grafo com $\Delta = 3$ infectado pelo conjunto S' à esquerda e pelo conjunto S à direita com os tempos de infecção dos vértices.

Sejam Y o conjunto de vértices em G que não estão nem em $V(P)$ e nem em S' e $S = S' \cup Y = V(G) \setminus V(P)$. Como todos os vértices em P são adjacentes só aos vértices em S' , temos que o conjunto S infecta os vértices em P no mesmo tempo que o conjunto S' . A Figura 14 ilustra esse argumento, na qual os vértices mais escuros são os que fazem parte de S' à esquerda e S à direita. Portanto, como em ambos os casos temos que v_t é infectado por S em tempo t e S é um conjunto de envoltória, temos que $t(G) \geq t \geq k$. \square

Como consequência direta do Lema 3.19, podemos computar em tempo polinomial o tempo máximo de infecção de um grafo de qualquer classe de grafos em que se possa computar em tempo polinomial os tamanhos dos caminhos induzidos máximos com as restrições de grau descritas no Lema 3.19. Particularmente, esse é o caso dos grafos grade sólidos.

Teorema 3.20. *O tempo máximo de infecção de um grafo grade sólido com grau máximo três pode ser computado em tempo $O(n^2)$.*

Esboço da demonstração. Todo grafo de grau máximo $\Delta = 3$ é livre de $K_{1,4}$. Portanto, temos que todo grafo grade sólido é composto de escadas e árvores conectando as extremidades dessas escadas. Na Figura 15, temos um exemplo de grafo grade sólido com grau máximo três.

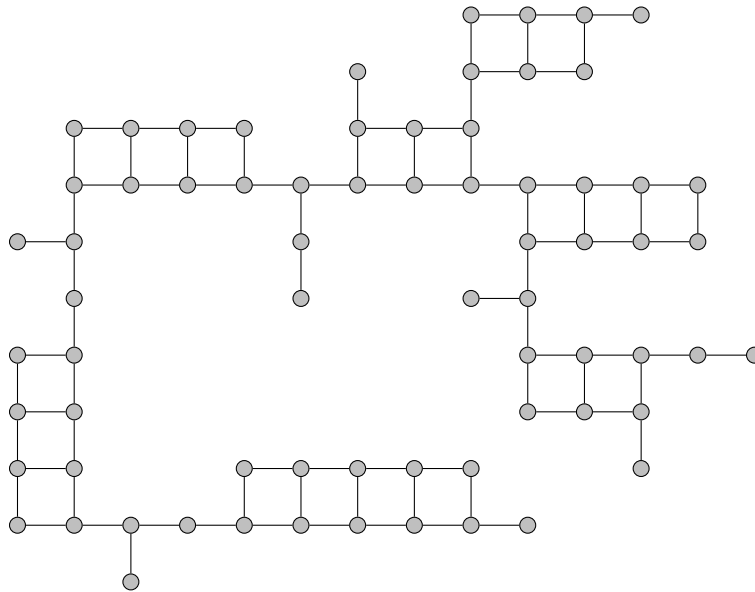


Figura 15 – Um grafo grade sólido com grau máximo três.

A ideia, então, é achar um caminho induzido máximo cujo todos os vértices possuam grau três e o caminho induzido máximo cujo todos os vértices possuam grau três, exceto a sua extremidade que deve ter grau dois, pois, assim, é possível invocar o Lema 3.19 para calcular $t(G)$.

Como os grafos grade sólidos de grau máximo três são compostos de escadas e árvores as conectando, a única dificuldade real em calcular os caminhos induzidos máximos é o cálculo dos caminhos induzidos máximos entre as extremidades das escadas.

É possível provar que, para todo $k > 0$, o tamanho do maior caminho induzido entre duas extremidades de um escada L_k é $k + 2 \cdot \lfloor (k + 1)/4 \rfloor - 1$, se as extremidades estão a uma distância $k - 1$ uma da outra, e $k + 2 \cdot \lfloor (k - 1)/4 \rfloor$, se as extremidades estão a uma distância k uma da outra.

Com isso, somos capazes de calcular o tamanho dos caminhos induzidos máximos entre duas extremidades que respeitam as restrições de grau, mesmo que hajam extremidades de grau dois.

O algoritmo então funciona assim: Primeiro transformamos o grafo original em um grafo com pesos nas arestas de forma que cada aresta entre dois vértices que não pertencem a

uma mesma escada continua no grafo com peso um e cada escada é transformada em um K_4 de forma que cada uma de suas arestas têm o peso correspondente ao tamanho do caminho máximo induzido no grafo original entre os vértices correspondentes na escada no qual todos os seus vértices têm grau três.

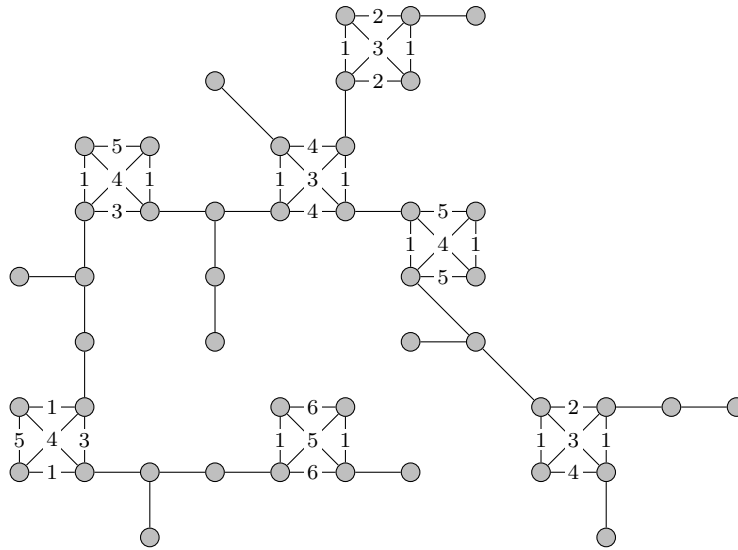


Figura 16 – O grafo resultante da transformação do grafo da Figura 15.

A Figura 16 ilustra o grafo resultante da transformação aplicada ao grafo da figura 15. Essa transformação pode ser feita em tempo linear no tamanho do grafo.

Ficamos agora com um grafo que só possui um vértice para cada vértice do grafo original, exceto os vértices internos das escadas, e possui apenas um caminho induzido entre qualquer par de vértices. Adicionalmente, temos também que se um vértice de um K_4 no grafo transformado tem grau quatro, significa que no grafo original ele possui grau três, e se um vértice de um K_4 no grafo transformado tem grau três, significa que no grafo original ele possui grau dois. Temos ainda que a soma dos pesos das arestas no único caminho induzido entre dois vértices no grafo transformado corresponde ao tamanho do caminho induzido máximo entre aqueles vértices no grafo original.

Note que, quando um caminho induzido máximo no grafo original acaba em um vértice interno de uma escada, esse vértice interno sempre é adjacente a uma extremidade da escada. Assim, podemos calcular o caminho máximo com as devidas restrições aos seus graus a partir de um vértice no grafo original calculando a soma dos pesos das arestas nos caminhos induzidos a partir daquele vértice no grafo transformado, sempre tomando cuidado com o fato de que, se um caminho acaba em um vértice de um K_4 de grau três, significa que ele tem grau dois no grafo original.

Algoritmo 1: Algoritmo que computa $t(G)$ para qualquer grafo grade sólido G com grau máximo três.

```

Algoritmo TempoMaximo( $G$ )
   $G' = \text{Transforma}(G)$ 
   $\text{maxTempoInfec} = 0$ 
  para todo  $u \in V(G')$  tal que  $d_G(u) \geq 2$  faça
    se  $d_G(u) = 2$  então
       $\lfloor \text{InfecTimeU} = \text{CaminhoInduzidoMaximo}(G', u) + 1$ 
    senão
       $\lfloor \text{InfecTimeU} = \lfloor (\text{CaminhoInduzidoMaximo}(G', u) + 2) / 2 \rfloor$ 
    se  $\text{maxTempoInfec} < \text{InfecTimeU}$  então
       $\lfloor \text{maxTempoInfec} = \text{InfecTimeU}$ 
  retorna  $\text{maxTempoInfec}$ 

```

No Algoritmo 1, a função `Transforma` transforma o grafo original G no grafo com pesos nas arestas G' e a função `CaminhoInduzidoMaximo` recebe um vértice u e retorna o caminho induzido máximo a partir de u cujo todos os vértices possuem grau três, exceto talvez pelo próprio u . Como ambas as funções podem executar em tempo $O(m)$, que é igual a $O(n)$, pois $\Delta = 3$, o Algoritmo 1 executa em tempo $O(n^2)$. \square

3.7 Algoritmo FPT no parâmetro $\Delta + k$

Nesta seção, nós mostramos que o problema TEMPO MÁXIMO DE INFECÇÃO é **FPT** quando parametrizado em $\Delta + k$. O próximo lema apresenta uma caracterização cuja prova é baseada em fatos simples e o algoritmo que a computa é polinomial.

Lema 3.21. $t(G) \geq k$ se e somente se existe um vértice u e um subconjunto $S \subseteq N_{<k}(u)$ tais que $S \cup N_{\geq k}(u)$ é um conjunto de envoltória que infecta u em tempo k .

Esboço da demonstração. Se existe um vértice u e um conjunto $S' \subseteq N_{<k}(u)$ tais que $S \cup N_{\geq k}(u)$ é um conjunto de envoltória que infecta u em tempo k então, trivialmente, temos que $t(G) \geq k$. Por outro lado, se $t(G) \geq k$, temos um conjunto de envoltória S' que infecta um vértice u em tempo k . Seja $S = S' \cap N_{<k}(u)$. Pelo Lema 3.4, temos então que o conjunto $S' \cup N_{\geq k}(u) = S \cup N_{\geq k}(u)$ é um conjunto de envoltória e infecta u em tempo k . \square

Teorema 3.22. O problema TEMPO MÁXIMO DE INFECÇÃO pode ser decidido em tempo $O(2^{\Delta^k} \cdot k \cdot \Delta \cdot n^2)$.

Demonstração. Para cada $u \in V(G)$, temos que $|N_{<k}(u)| \leq \Delta^k$ e, conseqüentemente, o conjunto das partes de $N_{<k}(u)$ tem tamanho $2^{|N_{<k}(u)|} \leq 2^{\Delta^k}$. Dessa forma, para determinar se $t(G) \geq k$, o nosso algoritmo verifica se existe um vértice u e um conjunto $S \subseteq N_{<k}(u)$ tal que $S \cup N_{\geq k}(u)$ infecta u em tempo k . Essa verificação leva tempo $O(k \cdot m)$, que é igual a $O(k \cdot \Delta \cdot n)$. Assim, o nosso algoritmo leva tempo $O(2^{\Delta^k} \cdot k \cdot \Delta \cdot n^2)$ para computar $t(G)$ e a sua corretude segue direto do Lema 3.21. \square

Na verdade, se $\Delta = 3$, nós podemos fazer melhor do que o algoritmo do Teorema 3.22. O teorema a seguir segue do Lema 3.19 e do fato de existirem $O(4^k)$ vértices distintos a distância no máximo $2k + 2$ de qualquer vértice em um grafo de grau máximo três.

Teorema 3.23. TEMPO MÁXIMO DE INFECÇÃO P_3 *pode ser decidido em tempo $O(4^k \cdot n)$ em grafos com grau máximo três.*

3.8 Resultados parametrizados na largura em árvore

Nesta seção, nós mostramos de duas formas distintas que o problema TEMPO MÁXIMO DE INFECÇÃO parametrizado por $tw(G) + k$ é **FPT**. Na primeira forma, usamos o celebrado Teorema de Courcelle [48]. Na segunda, nós provamos que existe um algoritmo XP na largura em árvore do grafo de entrada que computa $t(G)$, o que implica que o problema parametrizado TEMPO MÁXIMO DE INFECÇÃO parametrizado por $tw(G)$ é **XP** e apresentamos uma modificação do nosso algoritmo XP que executa em $O((50k)^{tw+1} \cdot tw^3 \cdot n)$, onde $n = |V(G)|$ e $tw = tw(G)$, provando assim que ele é **FPT** quando parametrizado por $tw(G) + k$.

3.8.1 Aplicação do Teorema de Courcelle

A lógica monádica de segunda ordem, do inglês *monadic second order logic* (MSOL ou MSO), é uma restrição da lógica de segunda ordem, onde só são aceitos quantificadores quando aplicados em conjuntos. Uma das conseqüências do Teorema de Courcelle é que qualquer propriedade em grafos que pode ser definida por uma fórmula φ pertencente a uma variação da MSO, conhecida como MSO_1 , pode ser reconhecida por um algoritmo FPT cujo parâmetro é $cw(G) + \|\varphi\|$, onde $\|\varphi\|$ é o tamanho da fórmula e $cw(G)$ é a largura em clique do grafo.

O parâmetro *largura em clique* de um grafo, introduzido por Courcelle, Engelfriet e Rozenberg [49], pode ser definida como a quantidade mínima de rótulos necessários para se criar tal grafo usando as seguintes quatro operações:

- criação de um novo vértice v com o rótulo i ;
- união disjunta de dois grafos rotulados;
- adição de uma aresta entre cada vértice com rótulo i e cada vértice com rótulo j ;
- rerrotulação dos vértices com rótulo i para o rótulo j ;

Em [50], foi provado que $cw(G) \leq 3 \cdot 2^{tw(G)-1}$, o que implica que se um problema parametrizado pela largura em clique de um grafo é **FPT**, então o mesmo problema parametrizado pela largura em árvore também é **FPT**. Isso acontece porque um algoritmo FPT no parâmetro largura em clique executa em $f(cw(G)) \cdot n^c = O(f(3 \cdot 2^{tw(G)-1}) \cdot n^c) = O(f'(tw(G)) \cdot n^c)$ passos, para alguma função f , o que implica que ele é FPT no parâmetro largura em árvore.

Na MSO_1 , o grafo é descrito pelo seu conjunto de vértices V e uma função de adjacência $adj(\cdot, \cdot)$ entre os vértices. Adicionalmente, na MSO_1 , além dos operadores usuais \wedge , \vee e \neg , o operador \in também pode ser usado e os quantificadores são aceitos apenas quando aplicados em vértices ou em um conjunto de vértices.

Considere a seguinte fórmula na lógica MSO_1 :

$$\begin{aligned} maxtime_k := & \exists V_{-1}, V_0, V_1, \dots, V_k : \forall w \in V : \neg(w \in V_{-1}) \wedge (w \in V_k) \wedge \\ & \bigwedge_{0 \leq i < k} : (w \in V_i \implies w \in V_{i+1}) \wedge (\exists v \in V : (v \in V_{i+1} \setminus V_i)) \wedge \\ & \left((w \in V_i) \vee \left((w \in V_{i+1}) \iff \right. \right. \\ & \left. \left. \exists y, z \in V : (y \in V_i \setminus V_{i-1}) \wedge (z \in V_i) \wedge adj(y, w) \wedge adj(w, z) \right) \right) \end{aligned}$$

Note que, apesar do operador \setminus não ser próprio da MSO_1 , a fórmula $x \in V \setminus S$ pode facilmente ser substituída pela fórmula equivalente na MSO_1 $(x \in V) \wedge (\neg(v \in S))$.

Adicionalmente, o operador $\bigwedge_{0 \leq i < k}$ é apenas uma abreviação que representa k vezes as fórmulas que vem após ele, modificando apenas o valor de i nas fórmulas e ligando-as por um \wedge .

Na primeira linha da fórmula $maxtime_k$, nós basicamente definimos $V_{-1} = \emptyset$ e $V_k = V$. Incluímos V_{-1} apenas para simplificar a fórmula. Na segunda linha, nós nos asseguramos que $V_0 \subset V_1 \subset \dots \subset V_k$. Na última linha, nós garantimos que, para todo $0 \leq i < k$, caso um vértice não seja infectado até o tempo i , ele é infectado em tempo $i + 1$ se e somente se ele possui um vizinho infectado em tempo i e outro infectado até o tempo i .

Assim, a fórmula $maxtime_k$ é verdadeira se e somente se existe um conjunto de envoltória V_0 que infecta o grafo em tempo exatamente k , ou seja, $t(G) \geq k$. Portanto, como $\|maxtime_k\| = \Theta(k)$, pelo Teorema de Courcelle, o TEMPO MÁXIMO DE INFECÇÃO é **FPT** quando parametrizado por $cw(G) + k$, o que implica que ele também é **FPT** quando parametrizado por $tw(G) + k$.

3.8.2 Algoritmos parametrizados

Na subseção anterior, nós mostramos que o problema TEMPO MÁXIMO DE INFECÇÃO é **FPT** quando parametrizado por $tw(G) + k$ através do Teorema de Courcelle. No entanto, os algoritmos oriundos desse tipo de resultado raramente são práticos, pois o seu tempo de execução é do tipo $O((\exp^{(t)}(f(tw(G), k)) \cdot n))$, onde k é o tamanho da fórmula, $\exp^{(0)}(x) = x$, $\exp^{(y)}(x) = 2^{\exp^{(y-1)}(x)}$, t depende linearmente do número de alternâncias de quantificadores na fórmula e f é uma função computável.

Por isso, nesta subseção, vamos apresentar um algoritmo que, dado um grafo G e uma decomposição em árvore de G de largura $tw(G)$, determina se $t(G) \leq k$ em tempo $O((50k)^{tw+1} \cdot tw^3 \cdot n)$ e determina $t(G)$ em tempo $O(50^{tw} \cdot tw^3 \cdot n^{tw+2})$, onde $n = |V(G)|$ e $tw = tw(G)$. Isso implica que o problema TEMPO MÁXIMO DE INFECÇÃO parametrizado por $tw(G)$ é **XP** e parametrizado por $tw(G) + k$ é **FPT**, pois existem algoritmos tratáveis por parâmetro fixo, onde o parâmetro é $tw(G)$, que computam uma decomposição em árvore de um grafo G de largura $tw(G)$ [26].

A técnica que usamos nesta seção para conseguir os algoritmos XP e FPT é uma técnica de programação dinâmica bastante utilizada [10, 25, 31, 47, 77].

Primeiro, vamos a algumas definições básicas. Sejam G o grafo que trabalhamos nessa seção e $n = |V(G)|$. Seja $(\mathcal{T} = (T, F), \mathcal{B} = \{B_j : j \in T \mid B_j \subseteq V(G)\})$ uma boa decomposição em árvore de G de largura $tw = tw(G)$, de forma que a árvore \mathcal{T} está enraizada no nó r . Seja G_j , para todo $j \in T$, o subgrafo de G induzido pela união das sacolas de todos os descendentes de j , incluindo o próprio j . Note então que $G_r = G$.

Para cada nó $j \in T$, sejam P_j o conjunto de todas as funções $p : B_j \rightarrow [n-1]_0$ e F_j o conjunto de todas as funções $f : B_j \rightarrow \{z, o_1, o_2, t_1, t_2\}$.

A ideia por trás das funções em P_j é representar o tempo em que os vértices em B_j são infectados. A ideia por trás das funções em F_j é representar a situação dos vizinhos de cada vértice v em B_j da seguinte maneira:

- $f(v) = z$ indica que v não tem nenhum vizinho em G_j infectado em tempo menor que

$p(v)$;

- $f(v) = o_1$ indica que v tem exatamente um vizinho em G_j infectado em tempo menor que $p(v)$ e esse vizinho é infectado em tempo menor que $p(v) - 1$;
- $f(v) = o_2$ indica que v tem exatamente um vizinho em G_j infectado em tempo menor que $p(v)$ e esse vizinho é infectado em tempo $p(v) - 1$;
- $f(v) = t_1$ indica que v possui dois ou mais vizinhos em G_j infectados em tempo menor que $p(v)$ e exatamente um deles é infectado em tempo menor que $p(v) - 1$;
- $f(v) = t_2$ indica que v tem dois ou mais vizinhos em G_j infectados em tempo menor que $p(v)$ e todos eles são infectado em tempo $p(v) - 1$.

Uma função $g : V(G_j) \rightarrow [n - 1]_0$ é uma função de *quasi-infecção* de G_j se, para todo $v \in V(G_j)$, existe no máximo um vértice $u \in N_{G_j}(v)$ tal que $g(u) < g(v) - 1$.

Para qualquer $j \in T$, $p \in P_j$ e $f \in F_j$, seja $W_j(p, f) = \max_g \max_{v \in V(G_j)} g(v)$ tal que g itera sobre todas as funções de *quasi-infecção* de G_j tais que:

- Todo vértice $v \in V(G_j) \setminus B_j$ tal que $g(v) > 0$ possui pelo menos dois vizinhos u_1 e u_2 tais que $g(u_1) < g(v)$ e $g(u_2) < g(v)$;
- A função p é g restrita a B_j e
- As funções g e f não se contradizem.

Dizemos que uma função de *quasi-infecção* g de G_j contradiz uma função $f \in F_j$ se, para algum vértice $v \in B_j$, $f(v)$ não condiz com a realidade dos valores de g na vizinhança de v . Por exemplo, se $f(v) = z$ e existe um vértice $u \in N_{G_j}(v)$ tal que $g(u) < g(v)$. Seja $W_j(p, f) = -1$ se não existir uma função de *quasi-infecção* de G_j que satisfaz essas três propriedades.

Note que $t(G) = \max_{p,f} W_r(p, f)$, onde p e f iteram sobre todas as funções em P_j e F_j tais que, para todo $v \in B_r$, ou $p(r) = 0$ ou $f(v) = t_1$ ou $f(v) = t_2$.

Para alguns índices (p, f) da tabela W_j , podemos ver diretamente que $W_j(p, f) = -1$. Dizemos que um índice (p, f) é inválido se, analisando apenas os valores de p nos vértices em B_j , podemos deduzir que qualquer extensão de p contraria f . Portanto, temos que, se (p, f) é um índice inválido, $W_j(p, f) = -1$. Adicionalmente, podemos determinar se um índice é inválido em tempo $O(tw^2)$.

Agora, podemos mostrar como calcular $W_j(p, f)$ dependendo do tipo de nó j .

Lema 3.24 (Nó Folha). *Sejam j um nó folha e (p, f) um índice válido de W_j . Temos que $W_j(p, f) = \max_{v \in B_j} p(v)$.*

Lema 3.25 (Nó Esquece). *Sejam j um nó esquece, j' o seu filho e (p, f) um índice válido de W_j . Temos que $W_j(p, f) = \max_{p', f'} W_{j'}(p', f')$ tal que $p' \in P_{j'}$ e $f' \in F_{j'}$ iteram sobre todas as extensões de p e f , respectivamente, tais que ou $f'(v) = t_1$ ou $f'(v) = t_2$ ou ainda $p'(v) = 0$.*

Lema 3.26 (Nó Introduz). *Sejam j um nó esquece, j' o seu filho e (p, f) um índice válido de W_j . Temos que, se $\max_{f'}(W_{j'}(p', f')) \geq 0$, então $W_j(p, f) = \max(p(v), \max_{f'}(W_{j'}(p', f')))$, onde p' é p restrito a $B_{j'}$ e f' itera sobre todas as funções em $F_{j'}$ tais que, para todo $r \in B_{j'} \cap N(v)$ tal que $p(r) > p(v)$, temos que:*

1. Se $p(v) < p(r) - 1$ e $f(r) = o_1$, então $f'(r) = z$;
2. Se $p(v) = p(r) - 1$ e $f(r) = o_2$, então $f'(r) = z$;
3. Se $p(v) = p(r) - 1$ e $f(r) = t_1$, então ou $f'(r) = o_1$ ou $f'(r) = t_1$;
4. Se $p(v) < p(r) - 1$ e $f(r) = t_1$, então ou $f'(r) = o_2$ ou $f'(r) = t_2$;
5. Se $p(v) = p(r) - 1$ e $f(r) = t_2$, então ou $f'(r) = o_2$ ou $f'(r) = t_2$;

e $f'(r) = f(r)$ para todos os outros vértices. Se, para todo f' descrito acima, $W_{j'}(p', f') = -1$, então $W_j(p, f) = -1$.

Lema 3.27 (Nó Junção). *Sejam j um nó junção, j_1 e j_2 os seus filhos e (p, f) um índice válido de W_j . Se existe um par (f_1, f_2) tal que $\min(W_{j_1}(p, f_1), W_{j_2}(p, f_2)) \geq 0$, então $W_j(p, f) = \max_{(f_1, f_2)} \max(W_{j_1}(p, f_1), W_{j_2}(p, f_2))$ onde o par (f_1, f_2) itera sobre todas as funções em F_j de forma que, para cada $r \in B_j$, existem inteiros $i \neq j \in \{1, 2\}$ tais que:*

1. Se $f(r) = z$, então $f_i(r) = z$ e $f_j(r) = z$;
2. Se $f(r) = o_1$, então $f_i(r) = z$ e $f_j(r) = o_1$;
3. Se $f(r) = o_2$, então $f_i(r) = z$ e $f_j(r) = o_2$;
4. Se $f(r) = t_1$, então:
 - a) $f_i(r) = z$ e $f_j(r) = t_1$; ou
 - b) $f_i(r) = o_1$ e $f_j(r) = o_2$; ou
 - c) $f_i(r) = o_1$ e $f_j(r) = t_2$; ou

d) $f_i(r) = o_2$ e $f_j(r) = t_1$; ou

e) $f_i(r) = t_1$ e $f_j(r) = t_2$.

5. Se $f(r) = t_2$, então:

a) $f_i(r) = z$ e $f_j(r) = t_2$; ou

b) $f_i(r) = o_2$ e $f_j(r) = o_2$; ou

c) $f_i(r) = o_2$ e $f_j(r) = t_2$; ou

d) $f_i(r) = t_2$ e $f_j(r) = t_2$.

Se não existe tal par (f_1, f_2) , $W_j(p, f) = -1$.

Agora que sabemos computar a tabela de qualquer tipo de nó, nós podemos apresentar o resultado principal da seção.

Teorema 3.28. *Dado uma decomposição em árvore com largura tw de um grafo G com n vértices, é possível determinar $t(G)$ em tempo $O(50^{tw+1} \cdot tw^3 \cdot n^{tw+2})$. Além disso, podemos determinar se $t(G) \geq k$ em $O((50k)^{tw+1} \cdot tw^3 \cdot n)$.*

Demonstração. Antes de tudo, o nosso algoritmo computa uma boa decomposição em árvore de largura tw com $O(tw \cdot n)$ nós em tempo $O(tw^2 \cdot n)$ a partir da decomposição em árvore dada.

Depois, o nosso algoritmo computa o valor de todas as tabelas W_j de “baixo para cima”, ou seja, ele computa as tabelas W_j , onde j é um nó folha e, após isso, só computa uma tabela W_j se a tabela do filho, ou filhos, de j já tiver sido computada.

Para cada $j \in T$, temos que $|P_j| = n^{tw+1}$ e $F_j = 5^{tw+1}$. Dado um índice (p, f) de uma tabela M_j , o algoritmo leva tempo $O((tw+1)^2)$ para verificar se (p, f) é válido e, no pior caso, que é alcançado quando j é um nó junção, leva tempo $O(10^{tw})$ para, de fato, computar $W_j(p, f)$. Portanto, como existem $O(tw \cdot n)$ tabelas, o nosso algoritmo leva tempo $O(50^{tw} \cdot tw^3 \cdot n^{tw+2} + tw^2 \cdot n) = O(50^{tw} \cdot tw^3 \cdot n^{tw+2})$ para computar a tabela W_r e achar o maior valor dentre todos os índices (p, f) tais que $f(v) = t_1$ ou $f(v) = t_2$ ou $p(v) = 0$ para todo $v \in B_r$.

Podemos facilmente modificar este algoritmo para que ele decida, dados um grafo G e um inteiro k , se $t(G) \geq k$. Para isso, note que, pelo Lema 3.3, se $t(G) \geq k$, então existe um conjunto de envoltória que infecta o grafo em tempo exatamente k . Por isso, basta que P_j seja o conjunto de todas as funções $p : B_j \rightarrow [k]_0$, onde $p(v)$ representaria o tempo de infecção de

$p(v)$, e que $W_j(p, f)$ armazene o valor um, se existe uma função de *quasi*-infecção f de G_j de forma que $\max_{v \in V(G_t)} f(v) = k$, e armazena zero, caso contrário. Os lemas correspondentes aos tipos de nó são então modificados de acordo. Por exemplo, no nó *introduz*, teríamos o seguinte:

$$W_j(p, f) = \begin{cases} 1 & \text{se } p(v) = k \\ \max_{f'}(W_{j'}(p', f')) & \text{caso contrário} \end{cases}$$

Dessa forma, como $|P_j| = O(k^{tw(G)+1})$, tal algoritmo decidiria se $t(G) \geq k$ em tempo $O((50k)^{tw+1} \cdot tw^3 \cdot n)$. \square

Com isso, mostramos novamente que o problema TEMPO MÁXIMO DE INFECÇÃO parametrizado por $tw(G) + k$ é **FPT**.

4 RESULTADOS DE COMPLEXIDADE PARA TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3

Neste capítulo, serão apresentados os nossos resultados relacionados à complexidade do problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 . Todos os resultados aqui apresentados complementam de alguma forma algum resultado algorítmico do Capítulo 3.

Como vamos trabalhar apenas a convexidade P_3 neste capítulo, daqui e diante, nós não mencionamos explicitamente a convexidade que estamos nos referindo, assim como no capítulo anterior. Também omitimos o subscrito das notações $t_{P_3}(G)$, $t_{P_3}(G, S)$ e $t_{P_3}(G, S, v)$ e abreviamos o nome do problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 para TEMPO MÁXIMO DE INFECÇÃO.

Na Seção 4.1, mostramos que TEMPO MÁXIMO DE INFECÇÃO é **NP**-completo mesmo quando $k = n - n^\varepsilon + 4$, para qualquer $0 < \varepsilon \leq 1$ fixo, estendendo, assim, a prova em [18] para $k = 4$. Esse resultado complementa o resultado da Seção 3.1, pois, na Seção 3.1, mostramos que o problema é polinomial para $k = n - c$ quando c é constante, porém se torna **NP**-completo quando $c = n^\varepsilon - 4$ para ε tão perto de zero quanto se queira.

Nós mostramos, na Seção 4.2, que TEMPO MÁXIMO DE INFECÇÃO é **NP**-completo para grafos bipartidos para qualquer $k \geq 5$ fixo. Esse resultado complementa os resultados para k fixo em grafos bipartidos das Seções 3.3 e 3.5, pois, com ele, nós mostramos que TEMPO MÁXIMO DE INFECÇÃO em grafos bipartidos é polinomial para todo valor fixo de $k \leq 4$ e **NP**-completo para todo valor fixo de $k \geq 5$. Adicionalmente, esse resultado melhora o resultado em [18], onde Benevides *et al.* provaram que TEMPO MÁXIMO DE INFECÇÃO é **NP**-completo para grafos bipartidos para qualquer $k \geq 7$ fixo.

Na Seção 4.3, mostramos que TEMPO MÁXIMO DE INFECÇÃO é **NP**-completo mesmo em grafos grade de grau máximo três. Esse resultado responde uma questão em aberta proposta em [18] sobre a complexidade do problema TEMPO MÁXIMO DE INFECÇÃO em grafos planares bipartidos. Esse resultado complementa o resultado da Seção 3.6, que é voltado a grafos grade sólidos de grau máximo três.

Na Seção 4.4, mostramos que TEMPO MÁXIMO DE INFECÇÃO é **NP**-completo mesmo

para grafos com grau máximo fixo em qualquer valor maior ou igual a quatro e $k = \Theta(\log n)$, onde $n = |V(G)|$ e G é o grafo de entrada. Esse resultado implica que, a não ser que a ETH seja falsa, o problema não possui um algoritmo que executa em tempo $O(2^{\Delta^{\varepsilon \cdot k}} \cdot p(n, \Delta, k))$, para qualquer função polinomial $p(\cdot, \cdot, \cdot)$ e constante $0 \leq \varepsilon < 1$. Dessa forma, temos que esse resultado complementa o resultado na Seção 3.7, que apresenta um algoritmo que decide o problema em tempo $O(2^{\Delta^k} \cdot k \cdot \Delta \cdot n^2)$.

Finalmente, na Seção 4.5, mostramos que TEMPO MÁXIMO DE INFECÇÃO é $\mathbf{W}[1]$ -difícil quando parametrizado em $tw(G)$, o que implica que, a não ser que $\mathbf{FPT} = \mathbf{W}[1]$, o problema não possui algoritmo FPT no parâmetro $tw(G)$ e, a não ser que a ETH seja falsa, o problema não pode ser decidido em tempo $f(tw(G)) \cdot n^{o(tw(G))}$, para qualquer função computável f , onde $n = |V(G)|$ e G é o grafo de entrada, complementando um dos resultados da Seção 3.8, o qual apresenta um algoritmo XP que decide o problema parametrizado por $tw(G)$ em tempo $O(50^{tw(G)} \cdot tw(G)^3 \cdot n^{tw(G)+2})$.

Nos resultados nas Seções 4.1, 4.2 e 4.4, nós utilizamos reduções inspiradas na redução de 3-SAT para TEMPO MÁXIMO DE INFECÇÃO por Benevides *et al.* em [18].

Neste capítulo, nós apresentamos o enunciado de cada lema e teorema e, quando se tratar de um lema ou teorema importante, as ideias centrais da demonstração. A demonstração completa do resultado de \mathbf{NP} -completude em grafos bipartidos para $k \geq 5$ fixo se encontra no Apêndice A. As demonstrações completas de todos os outros resultados se encontram no Apêndice B.

4.1 NP-Completude para $k = n - n^\varepsilon + 4$

Em [18], Benevides *et al.* provaram que o TEMPO MÁXIMO DE INFECÇÃO é \mathbf{NP} -completo para qualquer constante $k \geq 4$. Nesta seção, nós generalizamos esse resultado. A seguir, nós provamos que TEMPO MÁXIMO DE INFECÇÃO é \mathbf{NP} -completo para $k = n - n^\varepsilon + 4$ e qualquer $0 < \varepsilon \leq 1$ fixo através de uma redução similar a encontrada em [18].

Teorema 4.1. *O problema TEMPO MÁXIMO DE INFECÇÃO é \mathbf{NP} -completo quando $k = n - n^\varepsilon + 4$, para qualquer $0 < \varepsilon \leq 1$ fixo.*

Esboço da demonstração. Para provar que TEMPO MÁXIMO DE INFECÇÃO para $k = n - n^\varepsilon + 4$ é \mathbf{NP} -completo, nós vamos fazer uma redução a partir de 3-SAT.

Dada uma fórmula φ com M cláusulas $\mathcal{C} = \{C_1, \dots, C_M\}$ em N variáveis $X = \{x_1, \dots, x_N\}$ como uma instância de 3-SAT, sejam $\ell_{i,1}$, $\ell_{i,2}$ e $\ell_{i,3}$ os três literais da cláusula

C_i . Primeiro, para cada cláusula C_i , adicione o *gadget* da Figura 17. Depois, para cada par

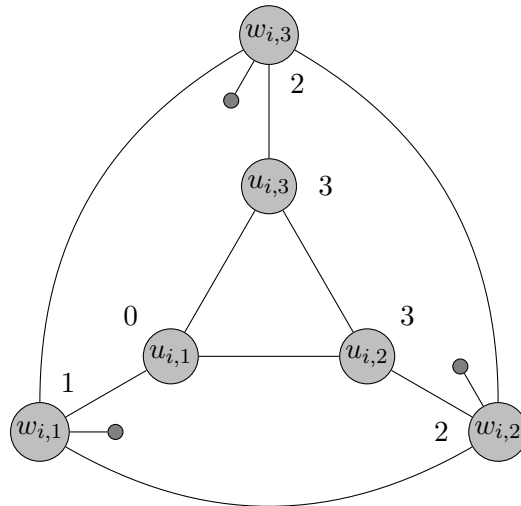


Figura 17 – *Gadget* da cláusula C_i com os tempos de infecção de seus vértices.

de literais $\ell_{i,a}, \ell_{j,b}$ tais que um é a negação do outro, adicione um vértice $y_{(i,a),(j,b)}$ adjacente à $w_{i,a}$ e à $w_{j,b}$. Seja Y o conjunto de todos os vértices criados dessa maneira. Até este ponto, nós adicionamos $9M + y$ vértices, onde $y = |Y|$. Sejam $U_i = \{u_{i,1}, u_{i,2}, u_{i,3}\}$, $U = \bigcup_{1 \leq i \leq M} U_i$ e $W = \bigcup_{1 \leq i \leq M} \{w_{i,1}, w_{i,2}, w_{i,3}\}$.

Sejam $c' = 9M + y + 4$, $t = c'^{1/\varepsilon}$ e $c = \lceil t \rceil - c'$. Note que, como $t \geq t^\varepsilon = c'$, nós temos que c é um inteiro não-negativo. Finalmente, adicione um caminho P_{c+1} com extremidades z e z' , adicione três vértices u, v, w , adicione uma aresta entre u e v , u e w , u e todos os vértices no caminho P_{c+1} e entre z e todos os vértices em Y . Note que o número de vértices desse grafo é $n = c + c'$. Seja T_0 o conjunto de vértices com grau um no grafo.

Nós adicionamos os *gadgets* da Figura 17 com o intuito de forçar a escolha de pelo menos um vértice de cada conjunto U_i para fazer parte de qualquer conjunto de envoltória e, para que um conjunto de envoltória atinja o tempo máximo, deve-se escolher exatamente um vértice de cada conjunto U_i . Assim, se um vértice $u_{i,j}$ está em um conjunto de envoltória, então tal conjunto de envoltória infecta $w_{i,j}$ em tempo um. Semelhantemente, se um vértice $u_{i,j}$ não está em um conjunto de envoltória, temos que tal conjunto de envoltória infecta $w_{i,j}$ em tempo dois.

Assim se φ é satisfável, então o conjunto $S = \{u_{i,k_i}^A : 1 \leq i \leq m\} \cup L$ é um conjunto de envoltória que infecta todos os vértices em Y em tempo três e, assim, infecta o grafo inteiro em tempo $4 + c$. Por outro lado, se φ é insatisfável, todos os conjuntos de envoltória irão infectar algum vértice em Y em tempo dois, o que implica que eles infectaram o grafo em tempo menor que $4 + c$.

No entanto, como $n = c + c'$ e $c = \lceil t \rceil - c'$, temos que $c = n - c' = n - t^\varepsilon$ e $n = \lceil t \rceil$. Portanto, temos que $c - 1 < n - n^\varepsilon \leq c$, pois $n - t^\varepsilon - 1 < n - n^\varepsilon = n - \lceil t \rceil^\varepsilon \leq n - t^\varepsilon$. Isso implica que φ é satisfatível se e somente se $t(G) \geq n - n^\varepsilon + 4$. \square

4.2 NP-Completeness for $k \geq 5$ in bipartite graphs

Também em [18], Benevides *et al.* mostraram que TEMPO MÁXIMO DE INFECÇÃO era NP-completo para qualquer $k \geq 7$ fixo em grafos bipartidos. Nesta seção, nós melhoramos esse resultado. Nós mostramos que o problema é NP-completo para qualquer $k \geq 5$ fixo em grafos bipartidos.

Teorema 4.2. *O problema TEMPO MÁXIMO DE INFECÇÃO é NP-completo para qualquer $k \geq 5$ fixo em grafos bipartidos.*

Esboço da demonstração. Para provar que TEMPO MÁXIMO DE INFECÇÃO para $k = 5$ em Grafos Bipartidos é NP-completo, nós vamos fazer um redução a partir de 3-SAT similar a anterior.

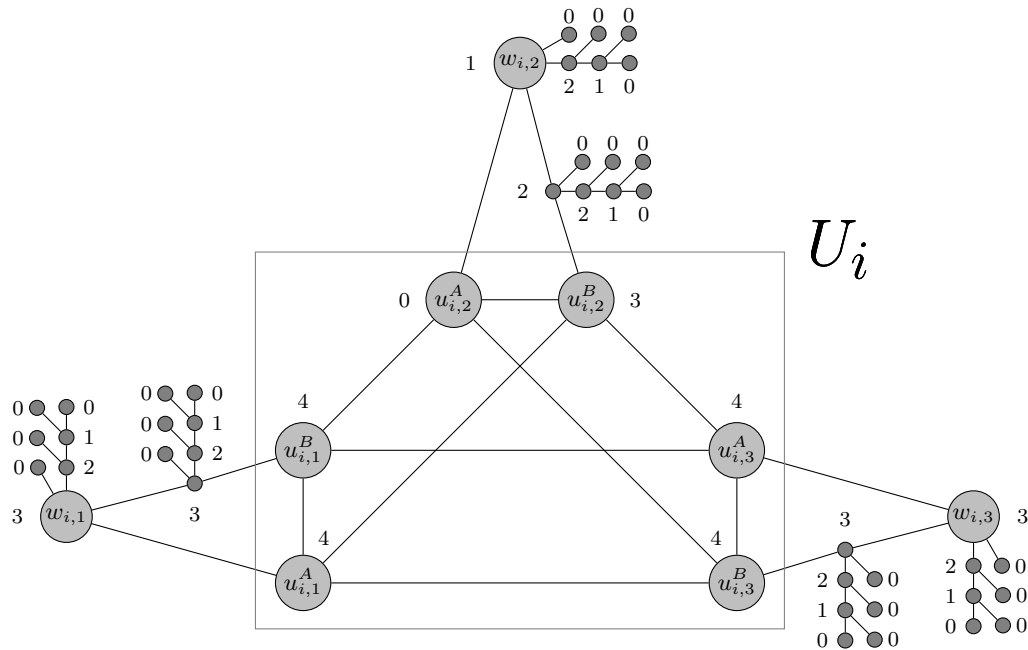


Figura 18 – *Gadget* bipartido adicionado para cada cláusula C_i com os tempos de infecção de seus vértices.

Dadas m cláusulas $\mathcal{C} = \{C_1, \dots, C_m\}$ em variáveis $X = \{x_1, \dots, x_n\}$ como uma instância de 3-SAT, sejam $\ell_{i,1}$, $\ell_{i,2}$ e $\ell_{i,3}$ os três literais da cláusula C_i . Primeiro, para cada

cláusula C_i , adicione o *gadget* da Figura 18. Depois, para cada par de literais $\ell_{i,a}, \ell_{j,b}$ tais que um é a negação do outro, adicione um vértice $y_{(i,a),(j,b)}$ adjacente à $w_{i,a}$ e à $w_{j,b}$. Seja Y o conjunto de todos os vértices criados dessa maneira.

Finalmente, adicione os vértices z e z' e faça z adjacente a todos os vértices no conjunto $Y \cup \{z'\}$. Sejam $U_i = \{u_{i,1}^A, u_{i,2}^A, u_{i,3}^A, u_{i,1}^B, u_{i,2}^B, u_{i,3}^B\}$ e $W_i = \{w_{i,1}, w_{i,2}, w_{i,3}\}$. Sejam $U = \cup_{1 \leq i \leq m} U_i$, $W = \cup_{1 \leq i \leq m} W_i$ e L o conjunto de todos os vértices de grau 1 em G .

Nós adicionamos os *gadgets* da Figura 18 com o mesmo intuito dos *gadgets* do Teorema 4.1, porém tomando o cuidado de manter o *gadgets* bipartidos.

Claramente, cada *gadgets* induz um subgrafo bipartido. Adicionalmente, podemos fazer com que cada vértice em W_i fique na mesma partição. Assim, se colocarmos os vértices em $Y \cup \{z'\}$ na partição diferente dos vértices em $W \cup \{z\}$, nós obtemos uma bipartição do grafo.

A prova que φ é satisfatível se e somente se G contém um conjunto de envoltória que infecta G em tempo pelo menos cinco é bastante similar a do Teorema 4.1.

Para valores de $k > 5$, é suficiente subdividir a aresta (z, z') em um caminho de tamanho $k - 5$, adicionando um vizinho pendente para cada vértice do caminho, inclusive z e z' . □

4.3 NP-Completo em grafos grade com grau máximo três

Em [18], foi provado que TEMPO MÁXIMO DE INFECÇÃO era NP-completo em grafos bipartidos e em grafos planares, porém deixaram a complexidade do problema em grafos bipartidos planares em aberto. Nesta seção, nós respondemos essa questão provando que o problema é NP-completo mesmo em grafos grade com grau máximo três. Para isso, nós nos baseamos na caracterização do Lema 3.19.

Teorema 4.3. *O problema TEMPO MÁXIMO DE INFECÇÃO é NP-completo mesmo em grafos grade de grau máximo três.*

Esboço da demonstração. Claramente, o problema está em NP. Para provar que ele também é NP-difícil, nós vamos mostrar uma redução do problema CAMINHO MÁXIMO em grafos grade com grau máximo três. O problema CAMINHO MÁXIMO consiste em, dado um grafo G e um inteiro k , determinar se G possui um caminho de tamanho pelo menos k . Esse problema é NP-completo, pois existe uma redução trivial a partir do problema CAMINHO HAMILTONIANO em grafos grade com grau máximo três que, por sua vez, também é NP-completo [97].

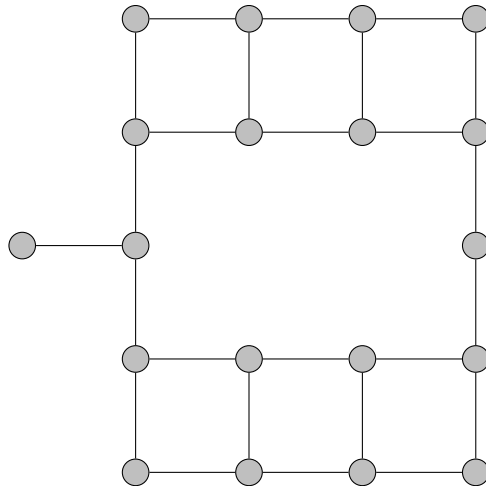


Figura 19 – Grafo grade com $\Delta = 3$.

Seja (G, k) a instância do problema CAMINHO MÁXIMO, onde $\Delta(G) = 3$ e G é um grafo grade. Primeiro, vamos apresentar a construção do grafo G' . Multiplique a escala de G por três. Dessa forma, cada aresta originalmente em G se torna em um P_4 . Chame de *vértice original* todo vértice originalmente em G . Nós fazemos essa multiplicação para que todo caminho P no grafo G se torne um caminho induzido em G' se seguirmos os mesmos vértices de P em G' .

Agora, adicione $3 - d(v)$ vizinhos a cada vértice original v em quaisquer posições livres vizinhas a dele na grade. Neste ponto, cada vértice original tem grau três. Chame de *vértice auxiliar* todo vértice que, até esse ponto, tem grau menor do que três. Note que cada vértice auxiliar é adjacente a exatamente um vértice original e cada vértice original é adjacente a exatamente três vértices auxiliares.

Após isso, para cada vértice auxiliar v , adicione um novo vértice adjacente a v da seguinte maneira: se o vértice original vizinho de v está acima dele, adicione um novo vértice a sua direita, se já não houver um vértice nessa posição, e faça-o vizinho de v . Se o vértice original vizinho de v está abaixo dele, adicione um novo vértice a sua esquerda, se já não houver um vértice nessa posição, e faça-o vizinho de v . Se o vértice original vizinho de v está à esquerda dele, adicione um novo vértice abaixo de v , se já não houver um vértice nessa posição, e faça-o vizinho de v . Se o vértice original vizinho de v está à direita dele, adicione um novo vértice acima de v , se já não houver um vértice nessa posição, e faça-o vizinho de v .

A Figura 20 ilustra como um bloco 4x4 fica em G' antes e depois da adição desses vértices.

Depois disso, para cada vértice auxiliar v com grau 2 até aqui, adicione um vértice

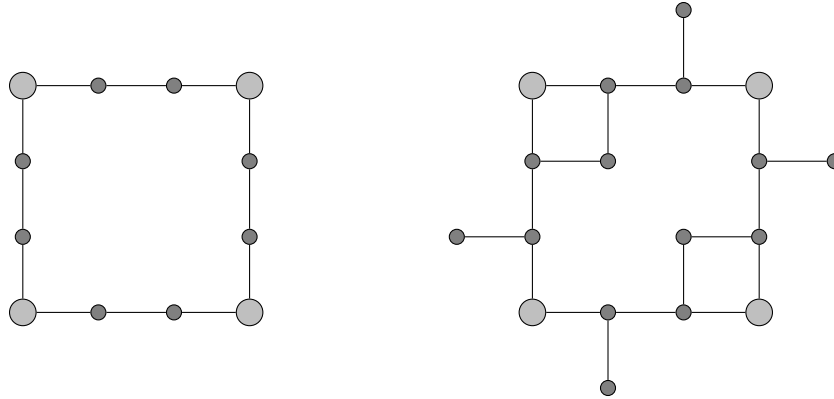


Figura 20 – Bloco 4x4 antes e depois da adição dos vizinhos dos vértices auxiliares.

adjacente a v na posição oposta da que o seu vizinho original está, onde um vizinho original é um vizinho que é um vértice original. Por exemplo, se o vizinho original de v está a sua esquerda, adicione um vértice adjacente a v a sua direita. Assim, temos que todos os vértices originais e auxiliares têm grau três em G' . Fazemos assim, para que todo caminho em G' entre vértices originais que passa somente pelas arestas criadas na multiplicação da grade passe somente por vértices que possuem grau três.

Assim, acabamos a construção de G' . Como G é um grafo grade, então G' também é. Chame de *vértice de canto* todos os vértices de grau dois em G' . Note que cada vértice de canto está à distância dois de exatamente um vértice original e vice-versa, o que acontece pelo fato de cada vértice original ter grau três. Seja f a função bijetora que mapeia os vértices originais aos vértices de canto dos qual ele está à distância dois. A Figura 21 mostra o grafo resultante da redução aplicada ao grafo grade da Figura 19.

Agora, vamos mostrar que G possui um caminho de tamanho pelo menos k se e somente se $t(G') \geq 3k + 2$.

Suponha que G possui um caminho de tamanho pelo menos k . Seja M um caminho em G de tamanho exatamente k . Assim, temos um caminho induzido P em G' de tamanho $3k$, que é o caminho que passa pelos mesmos vértices originais que M passa e na mesma ordem. Assim, temos que P passa apenas por vértices originais e auxiliares e as extremidades de P são vértices originais.

Sejam v e v' as extremidades de P e $f(v') = q'$. Como v é um vértice original, seja w qualquer um dos seus vizinhos auxiliares que não fazem parte de P . Note que, todos os vizinhos de w , exceto o próprio v , não estão em P . Seja r o vizinho auxiliar de v' que está em P . Claramente, existe pelo menos um caminho mínimo I entre r e q' que passa somente por vértices originais e auxiliares tal que o caminho P' formado por $w, P - v, I$ é induzido.

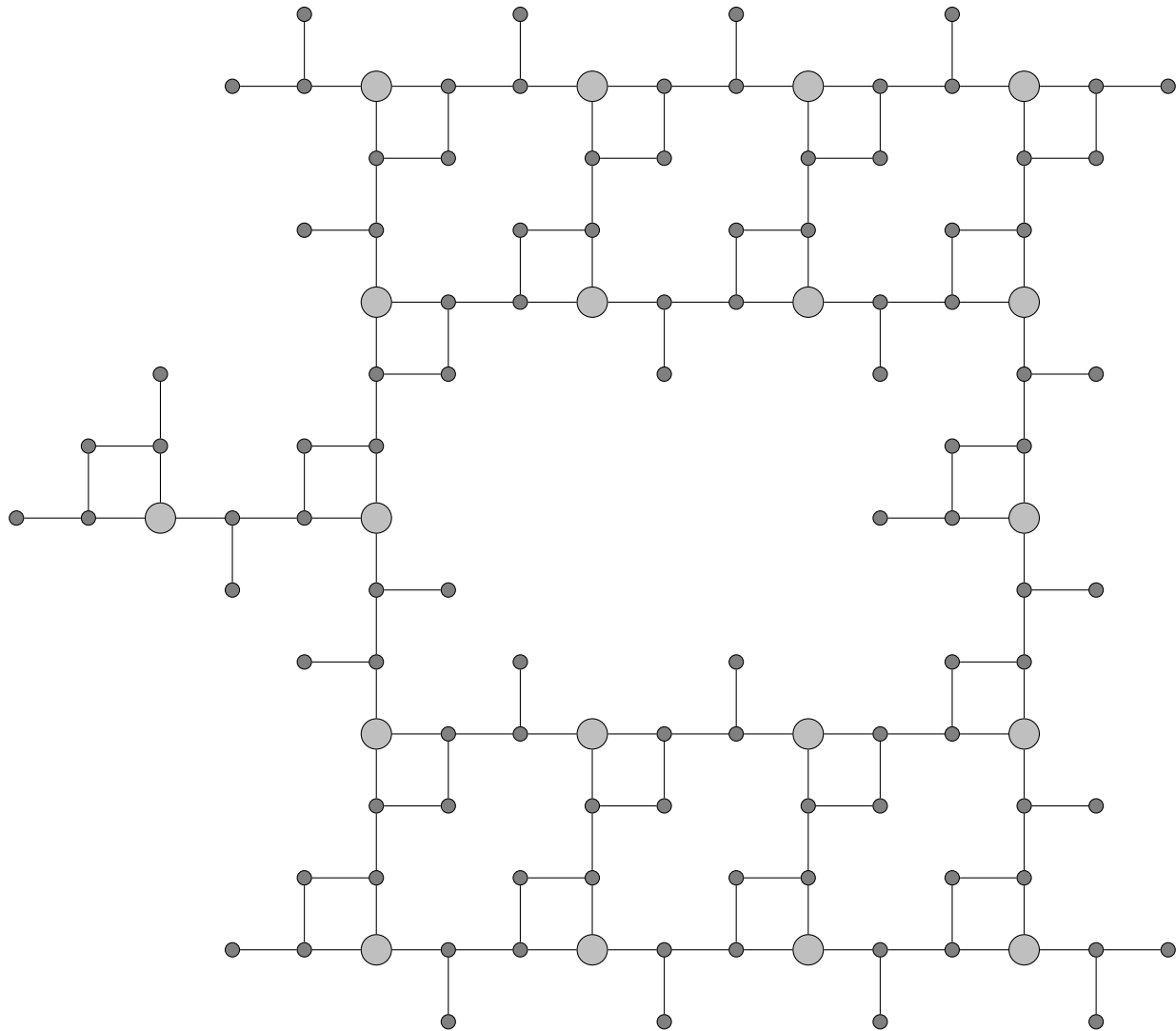


Figura 21 – Grafo grade resultante da aplicação da redução ao grafo da Figura 19.

Temos ainda que I tem tamanho um ou três. Assim, temos que P' é um caminho induzido de tamanho pelo menos $3k + 1$ tal que todos os seus vértices têm grau três, exceto a sua extremidade q' que possui grau 2. Portanto, pelo Lema 3.19, temos que $t(G') \geq 3k + 2$.

Agora, suponha que G' é o grafo resultante da aplicação da nossa redução a G e $t(G') \geq 3k + 2$. Como $t(G') \geq 3k + 2$, invocando o Lema 3.19, temos que G' possui um caminho induzido P tal que ou todos os vértices de P tem grau 3 e P tem tamanho pelo menos $6k + 2$ ou todos os vértices de P tem grau 3, exceto uma de suas extremidades a qual possui grau 2, e P tem tamanho pelo menos $3k + 1$. Note que todos os vértices internos de P são originais ou auxiliares. Note também que, nesse caso, para cada três vértices internos consecutivos em P , dois são auxiliares e um é original.

Primeiro suponha que P tem tamanho pelo menos $6k + 2$. Nesse caso, é fácil ver que há pelo menos $k + 1$ vértices originais em P . Assim, temos um caminho de tamanho pelo menos k em G .

Agora, suponha que P tem tamanho pelo menos $3k + 1$. É o bastante analisar o caso em que P tem tamanho exatamente $3k + 1$. Temos então que há pelo menos k vértices originais em P . Com uma análise de caso cuidadosa, é possível provar que, se não há $k + 1$ vértices originais em P , é possível estender P para que haja. Assim, temos um caminho de tamanho pelo menos k em G . \square

Mesmo que o problema CAMINHO MÁXIMO fosse **NP**-completo em grafos grade sólidos, questão essa que permanece em aberto [111], nós não conseguiríamos mostrar a **NP**-completude do TEMPO MÁXIMO DE INFECÇÃO em grafos grade sólidos usando a redução acima, pois, a redução acima se baseia fortemente no fato de que o grafo possui grau máximo três e ainda que o grafo grade de entrada fosse sólido, o grafo grade resultante não seria. De fato, mesmo que CAMINHO MÁXIMO seja polinomial em grafos grade sólidos, nós fazemos a seguinte conjectura.

Conjectura 4.4. *O problema TEMPO MÁXIMO DE INFECÇÃO é **NP**-completo em grafos grade sólidos.*

4.4 NP-Completo para $k = \Theta(\log n)$ em grafos com grau máximo limitado

Nesta seção, nós demonstramos que TEMPO MÁXIMO DE INFECÇÃO é **NP**-completo mesmo quando o grau máximo do grafo é $\Delta \geq 4$ fixo e $k = \log_{\Delta-2} n$, onde $n = |V(G)|$ e G é o grafo de entrada, e as consequências desse resultado quando assumimos a ETH.

Teorema 4.5. *O problema TEMPO MÁXIMO DE INFECÇÃO é **NP**-completo mesmo em grafos de grau máximo fixo $\Delta \geq 4$ e $k = \Theta(\log n)$.*

Esboço da demonstração. Nós vamos apresentar uma redução polinomial da variação do problema 3-SAT, onde cada variável aparece no máximo em quatro cláusulas, chamada 3,4-SAT. Tovey provou que 3,4-SAT é **NP**-completo [113]. Como $k = \Theta(\log n)$, suponha que $k = c \cdot \log_{\Delta-2} n$ para algum $c > 0$ real.

A redução é semelhante à redução da Seção 4.1. Dada uma fórmula φ em M cláusulas $\mathcal{C} = \{C_1, \dots, C_m\}$ e N variáveis $X = \{x_1, \dots, x_N\}$ como uma instância de 3-SAT, sejam $\ell_{i,1}$,

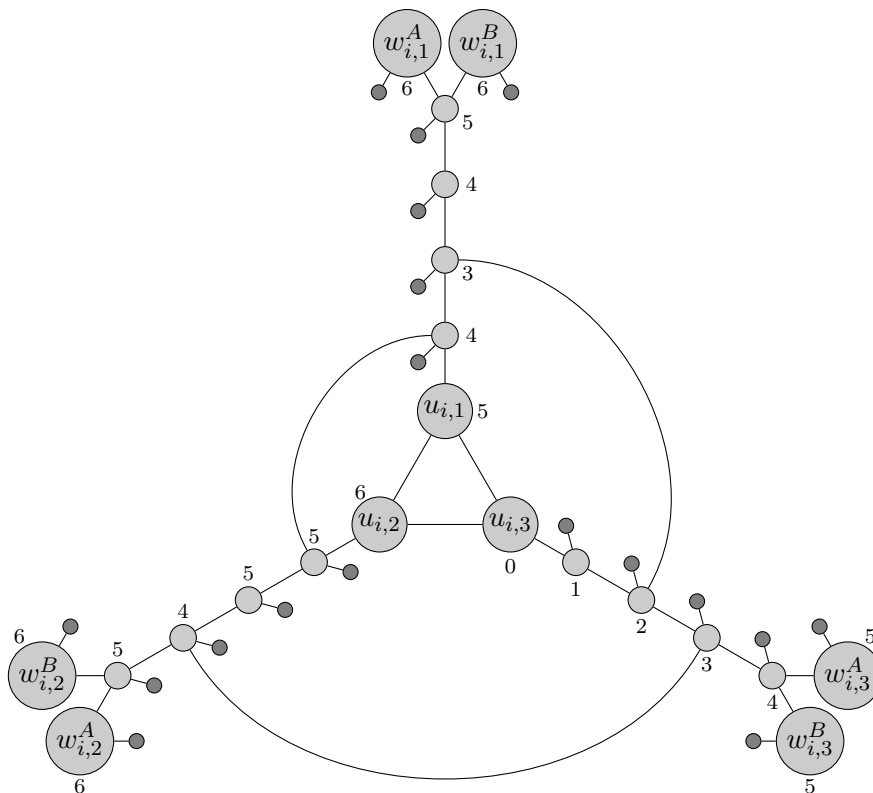


Figura 22 – *Gadget* de cada cláusula C_i com os tempos de infecção dos vértices para o conjunto $\{u_{i,3}\} \cup$ todos os vértices pendentes.

$\ell_{i,2}$ e $\ell_{i,3}$ os três literais da cláusula C_i . Suponha que $N \geq c^2$. Como cada variável está no máximo em quatro cláusulas, temos que $N/3 \leq M \leq 4N/3$.

Primeiro, para cada cláusula C_i , adicione o *gadget* da Figura 22. Depois, para cada par de literais $\ell_{i,a}, \ell_{j,b}$ tais que um é a negação do outro, adicione um vértice $y_{(i,a),(j,b)}$ e adicione um aresta entre ele e, ou o vértice $w_{i,a}^A$, ou o vértice $w_{i,a}^B$, e entre ele e, ou o vértice $w_{j,b}^A$, ou o vértice $w_{j,b}^B$, de modo que os graus de $w_{i,a}^A$, $w_{i,a}^B$, $w_{j,b}^A$ e $w_{j,b}^B$ sejam no máximo quatro. Como cada variável só aparece em quatro cláusulas, isso é possível. Seja Y o conjunto de todos os vértices criados dessa maneira. Temos que $y = |Y| \leq 4N$.

Agora, adicione uma árvore $(\Delta - 2)$ -ária cheia T com raiz z tal que o seu número de folhas f é tal que $y/(\Delta - 2) \leq f < y$ e adicione um vértice pendente adjacente a cada nó dessa árvore. Depois, adicione uma aresta entre cada folha de T e no mínimo um e no máximo $\Delta - 2$ vértices de Y de modo que cada vértice em Y seja adjacente a exatamente uma folha. Seja $t = 2 \cdot |V(T)|$. A Figura 23 ilustra um exemplo de árvore T para $\Delta = 4$ e $y = 6$. Note que $\text{altura}(T) = \lceil \log_{\Delta-2} y \rceil - 1$. Adicionalmente, temos, pela soma da progressão geométrica, que $t \leq 2 \cdot \frac{(\Delta-2)y-1}{\Delta-3} \leq 16N$.

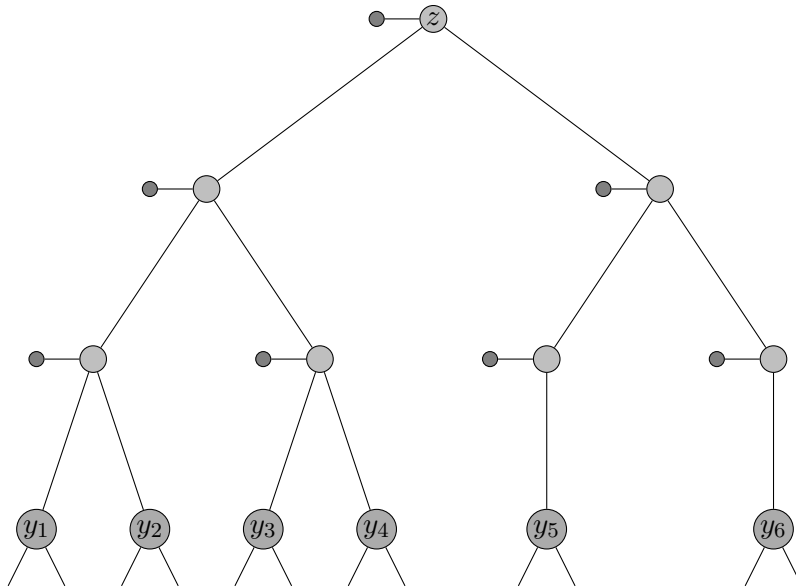


Figura 23 – Exemplo de árvore $(\Delta - 2)$ -ária cheia adicionada a G , para $\Delta = 4$ e $y = 6$.

Seja $r = \lceil \log_{\Delta-2}(4N) \rceil - \lceil \log_{\Delta-2} y \rceil \geq 0$. Se $r = 0$, seja $q = z$. Se $r > 0$, adicione um caminho com r vértices, adicione uma aresta entre uma de suas extremidades e z e seja q a outra extremidade. Adicione um vértice pendente adjacente a cada vértice do caminho.

Seja n' o número de vértices adicionados até agora, ou seja, $n' = 39M + y + t + 2r \leq 80N$. Sejam $c' = \min\{c, 1\}$ e $\alpha = (\Delta - 2)^{-7} \leq 1/128$. Finalmente, adicione $n'' = \lceil (4N/\alpha)^{1/c'} \rceil - n' > 900N$ vértices isolados. Pela nossa construção, o grafo G tem $n = n' + n'' = \lceil (4N/\alpha)^{1/c'} \rceil$ vértices.

Se $c \leq 1$, sejam $q' = q$ e $r' = 0$. Se $c > 1$, seja $r' = \lceil (c - 1) \log_{\Delta-2}(n) \rceil$ e, usando os vértices isolados recém introduzidos, construa um caminho $P_{r'}$, adicione uma aresta entre uma de suas extremidades e q e seja a outra q' e adicione uma aresta de cada vértice desse caminho para um vértice isolado diferente. Note que $2r' \leq n''$, pois $n'' \geq n/2 = \sqrt{n}\sqrt{n}/2 \geq 8\sqrt{N} \log_2 n \geq 8c \log_{\Delta-2} n \geq 2r'$.

Assim, temos um grafo cujo grau de cada vértice é no máximo Δ . Seguindo argumentos similares aos da prova do Teorema 4.2, é possível mostrar que existe um conjunto de envoltória que infecta z em tempo $height(T) + 8$ se e somente se φ é satisfável, o que implica que q' pode ser infectado em tempo $\lceil \log_{\Delta-2}(4N) \rceil + 7 + r'$ se e somente se φ é satisfável.

Como $4N = \alpha n^{c'}$, temos que φ é satisfável se e somente se $t(G) \geq c' \log_{\Delta-2}(n) + r'$. Assim, φ é satisfável se e somente se $t(G) \geq c \log_{\Delta-2} n$. \square

Uma consequência direta desse resultado é a seguinte.

Corolário 4.6. *O problema TEMPO MÁXIMO DE INFECÇÃO não pode ser decidido em tempo $O(2^{\Delta^{\varepsilon \cdot k}} \cdot p(n, \Delta, k))$, para qualquer função polinomial $p(\cdot, \cdot, \cdot)$ e qualquer constante $\varepsilon < 1$, a não ser que a ETH seja falsa.*

Demonstração. Pela prova do Teorema 2.3 em [113], temos que 3,4-SAT, a versão de 3-SAT em que cada variável aparece no máximo em quatro cláusulas, não pode ser decidida em tempo $O(2^{o(N)} \cdot p(N))$, para qualquer função polinomial $p(\cdot)$, a não ser que a ETH seja falsa, onde N é o número de variáveis da instância do 3,4-SAT. Isso acontece porque, na redução da prova de NP-completude do problema 3,4-SAT em [113], temos que $N' \leq N \leq 3M \leq 129M'$, onde N e M são o número de variáveis e cláusulas, respectivamente, da fórmula do problema 3,4-SAT resultante da redução e N' e M' são o número de variáveis e de cláusulas, respectivamente, da fórmula do problema 3-SAT sobre o qual a redução é aplicada.

Assim, se tivéssemos um algoritmo que decide TEMPO MÁXIMO DE INFECÇÃO em tempo $O(2^{\Delta^{\varepsilon \cdot k}} \cdot p(n, \Delta, k))$, para alguma constante $\varepsilon < 1$, poderíamos aplicar a redução polinomial do Teorema 4.4, fazendo $c = 1$, para obter um algoritmo que decide 3,4-SAT em tempo $O(2^{(512N)^{\varepsilon \cdot \log_{\Delta-2} \Delta}} \cdot N^{\Theta(1)})$. Como sempre podemos escolher um valor para Δ de forma que $\varepsilon \cdot \log_{\Delta-2} \Delta < 1$, o tempo de execução do nosso algoritmo seria $O(2^{o(N)} \cdot N^{\Theta(1)})$, o que contrariaria a ETH. \square

Portanto, de acordo com esse corolário, temos que, assumindo a ETH, o algoritmo apresentado na Seção 3.7, que decide o problema TEMPO MÁXIMO DE INFECÇÃO em tempo $O(2^{\Delta^k} \cdot k \cdot \Delta \cdot n^2)$, é ótimo no que diz respeito à constante multiplicativa de k no expoente de Δ .

4.5 W[1]-dificuldade no parâmetro largura em árvore

Nesta seção, nós mostramos que o problema TEMPO MÁXIMO DE INFECÇÃO parametrizado por $tw(G)$ é W[1]-difícil e, portanto, se $\mathbf{FPT} \neq \mathbf{W}[1]$, não existe um algoritmo FPT que o decide. Apresentamos também uma consequência desse resultado quando assumimos a ETH.

Na Seção 2.3, nós mencionamos que o problema CLIQUE MULTICOLORIDA é W[1]-difícil quando parametrizado pela entrada k [51]. Como ele é importante na demonstração do teorema a seguir, vamos defini-lo agora.

O problema CLIQUE MULTICOLORIDA consiste em, dados um grafo $G = (V, E)$, uma função $c : V \rightarrow [k]$ que determina as cores dos vértices de G e um inteiro k , determinar se

existe um clique de tamanho k em G de forma que cada vértice da clique recebe uma cor diferente da função c . Agora, vamos à demonstração do teorema.

Teorema 4.7. *O problema TEMPO MÁXIMO DE INFECÇÃO é $\mathbf{W}[1]$ -difícil quando parametrizado por $tw(G)$.*

Esboço da demonstração. Vamos apresentar uma redução parametrizada a partir do problema CLIQUE MULTICOLORIDA que, como dito na Seção 2.3, é $\mathbf{W}[1]$ -difícil quando parametrizada por k . Essa redução usa um raciocínio similar ao raciocínio da redução em [69], do problema CLIQUE MULTICOLORIDA parametrizado pela entrada k para o problema COLORAÇÃO POR LISTA, do inglês LIST COLORING, parametrizado pela largura em árvore do grafo.

Seja (G, c, k) a instância do problema CLIQUE MULTICOLORIDA que vamos trabalhar. Seja V_i o conjunto de vértices de G que recebem a cor i da função c . Para deixar as figuras mais sucintas e organizadas, daqui em diante, uma aresta de peso n representa a estrutura da Figura 24.

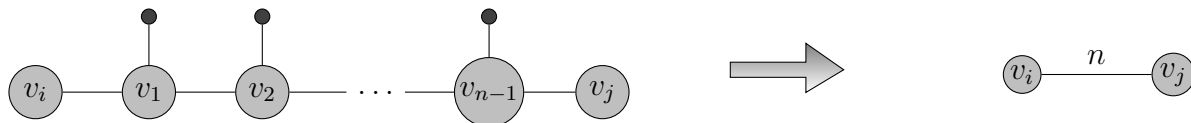


Figura 24 – Uma aresta com peso à direita e a sua estrutura real à esquerda.

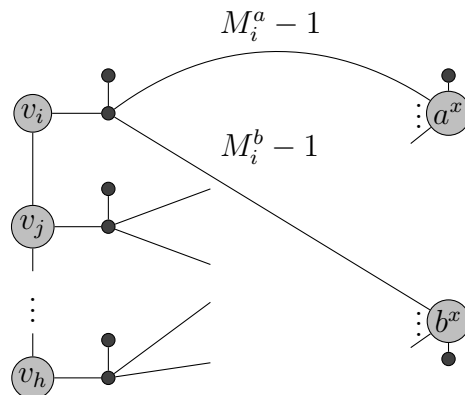


Figura 25 – Exemplo de *gadget* Q^x .

Seja $n = |V(G)|$. Primeiro, rotule de forma arbitrária os vértices de G com os rótulos v_1, v_2, \dots, v_n e defina, para cada $1 \leq i \leq k$, $M_i^a = 2n + 2i - 2$ e $M_i^b = 2n - 2i + 2$. Construa G' da seguinte forma: para cada $1 \leq x \leq k$, adicione o *gadget* Q^x , que vamos chamar de *gadget de cor*, como o da Figura 25, onde v_i, v_j, \dots, v_h são vértices que representam os vértices de V_x com o mesmo rótulo em G . Sejam $T = \bigcup_{1 \leq x \leq k} \{a^x, b^x\}$ e $V^x = \{v_1, v_2, \dots, v_n\} \cap V(Q^x)$.

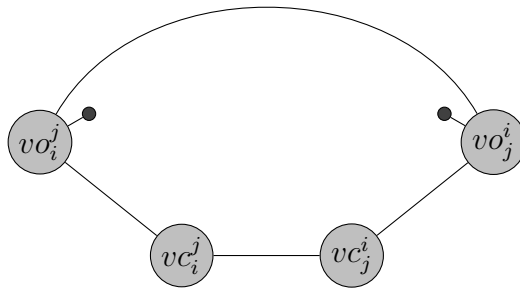


Figura 26 – *Gadget* que colocamos entre *gadgets* de cor sempre que **não** houver aresta entre os vértices v_i e v_j .

Note que os vértices em V^x foram um conjunto coconvexo. Para cada par de vértices $v_i \in V_x$ e $v_j \in V_y$ **não** adjacentes tais que $x \neq y$, adicione um *gadget* de escolha como o da Figura 26. Adicione ainda quatro vértices $a_i^j, b_i^j, a_j^i, b_j^i$ e adicione as seguintes arestas: $\{a_i^j, a^x\}$, $\{b_i^j, b^x\}$, $\{a_j^i, a^y\}$ e $\{b_j^i, b^y\}$. Depois disso, adicione uma aresta entre o vértice $v o_i^j$ e ambos os vértices a_i^j e b_i^j com pesos M_i^a e M_i^b , respectivamente, e entre o vértice $v o_j^i$ e ambos os vértices a_j^i e b_j^i com pesos M_j^a e M_j^b respectivamente. Seja $V C_i^j = V C_j^i = \{v c_i^j, v c_j^i\}$. Novamente, note que os vértices em $V C_i^j$ foram um conjunto coconvexo.

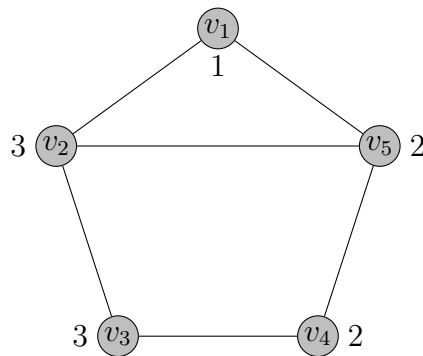


Figura 27 – Grafo G .

Seja E o número de vértices adicionados até esse ponto, incluindo os vértices escondidos nas arestas com pesos. Para cada vértice a_i^j e b_i^j , adicione um vértice l_i^j , faça-o vizinho de a_i^j através de uma aresta com peso $E - M_i^a - 2$ e de b_i^j através de uma aresta com peso $E - M_i^b - 2$. Seja L o conjunto de vértices l_i^j adicionados dessa forma. Adicione um vértice p e faça-o vizinho de cada vértice em L . Crie mais um vértice e faça-o vizinho de p . Seja D o número de vértices que adicionamos até esse ponto. Finalmente, adicione os vértices z e z' , adicione a aresta $\{z, z'\}$ e a aresta $\{p, z\}$ com peso D .

A Figura 29 ilustra esse último passo da construção. Claramente G' pode ser construído em tempo polinomial. A Figura 28 e 30 ilustram o grafo resultante da construção aplicada ao

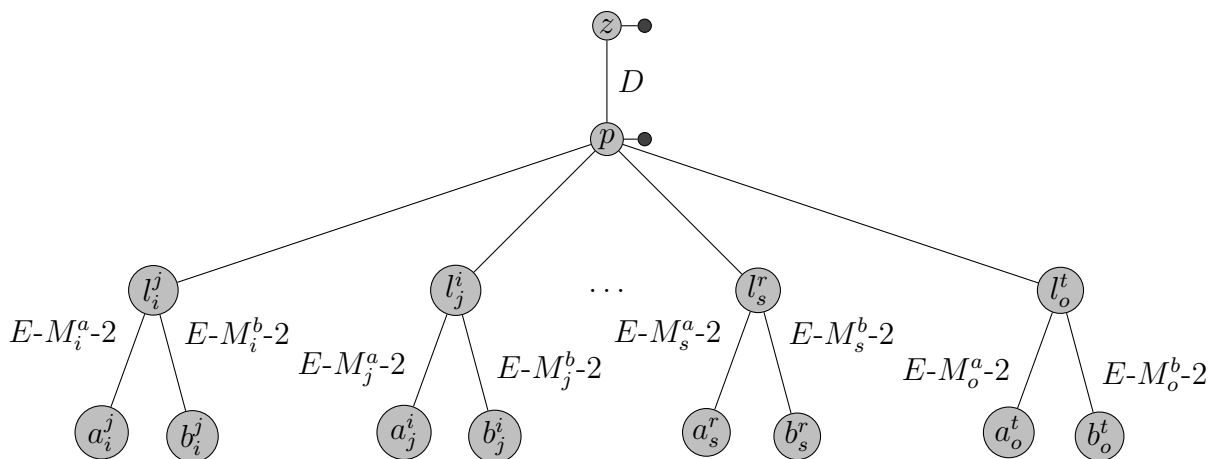


Figura 29 – Ligações dos vértices em L

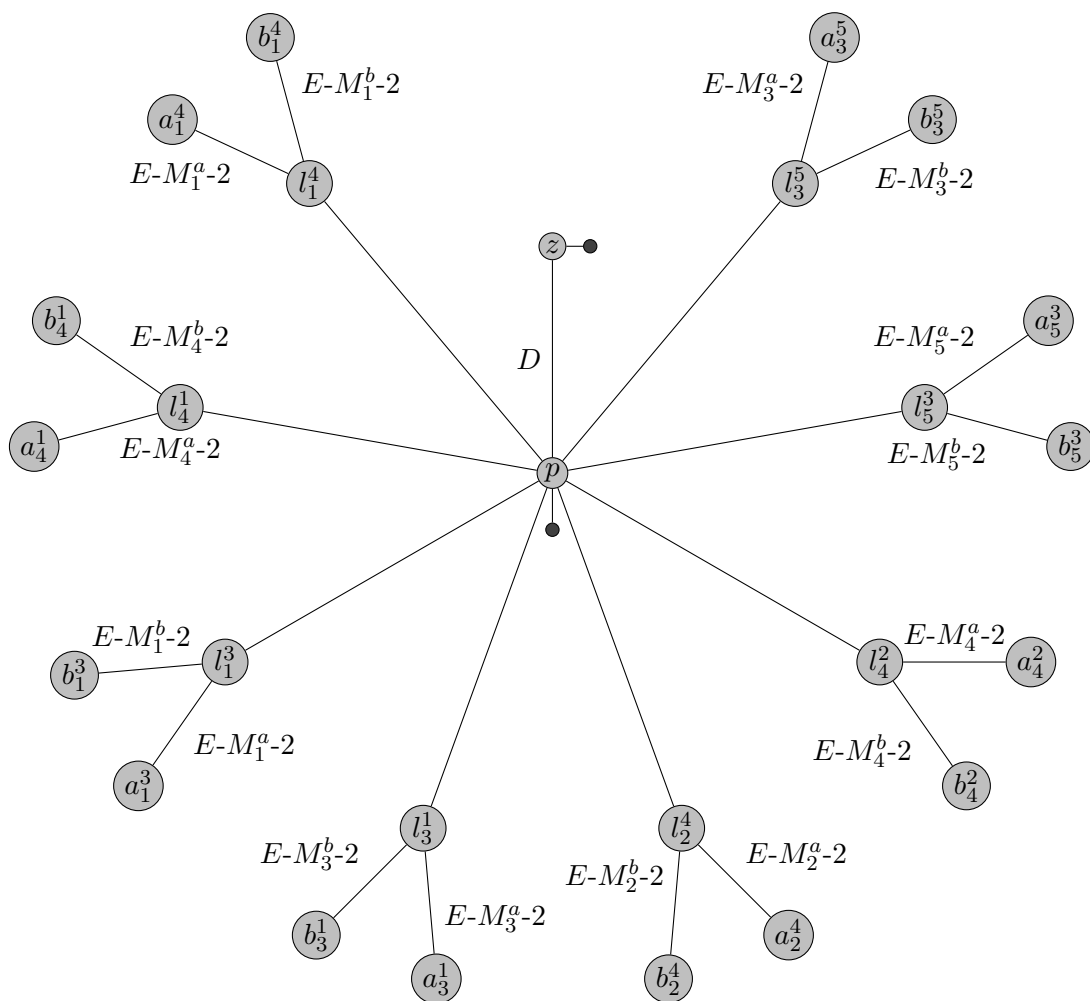


Figura 30 – Segunda parte do grafo G' resultante da aplicação da redução no grafo da Figura 27.

Claramente, S é um conjunto de envoltória. Adicionalmente, temos que cada vértice a^x é infectado em tempo $M_{v_{c_x}}^a$ e cada vértice b^x em tempo $M_{v_{c_x}}^b$ por S . Assim, pelas propriedades dos inteiros M_i^a e M_i^b , temos que ou S infecta a_i^j em tempo pelo menos $M_i^a + 2$ ou S infecta b_i^j em tempo pelo menos $M_i^b + 2$. Portanto, temos que todos os vértices em L são infectados por S em tempo pelo menos E . Logo, temos que S infecta z em tempo $D + E + 1$ e, portanto, $t(G) \geq D + E + 1$.

Agora, assumamos que não há uma clique multicolorida de tamanho k em G . Como, para um conjunto S ser um conjunto de envoltória, é necessário e suficiente escolher para fazer parte dele todos os vértices pendentos mais um vértice de cada conjunto V^x e VC_i^j , para provar o que $t(G) < D + E + 1$, é o bastante provar que cada conjunto de envoltória construído dessa forma infecta G em tempo no máximo $D + E$. Seja então S tal conjunto de envoltória arbitrário. É possível provar que z é o último vértice a ser infectado por S .

Como não há uma clique multicolorida de tamanho k em G , então existem vértices a_i^j e b_i^j que são infectados por S no máximo nos tempos $M_i^a + 1$ e $M_i^b + 1$ respectivamente. Assim, temos que existe um vértice em L infectado por S em tempo no máximo $E - 1$. Logo, temos que z é infectado por S em tempo no máximo $E + D$. Portanto, temos que $t(G) < D + E + 1$.

Adicionalmente, temos que o conjunto $S = T \cup \{p\}$ é um separador de G' e claramente, as componentes conexas de $G' - S$ têm largura em árvore no máximo 2. Portanto, temos que $tw(G') \leq |S| + 3 - 1 = 2k + 3$. \square

Como $tw(G') \leq 2k + 3$ no grafo G' da prova anterior, temos o seguinte corolário do teorema anterior.

Corolário 4.8. *O problema TEMPO MÁXIMO DE INFECÇÃO não pode ser decidido em tempo $f(tw(G)) \cdot n^{o(tw(G))}$, onde $n = |V(G)|$, para qualquer função computável f , a não ser que a ETH seja falsa.*

Demonstração. O corolário segue do fato de que, se tivéssemos um algoritmo que decide TEMPO MÁXIMO DE INFECÇÃO em tempo $f(tw(G)) \cdot n^{g(tw(G))}$ para algumas funções f e $g \in o(tw(G))$, poderíamos usar a redução do Teorema 4.7, que é um algoritmo FPT, para, a partir de um instância (G, c, k) do problema CLIQUE MULTICOLORIDA, gerar uma instância equivalente para o problema TEMPO MÁXIMO DE INFECÇÃO e, depois, decidir tal instância em tempo $O(f(2k + 3) \cdot n^{5 \cdot g(2k + 3)}) = f'(k) \cdot n^{o(k)}$ para alguma função f' em k . Assim, pela equivalência das instâncias, teríamos um algoritmo para decidir o problema CLIQUE MULTICOLORIDA em tempo $f'(k) \cdot n^{o(k)}$, o que contraria a ETH. \square

Assumindo a ETH, esse resultado nos mostra que o algoritmo XP da Seção 3.8, que decide o problema TEMPO MÁXIMO DE INFECÇÃO parametrizado por $tw(G)$ em $O(50^{tw(G)+1} \cdot tw(G)^3 \cdot n^{tw(G)+2})$, é assintoticamente ótimo no que diz respeito ao expoente de n .

5 RESULTADOS PARAMETRIZADOS PARA O NÚMERO DE ENVOLTÓRIA

Neste capítulo, serão apresentados alguns resultados parametrizados para o problema NÚMERO DE ENVOLTÓRIA nas convexidades P_3 e geodésica. O número de envoltória de um grafo G na convexidade C , denotado por $hn_C(G)$, é a cardinalidade do menor conjunto de envoltória de G . Assim, o problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE C é o problema de decisão definido da seguinte forma:

NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE C

Entrada: Um grafo G e um inteiro k .

Saída: Sim, se $hn_C(G) \leq k$. Não, caso contrário.

Em outras palavras, o problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE C consiste em determinar, dados um grafo G e um inteiro k , se existe um conjunto de envoltória de G na convexidade C com tamanho no máximo k . Neste capítulo, abreviamos o nome do problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE P_3 para NÚMERO DE ENVOLTÓRIA P_3 e abreviamos o nome do problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE GEODÉSICA para NÚMERO DE ENVOLTÓRIA GEODÉTICO.

Na Seção 5.1, nós mostramos que o problema NÚMERO DE ENVOLTÓRIA P_3 é $\mathbf{W}[1]$ -difícil quando parametrizado por k mesmo em grafos livres de K_3 e com grau máximo seis. Também provamos que o problema NÚMERO DE ENVOLTÓRIA GEODÉTICO é $\mathbf{W}[1]$ -difícil quando parametrizado por k mesmo em grafos com diâmetro dois. O resultado de $\mathbf{W}[1]$ -dificuldade para o problema NÚMERO DE ENVOLTÓRIA GEODÉTICO foi motivado pelo trabalho de Araújo *et al.* em [7].

Na Seção 5.2, nós mostramos que o problema NÚMERO DE ENVOLTÓRIA GEODÉTICO é $\mathbf{W}[1]$ -difícil mesmo quando parametrizado por $tw(G) + k$, ao contrário do problema NÚMERO DE ENVOLTÓRIA P_3 , o qual é \mathbf{FPT} quando parametrizado por $tw(G)$ [22].

Apresentamos as consequências dos três resultados anteriores quando aceitamos a Hipótese do Tempo Exponencial na Seção 5.3.

Finalmente, na Seção 5.4, nós apresentamos um algoritmo que computa $hn_g(G)$ em tempo $O((tw + 1)^{tw+6} \cdot n^{2tw+5})$, provando assim que NÚMERO DE ENVOLTÓRIA GEODÉTICO parametrizado somente por $tw(G)$ está em \mathbf{XP} .

Neste capítulo, encontram-se contribuições que ainda estão em fase de elaboração e constituirão um trabalho conjunto entre Júlio Araújo, Mamadou Kanté, Thiago Marcilon e Rudini Sampaio, a ser submetido em breve.

A seguir, vamos apresentar o enunciado e o esboço da demonstração de cada lema e teorema, exceto alguns da última seção por serem bastante longos e técnicos. As demonstrações completas dos resultados neste capítulo se encontram no Apêndice C.

5.1 $W[1]$ -dificuldade no parâmetro k nas convexidades P_3 e Geodésica

Nesta seção, nós apresentamos a $W[1]$ -dificuldade dos problemas NÚMERO DE ENVOLTÓRIA P_3 e NÚMERO DE ENVOLTÓRIA GEODÉTICO quando parametrizados por k , respondendo a questão proposta por Araújo *et al.* em [7] sobre a existência de um algoritmo FPT que decide o problema NÚMERO DE ENVOLTÓRIA GEODÉTICO parametrizado pelo tamanho da solução k .

Antes de apresentar os resultados desta seção, vamos introduzir algumas definições. Defina como uma *biárvore estritamente binária* (BAEB) todo grafo que pode ser construído seguindo o seguinte procedimento:

1. crie uma árvore estritamente binária T ;
2. enquanto houver nós em T que não possuem um gêmeo falso no grafo, escolha um desses vértices, chame-o de v e adicione no grafo um gêmeo falso de v .

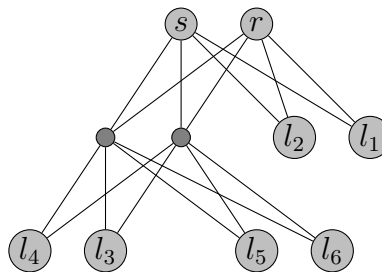


Figura 31 – BAEB com biraiz $\{r, s\}$ e três bifolhas $\{l_1, l_2\}$, $\{l_3, l_4\}$ e $\{l_5, l_6\}$.

A Figura 31 ilustra um exemplo de BAEB. Em uma BAEB, todo nó possui um gêmeo falso. Nós definimos a *biraiz* de uma BAEB como o conjunto dos dois vértices composto pela raiz da árvore estritamente binária original e o seu gêmeo falso. Similarmente, uma *bifolha* de uma BAEB é o conjunto de dois vértices composto por uma folha da árvore estritamente

binária original e o seu gêmeo falso. Note que uma BAEB tem grau máximo seis e é livre de K_3 .

Teorema 5.1. *O problema NÚMERO DE ENVOLTÓRIA P_3 é $\mathbf{W}[1]$ -difícil quando parametrizado por k mesmo em grafos livres de K_3 com grau máximo seis.*

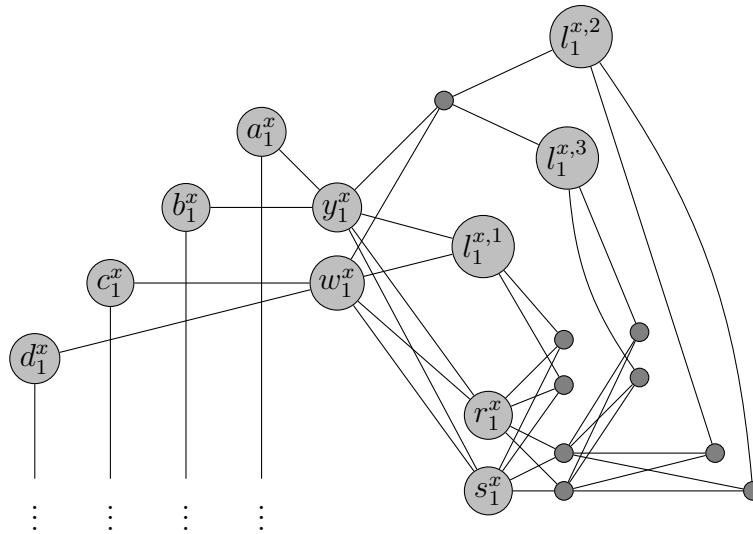
Esboço da demonstração. Vamos mostrar que há uma redução parametrizada do problema CLIQUE MULTICOLORIDA parametrizado por k , definido na Seção 4.5, para o problema NÚMERO DE ENVOLTÓRIA P_3 parametrizado por k mesmo em grafos livres de K_3 com grau máximo seis. Na verdade, nós vamos trabalhar com uma modificação do problema CLIQUE MULTICOLORIDA na qual o conjunto de vértices da mesma cor no grafo de entrada formam um conjunto independente, o grau mínimo do grafo é dois e, para todo par de cores x e y , existe um vértice com a cor x e outro com a cor y que são vizinhos. Não é difícil ver que essas restrições não alteram a dificuldade do problema.

Seja $(G = (V, E), c, k)$ uma instância de CLIQUE MULTICOLORIDA. Sejam $n = |V|$ e $m = |E|$. Sejam também $V_x = \{v_1^x, v_2^x, \dots, v_{t_x}^x\}$ o conjunto de vértices em G com a cor x , onde $t_x = |V_x|$, e $d_i^x = d_G(v_i^x)$.

Primeiro, para cada cor $1 \leq x \leq k$, construa um *gadget de cor* da seguinte forma:

1. Crie quatro caminhos $a_1^x, a_2^x, \dots, a_{t_x}^x; b_1^x, b_2^x, \dots, b_{t_x}^x; c_1^x, c_2^x, \dots, c_{t_x}^x$ e $d_1^x, d_2^x, \dots, d_{t_x}^x$ e sejam A_x, B_x, C_x e D_x o conjunto de vértices de cada caminho, respectivamente;
2. Para cada vértice v_p^x em V_x , adicione os vértices y_p^x e w_p^x . Depois disso, adicione arestas entre a_p^x e y_p^x , b_p^x e y_p^x , c_p^x e w_p^x , e d_p^x e w_p^x .
3. Para cada vértice v_p^x em V_x , adicione uma árvore estritamente binária enraizada no vértice y_p^x com d_p^x folhas $L_p^x = \{l_p^{x,1}, l_p^{x,2}, \dots, l_p^{x,d_p^x}\}$. Adicione uma aresta entre w_p^x e ambos os filhos de y_p^x . Seja T_p^x o conjunto de todos os nós dessa árvore juntamente com o vértice w_p^x ;
4. Para cada vértice v_p^x em V_x , adicione uma BAEB com birraiz $\{r_p^x, s_p^x\}$ e com d_p^x bifolhas. Ligue os vértices r_p^x e s_p^x à ambos os vértices y_p^x e w_p^x . Seja Q_p^x essa BAEB.
5. Finalmente, para cada bifolha $\{l, l'\}$ de Q_p^x , faça os vértices l e l' vizinhos de um vértice em L_p^x de forma que cada vértice em L_p^x seja vizinho de apenas dois vértices em Q_p^x .

A Figura 32 ilustra um *gadget* parcial para a cor x e $d_3^x = 3$. Sejam $L^x = \bigcup_{j=1}^{t_x} L_j^x$ e $L = \bigcup_{x=1}^k L^x$.


 Figura 32 – *Gadget de cor parcial para a cor x e $d_1^x = 3$.*

Finalmente, para cada par de cores diferentes $1 \leq x, x' \leq k$, construa o *gadget* de aresta da seguinte maneira:

1. Para cada aresta entre um par de vértices v_p^x e $v_{p'}^{x'}$ em G , adicione os vértices $e_{p,p'}^{x,x'}$ e $f_{p,p'}^{x,x'}$ e faça-os vizinhos ao mesmo vértice em L_p^x e ao o mesmo vértice em $L_{p'}^{x'}$. Isso deve ser feito de forma que cada vértice $e_{p,p'}^{x,x'}$ e $f_{p,p'}^{x,x'}$ em todos os *gadgets* de aresta sejam vizinhos de um único vértice em L_p^x e a um único vértice em $L_{p'}^{x'}$, ou seja, cada vértice em L deve ser vizinho de exatamente um par de vértices $e_{p,p'}^{x,x'}$ e $f_{p,p'}^{x,x'}$ em algum *gadget* de aresta.
2. Adicione uma BAEB qualquer de modo que suas bifolhas sejam os vértices $\{e_{p,p'}^{x,x'}, f_{p,p'}^{x,x'}\}$ do *gadget* de aresta.

Seja $E_{x,x'}$ a BAEB no *gadget* de aresta das cores x e x' . A Figura 33 ilustra um *gadget* de aresta com arestas entre os vértices v_1^x e $v_3^{x'}$, v_1^x e $v_4^{x'}$, e v_2^x e $v_3^{x'}$.

O grafo resultante dessa construção é livre de K_3 , tem grau máximo seis e tem $2n + 16m - 2\binom{k}{2}$ vértices. A construção em si pode ser executada em tempo $O(m + n + k^2)$.

A função de cada *gadget* de cor é fazer com que o conjunto $\{y_p^x, w_p^x\}$ infecte todos os vértices em L_p^x e vice-versa. Por isso, adicionamos a BAEB, para que o conjunto $\{y_p^x, w_p^x\}$ infectasse todos os vértices em L_p^x . Por sua vez, a função de cada *gadget* de aresta das cores x e x' é fazer com que qualquer par de vértices $l_p^{x,q}$ e $l_{p'}^{x',q'}$ tais que $\{v_p^x, v_{p'}^{x'}\} \in E(G)$ infecte todos os vértices em $L^x \cup L^{x'}$ que sejam vizinhos de algum vértice em $V(E_{x,x'})$.

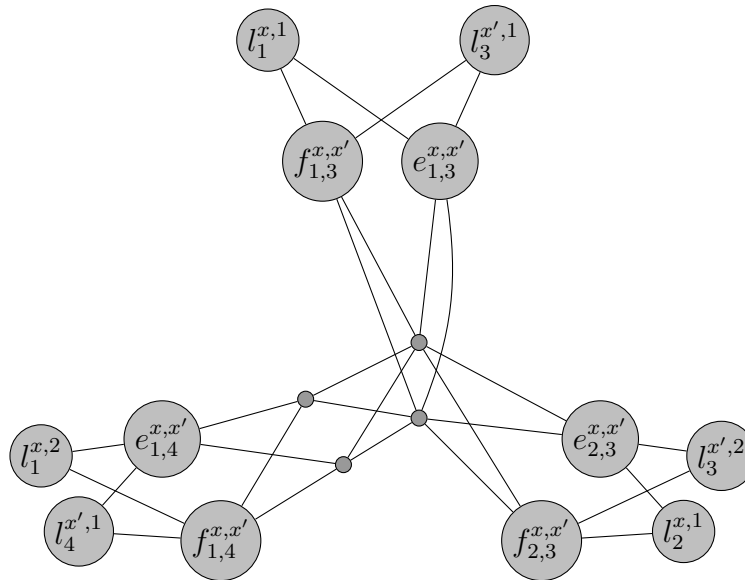


Figura 33 – *Gadget* de aresta das cores x e x' com arestas entre os vértices v_1^x e $v_3^{x'}$, v_1^x e $v_4^{x'}$, e v_2^x e $v_3^{x'}$.

Seja G' o grafo resultante da redução aplicada ao grafo G . Vamos mostrar que G possui uma clique multicolorida de tamanho k se e somente se $hn_{P_3}(G') \leq 4k$.

Suponha que $C = \{v_{c_1}^1, v_{c_2}^2, \dots, v_{c_k}^k\}$ é uma clique multicolorida de G , onde $v_{c_x}^x$ tem cor x . Seja $S = \bigcup_{x=1}^k \{a_{c_x}^x, b_{c_x}^x, c_{c_x}^x, d_{c_x}^x\}$. Temos que S infecta $y_{c_x}^x$ e $w_{c_x}^x$, para cada $1 \leq x \leq k$. O conjunto S também infecta os vértices em $T_{c_x}^x \cup V(Q_{c_x}^x)$. Como C é uma clique, temos que os vértices em $E_{x,x'}$, para todo $1 \leq x \neq x' \leq k$, são infectados por S , o que implica que os vértices em L são infectados por S . Finalmente, temos que S infecta todos os *gadgets* de cor e , assim, o grafo inteiro é infectado por S . Portanto, temos que $hn_{P_3}(G') \leq |S| = 4k$.

Agora, suponha que não há um clique multicolorida de tamanho k em G . Note que os conjuntos A_x, B_x, C_x, D_x , para todo $1 \leq x \leq k$, são conjuntos coconvexos disjuntos dois a dois e, portanto, qualquer conjunto de envoltória deve ter tamanho pelo menos $4k$ e qualquer conjunto de envoltória de tamanho $4k$ deve consistir em exatamente um vértice de cada um desses conjuntos.

Seja S um conjunto arbitrário que consiste em exatamente um vértice de cada um desses conjuntos. É possível provar, baseando-se no fato que G não possui clique multicolorida de tamanho k , que existe pelo menos um *gadget* de aresta não infectado por S , o que implica que não existe conjunto de envoltória de tamanho $4k$. Portanto, todos os conjuntos de envoltória devem ter tamanho pelo menos $4k + 1$ o que implica que $hn_{P_3}(G') > 4k$. \square

É importante ressaltar que o resultado por Nichterlein *et al.* em [94], onde eles mostram

que NÚMERO DE ENVOLTÓRIA é $\mathbf{W}[2]$ -difícil quando parametrizado pela entrada k mesmo em grafos bipartidos não implica no Teorema 5.1, pois, no Teorema 5.1, o grau máximo do grafo é seis, enquanto, no resultado de Nichterlein *et al.*, o grau máximo do grafo não é limitado.

Em [8], Araújo, Sampaio and Szwarcfiter introduziram a convexidade P_3^* . A convexidade P_3^* é uma convexidade de intervalo em grafos tal que sua função de intervalo $I[x, y]$ retorna todos os vértice em um caminho P_3 induzido entre os vértices x e y . No Teorema 3.1 de [6], eles declaram, dentre outras coisas, que para todo m e grafo G , $hn_g(G) = hn_{P_3^*}(G')$, onde $G' = G + K_m$ e $G_1 + G_2 = \{V(G_1) \cup V(G_2), E(G_1) \cup E(G_2) \cup \{\{u, v\} \mid u \in V(G_1), v \in V(G_2)\}\}$.

Também temos, em grafos G livres de K_3 , que $hn_{P_3}(G') = hn_{P_3^*}(G')$, pois todo caminho P_3 é um caminho induzido. Assim, temos o seguinte lema.

Lema 5.2 (Consequência do Teorema 3.1 em [8]). *Sejam $G = (V, E)$ um grafo livre de K_3 tal que $|V| \geq 3$ e $G' = G + K_1$. Então $hn_g(G) = hn_{P_3}(G')$.*

Portanto, como consequência dos dois resultados anteriores, temos o seguinte teorema.

Teorema 5.3. *O problema NÚMERO DE ENVOLTÓRIA GEODÉTICO é $\mathbf{W}[1]$ -difícil quando parametrizado por k mesmo em grafos de diâmetro dois.*

Demonstração. Sejam (G, k) uma instância do problema NÚMERO DE ENVOLTÓRIA P_3 , onde G é um grafo livre de K_3 . Seja ainda $H = G + K_1$ e, por isso, temos que H tem diâmetro dois. Como G é livre de K_3 , podemos invocar o Lema 5.2 para obter que $hn_g(H) = hn_{P_3}(G)$. Portanto, temos que $hn_{P_3}(G) \leq k$ se e somente se $hn_g(H) \leq k$.

Assim, nós temos uma redução parametrizada de NÚMERO DE ENVOLTÓRIA P_3 parametrizado por k em grafos livres de K_3 , que é $\mathbf{W}[1]$ -difícil pelo Teorema 5.1, para NÚMERO DE ENVOLTÓRIA GEODÉTICO parametrizado por k em grafos de diâmetro dois, o que implica que NÚMERO DE ENVOLTÓRIA GEODÉTICO parametrizado por k é $\mathbf{W}[1]$ -difícil mesmo em grafos de diâmetro dois. \square

O Teorema 5.3 responde a pergunta proposta por Araújo *et al.* em [7], pois, se assumirmos que $\mathbf{W}[1] \neq \mathbf{FPT}$, temos que não há algoritmo FPT para o problema NÚMERO DE ENVOLTÓRIA GEODÉTICO parametrizado no tamanho da solução k .

5.2 $W[1]$ -dificuldade no parâmetro $tw(G) + k$ na convexidade Geodésica

Nesta seção, nós vamos provar que NÚMERO DE ENVOLTÓRIA GEODÉTICO parametrizado por $tw(G) + k$ é $\mathbf{W}[1]$ -difícil através de uma redução parametrizada a partir do problema REVESTIMENTO DE GRADE parametrizado por sua entrada k . Na Seção 2.3, nós mencionamos que o problema REVESTIMENTO DE GRADE parametrizado por k é $\mathbf{W}[1]$ -difícil [51]. Como ele é importante na demonstração do próximo teorema, vamos defini-lo agora.

O problema REVESTIMENTO DE GRADE consiste em, dados inteiros positivos k e n e uma família \mathcal{S} de k^2 conjuntos de pares ordenados $S_{i,j} \subseteq [n]^2$, para $1 \leq i, j \leq k$, determinar se é possível achar um par $s_{i,j}$ de cada conjunto $S_{i,j}$ de modo que, se $s_{i,j} = (a, b)$ e $s_{i+1,j} = (a', b')$, então $a = a'$ e, se $s_{i,j} = (a, b)$ e $s_{i,j+1} = (a', b')$, então $b = b'$. Na Figura 34, temos uma instância do problema e uma possível solução em negrito.

	$(2, *)$	$(3, *)$	$(5, *)$
	S_{1,1} :	S_{1,2} :	S_{1,3} :
	(2, 1)	(3, 1)	(2, 4)
(*, 1)	(2, 2)	(1, 5)	(5, 1)
	(3, 3)	(4, 5)	(3, 4)
	S_{2,1} :	S_{2,2} :	S_{2,3} :
	(2, 4)	(1, 1)	(1, 4)
(*, 4)	(5, 2)	(3, 3)	(5, 4)
		(3, 4)	(2, 4)
	S_{3,1} :	S_{3,2} :	S_{3,3} :
	(2, 5)	(1, 1)	(2, 2)
(*, 5)	(4, 5)	(3, 5)	(5, 5)
	(5, 3)	(3, 3)	

Figura 34 – Instância do problema REVESTIMENTO DE GRADE com $k = 3$ e $n = 5$.

Agora, vamos ao resultado principal da seção.

Teorema 5.4. *O problema NÚMERO DE ENVOLTÓRIA GEODÉTICO é $\mathbf{W}[1]$ -difícil quando parametrizado por $tw(G) + k$.*

Esboço da demonstração. Vamos apresentar uma redução parametrizada do problema REVESTIMENTO DE GRADE no parâmetro k para NÚMERO DE ENVOLTÓRIA GEODÉTICO parametrizado por $tw(G) + k$.

Seja (k, n, \mathcal{S}) uma instância de REVESTIMENTO DE GRADE. Seja $Q = \bigcup_{S \in \mathcal{S}} \bigcup_{(x,y) \in S} \{x, y\}$, ou seja, Q é o conjunto de todos os diferentes inteiros positivos que aparecem em algum par ordenado de algum conjunto em \mathcal{S} , e seja $n' = |Q|$. Seja ainda $\pi : Q \rightarrow [n']$ uma função injetora qualquer. Note que $\sum_{S \in \mathcal{S}} |S| \geq |Q| = n'$.

Defina $A_i = 6n' + i$ e $B_i = 6n' - i$, para cada $1 \leq i \leq n'$. Note que, para todo $1 \leq i, j \leq n'$ tal que $i \neq j$, temos que $A_i + B_i = A_j + B_j = 12n'$ e $A_i - B_i \neq A_j - B_j$.

Sejam I a função de intervalo da convexidade geodésica e $f(a, b) = k(a-1)+b$. Dizemos que dois pares ordenados (i, j) e (i', j') são *vizinhos* se e somente se $|i - i'| + |j - j'| = 1$. Dizemos que um par ordenado (i, j) é um *par de canto* se e somente se $i \in \{1, k\}$ e $j \in \{1, k\}$. Finalmente, dizemos que um par ordenado (i, j) é um *par de fronteira* se e somente se ele não é um par de canto e, ou $i \in \{1, k\}$, ou $j \in \{1, k\}$.

A fim de não comprometer a clareza das figuras, vamos fazer com que um caminho P_{n+1} seja ilustrado nas figuras como uma aresta com peso n entre as extremidades do caminho. Assim, uma aresta com peso 1 entre dois vértices representa apenas uma aresta entre esses vértices. Nas figuras, os pesos das arestas serão apresentados como números perto das mesmas. Como a maioria das arestas nas figuras têm peso $A_{n'}$, ainda com intuito de melhorar a clareza das figuras, nós fazemos com que uma aresta sem nenhum número perto para indicar o seu peso represente uma aresta com peso $A_{n'}$.

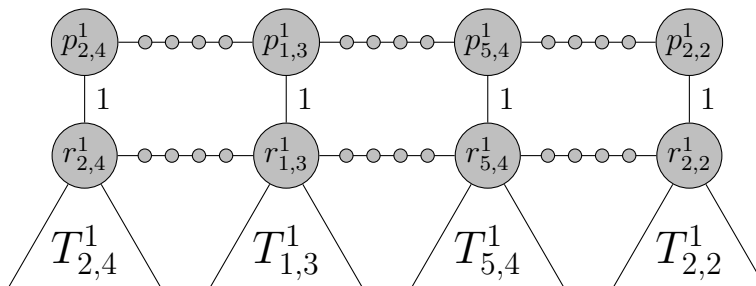


Figura 35 – Caminhos $P_{1,1}$ e $R_{1,1}$ para $S_{1,1} = \{(2, 4), (1, 3), (5, 4), (2, 2)\}$.

Agora, vamos descrever a construção do grafo G' , que é parte da instância do problema NÚMERO DE ENVOLTÓRIA GEODÉTICO. Primeiro, adicione vértices u , z e z' e adicione a aresta $\{z', z\}$. Para cada $1 \leq i, j \leq k$, construa o *gadget de conjunto* $G_{i,j}$ da seguinte forma. Sejam $f = f(i, j)$, $s = |S_{i,j}|$ e $S_{i,j} = \{(x_1, y_1), (x_2, y_2), \dots, (x_s, y_s)\}$:

1. Adicione dois caminhos $P_f = p_{x_1, y_1}^f, p_{x_2, y_2}^f, \dots, p_{x_s, y_s}^f$ e $R_f = r_{x_1, y_1}^f, r_{x_2, y_2}^f, \dots, r_{x_s, y_s}^f$ de tamanho s . Então, subdivida cada aresta $5A_{n'} - 1$ vezes em cada um dos caminhos. Ligue cada vértice $p_{x,y}^f \in P_f$ ao vértice $r_{x,y}^f \in R_f$. A Figura 35 possui um exemplo de caminhos P_1 e R_1 para $S_{1,1} = \{(2, 4), (1, 3), (5, 4), (2, 2)\}$;

2. Para cada $r_{x,y}^f \in R_f$, adicione uma árvore binária completa com quatro folhas $L_{x,y}^f = \{l_{x,y}^{f,e}, l_{x,y}^{f,w}, l_{x,y}^{f,n}, l_{x,y}^{f,s}\}$ e três nós internos $z_{x,y}^{f,0}, z_{x,y}^{f,1}$ e $r_{x,y}^f$, de forma que $l_{x,y}^{f,e}$ e $l_{x,y}^{f,w}$ são filhos de $z_{x,y}^{f,0}$, $l_{x,y}^{f,n}$ e $l_{x,y}^{f,s}$ são filhos de $z_{x,y}^{f,1}$ e $r_{x,y}^f$ é a raiz. Para cada $c \in \{e, w, n, s\}$, sejam $L_f^c = \{l_{x_1,y_1}^{f,c}, l_{x_2,y_2}^{f,c}, \dots, l_{x_s,y_s}^{f,c}\}$, $L_f = L_f^n \cup L_f^e \cup L_f^w$ e $L = \bigcup_{f=1}^{k^2} L_f$. Subdivida cada aresta entre $r_{x,y}^f$ e seus filhos $2A_{n'} - 1$ vezes e subdivida a aresta entre cada folha em $L_{x,y}^f$ e seu pai $A_{n'} - 1$ vezes. Seja $T_{x,y}^f$ a árvore enraizada em $r_{x,y}^f$, com folhas $L_{x,y}^f$.
3. Para cada conjunto $L_{x,y}^f$, adicione uma aresta entre cada folha em $L_{x,y}^f$ e u e subdivida cada uma dessas arestas $A_{n'} - 1$ vezes. Seja $L_r = (\bigcup_{j=1}^k L_{f(1,j)}^n \cup L_{f(k,j)}^s) \cup (\bigcup_{i=1}^k L_{f(i,1)}^w \cup L_{f(i,k)}^e)$. Ligue cada vértice em $L_{x,y}^f \cap L_r$ a z e, então, subdivida cada uma dessas arestas $A_{n'} - 1$ vezes. Note que os vértices criados a partir da subdivisão das arestas entre um vértice em $L_{x,y}^f$ e u e entre um vértice em $L_{x,y}^f$ e z estão em $V(G_{i,j})$ mas não em $V(T_{x,y}^f)$ enquanto os próprios vértices u e z não estão em nenhum dos dois conjuntos.

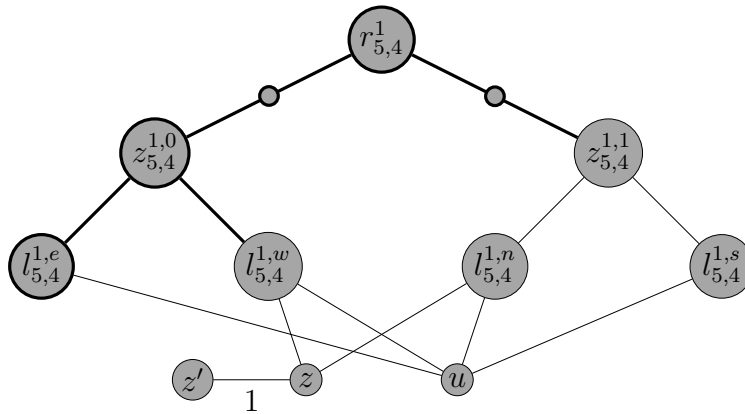


Figura 36 – Vértices u , z e z' e árvore $T_{5,4}^1$. A subárvore $T_{5,4}^{1,e}$ está destacada em negrito. Note que $l_{5,4}^{1,w}$ e $z_{5,4}^{1,1}$ não estão em $V(T_{5,4}^{1,e})$.

Sejam $T_f = V(T_{x_1,y_1}^f) \cup V(T_{x_2,y_2}^f) \cup \dots \cup V(T_{x_s,y_s}^f)$ e $T = \bigcup_{i=1,j=1}^k T_{f(i,j)}$. Seja $T_{x,y}^{f,c}$, para todo $c \in \{n, s, e, w\}$, a subárvore de $T_{x,y}^f$ induzida pelos vértices $(I[r_{x,y}^f, l_{x,y}^{f,c}] \cup I[r_{x,y}^f, l_{x,y}^{f,c'}] \cup I[r_{x,y}^f, z_{x,y}^{f,t}]) \setminus \{l_{x,y}^{f,c'}, z_{x,y}^{f,t}\}$, onde, se $c = e$ ou $c = w$, então $t = 1$, ou, caso contrário, $t = 0$ e c' representa a direção oposta de c , ou seja, se $c = n$, $c' = s$ e vice-versa e se $c = e$, $c' = w$ e vice-versa. A Figura 36 possui um exemplo que mostra a árvore $T_{5,4}^1$ e as arestas entre os vértices em $L_{5,4}^1$ e os vértices u e z . Ela também mostra a subárvore $T_{5,4}^{1,e}$ destacada em negrito.

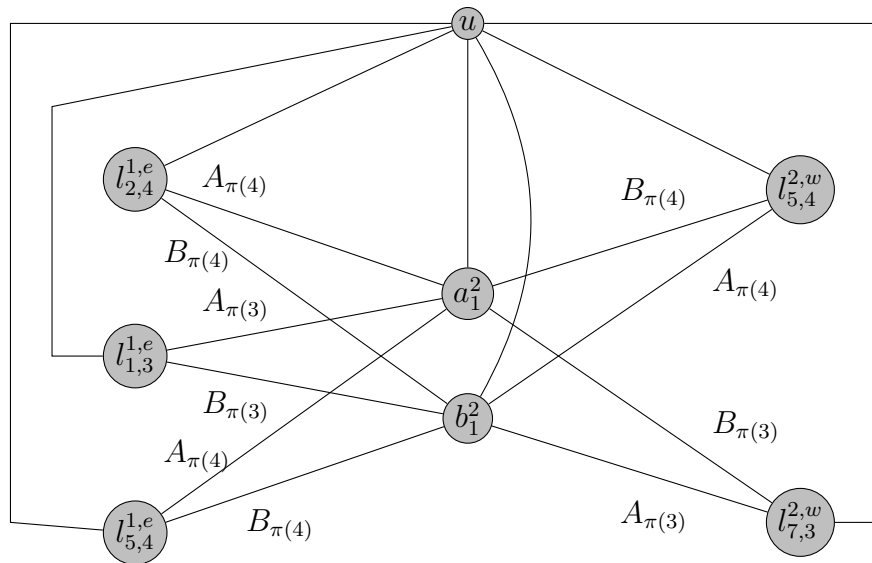


Figura 37 – *Gadget* de aresta $E_{1,2}$, onde $S_{1,1} = \{(2, 4), (1, 3), (5, 4)\}$ e $S_{1,2} = \{(5, 4), (7, 3)\}$.

Para cada par de pares ordenados vizinhos (i, j) e (i', j') , onde $i \leq i'$ e $j \leq j'$, sejam

- $c = e$ e $c' = w$, se $i = i'$;
- $c = s$ e $c' = n$, se $j = j'$; e
- $f = f(i, j)$ e $f' = f(i', j')$

e construa o *gadget de aresta* $E_{f,f'}$ da seguinte forma:

1. Adicione os vértices $a_f^{f'}$ e $b_f^{f'}$, faça-os vizinhos de u e subdivida cada uma das duas arestas $A_{n'}$ – 1 vezes;
2. Ligue cada vértice em L_f^c e em $L_{f'}^{c'}$ a ambos os vértices $a_f^{f'}$ e $b_f^{f'}$.
3. Para cada $l_{x,y}^{f,c} \in L_f^c$, subdivida a aresta entre ele e $a_f^{f'}$ ou $A_{\pi(y)}$ – 1 vezes, se $c = e$, ou $A_{\pi(x)}$ – 1 vezes, se $c = s$, e subdivida a aresta que entre ele e $b_f^{f'}$ ou $B_{\pi(y)}$ – 1 vezes, se $c = e$, ou $B_{\pi(x)}$ – 1 vezes, se $c = s$.
4. Para cada $l_{x',y'}^{f',c'} \in L_{f'}^{c'}$, subdivida a aresta que entre ele e $a_f^{f'}$ ou $B_{\pi(y')}$ – 1 vezes, se $c' = w$, ou $B_{\pi(x')}$ – 1 vezes, se $c' = n$, e a aresta entre ele e $b_f^{f'}$ ou $A_{\pi(y')}$ – 1 vezes, se $c' = w$, ou $A_{\pi(x')}$ – 1 vezes, se $c' = n$;

A Figura 37 contém um exemplo do *gadget* de aresta $E_{1,2}$, onde $S_{1,1} = \{(2, 4), (1, 3), (5, 4)\}$ e $S_{1,2} = \{(5, 4), (7, 3)\}$. A Figura 38 possui um exemplo mais completo, mostrando as ligação de dois *gadgets* de conjunto através de um *gadget* de aresta.

Seja E_f o conjunto de vértices resultantes da subdivisão da aresta entre um vértice l em L_f e os vértices $a_f^{f'}$ e $b_f^{f'}$, para todo f' tal que existem vértices $a_f^{f'}$ e $b_f^{f'}$. Note que l , $a_f^{f'}$ e $b_f^{f'}$ não estão em E_f .

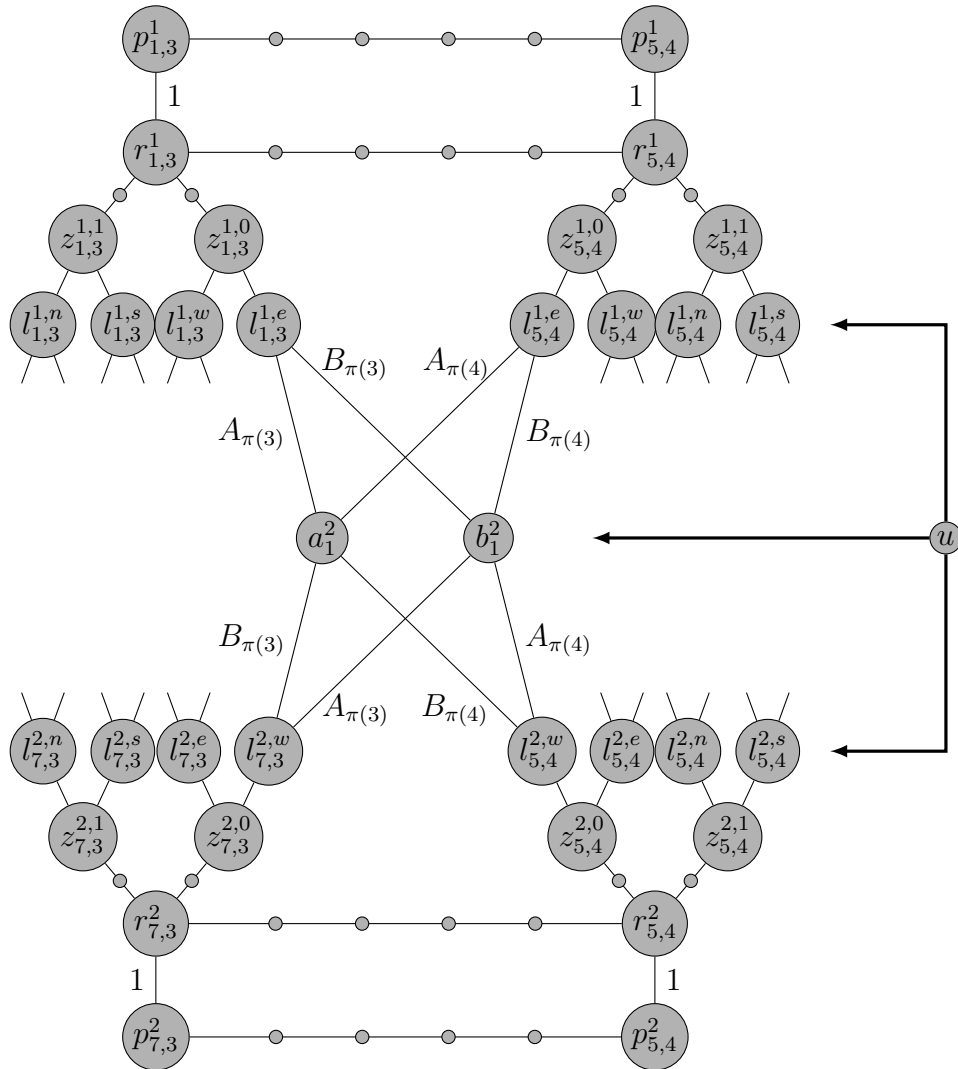


Figura 38 – Gadgets de conjunto $G_{1,1}$ e $G_{1,2}$ e gadget de aresta $E_{1,2}$ sem os vértices z e z' . As arestas entre u e os vértices em $L \cup \{a_1^2, b_1^2\}$ foram omitidas.

Seja G' o grafo resultante da nossa construção aplicada à instância (k, n, \mathcal{S}) do problema REVESTIMENTO DE GRADE. A redução executa em tempo $O(n'S) = O(S^2)$, onde $S = \sum_{S' \in \mathcal{S}} |S'|$, o que é polinomial no tamanho da entrada.

Baseando-nos no fato de que o conjunto de vértices

$$\{u, z, a_{f(i,j)}^{f(i+1,j)}, b_{f(i,j)}^{f(i+1,j)}, a_{f(i-1,j)}^{f(i,j)}, b_{f(i-1,j)}^{f(i,j)}, a_{f(i,j-1)}^{f(i,j)}, b_{f(i,j-1)}^{f(i,j)}, a_{f(i,j)}^{f(i,j+1)}, b_{f(i,j)}^{f(i,j+1)}\}$$

separa o gadget de conjunto $G_{i,j}$ dos outros, é possível provar que $tw(G) \leq 8k + 3$.

Agora, vamos mostrar que (k, n, \mathcal{S}) é uma instância SIM de REVESTIMENTO DE GRADE se e somente se $(G', k^2 + 1)$ é uma instância SIM de NÚMERO DE ENVOLTÓRIA GEODÉTICO.

A ideia principal desta prova é: (a) todos os conjuntos de vértices $V(P_f)$ são coconvexos e existem k^2 deles, (b) se S composto pelo vértice z juntamente com um vértice de cada caminho P_f , então o grafo todo é infectado por S se e somente se todos os vértices a_i^j e b_i^j estão em $I[S]$ e (c) os vértices $a_{f(i,j)}^{f(i',j')}$ e $b_{f(i,j)}^{f(i',j')}$ estão em $I[p_{x,y}^{f(i,j)}, p_{x',y'}^{f(i',j')}]$ se e somente se $x = x'$, caso $j = j'$, ou se $y = y'$, caso $i = i'$.

Suponha que (k, n, \mathcal{S}) é uma instância SIM de REVESTIMENTO DE GRADE e seja $H = \{s_{i,j} = (x_j, y_i) : 1 \leq i, j \leq k\}$ uma solução para essa instância do problema. Seja também $S = \{p_{x_j, y_i}^{f(i,j)} : 1 \leq i, j \leq k\} \cup \{z'\}$. Note que $|S| = k^2 + 1$.

Vamos mostrar que S é um conjunto de envoltória de G' . Inicialmente, para cada (i, j) vizinho de (i', j') , onde $i \leq i'$ e $j \leq j'$, temos que os vértices $a_{f(i,j)}^{f(i',j')}$, $b_{f(i,j)}^{f(i',j')}$ estão em $I[S]$, pois H é uma solução para a instância (k, n, \mathcal{S}) . Como $A_x + B_x = A_y + B_y$ para todo $1 \leq x, y \leq n'$, temos, então, que os vértices em L estão em $I^2[S]$. Assim, temos que todos os vértices em T estão em $I^4[S]$ e, desta forma, temos que $V(G') \subseteq I_h[S]$. Portanto, temos que $hn_g(G') \leq k^2 + 1$.

Note que cada conjunto P_f é um conjunto coconvexo. Portanto, como existem k^2 deles e z' tem grau 1, temos que qualquer conjunto de envoltória de G' tem tamanho pelo menos $k^2 + 1$ e qualquer conjunto de envoltória de G' de tamanho $k^2 + 1$ deve possuir z' e exatamente um vértice de cada conjunto P_f .

Então, suponha que (k, n, \mathcal{S}) é uma instância NÃO de REVESTIMENTO DE GRADE. Seja S um subconjunto arbitrário de $V(G')$ que possui z' e exatamente um vértice de cada conjunto P_f . Como (k, n, \mathcal{S}) é uma instância NÃO de REVESTIMENTO DE GRADE, temos que existem vértices $a_{f(i,j)}^{f(i',j')}$, $b_{f(i,j)}^{f(i',j')}$ tais que um dos dois não está em $I[S]$. Assim, é possível provar que tal vértice não pode ser mais infectado nos passos seguintes de infecção, o que implica que S não é um conjunto de envoltória de G' . Portanto, temos que $hn_g(G') > k^2 + 1$. \square

Nossa intuição, de acordo com os resultados até aqui, é de que, de uma forma geral, o problema NÚMERO DE ENVOLTÓRIA P_3 é mais fácil do que o problema NÚMERO DE ENVOLTÓRIA GEODÉTICO quando são parametrizados pelo mesmo parâmetro, sobretudo quando tal parâmetro não influencia nos caminhos mínimos de um grafo. Por exemplo, temos que o problema NÚMERO DE ENVOLTÓRIA P_3 parametrizado por $tw(G)$ é **FPT** [22], porém mostramos que o problema NÚMERO DE ENVOLTÓRIA GEODÉTICO parametrizado por $tw(G) + k$ é **W[1]**-difícil.

Por isso, dado que o problema NÚMERO DE ENVOLTÓRIA P_3 é $\mathbf{W}[1]$ -difícil quando parametrizado por k , mesmo em grafos de grau máximo limitado, nós formulamos a seguinte conjectura.

Conjectura 5.5. NÚMERO DE ENVOLTÓRIA GEODÉTICO parametrizado por k é $\mathbf{W}[1]$ -difícil mesmo em grafos de grau máximo limitado.

5.3 Consequências da Hipótese do Tempo Exponencial

Nesta seção, nós apresentamos as consequências das provas de $\mathbf{W}[1]$ -dificuldade nas Seções 5.1 e 5.2 quando assumimos a ETH.

Duas consequências conhecidas da ETH são que, como mencionado na Seção 2.4, o problema CLIQUE MULTICOLORIDA não pode ser decidido em tempo $f(k) \cdot |V(G)|^{o(k)}$ e o problema REVESTIMENTO DE GRADE não pode ser decidido em tempo $f(k) \cdot n^{o(k)}$, para qualquer função computável f .

Assim, os Corolários 5.6, 5.7 e 5.8 são subprodutos das provas dos Teoremas 5.1, 5.3 e 5.4 respectivamente. Sejam n o número de vértices e tw a largura em árvore do grafo na instância do problema em questão.

Corolário 5.6. O problema NÚMERO DE ENVOLTÓRIA P_3 não pode ser decidido em tempo $f(k) \cdot n^{o(k)}$, mesmo em grafos livres de K_3 com grau máximo seis, para qualquer função computável f , a não ser que a ETH seja falsa.

Demonstração. O corolário segue do fato de que, se tivéssemos um algoritmo que decide NÚMERO DE ENVOLTÓRIA P_3 em tempo $f(k) \cdot n^{g(k)}$ para algumas funções f e $g \in o(k)$, poderíamos usar a redução do Teorema 5.1, que é um algoritmo FPT, para, a partir de um instância (G, c, k) do problema CLIQUE MULTICOLORIDA, gerar uma instância equivalente $(G', 4k)$ para o problema NÚMERO DE ENVOLTÓRIA P_3 e, depois, decidir a instância $(G', 4k)$ em tempo $O(f(4k) \cdot (n + m + k^2)^{g(4k)}) = f'(k) \cdot n^{o(k)}$ para alguma função f' em k . Assim, pela equivalência das instâncias, teríamos um algoritmo para decidir o problema CLIQUE MULTICOLORIDA em tempo $f'(k) \cdot n^{o(k)}$, o que contraria a ETH. \square

Com um argumento mais simples e similar ao da prova do corolário acima, temos o seguinte corolário.

Corolário 5.7. O problema NÚMERO DE ENVOLTÓRIA GEODÉTICO não pode ser decidido em tempo $f(k) \cdot n^{o(k)}$, mesmo em grafos de diâmetro dois, para qualquer função computável f , a não ser que a ETH seja falsa.

Os Corolários 5.6 e 5.7 mostram que, em ambas as convexidades, o algoritmo que enumera os $O(n^k)$ conjuntos de tamanho no máximo k e testa se algum deles é um conjunto de envoltória é o melhor que podemos conseguir no expoente de n em termos assintóticos, pois tem complexidade de tempo $O(m \cdot n^{k+1})$.

Corolário 5.8. *O problema NÚMERO DE ENVOLTÓRIA GEODÉTICO não pode ser decidido em tempo $f(tw, k) \cdot n^{o(tw)+o(\sqrt{k})}$, para qualquer função computável f , a não ser que a ETH seja falsa.*

Demonstração. O corolário segue do fato de que, se tivéssemos um algoritmo que decide NÚMERO DE ENVOLTÓRIA GEODÉTICO em tempo $f(k) \cdot n^{g(tw)+h(k)}$ para algumas funções $f, h \in o(\sqrt{k})$ e $g \in o(tw)$, poderíamos usar a redução do Teorema 5.4, que é um algoritmo FPT, para, a partir de um instância (k, n, \mathcal{S}) do problema REVESTIMENTO DE GRADE, gerar uma instância equivalente $(G', k^2 + 1)$, onde $tw(G') \leq 8k + 3$, para o problema NÚMERO DE ENVOLTÓRIA GEODÉTICO e, depois, decidir a instância $(G', k^2 + 1)$ em tempo $O(f(k^2 + 1) \cdot (k^2 \cdot n^3)^{g(k)+h(k^2)}) = f'(k) \cdot n^{o(k)}$ para alguma função f' em k . Portanto, pela equivalência das instâncias, teríamos um algoritmo para decidir o problema REVESTIMENTO DE GRADE em tempo $f'(k) \cdot n^{o(k)}$, contrariando a ETH. \square

5.4 Algoritmo XP no parâmetro $tw(G)$ na convexidade Geodésica

Na Seção 5.2, nós vimos que NÚMERO DE ENVOLTÓRIA GEODÉTICO parametrizado por $tw(G) + k$ é $\mathbf{W}[1]$ -difícil. Isso significa que não existe algoritmo FPT que o decida a não ser que $\mathbf{FPT} = \mathbf{W}[1]$. Por outro lado, como podemos computar a envoltória na convexidade geodésica de um conjunto qualquer S em tempo $O(m \cdot n)$ [53], temos que o problema NÚMERO DE ENVOLTÓRIA GEODÉTICO parametrizado por k é \mathbf{XP} , pois podemos verificar se um dos $O(n^k)$ conjuntos S de tamanho no máximo k é um conjunto de envoltória. Adicionalmente, temos que o problema NÚMERO DE ENVOLTÓRIA P_3 parametrizado por $tw(G)$ é \mathbf{FPT} [22].

Motivados por esses resultados, nesta seção, nós provamos que o problema NÚMERO DE ENVOLTÓRIA GEODÉTICO é \mathbf{XP} quando parametrizado pela largura em árvore do grafo de entrada.

Para isso, apresentamos um algoritmo que, dado um grafo G e uma decomposição em árvore de G de largura $tw(G)$, computa $hn_g(G)$ em tempo $O((tw + 1)^{tw+6} \cdot n^{2tw+5})$, onde $n = |V(G)|$ e $tw = tw(G)$.

Como existem algoritmos tratáveis por parâmetro fixo, onde o parâmetro é $tw(G)$, que computam uma decomposição em árvore de um grafo G de largura $tw(G)$ [26], tal

algoritmo implica diretamente que o problema NÚMERO DE ENVOLTÓRIA GEODÉTICO é **XP** quando parametrizado por $tw(G)$. Como, nesta seção, trabalhamos apenas com a convexidade geodésica, doravante, omitimos qualquer menção a convexidade usada nesta seção.

A técnica que usamos nesta seção para conseguir o nosso algoritmo XP parametrizado pela largura em árvore nesta seção é a mesma técnica da Seção 3.8 e é uma técnica de programação dinâmica bastante difundida [10, 25, 31, 47, 77].

Primeiro, vamos a algumas definições básicas. Sejam $n = |V(G)|$ e $(\mathcal{T} = (T, F), \mathcal{B} = \{B_j : j \in T \mid B_j \subseteq V(G)\})$ uma boa decomposição em árvore de G de largura $tw = tw(G)$, de forma que a árvore \mathcal{T} está enraizada no nó r . Seja G_j , para todo $j \in T$, o subgrafo de G induzido pela união das sacolas de todos os descendentes de j , incluindo o próprio j . Note então que $G_r = G$. Também, seja \overline{G}_j , para todo $j \in T$, o subgrafo de G induzido pela união das sacolas de todos os antepassados de j , incluindo o próprio j . Seja $Int_j(P) = Int(P) \cap V(G_j)$, onde P é um caminho de G .

Para qualquer $S \subseteq V(G_j)$ e $j \in T$, sejam $U_j(S) = (V(G) \setminus V(G_j)) \cup S$ e $R_j(S)$ o conjunto de todos os caminhos mínimos P entre dois vértices em S tal que $Int(P) \cap B_j = \emptyset$ e $Int_j(P) \setminus S \neq \emptyset$.

Para todo $j \in T$, seja H_j o conjunto de todas as funções $h : B_j \rightarrow [tw + 1]_0$. Nós dizemos que duas funções $h \in H_j$ e $h' \in H_{j'}$, onde j e j' são nós em T tais que $B_j \subseteq B_{j'}$, têm a mesma ordem relativa se e somente se

- $\{u \in B_j \mid h(u) = 0\} = \{u \in B_j \mid h'(u) = 0\}$ e,
- para todo $v, u \in B_j$, $h(v) < h(u)$ se e somente se $h'(v) < h'(u)$.

Dizemos também que uma função $h \in H_j$ é *legal* se e somente se existe um inteiro k tal que ou $Im(h) = [k]_0$ ou $Im(h) = [k]$.

Para todo $j \in T$ e $h \in H_j$, seja $F_{j,h}$ o conjunto de todas as funções $f : Im(h) \setminus \{0\} \rightarrow \binom{V(G)}{2}$. Seja $Q_j^{h,f}(x)$ o conjunto de todos os caminhos mínimos P entre os vértices em $f(x)$ tais que $Int(P) \cap B_j = \{v \in B_j \mid h(v) = x\}$.

Para todo $j \in T$, $h \in H_j$ e $f \in F_{j,h}$, seja $K_j^h(S) = \{v \in B_j \setminus S \mid \forall w \in B_j \setminus S : h(v) \leq h(w)\}$. Também, se $B_j \setminus S \neq \emptyset$, seja $A_j^h(S) = \min\{h(v) \mid v \in B_j \setminus S\}$. Defina \mathcal{P} como sendo o conjunto de todos os caminhos em G .

A ideia por trás das funções em H_j é representar a ordem relativa na qual os vértices em B_j são infectados, de forma que, se dois vértices v e u são tais que $h(u) = h(v)$, então eles são infectados no mesmo tempo. A ideia por trás das funções f em $F_{j,h}$ é, dado um inteiro x ,

representar as extremidades de um dos caminhos que infectou os vértices v em G tais que $h(v) = x$.

Sejam S_1 e S_2 dois conjuntos de envoltória de um grafo G que obedecem duas funções $h \in H_j$ e $f \in F_{j,h}$ no sentido de que os valores de f realmente definem a ordem relativa em que os vértices em B_j são infectados pelos conjuntos S_1 e S_2 e os valores de h definem os vértices que infectaram os vértices em B_j . Intuitivamente, podemos ver que o conjunto $S_3 = (S_1 \cap V(G_j)) \cup (S_2 \cap V(\overline{G}_j))$ também é um conjunto de envoltória de G que obedece h e f . Na verdade, para que S_3 seja um conjunto de envoltória de G que obedece h e f , S_1 e S_2 não precisam ser conjuntos de envoltória, basta que eles obedeçam h e f , S_1 infecte todos os vértices em $V(G_j)$ e S_2 infecte todos os vértices em $V(\overline{G}_j)$.

Todas essas ideias e observações serão formalizadas a seguir e, baseada nelas, nós derivamos um algoritmo de programação dinâmica que calcula o número de envoltória de um grafo G . No entanto, poderíamos usar basicamente o mesmo algoritmo com poucas modificações para adicionalmente computar um conjunto de envoltória mínimo de G .

Definição 5.9. Para todo $j \in T$, $h \in H_j$ e $f \in F_{j,h}$, seja $C_j^{h,f}(S)$, para todo $S \subseteq V(G_j)$, o conjunto de todos os caminhos P que satisfazem uma das três condições a seguir:

1. $P \in R_j(S)$;
2. $Ext(P) \subseteq U_j(S)$ e $P \in Q_j^{h,f}(A_j^h(S))$;
3. $Int(P) = \emptyset$.

Definição 5.10. Para todo $j \in T$, $h \in H_j$ e $f \in F_{j,h}$, seja $C_j^{h,f}$ o conjunto de todas as funções $c : 2^{V(G_j)} \rightarrow \mathcal{P}$ tais que, para todo $S \subseteq V(G_j)$, $c(S) \in C_j^{h,f}(S)$.

Definição 5.11. Para todo $j \in T$, $h \in H_j$, $c : 2^{V(G_j)} \rightarrow \mathcal{P}$ e conjunto $S \subseteq V(G_j)$, seja $I_j^c(S) = S \cup Int_j(c(S))$.

Sejam $I_j^c(S, 0) = S$, $I_j^c(S, i) = I_j^c(I_j^c(S, i-1))$, para qualquer $i > 0$, e $Y_j^c(S) = I_j^c(S, i)$, onde i é o um inteiro positivo tal que $I_j^c(S, i+1) = I_j^c(S, i)$. Pela definição, podemos ver que, para qualquer $i \geq 0$, $I_j^c(S, i) = I_j^c(I_j^c(S, i_1), i_2)$, onde $i = i_1 + i_2$.

Seja $\mathcal{Z}_k^{j,h}$ a família de todos os subconjuntos Z of $V(G_j)$ tais que $Z \cap B_j = \{v \in B_j \mid h(v) \leq k\}$.

Definição 5.12. Sejam $j \in T$, $h \in H_j$ e $f \in F_{j,h}$. Um conjunto S é um (h, f) -conjunto de envoltória de um subgrafo G_j se e somente se $S \in \mathcal{Z}_0^{j,h}$ e existe uma função $c \in C_j^{h,f}$ tal que $Y_j^c(S) = V(G_j)$.

Essas definições formalizam a ideia de um conjunto “obedecer” as funções $h \in H_j$ e $f \in F_{j,h}$. No entanto, note que, para computar a envoltória, em cada passo, em vez de acrescentar a S os vértices em todos os caminhos mínimos entre quaisquer dois vértices de S , a função c escolhe um par de vértices de S e então adicionamos os vértices no caminho $c(S)$ a S . Essa mudança diminui a complexidade das provas e é segura, ou seja, se um conjunto S infecta todos os vértices de um grafo G na convexidade geodésica, ele também infecta todos os vértices de G nessa convexidade geodésica modificada.

Para cada $j \in T$, $h \in H_j$ e $f \in F_{j,h}$, denote por $M_j(h, f)$ a cardinalidade do (h, f) -conjunto de envoltória mínimo de G_j ou ∞ , caso G_j não possua um. Recorde-se que r é a raiz de T . O próximo lema relaciona $hn_g(G)$ com os valores da tabela M_r .

Lema 5.13. *Um conjunto S é um conjunto de envoltória de G se e somente se existem funções $h \in H_r$ e $f \in F_{r,h}$ tais que S é um (h, f) -conjunto de envoltória de G_r .*

Uma consequência do Lema 5.13 é que $hn_g(G) = \min_{h,f} M_r(h, f)$, onde h e f iteram sobre todas as funções em H_r e $F_{r,h}$ respectivamente.

Antes de explicarmos como computamos as tabelas M_j , vamos definir o conceito de índice válido. Seja (h, f) um índice da tabela M_j . Dizemos que (h, f) é *inválido* se e somente se existe um inteiro $x \in \text{Im}(h) \setminus \{0\}$ tal que $Q_j^{h,f}(x) = \emptyset$ ou existe um vértice $v \in B_j$ e um inteiro $x \in \text{Im}(h) \setminus \{0\}$ tal que $v \in f(x)$ e $h(v) \geq x$. Dizemos que um índice de uma tabela é válido se ele não é inválido.

Se há um inteiro $x \in \text{Im}(h) \setminus \{0\}$ tal que $Q_j^{h,f}(x) = \emptyset$, então, para qualquer conjunto $S \subseteq V(G_j)$ e função $c \in C_j^{h,f}$, pela Definição 5.9, temos que $Y_j^c(S) \cap \{v \in B_j \mid h(v) = x\} = \emptyset$, o que implica que G_j não possui (h, f) -conjunto de envoltória.

Se existe um vértice $v \in B_j$ e um inteiro $x \in \text{Im}(h) \setminus \{0\}$ tais que $v \in f(x)$ e $h(v) \geq x$, temos que, para qualquer conjunto $S \subseteq V(G_j)$ e função $c \in C_j^{h,f}$, por um lado, para que os vértices em $\{v \in B_j \mid h(v) = x\}$ estejam em $I_j^c(S, i) \setminus I_j^c(S, i - 1)$, temos que $v \in I_j^c(S, i - 1)$. No entanto, por outro lado, como $h(v) \geq x$, temos que v é infectado por S depois dos vértices em $\{v \in B_j \mid h(v) = x\}$, ou seja, $v \notin I_j^c(S, i - 1)$. Assim, nesse caso, também temos que G_j não possui (h, f) -conjunto de envoltória.

Portanto, se um índice (h, f) de uma tabela M_j é inválido, então $M_j(h, f) = \infty$. Adicionalmente, dada a tabela das distâncias em G , podemos verificar se um índice (h, f) de uma tabela M_j é inválido em tempo $O(tw^2)$.

Os próximos quatro lemas mostram como computar uma tabela M_j dependendo do tipo do nó j .

Lema 5.14 (Nó Folha). *Seja $j \in T$ um nó folha. Para todo $h \in H_j$ e $f \in F_{j,h}$ tais que (h, f) é um índice válido de M_j , temos que $M_j(h, f) = |\{v \in B_j \mid h(v) = 0\}|$.*

Lema 5.15 (Nó Esquece). *Sejam $j \in T$ um nó esquece e j' o seu filho de modo que $B_j = B_{j'} \setminus \{v\}$. Para todo $h \in H_j$ e $f \in F_{j,h}$ tais que (h, f) é um índice válido de M_j , temos que*

$$M_j(h, f) = \min_{h', f'} M_{j'}(h', f'),$$

onde h' itera sobre todas as funções em $H_{j'}$ tais que h' é legal e tem a mesma ordem relativa de h , e f' itera sobre todas as funções em $F_{j',h'}$ tais que, para todo $u \in \text{Dom}(h)$, $f'(h'(u)) = f(h(u))$.

Lema 5.16 (Nó Introduz). *Sejam $j \in T$ um nó introduz e j' o seu filho de modo que $B_j = B_{j'} \cup \{v\}$. Para todo $h \in H_j$ e $f \in F_{j,h}$ tais que (h, f) é um índice válido de M_j , temos que*

$$M_j(h, f) = \begin{cases} M_{j'}(l, g) + 1 & \text{se } h(v) = 0 \\ M_{j'}(l, g) & \text{caso contrário,} \end{cases}$$

onde l é h restrito a $B_{j'}$ e $g \in F_{j',l}$ é tal que, para todo $x \in \text{Im}(l) \setminus \{0\}$, $g(x) = f(x)$.

Lema 5.17 (Nó Junção). *Sejam $j \in T$ um nó junção e j_1 e j_2 os seus filhos de modo que $B_j = B_{j_1} \cup B_{j_2}$. Para todo $h \in H_j$ e $f \in F_{j,h}$ tais que (h, f) é um índice válido de M_j , temos que $M_j(h, f) = M_{j_1}(h, f) + M_{j_2}(h, f) - |\{v \in B_j \mid h(v) = 0\}|$.*

Na prova desses quatro lemas, sempre mostramos como estender um (l, g) -conjunto de envoltória de $G_{j'}$ para um (h, f) -conjunto de envoltória de G_j , onde l e g dependem do tipo de nó. Por exemplo, no caso do nó junção, mostramos basicamente que, se S_1 é um (h, f) -conjunto de envoltória de G_{j_1} e S_2 é um (h, f) -conjunto de envoltória de G_{j_2} , então $S_1 \cup S_2$ é um (h, f) -conjunto de envoltória de G_j . Para provar o lema do nó introduz, nos valem fortemente do fato de que $B_{j'}$ separa v dos vértices em $V(G_j) \setminus B_j$.

Teorema 5.18. *O número de envoltória na convexidade geodésica de um grafo G pode ser computado em tempo $O((tw + 1)^{tw+6} \cdot n^{2tw+5})$, dada uma decomposição em árvore de G de largura tw .*

Demonstração. Antes de tudo, o nosso algoritmo computa a tabela com as distâncias entre todos os pares de vértices de G . Como G é um grafo não-direcionado e sem pesos nas arestas, e $|E(G)| \leq tw \cdot n$, podemos obter essa tabela em tempo $O(tw \cdot n^2)$ aplicando a busca em largura partindo de cada vértice em G . Então, o algoritmo computa uma boa decomposição

em árvore de largura tw com $O(tw \cdot n)$ nós em tempo $O(tw^2 \cdot n)$ a partir da decomposição em árvore dada.

Depois, o nosso algoritmo computa o valor de todas as tabelas M_j de “baixo para cima”, ou seja, ele computa as tabelas M_j , onde j é um nó folha e, após isso, só computa uma tabela M_j se a tabela do filho, ou filhos, de j já tiver sido computada.

Para cada $j \in T$, existem $O((tw + 1)^{tw+2})$ funções em H_j e, para cada função $h \in H_j$, existem $O(n^{2tw+2})$ funções em $F_{j,h}$. Dado um índice (h, f) de uma tabela M_j , o algoritmo leva tempo $O((tw + 1)^2)$ para verificar se (h, f) é válido e, no pior caso, que é alcançado quando j é um nó esquece, leva tempo $O((tw + 1) \cdot n^2)$ para computar $M_j(h, f)$. Assim, para computar a tabela M_j inteira, o algoritmo leva tempo $O((tw + 1)^{tw+5} \cdot n^{2tw+4})$. Portanto, como existem $O(tw \cdot n)$ tabelas, o nosso algoritmo leva tempo $O((tw + 1)^{tw+6} \cdot n^{2tw+5} + tw \cdot n^2 + tw^2 \cdot n) = O((tw + 1)^{tw+6} \cdot n^{2tw+5})$ para computar a tabela M_r e achar o menor valor nela. \square

Em [12], Baker provou que, se G é um grafo planar, então temos que $tw(G) \leq 3d - 1$ e que uma decomposição em árvore de G pode ser computada em tempo $O(d \cdot n)$, onde $n = |V(G)|$ e d é o diâmetro de G . Já sabemos, de acordo com o Teorema 5.18, que NÚMERO DE ENVOLTÓRIA GEODÉTICO parametrizado por $tw(G)$ é um problema **XP**. Assim, temos que o problema NÚMERO DE ENVOLTÓRIA GEODÉTICO parametrizado pelo diâmetro do grafo de entrada em grafos planares também é **XP**. No entanto, acreditamos que é possível fazer melhor para o problema NÚMERO DE ENVOLTÓRIA GEODÉTICO parametrizado pelo diâmetro do grafo em grafos planares. Assim, formulamos a seguinte conjectura.

Conjectura 5.19. *O problema NÚMERO DE ENVOLTÓRIA GEODÉTICO parametrizado pelo diâmetro do grafo é **FPT** em grafos planares.*

6 CONCLUSÃO

Nesta tese, apresentamos diversos resultados para os problemas NÚMERO DE ENVOLTÓRIA nas convexidades P_3 e geodésica e para o problema TEMPO MÁXIMO DE INFECÇÃO na convexidade P_3 obtidos entre os anos 2013 e 2016.

Nós provamos que TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 é **NP**-completo mesmo em grafos grade de grau máximo 3, em grafos bipartidos e $k \geq 5$ fixo, quando $k = n - n^\varepsilon + 4$, para todo $0 < \varepsilon \leq 1$ fixo, e em grafos com grau máximo $\Delta \geq 4$ fixo e $k = \Theta(\log n)$. Mostramos também que ele é **W[1]**-difícil quando parametrizado por $tw(G)$.

Sobre o problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE P_3 , nós mostramos que ele é **W[1]**-difícil quando parametrizado pela entrada k mesmo em grafos livres de K_3 de grau máximo seis. Mostramos ainda que o problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE GEODÉSICA é **W[1]**-difícil no parâmetro k mesmo em grafos com diâmetro dois e no parâmetro $tw(G) + k$. Como consequência da ETH, temos que o problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE P_3 não pode ser decidido em tempo $f(tw(G)) \cdot n^{o(tw(G))}$ e o problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE GEODÉSICA não pode ser decidido nos tempos $f(k) \cdot n^{o(k)}$ e $f(tw(G), k) \cdot n^{o(tw(G)+o(\sqrt{k}))}$, para qualquer função computável f , onde G é o grafo de entrada, k é o inteiro da entrada e $n = |V(G)|$.

Esses resultados mostram o quão difícil os problemas são mesmo em classes de grafos bastante restritas. Intuitivamente, temos que, na convexidade P_3 , o problema NÚMERO DE ENVOLTÓRIA é mais fácil que o problema TEMPO MÁXIMO DE INFECÇÃO, dado que temos um algoritmo polinomial que decide NÚMERO DE ENVOLTÓRIA para k fixo, que é o algoritmo XP que checa se existe algum conjunto de envoltória dentre todos os $O(n^k)$ conjuntos de tamanho no máximo k , enquanto o TEMPO MÁXIMO DE INFECÇÃO é **NP**-completo mesmo para $k \geq 4$ fixo, o que implica que ele não possui algoritmos FPT e nem mesmo XP no parâmetro natural a não ser que **P** = **NP**. Intuitivamente, temos também que o problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE P_3 é mais fácil do que o NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE GEODÉSICA, pois, ainda que ambos sejam **NP**-completos e **W[1]**-difíceis no parâmetro natural, temos que o problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE P_3 parametrizado por k se reduz ao problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE GEODÉSICA parametrizado por k através de uma redução polinomial, e temos também um algoritmo FPT para o problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE P_3 no parâmetro $tw(G)$ por Ben-Zwi [22], enquanto nós mostramos que o

problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE P_3 no parâmetro $tw(G) + k$ continua sendo $\mathbf{W}[1]$ -difícil.

Apesar de muitos resultados de dificuldade, nós também apresentamos alguns algoritmos eficientes para os problemas trabalhados. Nós apresentamos algoritmos polinomiais que decidem TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 para $k = 3$ e, em grafos bipartidos, para $k = 3, 4$. Também, para o mesmo problema, apresentamos um algoritmo para grafos grade sólidos com grau máximo 3. Finalmente, apresentamos alguns algoritmos parametrizados. Para o problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 , apresentamos algoritmos FPT para os parâmetros $\Delta(G) + k$ e $tw(G) + k$. Apresentamos também um algoritmo XP no parâmetro $tw(G)$. Finalmente, mostramos um algoritmo XP para o problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE GEODÉSICA parametrizado por $tw(G)$.

Os resultados no parágrafo anterior mostram que, apesar dos resultados negativos obtidos, podemos conseguir resultados positivos se restringirmos a entrada. Mais especificamente, para o problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 , se escolhermos um parâmetro que restringe os graus dos vértices do grafo, seja de forma direta como o parâmetro grau máximo do grafo ou de forma indireta como o parâmetro largura em árvore do grafo (lembre que $|E(G)|/|V(G)| < tw(G)$), e o combinarmos com outro parâmetro, temos boas chances de obter um algoritmo FPT. Adicionalmente, o Lema 3.19 também nos permite calcular $t_{P_3}(G)$ de qualquer grafo de grau máximo três em que se possa calcular o tamanho dos maiores caminhos induzidos que satisfazem as restrições do lema, o que acontece em classes de grafos que possui ciclos “comportados”, como cactos e pseudoflorestas de grau máximo três.

De uma forma geral, os nossos resultados positivos também reforçam a importância do parâmetro largura em árvore no desenvolvimento de algoritmos parametrizados, visto que, os problemas NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE GEODÉSICA e TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 parametrizados pela largura em árvore do grafo possuem algoritmos XP enquanto, de acordo com [22], o problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE P_3 parametrizado pela largura em árvore do grafo possui um algoritmo FPT.

6.1 Questões em aberto

Apesar de obter resultados que, em sua maioria, complementam-se, alguns pontos nos nossos resultados podem ser melhorados ou estendidos. Algumas questões em aberto relacionadas diretamente com os nossos resultados são:

- os resultados algorítmicos do Capítulo 3 relacionados a k fixo são consequências de caracterizações estruturais. Sendo assim, os tamanhos dos conjuntos F podem ser reduzidos para que o tempo dos algoritmos seja melhorado nos casos em que $k = 3$ em grafos gerais e $k = 4$ em grafos bipartidos? Que relação, se existe alguma, o tamanho de F possui com o Número de Carathéodory do grafo de entrada?
- O problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 em grafos grade sólidos com grau máximo quatro é polinomial ou **NP**-completo?
- Os tempos dos algoritmos FPT e XP envolvendo o parâmetro largura em árvore podem ser melhorados através de técnicas como, por exemplo, a convolução de subconjunto rápida, do inglês *fast subset convolution* [24]?
- O problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE GEODÉSICA parametrizado por k é **W[1]**-difícil ou **FPT** em grafos de grau máximo limitado?
- O problema NÚMERO DE ENVOLTÓRIA NA CONVEXIDADE GEODÉSICA parametrizado pelo diâmetro do grafo é **W[1]**-difícil ou **FPT** em grafos planares?

Como há muitos assuntos que tangem os tópicos tratados nessa tese, são várias as questões em aberto mais gerais sobre convexidade. Há questões em aberto relacionadas a outras classes de grafos que não trabalhamos, como grafos perfeitos ou grafos cordais. Também há questões em aberto relacionadas a outras convexidades, como a convexidade monofônica ou a convexidade de desvio [38], do inglês *detour convexity*, também conhecida como convexidade do maior caminho. Podemos ainda explorar vários outros parâmetros de convexidade, como o número de intervalo $in(G)$, que corresponde o tamanho do conjunto mínimo que infecta o grafo em tempo um, ou o número de convexidade $con(G)$ que é o tamanho do maior subconjunto próprio de vértices convexo. Pode-se ainda considerar vários outros parâmetros para se derivar resultados parametrizados como o diâmetro, largura em caminho e diversidade de vizinhança. A seguir, listamos alguns problemas em aberto, todos computacionais, que achamos particularmente interessantes:

- Determinar se $t_{P_3}(G) \geq k$ é polinomial ou **NP**-completo em grafos cordais?
- Determinar se $t_{P_3}(G) \geq k$ é polinomial ou **NP**-completo em grafos planares quando k é constante?
- Determinar se $hn_g(G) \leq k$ é **NP**-completo ou polinomial em grafos planares? E em grafos perfeitos? No caso de ser **NP**-completo para uma dessas classes, ele é **W[1]**-difícil ou **FPT** no parâmetro natural ou diâmetro do grafo de entrada para tal classe?

- Determinar se $t_g(G) \geq k$, $in_g(G) \leq k$ e $con_g(G) \geq k$ pode ser feito em tempo polinomial em grafos planares ou é **NP**-completo? E em grafos de grau máximo limitado? Esses problemas são **FPT** quando parametrizados pelo diâmetro, largura em árvore e largura em caminho? Em quais outros parâmetros esses problemas possuem algoritmos FPT?
- Uma generalização natural dos parâmetros número de envoltória e número de intervalo de um grafo é a cardinalidade do menor conjunto que infecta o grafo em, no máximo, t passos. Nesse caso, o número de intervalo corresponde a esse valor para $t = 1$ e o número de envoltória para $t = |V(G)|$. Para valores intermediários de t fixo, qual a complexidade de se decidir se esse valor é no máximo k nas convexidades P_3 , geodésica e monofônica? Qual a complexidade de se decidir se esse valor é no máximo k para t como parte da instância em árvores ou ainda para grafos com largura em árvore limitada?

6.2 Trabalhos futuros

Como visto na seção anterior, a área de convexidade oferece muitos caminhos de pesquisa. Um possível caminho de pesquisa a ser explorado seria restringir as classes de grafos a fim de se encontrar algoritmos polinomiais para o problema TEMPO MÁXIMO DE INFECÇÃO. Algumas classes de grafos particularmente interessantes seriam grafos com poucos ciclos ou com ciclos “comportados”, como cactos e pseudoflorestas.

Outro possível caminho a ser seguido é explorar, para os problemas estudados, o parâmetro largura em caminho ou número de cobertura de vértices para resultados de **W[1]**-dificuldade, por ser um parâmetro que restringe mais do que a largura em árvore, e o parâmetro largura em *rank* [96] ou, equivalentemente, largura em clique [49] para resultados algorítmicos, por serem parâmetros que restringem menos do que a largura em árvore.

Ainda outro possível caminho de pesquisa a ser seguido seria explorar os diversos parâmetros de convexidade para a convexidade monofônica. Conjuntamente, poderia-se analisar diversos parâmetros em grafos para se derivar resultados parametrizados.

REFERÊNCIAS BIBLIOGRÁFICAS

- 1 ABRAHAMSON, K. R.; DOWNEY, R. G.; FELLOWS, M. R. Fixed-parameter tractability and completeness IV: On completeness for $W[P]$ and PSPACE analogues. *Annals of Pure and Applied Logic*, v. 73, n. 3, p. 235–276, 1995.
- 2 ADLER, J.; AHARONY, A. Diffusion percolation. I. Infinite time limit and bootstrap percolation. *Journal of Physics A*, v. 21, n. 6, 1387p, 1988.
- 3 AIZENMAN M.; LEBOWITZ J. L. Metastability effects in bootstrap percolation. *Journal of Physics A*, v. 21, p. 3801–3813, 1988.
- 4 ALBENQUE N.; KNAUER K. Convexity in partial cubes: The hull number. *Discrete Mathematics*, v. 339, n. 2, p. 866–876, 2016.
- 5 AMINI, H. Bootstrap Percolation in Living Neural Networks. *Journal of Statistical Physics*, v. 141, n. 3, p. 459–475, 2010.
- 6 ARAUJO, J.; CAMPOS, V.; GIROIRE, F.; SAMPAIO, L.; SOARES, R. On the hull number of some graph classes. *Theoretical Computer Science*, v. 475, p. 1–12, 2013.
- 7 ARAUJO, J.; MOREL, G.; SAMPAIO, L.; SOARES, R.; WEBER, V. Hull number: P_5 -free graphs and reduction rules. *Discrete Applied Mathematics*, v. 210, p. 171–175, 2016.
- 8 ARAÚJO, R. T.; SAMPAIO, R. M.; SZWARCFITER J. L. The convexity of induced paths of order three. *Electronic Notes in Discrete Mathematics*, v. 44, p. 109–114, 2013.
- 9 ARNBORG, S.; CORNEIL, D. G.; PROSKUROWSKI, A. Complexity of finding embeddings in a k -tree. *SIAM Journal on Algebraic and Discrete Methods*, v. 8, n. 2, p. 277–284, 1987.
- 10 ARNBORG, S.; PROSKUROWSKI, A. Linear time algorithms for NP-hard problems restricted to partial k -trees. *Discrete Applied Mathematics*, v. 23, n. 1, p. 11–24, 1989.
- 11 BABEL, L.; OLARIU, S. On the structure of graphs with few P_4 's. *Discrete Applied Mathematics*, v. 84, p. 1–13, 1998.
- 12 BAKER, B. S. Approximation algorithms for NP-complete problems on planar graphs. *Journal of Association for Computing Machinery*, v. 41, n. 1, p. 153–180, 1994.

- 13 BALOGH, J.; BOLLOBÁS, B. Bootstrap percolation on the hypercube. *Probability Theory Related Fields*, v. 134, n. 4, p. 624–648, 2006.
- 14 BALOGH, J.; BOLLOBÁS, B.; MORRIS, R. Bootstrap percolation in three dimensions. *Annals of Probability*, v. 37, n. 4, p. 1329–1380, 2009.
- 15 BALOGH, J.; BOLLOBÁS, B.; MORRIS, R. Bootstrap percolation in high dimensions. *Combinatorics, Probability and Computing*, v. 19, n. 5–6, p. 643–692, 2010.
- 16 BALOGH, J.; BOLLOBÁS, B.; DUMINIL-COPIN, H.; MORRIS, R. The sharp threshold for bootstrap percolation in all dimensions. *Transactions of the American Mathematical Society*, v. 364, n. 5, p. 2667–2701, 2012.
- 17 BARBOSA, R. M.; COELHO, E. M. M.; DOURADO, M. C.; RAUTENBACH, D.; SZWARCFITER, J. L. On the Carathéodory number for the convexity of paths of order three. *SIAM Journal on Discrete Mathematics*, v. 26, p. 929–939, 2012.
- 18 BENEVIDES, F.; CAMPOS, V.; DOURADO, M. C.; SAMPAIO, R. M.; SILVA, A. The maximum time of 2-neighbor bootstrap percolation: algorithmic aspects. *European Journal of Combinatorics*, v. 48, p. 88–99, 2015.
- 19 BENEVIDES, F.; PRZYKUCKI, M. On slowly percolating sets of minimal size in bootstrap percolation. *The Electronic Journal of Combinatorics*, v. 20, n. 2, 46p, 2013.
- 20 BENEVIDES, F.; PRZYKUCKI, M. Maximum percolation time in two-dimensional bootstrap percolation. *SIAM Journal on Discrete Mathematics*, v. 29, n. 1, p. 224–251, 2015.
- 21 BENEVIDES, F.; CAMPOS, V.; DOURADO, M. C.; SAMPAIO, R. M.; SILVA, A. The maximum infection time in the geodesic and monophonic convexities. *Theoretical Computer Science*, v. 609, n. 2, p. 287–295, 2016.
- 22 BEN-ZWI, O.; HERMELIN, D.; LOKSHTANOV, D.; NEWMAN, I. Treewidth governs the complexity of target set selection. *Discrete Optimization*, v. 8, n. 1, p. 87–96, 2011.
- 23 BERTELÈ, U.; BRIOSCHI, F. On non-serial dynamic programming. *Journal of Combinatorial Theory A*, v. 14, n. 2, p. 137–148, 1973.
- 24 BJÖRKLUND, A.; KASKI, P.; HUSFELDT, T. Fourier meets Möbius: fast subset convolution. In: 39th ANNUAL ACM SYMPOSIUM ON THEORY OF COMPUTING, 39, 2007, San Diego, EUA. Anais... Nova Iorque: ACM, 2007. p. 67–74.

- 25 BODLAENDER, H. L. Dynamic programming on graphs with bounded treewidth. In: 15th INTERNATIONAL COLLOQUIUM, 15, 1988, Tampere, Finlândia. *Lecture Notes in Computer Science*, v. 317, p. 105–118, 1988.
- 26 BODLAENDER, H. L. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, v. 25, p. 1305–1317, 1996.
- 27 BODLAENDER, H. L.; CYGAN, M.; KRATSCH, S.; NEDERLOF, J. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. In: 40th INTERNATIONAL COLLOQUIUM ON AUTOMATA, LANGUAGES AND PROGRAMMING, 40, 2013, Riga, Letônia. *Information and Computation*, v. 243, p. 86–111, 2015.
- 28 BOLLOBÁS, B.; SMITH, P. J.; UZZELL, A. J. The time of bootstrap percolation with dense initial sets for all thresholds. *Random Structures & Algorithms*, v. 47, n. 1, p. 1–29, 2015.
- 29 BROADBENT, S. R.; HAMMERSLEY, J. M. Percolation processes: I. Crystals and mazes. *Mathematical Proceedings of the Cambridge Philosophical Society*, v. 53, n. 3, p. 629–641, 1957.
- 30 CALDER, J. Some elementary properties of interval convexities. *Journal of the London Mathematical Society*, v. 3, p. 422–428, 1971.
- 31 CAMPOS, V.; KLEIN, S.; SAMPAIO, R. M.; SILVA, A. Fixed-parameter algorithms for the cocoloring problem. *Discrete Applied Mathematics*, v. 167, p. 52–60, 2014.
- 32 CAMPOS, V.; SAMPAIO, R. M.; SILVA, A.; SZWARCFITER, J. L. Graphs with few P_4 's under the convexity of paths of order three. *Discrete Applied Mathematics*, v. 192, p. 28–39, 2015.
- 33 CARATHEODORY, C. Über den Variabilitätsbereich der Fourierschen Konstanten von positiven harmonischen Funktionen. *Rendiconti del Circolo Matematico di Palermo*, v. 32, p. 193–217, 1911.
- 34 CÁRCERES, J.; HERNANDO, C.; MORA, M.; PELAYO, I.; PUERTAS, M.; SEARA, C. On geodetic sets formed by boundary vertices. *Discrete Mathematics*, v. 306, p. 188–198, 2006.
- 35 CENTENO, C. C.; DOURADO, M. C.; PENSO, L. D.; RAUTENBACH, D.; SZWARCFITER, J. L. Irreversible Conversion of Graphs. *Theoretical Computer Science*, v. 412, p. 3693–3700, 2011.

- 36 CENTENO, C. C.; PENSO, L. D.; RAUTENBACH, D.; DE SÁ, V. G. P. Geodetic Number Versus Hull Number in P_3 -Convexity. *SIAM Journal on Discrete Mathematics*, v. 27, n. 2, p. 717–731, 2013.
- 37 CHALUPA, J.; LEATH, P. L.; REICH, G. R. Bootstrap percolation on a Bethe lattice. *Journal of Physics C*, v. 12, n. 1, p. 31–35, 1979.
- 38 CHANGAT, M.; NARASIMHA-SHENOI, P. G.; PELAYO, I. M. The longest path transit function of a graph and betweenness. *Utilitas Mathematica*, v. 82, p. 111–127, 2010.
- 39 CHARTRAND, G.; WALL, C. E.; ZHANG, P. The Convexity Number of a Graph. *Graphs and Combinatorics*, v. 18, n. 2, p. 209–217, 2002.
- 40 CHARTRAND, G.; FINK, J. F.; ZHANG, P. Convexity in oriented graphs. *Discrete Applied Mathematics*, v. 116, n. 1–2, p. 115–126, 2002.
- 41 CHARTRAND, G.; FINK, J. F.; ZHANG, P. The hull number of an oriented graph. *International Journal of Mathematics and Mathematical Sciences*, v. 2003, n. 36, p. 2265–2275, 2003.
- 42 CHEN, N. On the Approximability of Influence in Social Networks. *SIAM Journal on Discrete Mathematics*, v. 23, n. 3, p. 1400–1415, 2009.
- 43 CHEN, J.; CHOR, B.; FELLOWS, M.; HUANG, X.; JUEDES, D. W.; KANJ, I. A.; XIA, G. Tight lower bounds for certain parameterized np-hard problems. *Information and Computation*, v. 201, n. 2, p. 216–231, 2005.
- 44 CHEN, J.; KANJ, I. A.; XIA, G. Improved upper bounds for vertex cover. *Theoretical Computer Science*, v. 411, n. 40, p. 3736–3756, 2010.
- 45 COELHO, E. M. M.; DOURADO, M. C.; RAUTENBACH, D.; SZWARCFITER, J. L. The Carathéodory Number of the P_3 Convexity of Chordal Graphs. *Discrete Applied Mathematics*, v. 172, p. 104–108, 2014.
- 46 COELHO, E. M. M.; DOURADO, M. C.; SAMPAIO, R. M. Inapproximability results for graph convexity parameters. *Theoretical Computer Science*, v. 600, p. 49–58, 2015.
- 47 CORNEIL, D. G.; KEIL, J. M. A Dynamic Programming Approach to the Dominating Set Problem on k-Trees. *SIAM Journal on Algebraic and Discrete Methods*, v. 8, n. 4, p. 535–543, 1987.

- 48 COURCELLE, B. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, v. 85, n. 1, p. 12–75, 1990.
- 49 COURCELLE, B.; ENGELFRIET, J.; ROZENBERG, G. Handle-rewriting hypergraph grammars. *Journal of Computer and System Sciences*, v. 46, n. 2, p. 218–270, 1993.
- 50 CORNEIL, D. G.; ROTICS, U. On the Relationship Between Clique-Width and Treewidth. *SIAM Journal on Computing*, v. 34, n. 4, 825–847, 2006.
- 51 CYGAN, M.; FOMIN, F. V.; KOWALIK, Ł.; LOKSHTANOV, D.; MARX, D.; PILIPCZUK, M.; PILIPCZUK, M.; SAURABH, S. Parameterized Algorithms. [S.l.]: Springer, 2015. 613p.
- 52 DOMINGOS, P.; RICHARDSON, M. Mining the network value of customers. In: ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 7, 2001, São Francisco, Califórnia. Anais... Nova Iorque: ACM, 2001. p. 57–66.
- 53 DOURADO, M. C.; GIMBEL, J. G.; KRATOCHVÍL, J.; PROTTI, F.; SZWARCFITER, J. L. On the computation of the hull number of a graph. In: COMBINATORICS 2006: A MEETING IN CELEBRATION OF PAVOL HELL'S 60TH BIRTHDAY, 2006, Ischia, Itália. *Discrete Mathematics*, v. 309, n. 18, p. 5668–5674, 2009.
- 54 DOURADO, M. C.; PROTTI F.; RAUTENBACH D.; SZWARCFITER J. L. On the hull number of triangle-free graphs. *SIAM Journal on Discrete Mathematics*, v. 23, n. 4, p. 2163–2172, 2010.
- 55 DOURADO, M. C.; PROTTI F.; RAUTENBACH D.; SZWARCFITER J. L. Some remarks on the geodetic number of a graph. *Discrete Mathematics*, v. 310, n. 4, p. 832–837, 2010.
- 56 DOURADO, M. C.; RAUTENBACH, D.; SÁ, V. G.; SZWARCFITER, J. L. On the geodetic Radon number of grids. *Discrete Mathematics*, v. 313, p. 111–121, 2013.
- 57 DOURADO, M. C.; PENSO, L. D.; RAUTENBACH, D. On the geodetic hull number of P_k -free graphs. *Theoretical Computer Science*, v. 640, p. 52–60, 2016.
- 58 DOWNEY, R. G.; FELLOWS, M. R. Fixed-parameter intractability. In: 7th ANNUAL STRUCTURE IN COMPLEXITY THEORY CONFERENCE, 7, 1992, Boston, EUA. Anais... Califórnia: IEEE Computer Society, 1992. p. 36–49.

- 59 DOWNEY, R. G.; FELLOWS, M. R. Fixed-parameter tractability and completeness. 21st MANITOBA CONFERENCE ON NUMERICAL MATHEMATICS AND COMPUTING, 21, 1992, Winnipeg, Canada. *Congressus numerantium*, v. 87, p. 161–178, 1992.
- 60 DOWNEY, R. G.; FELLOWS, M. R. Fixed-parameter tractability and completeness I: Basic results. *SIAM Journal on Computing*, v. 24, n. 4, p. 873–921, 1995.
- 61 DOWNEY, R. G.; FELLOWS, M. R. Fixed-parameter tractability and completeness II: On completeness for $W[1]$. *Theoretical Computer Science*, v. 141, n. 1–2, p. 109–131, 1995.
- 62 DOWNEY, R. G.; FELLOWS, M. R. Parameterized Complexity. [S.l.]: Springer, 1999. 533p.
- 63 DREYER JR., P. A.; ROBERTS, F. S. Irreversible k-threshold processes: Graph-theoretical threshold models of the spread of disease and of opinion. *Discrete Applied Mathematics*, v. 157, n. 7, p. 1615–1627, 2009.
- 64 DUCHET, P. Convex sets in graphs, II. Minimal path convexity. *Journal of Combinatorial Theory B*, v. 44, p. 307–316, 1988.
- 65 ECKMANN, P.; FEINERMAN, O.; GRUENDLINGER, L.; MOSES, E.; SORIANO, J.; TLUSTY, T. The physics of living neural networks. *Physics Reports: Nonequilibrium physics: From complex fluids to biological systems III. Living systems*, v. 449, n. 1–3, 54–76, 2007.
- 66 ERDŐS, P.; FRIED, E.; HAJNAL, A.; MILNER, E. C. Some remarks on simple tournaments. *Algebra Universalis*, v. 2, p. 238–245, 1972.
- 67 EVERETT, M.; SEIDMAN, S. The hull number of a graph. *Discrete Mathematics*, v. 57, p. 217–223, 1985.
- 68 FARBER, M.; JAMISON, R. E. Convexity in graphs and hypergraphs. *SIAM Journal on Algebraic Discrete Methods*, v. 7, p. 433–444, 1986.
- 69 FELLOWS, M. R.; FOMIN, F. V.; LOKSHTANOV, D.; ROSAMOND, F.; SAURABH, S.; SZEIDER, S.; THOMASSEN, C. On the complexity of some colorful problems parameterized by treewidth. *Information and Computation*, v. 209, n. 2, p. 143–153, 2011.
- 70 FLUM, J.; GROHE, M. Parameterized Complexity Theory. [S.l.]: Springer, 2010. 495p.
- 71 HALIN, R. S-functions for graphs. *Journal of Geometry*, v. 8, n. 1, p. 171–186, 1976.
- 72 HARARY, F.; NIEMINEN, J. Convexity in graphs. *Journal of Differential Geometry*, v. 16, p. 185–190, 1981.

- 73 HARARY, F.; LOUKAKIS, E.; TSOUROS, C. The geodetic number of a graph. *Mathematical and Computer Modelling: An International Journal*, v. 17, n. 11, p. 89–95, 1993.
- 74 HOLROYD, A. E. Sharp metastability threshold for two-dimensional bootstrap percolation. *Probability Theory Related Fields*, v. 125, n. 2, p. 195–224, 2003.
- 75 IMPAGLIAZZO, R.; PATURI, R. On the complexity of k-sat. *Journal of Computer and System Sciences*, v. 62, p. 367–375, 2001.
- 76 IMPAGLIAZZO, R.; PATURI, R.; ZANE, F. Which problems have strongly exponential complexity. *Journal of Computer and System Sciences*, v. 63, n. 4, p. 512–530, 2001.
- 77 KAMMER, F.; THOLEY, T. The complexity of minimum convex coloring. *Discrete Applied Mathematics*, v. 160, n. 6, p. 810–833, 2009.
- 78 KANTÉ, M. M.; NOURINE, L. Polynomial Time Algorithms for Computing a Minimum Hull Set in Distance-Hereditary and Chordal Graphs. *SIAM Journal on Discrete Mathematics*, v. 30, n. 1, p. 311–326, 2016.
- 79 KEMPE, D.; KLEINBERG, J.; TARDOS, E. Maximizing the spread of influence through a social network. In: ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 9, 2003, Washington, EUA. Anais... Nova Iorque: ACM, 2003. p. 137–146.
- 80 KIRKPATRICK, S.; WILCKE, W. W.; GARNER, R. B.; HUELS, H. Percolation in dense storage arrays. *Physica A: Statistical Mechanics and its Applications*, v. 314, n. 1–4, p. 220–229, 2002.
- 81 KLOKS, T. Treewidth: computation and approximation. [S.l.]: Springer, 1994. p. 218.
- 82 KUTTEN, S.; PELEG, D. Fault-local distributed mending. *Journal on Algorithms*, v. 30, n. 1, p. 144–165, 1999.
- 83 LAMPIS, M. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, v. 64, p. 19–37, 2012.
- 84 LEVI, F. W. On Helly’s theorem and the axioms of convexity. *Journal Indian Mathematical Society*, v. 15, p. 65–76, 1951.
- 85 LOKSHTANOV, D.; MARX, D.; SAURABH, S. Lower bounds based on the Exponential Time Hypothesis. *Bulletin of the EATCS*, v. 105, p. 41–72, 2011.

- 86 MALVESTUTO, F. M.; MEZZINI, M.; MOSCARINI, M. Computing simple-path convex hulls in hypergraphs. *Information Processing Letters*, v. 111, n. 5, p. 231–234, 2011.
- 87 MARCILON, T. B.; ARAÚJO, S. N. de.; SAMPAIO, R. M. The maximum time of 2-neighbour bootstrap percolation: complexity results. In: 40th WORKSHOP ON GRAPH-THEORETIC CONCEPTS IN COMPUTER SCIENCE, 40, 2014, Orleans. *Lecture Notes in Computer Science*, v. 8747, p. 372–383, 2014.
- 88 MARCILON, T. B.; SAMPAIO, R. M. The maximum time of 2-neighbour bootstrap percolation in grid graphs and parametrized results. In: 41st WORKSHOP ON GRAPH-THEORETIC CONCEPTS IN COMPUTER SCIENCE, 41, 2015, Munique. *Lecture Notes in Computer Science*, v. 9224, p. 169–185, 2016.
- 89 MARX, D. On the optimality of planar and geometric approximation schemes. In: SYMPOSIUM ON FOUNDATIONS OF COMPUTER SCIENCE, 48., 2007, Providence. Anais... [S.l.]: IEEE, 2007. p. 338–348.
- 90 MILLER, J. C. Percolation and epidemics in random clustered networks. *American Physical Society*, v. 80, n. 2, 4p, 2009.
- 91 MOORE, C.; NEWMAN, M. E. J. Epidemics and percolation in small-world networks. *American Physical Society*, v. 61, n. 5, p. 5678–5682, 2000.
- 92 MOON, J. Embedding tournaments in simple tournaments. *Discrete Mathematics*, v. 2, n. 4, p. 389–395, 1972.
- 93 MUSTAFA, N.; PEKEČ, A. Listen to your neighbors: How (not) to reach a consensus. *SIAM Journal on Discrete Mathematics*, v. 17, p. 634–660, 2004.
- 94 NICTERLEIN, A.; NIEDERMEIER, R.; UHLMANN, J.; WELLER, M. On tractable cases of Target Set Selection. *Social Network Analysis and Mining*, v. 3, n. 2, p. 233–256, 2013.
- 95 NIEDERMEIER, R. Invitation to Fixed-Parameter Algorithms. Oxford: Oxford University Press, 2006. p. 316.
- 96 OUM, S.; SEYMOUR, P. Approximating clique-width and branch-width. *Journal of Combinatorial Theory B*, v. 96, n. 4, p. 514–528, 2006.
- 97 PAPADIMITRIOU, C. H.; VAZIRANI, U. V. On two geometric problems related to the travelling salesman problem. *Journal of Algorithms*, v. 5, n. 2, p. 231–246, 1984.

- 98 PARKER D. B.; WESTHOFF, R. F.; WOLF, M. J. On two-path convexity in multipartite tournaments. *European Journal of Combinatorics*, v. 29, n. 3, p. 641–651, 2008.
- 99 PELAYO, I. M. Geodesic Convexity in Graphs. [S.l.]: Springer, 2013. p. 112.
- 100 PENSO, L. D.; PROTTI, F.; RAUTENBACH, D.; SOUZA, U. S. Complexity analysis of P_3 -convexity problems on bounded-degree and planar graphs. *Theoretical Computer Science*, v. 607, p. 83–95, 2015.
- 101 PFALTZ, J. L. Convexity in directed graphs. *Journal of Combinatorial Theory B*, v. 10, n. 2, p. 143–162, 1971.
- 102 POLJAK, S.; SUURA, M. On periodical behaviour in societies with symmetric influences. *Combinatorica*, v. 3, p. 119–121, 1983.
- 103 PRZYKUCKI, M. Maximal Percolation Time in Hypercubes Under 2-Bootstrap Percolation. *The Electronic Journal of Combinatorics* 19, n. 2, 41p, 2012.
- 104 RADON, J. Mengen konvexer Körper, die einen gemeinsamen Punkt enthalten. *Mathematische Annalen*, v. 83, p. 113–115, 1921.
- 105 ROBERTS, F. S. Challenges for discrete mathematics and theoretical computer science in the defense against bioterrorism. In: BANKS, H. T.; CASTILLO-CHAVEZ, C. (Eds.). *Bioterrorism: Mathematical modeling applications in homeland security*. Philadelphia: SIAM, 2003. p. 1–34.
- 106 ROBERTSON, N.; SEYMOUR, P. Graph minors. I. Excluding a forest. *Journal of Combinatorial Theory B*, v. 35, n. 1, p. 39–61, 1983.
- 107 ROBERTSON, N.; SEYMOUR, P. Graph minors. III. Planar tree-width. *Journal of Combinatorial Theory B*, v. 36, n. 1, p. 49–64, 1984.
- 108 ROBERTSON, N.; SEYMOUR, P. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, v. 7, n. 3, p. 309–322, 1986.
- 109 ROBERTSON, N.; SEYMOUR, P. Graph minors. V. Excluding a planar graph. *Journal of Combinatorial Theory B*, v. 41, n. 1, p. 92–114, 1986.
- 110 ROCKAFELLAR, R. T. Convex analysis. Princeton: Princeton University Press, 1970. p. 451. (Princeton Mathematical Series, v. 28).

- 111 SARDROUD, A. A.; BAGHERI, A. An approximation algorithm for the longest path problem in solid grid graphs. *Optimization Methods and Software*, v. 31, n. 3, p. 479–493, 2016.
- 112 SEYMOUR, P. D.; THOMAS, R. Graph searching and a min-max theorem for treewidth. *Journal of Combinatorial Theory Series B*, v. 58, p. 22–33, 1993.
- 113 TOVEY, C. A. A simplified NP-complete satisfiability problem. *Discrete Applied Mathematics*, v. 8, n. 1, p. 85–89, 1984.
- 114 VAN DE VEL, M. L. J. *Theory of Convex Structures*. [S.l.]: Elsevier, 1993. p. 539.
- 115 VAN ROOIJ, J. M. M.; BODLAENDER, H. L.; ROSSMANITH, P. Dynamic Programming on Tree Decompositions Using Generalised Fast Subset Convolution. In: 17th ANNUAL EUROPEAN SYMPOSIUM, Copenhagen, Dinamarca. *Lecture Notes in Computer Science*, v. 5757, p. 566–577, 2009.
- 116 VARLET, J. C. Convexity in tournaments. *Bulletin de la Societe Royale des Sciences de Liège*, v. 45, p. 570–586, 1976.
- 117 WEST, D. B. *Introduction to Graph Theory*. New Jersey: Prentice Hall, 2001. p. 588.

Apêndices

APÊNDICE A – THE MAXIMUM TIME OF 2-NEIGHBOR BOOTSTRAP PERCOLATION: COMPLEXITY RESULTS

Neste apêndice, são apresentadas as provas completas em inglês dos resultados submetidos e aceitos no congresso *40th International Workshop on Graph-Theoretic Concepts in Computer Science* (WG-2014) em 2014 cujos anais foram publicados no *Lecture Notes in Computer Science* [87]. Tais resultados também foram submetidos ao periódico *Theoretical Computer Science* e estão todos relacionados ao problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 . Eles são:

- A **NP**-completude para grafos bipartidos e $k \geq 5$ fixo;
- Um algoritmo polinomial para $k = 3$ em grafos gerais;
- Um algoritmo polinomial para $k = 3$ em grafos bipartidos; e
- Um algoritmo polinomial para $k = 4$ em grafos bipartidos.

The maximum time of 2-neighbor bootstrap percolation: complexity results

Thiago Marcilon^a, Rudini Sampaio^a

^a*Departamento de Computação, Universidade Federal do Ceará, Fortaleza, Ceará, Brazil*

Abstract

In 2-neighborhood bootstrap percolation on a graph G , given an initially infected set of vertices of G , an infection spreads according to the following deterministic rule: once a vertex of G is infected, it remains infected forever and in consecutive rounds healthy vertices with at least 2 already infected neighbors becomes infected. Percolation occurs if eventually every vertex is infected. The maximum time $t(G)$ is the maximum number of rounds needed to eventually infect the entire vertex set. In 2015, it was proved [7] that deciding whether $t(G) \geq k$ is polynomial time solvable for $k = 2$, but is NP-Complete for $k = 4$ and, if the problem is restricted to bipartite graphs, it is NP-Complete for $k = 7$. In this paper, we solve the open questions. We obtain an $O(mn^5)$ -time algorithm to decide whether $t(G) \geq 3$. For bipartite graphs, we obtain an $O(mn^3)$ -time algorithm to decide whether $t(G) \geq 3$, an $O(m^2n^9)$ -time algorithm to decide whether $t(G) \geq 4$ and we prove that $t(G) \geq 5$ is NP-Complete.

Keywords: Hull set, P3 convexity

1

1. Introduction

We consider a problem in which an infection spreads over the vertices of a connected simple graph G following a deterministic spreading rule in such a way that an infected vertex will remain infected forever. Given a set $S \subseteq V(G)$ of initially infected vertices, we build a sequence $S_{(0)}, S_{(1)}, S_{(2)}, \dots$ in which $S_{(0)} = S$ and $S_{(i+1)}$ is obtained from $S_{(i)}$ by adding to it the vertices of G which have at least 2 neighbors in $S_{(i)}$. Given a set S of vertices and a vertex v of G , let $t(G, S, v)$ be the minimum t such that v belongs to $S_{(t)}$ (let $t(G, S, v) = \infty$ if there is no such t). We say that a set $S_{(0)}$ infects G if eventually every vertex of G becomes infected, that is, there exists t such that $S_{(t)} = V(G)$. We say that S is a hull set (or a percolating set) of G if S infects G .

If S is a percolating set of G , then we define $t(G, S)$ as the minimum t such that $S_{(t)} = V(G)$. Also, define the *percolation time of G* as $t(G) = \max\{t(G, S) : S \text{ infects } G\}$.

Bootstrap percolation was introduced by Chalupa, Leath and Reich [15] as a model for certain interacting particle systems in physics. Since then it has found applications in clustering phenomena, sandpiles [22], and many other areas of statistical physics, as well as in neural networks [1] and computer science [18].

There are two broad classes of questions one can ask about bootstrap percolation. The first, and the most extensively studied, is what happens when the initial configuration $S_{(0)}$ is chosen randomly under some probability distribution? For example, vertices are included in $S_{(0)}$ independently with some fixed probability p . One would like to know how likely percolation is to occur, and if it does occur, how long it takes.

The answer to the first of these questions is now well understood for various graphs. The existence of thresholds first appeared in papers by Holroyd, Balogh, Bollobás, Duminil-Copin and Morris [24, 5, 3]. Sharp thresholds have also been proved for the hypercube (Balogh and Bollobás [2], and Balogh, Bollobás and Morris [6]). There are also very recent results due to Bollobás, Smith and Uzzell [11], about the time percolation take on the discrete torus $\mathbb{T}_n^d = (\mathbb{Z}/n\mathbb{Z})^d$ for a randomly chosen set $S_{(0)}$.

The second broad class of questions is the one of extremal questions. For example, what is the smallest or largest size of a percolating set with a given property? The size of the smallest percolating set in the d -dimensional grid, $[n]^d$, was studied by Pete and a summary can be found in [4]. Morris [27] and Riedl [29], studied the maximum size of minimal percolating sets on the square grid and the hypercube $\{0, 1\}^d$, respectively, answering a question posed by Bollobás. However, it was proved in [16, 13] that finding the smallest percolating set is NP-complete for general graphs. Another type of question is: what is the minimum or maximum time that percolation can take, given that $S_{(0)}$ satisfies certain properties? Recently, Przykucki [28] determined the precise value of the maximum percolation time on the hypercube $2^{[n]}$ as a function of n , and Benevides and Przykucki [8, 9] have similar results for the square grid, $[n]^2$, also answering a question posed by Bollobás.

Email addresses: thiagomarcilon@lia.ufc.br (Thiago Marcilon), rudini@lia.ufc.br (Rudini Sampaio)

Here, we consider the decision version of the percolation time problem, as stated below.

PERCOLATION TIME

Input: A graph G and an integer k .

Question: Is $t(G) \geq k$?

In 2015, Benevides, Campos, Dourado, Sampaio and Silva [7], among other results, proved that deciding whether $t(G) \geq k$ is polynomial time solvable when the parameter k is fixed in 2, but is NP-Complete when the parameter k is fixed in 4 and, when we restrict the problem to allow only bipartite graphs, is NP-Complete when the parameter k is fixed in 7, thus, leaving open the questions: $t(G) \geq k$ is polynomial time solvable when the parameter k is fixed in 3? When we restrict the problem to bipartite graphs, what is the threshold, regarding the parameter k , between polynomiality and NP-completeness? In this paper, we solve these questions.

In Section 2, we show that the Percolation Time Problem is NP-Complete when restricted to bipartite graphs and $k \geq 5$. In Section 3, we discuss the similarities between the structural characterizations presented in Sections 4, 6 and 5 and the arguments used in the proofs in these sections. In Sections 4 and 6, we show that the problem is polynomial time solvable for $k = 3$ and $k = 4$ by showing a structural characterization that can be computed in time $O(mn^3)$ and $O(m^2n^9)$ respectively. For general graphs, we show in Section 5 that the problem is polynomial time solvable for $k = 3$ by showing a structural characterization that can be computed in time $O(mn^5)$.

1.1. Related works and some notation

It is interesting to notice that infection problems appear in the literature under many different names and were studied by researchers of various fields. The particular case in which $r = 2$ in r -neighborhood bootstrap percolation is also a particular case of a infection problem related to convexities in graph.

A finite *convexity space* [25, 30] is a pair (V, \mathcal{C}) consisting of a finite ground set V and a set \mathcal{C} of subsets of V satisfying $\emptyset, V \in \mathcal{C}$ and if $C_1, C_2 \in \mathcal{C}$, then $C_1 \cap C_2 \in \mathcal{C}$. The members of \mathcal{C} are called *\mathcal{C} -convex sets* and the *convex hull* of a set S is the minimum convex set $H(S) \in \mathcal{C}$ containing S . Also, we say that a set S is a *hull set* if the convex hull of S is V .

A convexity space (V, \mathcal{C}) is an *interval convexity* [12] if there is a so-called *interval function* $I : \binom{V}{2} \rightarrow 2^V$ such that a subset C of V belongs to \mathcal{C} if and only if $I(\{x, y\}) \subseteq C$ for every two distinct elements x and y of C . With no risk of confusion, for any $S \subseteq V$, we also denote by $I(S)$ the union of S with $\bigcup_{x, y \in S} I(\{x, y\})$. In interval convexities, the convex hull of a set S can be computed by exhaustively applying the corresponding interval function until obtaining a convex set.

The most studied graph convexities defined by interval functions are those in which $I(\{x, y\})$ is the union of paths between x and y with some particular property. Some common examples are the P_3 -convexity [20], geodetic convexity [21] and monophonic convexity [19]. We observe that the spreading rule in 2-neighbors bootstrap percolation is equivalent to $S_{(i+1)} = I(S_{(i)})$ where I is the interval function which defines the P_3 -convexity: $I(S)$ contains S and every vertex belonging to some path of 3 vertices whose extreme vertices are in S . For this reason, we will use the terms percolating set and hull set, which represent the same concept, interchangeably.

In geodetic convexity, where the interval of S contains S and every vertex lying in some geodesic joining two vertices of S , it was defined the *geodetic iteration number of a graph* [14, 23] which is similar to the definition of percolation time.

Given a vertex v , let $N(v)$ be the set of neighbors of v and let $N[v] = N(v) \cup \{v\}$. Also, given an integer $i \geq 0$, $N_i(v)$ denotes the set of vertices at distance i of v , $N_{\geq i}(v)$ denotes the set of vertices at distance greater or equal to i of v , and $N_{\leq i}(v)$ denotes the set of vertices at distance less or equal to i of v .

2. $t(G) \geq 5$ is NP-Complete in bipartite graphs

In [7], it was proved that deciding if $t(G) \geq 7$ is NP-Complete in bipartite graphs. The following theorem improves this result.

Theorem 2.1. *Deciding whether $t(G) \geq k$ is NP-Complete in bipartite graphs for any $k \geq 5$.*

PROOF. Let us prove that the Percolation Time Problem is NP-Complete by showing a polynomial reduction from the problem **3-SAT**. Given m clauses $\mathcal{C} = \{C_1, \dots, C_m\}$ on variables $X = \{x_1, \dots, x_n\}$ of an instance of **3-SAT**, let us denote the three literals of C_i by $\ell_{i,1}$, $\ell_{i,2}$ and $\ell_{i,3}$. The reduction is as follows:

For each clause C_i of \mathcal{C} , add to G a gadget like the one in Figure 1. Then, for each pair of literals $\ell_{i,a}, \ell_{j,b}$ such that one is the negation of the other, add a vertex $y_{(i,a),(j,b)}$ adjacent to $w_{i,a}$ and $w_{j,b}$. Let Y be the set of all vertices created this way. Finally, add a vertex z adjacent to all vertices in Y and a vertex z' adjacent only to z . Denote the sets $\{u_{i,1}^A, u_{i,2}^A, u_{i,3}^A\}$, $\{u_{i,1}^B, u_{i,2}^B, u_{i,3}^B\}$ and $\{w_{i,1}, w_{i,2}, w_{i,3}\}$ by U_i^A, U_i^B and W_i , respectively. Let

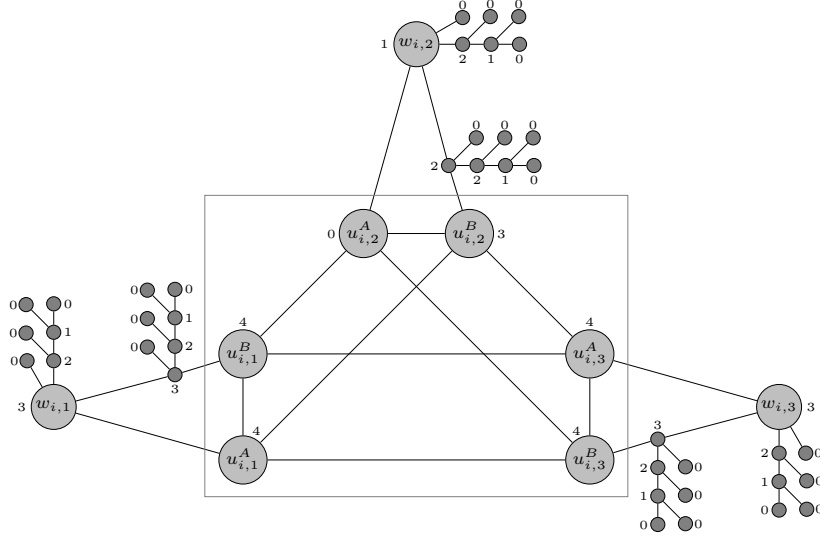


Figure 1: Bipartite gadget for each clause C_i . The numbers outside the vertices indicate the infection times.

$U_i = U_i^A \cup U_i^B$, $U = \bigcup_{1 \leq i \leq m} U_i$ and $W = \bigcup_{1 \leq i \leq m} W_i$. Also, let T be the smaller and darker vertices in the Figure 1 in all the gadgets and let T_0 be the vertices of T with degree one.

We have that each gadget of G is bipartite. Consider a bipartition where all vertices in $U_i^B \cup W$, for all $1 \leq i \leq m$, are in the same partition, and all vertices in U_i^A , for all $1 \leq i \leq m$, are on the other partition. If you consider $Y \cup \{z'\}$ to be in the same partition as the vertices in U_i^A and z to be in the same partition as the vertices in $U_i^B \cup W$, we can always choose in which partition to put the vertices in T so we can have a bipartition for G . Thus, G is bipartite.

Now, let us show that \mathcal{C} is satisfiable if and only if G contains a hull set that infects all vertices in G in time at least k , for any $k \geq 5$. First, we will consider the case $k = 5$.

Suppose that \mathcal{C} has a truth assignment. For each clause C_i , let K_i denote the subset of $\{1, 2, 3\}$ such that $\ell_{i,k}$ is true for all $k \in K_i$. Let $S = \{z'\} \cup \{u_{i,k}^A : k \in K_i, 1 \leq i \leq m\} \cup T_0$. We can see in Figure 1 that all vertices in the clause gadgets are infected by S in time at most 4. Also, we have that S infects $w_{i,k}$, for any $k \in K_i$, in time 1 and infects $w_{i,k'}$, for any $k' \notin K_i$, in time 3 for every $1 \leq i \leq m$. Since we used a truth assignment, we have that all vertices of Y are infected in time exactly 4 because any vertex $y \in Y$ is adjacent to exactly one vertex in W infected at time 1 and another vertex in W infected at time 3. Thus, the vertex z is infected in time 5 and, therefore, $t(G) \geq 5$.

Now, suppose that there is a instance \mathcal{C} is not satisfiable. Let G be the graph resulting from the reduction applied to \mathcal{C} . Let us prove that $t(G) < 5$. Let S be any hull set of G . Since S is a hull set, then it must have at least one vertex in each set U_i , because each vertex in U_i have only one neighbor outside U_i , and all vertices of degree one, for the same reason. Thus, since \mathcal{C} is not satisfiable, then there is a vertex $y \in Y$ such that S infects y at time ≤ 3 . This is because, if S infects all vertices in Y at time 4, then we can find an assignment to the variables of \mathcal{C} that satisfies \mathcal{C} . Consider the following assignment: If a vertex $u_{i,j}$ is in S and $\ell_{i,j}$ is a positive literal then assign true to the variable that it represents and, hence, if a vertex $u_{i,j}$ is in S and $\ell_{i,j}$ is a negative literal then assign false to the variable that it represents. After that, if there is a variable that does not yet have a value, assign true to that variable. This assignment satisfies \mathcal{C} because since S has at least one vertex in each set U_i , then each clause will have at least one literal that evaluates to true. Furthermore, since all vertices in Y are infected at time 4, then we have that two vertices in U that represent the same variable but negate each other, cannot be in S simultaneously and, therefore, we do not risk to assign different values to the same variable.

Thus, we have that z is infected by S at time ≤ 4 . Since any hull set S infects all vertices in $Y \cup U$ at time ≤ 4 , then z is the only possible vertex that can be infected at time greater than 4. Since, for any hull set S , S infects z at time ≤ 4 , hence all hull sets infect G at time ≤ 4 . Therefore, $t(G) \leq 4$.

For values $k > 5$, it suffices to alter the reduction by adding a path P of length $k - 5$ and linking one end of P to the vertex z , appending a new leaf vertex to each vertex in P . The proof remains the same. \blacksquare

In [7], by adapting the NP-Completeness result of $t(G) \geq 7$, it was also proved that $t(G)$ is NP-hard in planar graphs. Unfortunately, Theorem 2.1 cannot be used to prove NP-hardness in planar bipartite graphs, since our clause gadget is not planar.

3. Discussion about the polynomial results and some useful lemmas

In the next sections, we prove that the Percolation Time Problem for $k = 3$ in general graphs and for $k = 3$ and $k = 4$ for bipartite graphs is polynomial time solvable. In each of these cases, we obtain algorithms that are by-product of a structural characterization that can be computed in polynomial time. All the structural characterizations are very similar to each other: a graph G has percolation time at least k if and only if there is a set F of vertices with size $|F| = O(1)$, a vertex $u \in V(G)$ and a set T_0 of vertices satisfying some specific restrictions, such that the set $F \cup T_0 \cup N_{\geq k}(u)$ infects some vertex x at time k . Since $|F| = O(1)$, the difficulty to compute these characterizations resides in how many sets T_0 we have to test. In each of the next sections, we define the set T_0 and, when it is not obvious, we prove that the number of sets T_0 that we have to test and the construction of a set T_0 can be done in polynomial time.

Also, in the proofs of the next sections, we will often prove that some vertex v is infected by some set S at time k . To do that, we will prove that $t(G, S, v) \geq k$ and $t(G, S, v) \leq k$. Recall the infection process $S_{(0)}, S_{(1)}, S_{(2)}, \dots$ beginning with $S_{(0)} = S$. Besides Lemmas 3.3 and 3.4, there are essentially two arguments that we will use to prove these two points. The first argument is that, if $S_{(k-1)}$ contains at least two vertices in the neighborhood of v , then $t(G, S, v) \leq k$ (that is, $v \in S_{(k)}$). The second argument is that, if $S_{(k-1)}$ contains at most one vertex in the neighborhood of v , then either v is not infected by S or v is infected by S at time at least k , i.e., $t(G, S, v) \geq k$. Also, one argument used many times is that, if either S is a hull set or a subset of S infects v , then $t(G, S, v) \geq k$ means that S infects v at time at least k .

The following useful lemma show us that we can obtain a hull set for any time smaller or equal to $t(G)$.

Lemma 3.1. *Let G be a graph. Then, for any $0 \leq k \leq t(G)$, there exists a hull set S of G such that $t(G, S) = k$.*

PROOF. Given a hull set S' of G such that $t(G, S') \geq 1$, let $S = S' \cup S'_1$, where S'_1 is the set of vertices infected by S' at time 1. Clearly, $t(G, S) = t(G, S') - 1$. Applying this fact, we are done. ■

We say that a set T of vertices is *co-convex* if every vertex of T has at most one neighbor outside T . The next lemma proves that every hull set must contain at least one vertex of any co-convex set.

Lemma 3.2. *Let S be a hull set of G and let T be a co-convex set of G . Then S contains a vertex of T . Consequently, S contains all vertices of G with degree 1. Moreover, for all pair of adjacent vertices x, y with degree two, S contains at least one of them.*

PROOF. Suppose, by contradiction, that $S \cap T = \emptyset$. Since S is a hull set, then S infects all the vertices in T . Let v be the first vertex in T infected by S , i.e., $t(G, S, v) = \min_{v' \in T} t(G, S, v')$. Let $t(G, S, v) = t$. Since v is infected at time $t > 0$ by S , then v must have two neighbor vertices v_1 and v_2 infected at time $t(G, S, v_1) = t - 1$ and $t(G, S, v_2) \leq t - 1$. Clearly $t(G, S, v_1) < t(G, S, v)$ and $t(G, S, v_2) < t(G, S, v)$. Since v has at most one neighbor vertex outside T , then at least one of the two vertices v_1 and v_2 is in T , which is a contradiction because there is no vertex in T infected earlier than v . Therefore, there must be a vertex in $T \cap S$. If z is a vertex of degree 1, then $T = \{z\}$ is clearly co-convex. If x, y are adjacent vertices with degree two, then $T = \{x, y\}$ is clearly co-convex. ■

The following lemma shows us that, if a vertex w is infected at time at least k by a set S , then this also happens for $S \cup \{z\}$ for any $z \in N_{\geq k}(w)$. That is, z cannot help a faster infection of w because z is too far from w .

Lemma 3.3. *Let S be a set of vertices of a graph G and w be a vertex of G . If $t(G, S, w) \geq k$, then $t(G, S, w) \geq t(G, S \cup \{z\}, w) \geq k$ for any $z \in N_{\geq k}(w)$. Consequently, $t(G, S, w) \geq t(G, S \cup N_{\geq k}(w), w) \geq k$.*

PROOF. Let $S' = S \cup \{z\}$. Clearly, $t(G, S', w) \leq t(G, S, w)$. Let us prove that $t(G, S', w) \geq k$ by induction on k . The proof for $k = 0$ is trivial. Now, let $k > 0$ and suppose that this lemma holds for any value smaller than k . Let us prove that it also holds for k . Let $z \in N_{\geq k}(w)$ and suppose that $t(G, S, w) \geq k$. Thus, if $u \in N(w)$, then $z \in N_{\geq k-1}(u)$. By the inductive hypothesis, for every $u \in N(w)$, if $t(G, S, u) \geq k - 1$, then $t(G, S', u) \geq k - 1$. Therefore, since we have at most one vertex $x \in N(w)$ such that $t(G, S, x) < k - 1$ (otherwise, $t(G, S, w) < k$), there is at most one vertex x in $N(w)$ such that $t(G, S', x) < k - 1$. Thus, $t(G, S', w) \geq k$ and we are done. ■

Lemma 3.4. *Let G be a graph and let $S \subseteq S' \subseteq V(G)$. Then $t(G, S, u) \geq t(G, S', u)$ for any vertex u .*

PROOF. If $t(G, S, u) = \infty$, we are done. Suppose that $t(G, S, u) = k$. For any $0 \leq i \leq k$, let $S_{(i)}$ and $S'_{(i)}$ be the sets of vertices infected by S and S' at time i . Notice that, if $S_{(i)} \subseteq S'_{(i)}$, then $S_{(i+1)} \subseteq S'_{(i+1)}$. Since $S = S_{(0)} \subseteq S'_{(0)} = S'$, then we have by induction that $S_{(k)} \subseteq S'_{(k)}$ and we are done. ■

4. Maximum Percolation Time with $k = 3$ in bipartite graphs

The following theorem is the main result of this section.

Theorem 4.1. *Deciding whether $t(G) \geq 3$ is $O(mn^3)$ -time solvable in bipartite graphs.*

To prove this, we first show an important structural result.

Lemma 4.2. *Let G be a connected bipartite graph and T_0 be the set of vertices of G that have degree 1. $t(G) \geq 3$ if and only if there are three vertices $u, v \in N(u)$ and $s \in N_2(u)$ such that $T_0 \cup N_{\geq 3}(u) \cup \{v, s\}$ infects u at time 3.*

PROOF. Firstly, suppose that $t(G) \geq 3$. From Lemma 3.1, there exists a hull set S'' that infects G at time 3 and a vertex u that is infected by S'' at time 3. From Lemma 3.2, $T_0 \subseteq S''$. From Lemma 3.3, $S' = S'' \cup N_{\geq 3}(u)$ is also a hull set that infects u at time 3. If S' contains a vertex in $N(u)$, let $S = S'$. Otherwise, let v be a neighbor of u with the smallest infection time with respect to the hull set S' and let $S = S' \cup \{v\}$. Since G is bipartite, the distance from v to any other vertex of $N(u)$ is at least two. Since all vertices in $N(u) - \{v\}$ are infected at time ≥ 2 by S' , then by Lemma 3.3 all vertices in $N(u) - \{v\}$ are infected at time ≥ 2 by S . Thus, S infects u at time ≥ 3 . In fact, S infects u at time exactly 3, because $S' \subseteq S$ and by Lemma 3.4. Note that, since v is the only vertex of $N(u)$ infected by S at time 0 and G is bipartite, then any vertex in $N_2(u)$ infected at time 1 by S is also infected at time 1 by the set $\{v\} \cup N_{\geq 3}(u) \cup T_0$. Now, we split the proof in two cases.

The first case occurs when no vertex of $N(u) - \{v\}$ has a neighbor in S . Let $w \in N(u) - \{v\}$ be a vertex such that $t(G, S, w) = 2$ and let s, z be vertices of $N(w)$ infected by S at time 1. Since G is bipartite, $s, z \in N_2(u)$. Since no vertex of $N(u) - \{v\}$ has a neighbor in S , then any vertex of $N(u) - \{v\}$ has at most one neighbor in $S \cup \{s\}$. Therefore, every vertex of $N(u) - \{v\}$ is infected by $S \cup \{s\}$ at time ≥ 2 . Thus, $S \cup \{s\}$ infects u at time ≥ 3 . In fact, by Lemma 3.4, $S \cup \{s\}$ infects u at time exactly 3. Since $t(G, S, z) = 1$, then the set $\{v, s\} \cup N_{\geq 3}(u) \cup T_0$ infects z at time 1. Moreover, $\{s, v\} \cup N_{\geq 3}(u) \cup T_0 \subseteq S \cup \{s\}$ and, by Lemma 3.4, the set $\{s, v\} \cup N_{\geq 3}(u) \cup T_0$ infects w at time exactly 2. With the same arguments, we have that $\{s, v\} \cup N_{\geq 3}(u) \cup T_0$ infects u at time 3, through the vertices w and v .

The second case occurs when there is a vertex $w \in N(u) - \{v\}$ with a neighbor $s \in S$. Let z be any vertex of $N(w)$ infected by S at time 1. Again, we have that $s, z \in N_2(u)$. Similarly to the previous case, since the set $\{v, s\} \cup N_{\geq 3}(u) \cup T_0$ infects z at time 1 and $\{v, s\} \cup N_{\geq 3}(u) \cup T_0 \subseteq S$, then by Lemma 3.4 the set $\{s, v\} \cup N_{\geq 3}(u) \cup T_0$ infects w at time 2 and u at time 3. With this, we conclude the proof of the necessary condition.

Now, let us prove the sufficient condition. Suppose that there are three vertices u, v and s such that $v \in N(u)$, $s \in N_2(u)$ and the set $S_0 = T_0 \cup N_{\geq 3}(u) \cup \{v, s\}$ infects u at time 3. We will construct a hull set $S \supseteq S_0$ in rounds. Let S_i be the set obtained in round i . Consider we are in round $i \geq 0$. If S_i is a hull set, let $S = S_i$ and we stop. Otherwise, suppose that there is a vertex $q \in N_2(u)$ not infected by S_i . Notice that q is adjacent to a vertex $w \in N(u)$ not infected by S_i , because q has degree at least 2 and G is bipartite. Let $S_{i+1} = S_i \cup \{q\}$. Since q is at distance 2 of u , then, by Lemma 3.3, $2 \leq t(G, S_{i+1}, u) \leq 3$. Therefore, S_{i+1} infects w at time ≥ 3 , since all neighbors of w are at distance ≥ 2 of q .

Since $N_2(u)$ is finite, this procedure will finish and all vertices of $N_2(u)$ are infected by S and consequently S is a hull set. Moreover, $2 \leq t(G, S, u) \leq 3$ and S infects at time ≥ 3 some vertex $w \in N(u)$. ■

Given the vertices u, v, s such that $v \in N(u)$, $s \in N_2(u)$ and a set $S_0 = T_0 \cup N_{\geq 3}(u) \cup \{v, s\}$ that infects u at time 3, the construction to build a hull set S such that $t(G, S) \geq 3$ that is described in Lemma 4.2 can be implemented to run in time $O(mn^2)$. This is because, since G is connected, the sets $N(u), N_2(u)$ and $N_{\geq 3}(u)$ can be computed in time $O(m)$ and, in each step $i = 1, 2, \dots$, we only have to mark the vertices infected by S_{i-1} (which can be done in time $O(mn)$), find one vertex $q \in N_2(u)$ not infected by S_{i-1} (which can be done in time $O(n)$), if there is any, and set $S_i = S_{i-1} \cup \{q\}$. Now, let us prove the main theorem of the section.

Theorem 4.1. *Deciding whether $t(G) \geq 3$ is $O(mn^3)$ -time solvable in bipartite graphs.*

PROOF. Consider the following algorithm:

The Algorithm 1 outputs **Yes** if and only if, given a bipartite graph G , there are vertices $u, v \in N(u)$ and $s \in N_2(u)$ such that the set $\{s, v\} \cup N_{\geq 3}(u) \cup T_0$ infects u at time 3. Thus, by Lemma 4.2, the algorithm outputs **Yes** if and only if $t(G) \geq 3$. Furthermore, the Algorithm 1 is an $O(mn^3)$ -time algorithm. This is because in $O(n)$ -time the set T_0 can be computed, given a vertex u , it is necessary $O(m)$ -time to compute the sets $N(u), N_2(u)$ and $N_{\geq 3}(u)$, and given u, v and s , it is necessary $O(m)$ -time to test whether the set $\{s, v\} \cup N_{\geq 3}(u) \cup T_0$ infects u at time 3. Therefore, the algorithm 1 decides whether $t(G) \geq 3$, for any bipartite connected graph G , in $O(mn^3)$ -time. ■

Algorithm 1: Algorithm that solves the Maximum Percolation Problem for $k = 3$ in bipartite graphs

Input: A bipartite connected graph G .

Output: **Yes**, if $t(G) \geq 3$. **No**, otherwise.

```

forall  $u \in V(G)$  do
  forall  $v \in N(u)$  do
    forall  $s \in N_2(u)$  do
      if  $N_{\geq 3}(u) \cup T_0 \cup \{s, v\}$  infects  $u$  at time 3 then
        return Yes
return No

```

5. Maximum Percolation Time with $k = 3$ in general graphs

The following theorem is the main result of this section.

Theorem 5.1. *Deciding whether $t(G) \geq 3$ is $O(mn^5)$ -time solvable.*

In order to prove it, we first define the family of sets \mathcal{T}_0^u and prove a structural result.

Definition 5.2. Let \mathcal{T}_0^u be the family of subsets of $V(G)$ such that a set of vertices $T_0 \in \mathcal{T}_0^u$ if and only if, for every separator v and every connected component $H_{v,i}$ of $G - v$ such that $u \notin V(H_{v,i})$ and $V(H_{v,i}) \subseteq N(v)$, T_0 contains exactly one vertex of $H_{v,i}$ and every vertex of T_0 satisfies this property (that is, for every vertex $w \in T_0$, there exists a separator v and a connected component $H_{v,i}$ of $G - v$ such that $w \in V(H_{v,i})$, $u \notin V(H_{v,i})$ and $V(H_{v,i}) \subseteq N(v)$).

Figure 2 illustrates Definition 5.2. In Figure 2, we see that the vertices x, y and z are separators of the graph and $t_1 \in V(H_{x,1})$, $t_2 \in V(H_{x,2})$, $t_3 \in V(H_{z,1})$ and $t_4 \in V(H_{y,1})$. Also, we have that $u \notin V(H_{x,1}) \subseteq N(x)$, $u \notin V(H_{x,2}) \subseteq N(x)$, $u \notin V(H_{y,1}) \subseteq N(y)$ and $u \notin V(H_{z,1}) \subseteq N(z)$. Therefore, we have that $\{t_1, t_2, t_3, t_4\} \in \mathcal{T}_0^u$.

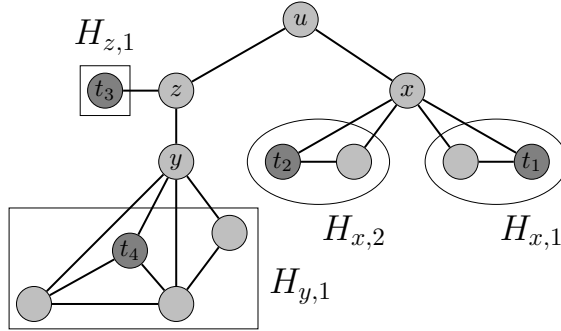


Figure 2: The set $\{t_1, t_2, t_3, t_4\}$ is in \mathcal{T}_0^u , where $t_1 \in V(H_{x,1})$, $t_2 \in V(H_{x,2})$, $t_3 \in V(H_{z,1})$ and $t_4 \in V(H_{y,1})$.

Lemma 5.3. *Let S a hull set of a graph G . For every $u \in V(G)$, there is a set $T_0 \in \mathcal{T}_0^u$ such that $T_0 \subseteq S$.*

PROOF. Notice that, if v is a separator and $H_{v,i}$ is a connected component of $G - v$ such that $u \notin V(H_{v,i})$ and $V(H_{v,i}) \subseteq N(v)$, then $V(H_{v,i})$ is co-convex. Then, by Lemma 3.2, S contains one vertex of $H_{v,i}$. Let T_0 be the set consisting of all such vertices. We have that $T_0 \subseteq S$ and, by Definition 5.2, that $T_0 \in \mathcal{T}_0^u$. ■

Lemma 5.4. *Let G be a connected graph. $t(G) \geq 3$ if and only if there is a vertex u , a set $T_0 \in \mathcal{T}_0^u$ and a set F with at most 4 vertices such that $T_0 \cup N_{\geq 3}(u) \cup F$ infects u at time 3.*

PROOF. First, let us prove the necessary condition. Suppose that $t(G) \geq 3$. Then there exists a hull set S' of G and a vertex u such that $t(G, S', u) = 3$. By Lemma 3.3, $S = S' \cup N_{\geq 3}(u)$ is a hull set such that $t(G, S, u) = 3$. Since S infects u at time 3 and, to be infected, each vertex needs two infected neighbors, then there is a set $F \subseteq S$ with size $|F| \leq 2^3 = 8$ which infects u at time 3. Thus, since, by Lemma 5.3, there is a set $T_0 \in \mathcal{T}_0^u$ such that $T_0 \subseteq S$, then $F \subseteq T_0 \cup N_{\geq 3}(u) \cup F \subseteq S$. By Lemma 3.4, we have that $3 = t(G, S, u) \leq t(G, T_0 \cup N_{\geq 3}(u) \cup F, u) \leq t(G, F, u) = 3$. Therefore, $T_0 \cup N_{\geq 3}(u) \cup F$ infects u at time 3. Furthermore, we can decrease the size of F to 4 by the following claim, proved in Section 7.

Claim 5.4.1. *Let G be a graph and $u \in V(G)$. If G has a hull set which infects u at time 3, then G has a set F with at most 4 vertices and a hull set S such that $F \subseteq S$ and $t(G, F, u) = t(G, S, u) = 3$.*

With this, we conclude the proof of the necessary condition.

Now, let us prove the sufficient condition. Suppose that there is a vertex u , a set $T_0 \in \mathcal{T}_0^u$ and a set $F \subseteq V(G)$ with $|F| \leq 4$ such that $S_0 = T_0 \cup N_{\geq 3}(u) \cup F$ infects u at time 3. If S_0 is a hull set, we are done. Thus, assume that S_0 is not a hull set. We will show how to construct a hull set S such that $t(G, S) \geq 3$. We begin with $S = S_0$. Let S_i be the constructed set at step i . Each step adds one vertex to S and, at the end of each step, it is guaranteed that S_i infects some vertex u_i at time ≥ 3 ($u_0 = u$) and infects u at time ≥ 2 . Let X_i be the set of vertices infected by S_i , and let $Y_i = V(G) - X_i$. It is worth noting that a vertex in Y_i is adjacent to at most one vertex in X_i and, since it must have degree at least two, it is also adjacent to at least one vertex in Y_i .

For $i \geq 0$, in the $(i + 1)$ th step of the construction, assume that there exists a vertex $y_i \in Y_i \cap N_2(u_i)$ with no neighbor infected by S_i at time ≥ 2 . Let $S'_{i+1} = S_i \cup \{y_i\}$. Since S_i infects u_i at time ≥ 3 , u_i has at most one neighbor infected by S_i at time ≤ 1 and, since $y_i \in N_2(u_i)$, u_i is not adjacent to y_i . Furthermore, we have that:

- Every neighbor of u_i infected by S_i at time ≥ 2 is also infected by S'_{i+1} at time ≥ 2 . This is because y_i is at distance ≥ 2 of every vertex infected by S_i at time ≥ 2 , and, therefore, by Lemma 3.3, every vertex infected by S_i at time ≥ 2 , including the neighbors of u_i , is infected by S'_{i+1} at time ≥ 2 .
- Every neighbor of u_i not infected by S_i is either not infected by S'_{i+1} or infected by S'_{i+1} at time ≥ 2 . This is because to a vertex $q \in Y_i \cap N(u_i)$ be infected at time ≤ 1 by S'_{i+1} either it must be y_i itself, which cannot happen because $y_i \in N_2(u_i)$, or q must be adjacent to a vertex infected at time 0 by S_i , which cannot happen either because, if q were adjacent to a vertex infected at time 0 by S_i , since it is also adjacent to u_i , then q would be in X_i .

Therefore, we have that $t(G, S'_{i+1}, u_i) \geq 3$. Let $S_{i+1} = S'_{i+1} \cup N_{\geq 3}(u_i)$. By Lemma 3.3, we have that $t(G, S_{i+1}, u_i) \geq 3$. Thus, letting $u_{i+1} = u_i$, we have that $t(G, S_{i+1}, u_{i+1}) \geq 3$.

Now, assume that every vertex in $Y_i \cap N_2(u_i)$ has exactly one neighbor in X_i , which is infected at time ≥ 2 by S_i . Let y_i be any vertex in $Y_i \cap N_2(u_i)$, let C_i be the connected component of $G[Y_i]$ that contains y_i and let z_i be the neighbor of y_i outside C_i , which is infected time ≥ 2 by S_i . If every vertex of C_i is adjacent to z_i , then C_i has only vertices of $N_2(u)$ or only vertices of $N(u)$ because, otherwise, since z_i would be equal to u , then there would be a vertex in $N_2(u)$ adjacent to u . Also, every vertex of C_i has no neighbor in $N_3(u)$. Hence, either way, z_i is a separator and C_i would be a connected component of $G - z_i$ such that $V(C_i) \subseteq N(z_i)$ and $u \notin V(C_i)$. Therefore, T_0 has a vertex in C_i , which is a contradiction since all vertices of C_i are in Y_i .

We then conclude that there is a vertex y'_i in C_i whose neighbor z'_i outside C_i is distinct from z_i . Let $S'_{i+1} = S_i \cup \{y_i\}$. All vertices in C_i are infected by S_{i+1} because all vertices in C_i have one neighbor infected by S_i . Let us prove that y'_i is infected by S'_{i+1} at time ≥ 3 . First, we have that all vertices in C_i , except for y_i , will be infected at time ≥ 2 by S'_{i+1} because a vertex in $V(C_i) - \{y_i\}$ can only be infected at time ≤ 1 by S'_{i+1} if it is adjacent to 2 vertices infected at time 0 by S'_{i+1} , which cannot happen because all vertices in C_i are adjacent to only one vertex outside of C_i , which is infected at time ≥ 1 by S'_{i+1} , and the only vertex in C_i infected at time 0 by S'_{i+1} is y_i . Additionally, since z'_i is at distance ≥ 2 of y_i and S_i infects z'_i at time ≥ 2 , by Lemma 3.3, z'_i is infected at time ≥ 2 by S'_{i+1} . Therefore, since all vertices in C_i are infected at time ≥ 2 by S'_{i+1} and S'_{i+1} infects z'_i at time ≥ 2 , then y'_i is infected at time ≥ 3 by S'_{i+1} . In Figure 3, we can see an example of how the infection times of the vertices in C_i will be like before and after we add y_i to S_i .

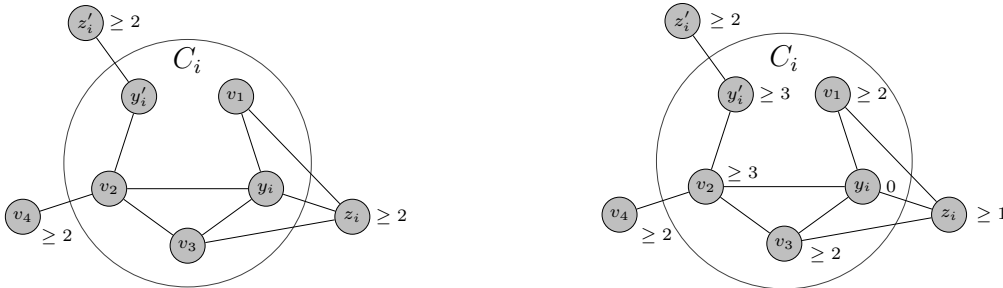


Figure 3: Vertices of the component C_i before and after the addition of y_i to S_i .

Let $S_{i+1} = S'_{i+1} \cup N_{\geq 3}(y'_i)$. By Lemma 3.3, we have that $t(G, S_{i+1}, y'_i) \geq 3$. Thus, letting $u_{i+1} = y'_i$, we have that $t(G, S_{i+1}, u_{i+1}) \geq 3$.

If we have that $N_2(u_i) \cap Y_i = \emptyset$, then $X_i = V(G)$. This is because $\{u_i\} \cup N_2(u_i) \cup N_{\geq 3}(u_i) \subseteq X_i$ and, for each vertex $q \in N(u_i)$, we have:

- If q has a neighbor in $N_2(u_i)$, then q is infected by this neighbor and u_i ;

- If q has degree one, then $q \in T_0$ and, hence, $q \in X_i$;
- If q does not have a neighbor in $N_2(u_i)$ and has degree ≥ 2 , then let C be the connected component of the subgraph induced by the vertices in $N(u_i)$ such that $q \in V(C)$. If $u \in V(C)$ then all vertices in C are also infected, since all vertices are adjacent to $u_i \in X_i$. If $u \notin V(C)$, then either there is a vertex v in C that has a neighbor in $N_2(u_i)$ or every vertex in C do not have a neighbor in $N_2(u_i)$. If there is a vertex v in C that has a neighbor in $N_2(u_i)$ then v is infected by his neighbor in $N_2(u_i)$ and u_i and, hence, all vertices in C are in X_i . If every vertex in C do not have a neighbor in $N_2(u_i)$ then C is a connected component of the subgraph induced by $G - u_i$, where $V(C) \subseteq N(u_i)$, and $u \notin V(C)$, which means that there is a vertex v in $V(C) \cap T_0$ and, thus, all vertices in C are infected.

Thus, since all vertices in $N_2(u_i)$ are in X_i , then $X_i = V(G)$. In this case, the construction is over. The construction will eventually end. This is because, in each step, we add one vertex of $N_2(u_i)$ to S and, even though the vertex u_i potentially changes with each i , eventually $N_2(u_i) \cap Y_i = \emptyset$ because eventually there will be no more vertices of G not infected. Since each step ensures that S infects u_i at time ≥ 3 , then, by the end of the construction, S will be a hull set such that $t(G, S) \geq 3$. ■

Given a vertex u , a set $T_0 \in \mathcal{T}_0^u$ and a set of vertices F , such that $T_0 \cup F \cup N_{\geq 3}(u)$ infects u at time 3, the construction of a hull set S such that $t(G, S) \geq 3$ that is described in Lemma 5.4 can be implemented to run in time $O(mn^2)$. This is because, in each step i , since G is connected, the sets $N(u_i)$, $N_2(u_i)$ and $N_{\geq 3}(u_i)$ can be computed in time $O(m)$; finding one vertex, if there is any, belonging to the set $N_2(u_i) \cap Y_i$ that has at least one neighbor that is either not infected or infected at time ≤ 1 by the current set S can be done in time $O(n)$; and, finally, the set Y_i can be updated in time $O(mn)$ (recall that Y_i are the vertices not infected by S_i).

However, in order to decide if $t(G) \geq 3$, Lemma 5.4 suggests that we have to search for every set $T_0 \in \mathcal{T}_0^u$, which clearly cannot be done in polynomial time. The next lemma shows us that testing just one such T_0 is sufficient.

Lemma 5.5. *If there is a vertex u , a set $T_0 \in \mathcal{T}_0^u$ and a set of vertices F , with $|F| \leq 4$, such that $T_0 \cup N_{\geq 3}(u) \cup F$ infects u at time 3 then for every set $T'_0 \in \mathcal{T}_0^u$ there is a set of vertices F' with $|F'| \leq 4$ such that the set $T'_0 \cup N_{\geq 3}(u) \cup F'$ infects u at time 3.*

PROOF. It is sufficient to show that, for every $T'_0 \in \mathcal{T}_0^u$ such that $T'_0 - \{w'\} = T_0 - \{w\}$, where $w \in T_0$ and $w' \in T'_0$, there is a set F' with $|F'| \leq 4$ such that $F' \cup T'_0 \cup N_{\geq 3}(u)$ also infects u at time 3. By definition 5.2, there is a separator v of G such that the vertices w and w' belong to the same connected component C of the graph $G - v$, where $V(C) \subseteq N(v)$. If $u \neq v$, let D be the connected component of $G - v$ such that $u \in V(D)$. Also, let $R = T_0 \cup N_{\geq 3}(u)$ and $R' = T'_0 \cup N_{\geq 3}(u)$. We can assume that $F \cap R = \emptyset$ (otherwise, let $F = F - R$).

First, suppose that $w' \in F$. Let $F' = (F - \{w'\}) \cup \{w\}$. Since $F' \cup R' = F \cup R$ and $t(G, F \cup R, u) = 3$, then $t(G, F' \cup R', u) = 3$.

Finally, suppose that $w' \notin F$. Since $w, w' \in V(C) \subseteq N(v)$ and C is a connected component, we have directly that $t(G, F \cup R', v) = t(G, F \cup R, v)$. Therefore, $t(G, F \cup R', u) = t(G, F \cup R, u) = 3$. This is because, either $u = v$ (and we are done), or $u \neq v$ and consequently the infection time of any vertex not in D can only affect the infection time of u through v . ■

Theorem 5.1. *Deciding whether $t(G) \geq 3$ is $O(mn^5)$ -time solvable.*

PROOF. Consider the following algorithm:

Algorithm 2: Algorithm that solves the Maximum Percolation Problem for $k = 3$

Input: A connected graph G .

Output: **Yes**, if $t(G) \geq 3$. **No**, otherwise.

forall $u \in V(G)$ **do**

 Find a set $T_0 \in \mathcal{T}_0^u$

forall $F \subseteq V(G)$, where $|F| \leq 4$, **do**

if $F \cup T_0 \cup N_{\geq 3}(u)$ infects u at time 3 **then**

return Yes

return No

When the Algorithm 2 outputs **Yes**, it means that there is a vertex u , a set $T_0 \in \mathcal{T}_0^u$, and a set $F \subseteq V(G)$, where $|F| \leq 4$, such that $F \cup T_0 \cup N_{\geq 3}(u)$ infects u at time 3. Thus, by Lemma 5.4, if the algorithm outputs **Yes**, then $t(G) \geq 3$. On the other hand, when the Algorithm 2 outputs **No**, it means that, for every vertex u ,

there is a set $T_0 \in \mathcal{T}_0^u$ such that for all sets F , where $|F| \leq 4$, we have that $F \cup T_0 \cup N_{\geq 3}(u)$ infects u at time $\neq 3$. Thus, applying the contrapositive of Lemma 5.5, we have that, for every vertex u , all sets $T_0 \in \mathcal{T}_0^u$ and all sets of vertices F , where $|F| \leq 4$, the set $F \cup T_0 \cup N_{\geq 3}(u)$ infects u at time $\neq 3$. Therefore, by Lemma 5.4, if the algorithm outputs **No** then $t(G) < 3$. Additionally, the Algorithm 2 is an $O(mn^5)$ -time algorithm because, given a vertex u , to find an arbitrary set in \mathcal{T}_0^u and compute the set $N_{\geq 3}(u)$, it is necessary $O(m)$ -time, and, given a set F , to test whether the set $F \cup T_0 \cup N_{\geq 3}(u)$ infects u at time 3, it is necessary $O(m)$ -time. Therefore, the Algorithm 2 decides whether $t(G) \geq 3$, for any connected graph G , in $O(mn^5)$ time. \blacksquare

6. Maximum Percolation Time with $k = 4$ in bipartite graphs

The following theorem is the main result of this section.

Theorem 6.1. *Deciding whether $t(G) \geq 4$ is $O(m^2n^9)$ -time solvable in bipartite graphs.*

In order to prove this, we first define some important sets.

Definition 6.2. Let u, v be vertices of G . If $v \in N_{\geq 2}(u)$, define $M_v^u = P_v^u = Q_v^u = \emptyset$. If $v \in N[u]$, define M_v^u to be the set of all vertices x, y such that $x \neq u, y \neq u, vxy$ is an induced P_3 , and x and y have degree two. Also, define $P_v^u = M_v^u \cap N(v)$ and $Q_v^u = M_v^u - P_v^u$.

Note that, if $v \in N(u)$, then $P_v^u = M_v^u \cap N_2(u)$, and $Q_v^u = M_v^u \cap N_2(u)$. Also, for any $v \in N[u]$, since at most two vertices in P_v^u may have the same neighbor in Q_v^u , we have that $|P_v^u|/2 \leq |Q_v^u| \leq |P_v^u|$.

Definition 6.3. Let $u \in V(G)$, $v \in N(u)$ and $k_v = |P_v^u|$. Consider the set $P_v^u = \{x_{v,1}, \dots, x_{v,k_v}\}$ and let $y_{v,i}$ be the neighbor of $x_{v,i}$ in Q_v^u , for every $i \in \{1, \dots, k_v\}$. Let $i \in \{1, \dots, k_v\}$.

- Let T_0^u the set of all vertices in $M_v^u \cap N_2(u)$, for every $v \in N[u]$, and all vertices with degree one;
- let T_0^v the set of all vertices in $(M_{v'}^u - M_v^u) \cap N_2(u)$, for every $v' \in N[u] - \{v\}$, all vertices in Q_v^u and all vertices with degree one;
- let $T_0^{v,i}$ the set of all vertices in $(M_{v'}^u - M_v^u) \cap N_2(u)$, for every $v' \in N[u] - \{v\}$, all vertices in $(Q_v^u - \{y_{v,i}\}) \cup \{x_{v,i}\}$ and all vertices with degree one.

Let Γ_0^u be the family with T_0^u, T_0^v and $T_0^{v,i}$ for every $v \in N(u)$ and every $i \in \{1, \dots, k_v\}$.

It is worth noting that, since two vertices in P_v^u may have the same neighbor in Q_v^u , there may be integers $1 \leq i \neq j \leq k_v$ such that $y_{v,i} = y_{v,j}$.

Lemma 6.4. *Given a vertex u , $|\Gamma_0^u| = O(m)$ and any set of the family Γ_0^u can be computed in time $O(m)$.*

PROOF. First, we can obtain the set $N_2(u)$ in time $O(m)$ by breadth-first search. Following Definition 6.3, we can build the set T_0^u in time $O(n)$ by checking, for any $q \in N_2(u)$, if q has degree two and has a neighbor that also has degree two. Also, add to T_0^u all vertices with degree one.

For each $v \in N(u)$, we can build the set T_0^v in time $O(n)$ by checking, for each vertex $q \in N_2(u)$, if q has degree two and a neighbor that also has degree two. In this case, add q to T_0^v , if q and v are non-adjacent; otherwise, add the neighbor of q that is not v to T_0^v . Also, add to T_0^v all vertices with degree one.

For each $v \in N(u)$ and $i \in \{1, \dots, k_v\}$, we can build the set $T_0^{v,i}$ in time $O(n)$ by checking, for each vertex $q \in N_2(u)$, if q has degree two and neighbor with degree two. In this case, add q to $T_0^{v,i}$, if either q and v are non-adjacent or $q = x_{v,i}$; otherwise, add the neighbor of q that is not v to $T_0^{v,i}$. Also, add to $T_0^{v,i}$ all vertices with degree one.

Therefore, since G is connected and, thus, $n = O(m)$, any set in Γ_0^u can be computed in time $O(m)$ and, since for each $v \in W$, there is $O(|N(v)|)$ sets in Γ_0^u , then $|\Gamma_0^u| = O(m)$. \blacksquare

Lemma 6.5. *Let G be a connected bipartite graph. $t(G) \geq 4$ if and only if there is a vertex u , a subset $T_0 \in \Gamma_0^u$ and a set of vertices F with $|F| \leq 8$ such that $T_0 \cup F \cup N_{\geq 4}(u)$ infects some vertex at time 4.*

PROOF. First, let us prove the necessary condition. If $t(G) \geq 4$, then there is a hull set Z and a vertex u such that $t(G, Z, u) = 4$. By Lemma 3.2, we have that all vertices with degree one are in Z . By Lemma 3.3, we have that the hull set $S'' = Z \cup N_{\geq 4}(u)$ infects u at time 4. Let us prove that there is a hull set $S \supseteq S''$ which infects some vertex x at time 4 and contains a set $T_0 \in \Gamma_0^u$. Let us divide the proof in two cases.

The first case occurs when there is a vertex $v \in N(u)$ such that $t(G, S'', v) \geq 3$ and either $|P_v^u| \geq 2$ or $|P_v^u| = 1$ and the only vertex x in P_v^u is not in S'' . Since $t(G, S'', v) \geq 3$, we have that v does not have two neighbors in S'' . Thus, if $|P_v^u| \geq 2$, by Lemma 3.2, there is at least one vertex x in P_v^u infected at time ≥ 4 by

S'' . On the other hand, if $|P_v^u| = 1$, then the only vertex x in P_v^u is not in S'' and, by Lemma 3.2, we have that x is infected at time ≥ 4 by S'' . At any case, there is a vertex $x \in P_v^u$ infected at time ≥ 4 by S'' . By Lemma 3.1, we have that there is a hull set $S' \supseteq S''$ that infects x at time 4 and v at time 3. Recall that $v \in N(u)$ and $x \in N_2(u)$.

Let $S = S' \cup (N_2(u) \cap \bigcup_{v' \in \{u\} \cup (N(u) - \{v\})} (M_v^u - M_v^u))$. Note that, since v is infected by S'' at time 3 and x at time 4, then v must have degree at least 3. Therefore, $v \notin M_v^u$ and, thus, $x \notin M_v^u$. Since all vertices in $S - S'$ are in $N_2(u)$, G is bipartite and $v \notin S - S'$, then each vertex $w \in S - S'$ is either at distance 2 or at distance ≥ 4 of x . If w is at distance 2 of x , then there is a vertex z adjacent to both w and x and, since z cannot be v , because w would be in M_v^u , and x has degree two, then z is in Q_v^u and also in S' , which means that w is at distance ≥ 3 of v . Therefore, when we add w to S' to form the set S , we have that the resulting set still infects z at time 0 and, by Lemma 3.3, v at time 3 and, thus, infects x at time 4. On the other hand, if w is at distance ≥ 4 of x , and, therefore, at distance ≥ 3 of v , by Lemma 3.3 applied for each vertex of $S - S'$, we have that S infects v at time 3 and x at time 4. In this case, note that either there is no vertex in $S \cap P_v^u$, which implies that $Q_v^u \subseteq S$ or there is exactly one vertex $x_{v,i}$ in $S \cap P_v^u$, which implies that $(Q_v^u - \{x_{v,i}\}) \cup \{x_{v,i}\} \subseteq S$. Therefore, we have that, by Definition 6.3, there is a set $T_0 \in \Gamma_0^u$ such that $T_0 \subseteq S$.

The second case occurs when, for every vertex $v \in N(u)$, $P_v^u = \emptyset$ or $t(G, S'', v) \leq 2$ or $|P_v^u| = 1$ and the only vertex x in P_v^u is in S'' . Let $S = S'' \cup (N_2(u) \cap \bigcup_{v \in N(u)} M_v^u)$. Then, for each vertex $w \in S - S''$, we have that either w is in P_v^u for some vertex $v \in N(u)$ or w is in Q_v^u .

If w is in P_v^u for some vertex $v \in N(u)$, then either $|P_v^u| = 1$ or S'' infects v at time ≤ 2 . However, since $w \notin S''$, we cannot have $|P_v^u| = 1$, since, if $|P_v^u| = 1$, w would have to be in S'' . Thus, v is infected at time ≤ 2 by S'' . Note that, in this case, v is the only vertex in $N(u)$ infected at time ≤ 2 by S'' . Thus, for each vertex $v' \in N(u) - \{v\}$, since G is bipartite and w has degree two, we have either w is at distance ≥ 3 of v' or w is adjacent to v' . But, in fact, we have that w cannot be adjacent to v' . This is because, if w is adjacent to v' , then v' has degree two, because, since $w \in P_v^u$, v' would be in Q_v^u , and is adjacent only to u and w . Thus, since $w \notin S''$, by Lemma 3.2, we would have that $v' \in S''$, and, therefore, since v' and v are infected respectively at times 0 and 2 by S'' , then u would be infected by S'' at time ≤ 3 . Therefore w is at distance ≥ 3 of v' and, since v is infected at time ≤ 2 and u at time 4 by S'' , then v' is infected at time ≥ 3 by S'' , and, therefore, by Lemma 3.3, when we add w to S'' to form S , the resulting set infects all vertices in $N(u)$ that are infected at time ≥ 3 by S'' at time ≥ 3 .

On the other hand, suppose that w is in Q_v^u . Let z be the only vertex in $N(w) \cap P_u^u$. Assume, by contradiction, that w is adjacent to a vertex q in $N(u) \setminus \{z\}$. Since $w \notin S''$, by Lemma 3.2, then $z \in S''$. Therefore, since S'' infects u at time 4, q is infected by S'' at time ≥ 3 by S'' . Also, we have that $w \in P_q^u$. Thus, we have that S'' infects q at time ≥ 3 , and either $|P_q^u| \geq 2$ or $|P_q^u| = 1$ and the only vertex in P_q^u is not in S'' , which is a contradiction. Therefore, z is the only neighbor of w in $N(u)$.

Therefore, since G is bipartite and w is not adjacent to any vertex in $N(u) - \{z\}$, then w is at distance ≥ 3 of all vertices in $N(u) - \{z\}$. Therefore, by Lemma 3.3, when we add w to S'' to form S , the resulting set infects all vertices of $N(u)$ that are infected at time ≥ 3 by S'' at time ≥ 3 .

Hence, we have that S infects all vertices in $N(u)$, except at most by one, at time ≥ 3 and, therefore, S infects u at time ≥ 4 . Also, by Lemma 3.4, S infects u at time ≤ 4 and, therefore, S infects u at time 4. Also, we have that, by Definition 6.3, there is a set $T_0 \in \Gamma_0^u$ such that $T_0 \subseteq S$, which is T_0' .

Since S infects some vertex x at time 4 and, to be infected, each vertex needs only two neighbors infected, then there is a set $F \subseteq S$ that infects x at time 4 such that $|F| \leq 2^4 = 16$. Thus, since there is a set $T_0 \in \Gamma_0^u$ such that $T_0 \subseteq S$ and $N_{\geq 4}(u) \subseteq S$, then $F \subseteq T_0 \cup N_{\geq 4}(u) \cup F \subseteq S$. By Lemma 3.4, we have that $4 = t(G, S, x) \leq t(G, T_0 \cup N_{\geq 4}(u) \cup F, x) \leq t(G, F, x) = 4$. Therefore, there is a set $T_0 \in \Gamma_0^u$, such that the set $T_0 \cup N_{\geq 4}(u) \cup F$, with $|F| \leq 16$, infects the vertex x at time 4. Furthermore, we can decrease the size of F to 8 by the following claim, which is proved in Section 7.

Claim 6.5.1. *Let G be a bipartite graph and $u, x \in V(G)$. If G has a hull set which contains a set $T_0 \in \Gamma_0^u$ and infects x at time 4, then there exists a set F with at most 8 vertices and a hull set S such that $F \subseteq S$ and $t(G, F, x) = t(G, S, x) = 4$.*

Therefore, $T_0 \cup N_{\geq 4}(u) \cup F$ infects the vertex x at time 4 and we conclude the proof of the necessary condition.

Now, let us prove the sufficient condition. Suppose that there is a vertex u , a set F , with $|F| \leq 8$, and a set $T_0 \in \mathcal{T}_0^u$ such that the set $S_0 = T_0 \cup N_{\geq 4}(u) \cup F$ infects some vertex x at time 4. We will show how to construct a hull set S such that $t(G, S) \geq 4$. We begin with $S = S_0$. Let S_i be the constructed set at step i . Each step adds one vertex to S and, at the end of each step, it is guaranteed that S_i infects, at first, x at time 4 and, in the last step, some vertex at time ≥ 4 . Let X_i be the set of vertices infected by S_i , and let $Y_i = V(G) - X_i$. It is worth noting that a vertex in Y_i is adjacent to at most one vertex in X_i and, since it must have degree at least two, it is also adjacent to at least one vertex in Y_i . If S_0 is a hull set, we stop the construction because, since $t(G, S_0, x) = 4$, then $t(G, S_0) \geq 4$.

If not, for $i \geq 0$, in the $(i + 1)$ th step of the construction, suppose that there exists a vertex $y_i \in Y_i \cap N_2(x)$ with no neighbor infected by S_i at time ≥ 2 . Let $S_{i+1} = S_i \cup \{y_i\}$. Clearly, x has at most one neighbor infected by S_i at time ≤ 2 and, by the choice of y_i , x is not adjacent to y_i . Let us prove that $t(G, S_{i+1}, x) = 4$. We have that:

- For every vertex $v \in N(x) \cap N(y_i)$ that is not infected by S_i , we have that $t(G, S_{i+1}, v) \geq 3$. This is because, since every neighbor of v , except x , is not infected by S_i and, since G is bipartite, every neighbor of v is at distance ≥ 2 of y_i , then, by Lemma 3.3, for every vertex $s \in N(v) - \{x\}$, $t(G, S_{i+1}, s) \geq 2$ and, thus, $t(G, S_{i+1}, v) \geq 3$.
- For every vertex $v \in N(x) \cap N_{\geq 3}(y_i)$ such that $t(G, S_i, v) \geq 3$, by Lemma 3.3, we have that $t(G, S_{i+1}, v) \geq 3$.

Additionally, there is at most one vertex in $N(x)$ infected at time ≤ 2 by S_i and, since $y_i \in N_2(x)$ and G is bipartite, $N(x) \cap N_2(y_i) = \emptyset$. Thus, we have that $t(G, S_{i+1}, x) \geq 4$. Also, since $S_i \subseteq S_{i+1}$, by Lemma 3.4, we have that $t(G, S_{i+1}, x) \leq 4$. Therefore $t(G, S_{i+1}, x) = 4$.

If all the vertices in the set $Y_i \cap N_2(x)$ have a neighbor infected by S_i at time ≥ 2 , suppose that there is a vertex $y_i \in Y_i \cap N_3(x)$ with no neighbor infected by S_i at time ≥ 2 . Let $S_{i+1} = S_i \cup \{y_i\}$. Let us prove that $t(G, S_{i+1}, x) = 4$. Let v be any vertex in $N(x)$ such that $t(G, S_i, v) \geq 3$. We have that:

- For every vertex $s \in N(v) \cap N(y_i)$ that is not infected by S_i , we have that $t(G, S_{i+1}, s) \geq 2$. This is because, since s is in $N_2(x)$, then every neighbor of s , except one, say q , which is infected at time ≥ 2 by S_i , is not infected by S_i and, additionally, every neighbor of s is at distance ≥ 2 of y_i , then, by Lemma 3.3, for every vertex $r \in N(s)$, $t(G, S_{i+1}, r) \geq 2$ and, thus, $t(G, S_{i+1}, s) \geq 3 \geq 2$.
- For every vertex $s \in N(v) \cap N_{\geq 3}(y_i)$ such that $t(G, S_i, s) \geq 2$, by Lemma 3.3, we have that $t(G, S_{i+1}, s) \geq 2$.

Additionally, there is at most one vertex in $N(v)$ infected at time ≤ 1 by S_i and, since $y_i \in N_3(x)$ and G is bipartite, $N(v) \cap N_2(y_i) = \emptyset$. Thus, we have that $t(G, S_{i+1}, v) \geq 3$ and, therefore, since there is at most one vertex in $N(x)$ infected at time ≤ 2 by S_i , $t(G, S_{i+1}, x) \geq 4$. Also, since $S_i \subseteq S_{i+1}$, by Lemma 3.4, we have that $t(G, S_{i+1}, x) \leq 4$. Therefore $t(G, S_{i+1}, x) = 4$. It is worth noting that y_i is at distance ≥ 2 of every vertex that is infected at time ≥ 2 by S_i , which, by Lemma 3.3, implies that it is not possible to go back to the previous state, i.e., it is not possible that there is a vertex in $Y_{i+1} \cap N_2(x)$ with no neighbor infected by S_{i+1} at time ≥ 2 .

When all vertices in the set $Y_i \cap N_2(x)$ and in the set $Y_i \cap N_3(x)$ have a neighbor infected by S_i at time ≥ 2 , let C_i be any connected component of $G[Y_i]$. We have that every vertex of C_i has exactly one neighbor outside C_i , which is infected at time ≥ 2 by S_i .

We have that C_i has at least 3 vertices because, otherwise, we would have in C_i either only one vertex of degree one, which would be in T_0 , or two adjacent vertices of degree two, that are not u because $u \in X_i$. If the latter happens, we have that at least one of the two vertices in C_i would be in S_i because of the following:

Claim 6.5.2. *For all pair of neighbor vertices $x, y \neq u$ that have degree 2, we have that $\{x, y\} \cap (T_0 \cup N_{\geq 4}(u)) \neq \emptyset$.*

PROOF. Since G is bipartite, we can assume, without loss of generality, that $x \in N_k(u)$, $y \in N_{k+1}(u)$. If $k = 3$, then $y \in N_{\geq 4}(u)$. So, assume that $1 \leq k \leq 2$. Since $x \in N_k(u)$, then there a vertex $v \in N_{k-1}(u)$ that is neighbor of x . Thus, $x \in P_v^u$ and $y \in Q_v^u$. Since, by Definition 6.3, for each vertex $v \in N[u]$, $w \in P_v^u$ and $z \in Q_v^u$, T_0 has either w or z , then either x or y is in T_0 . ■

Therefore, in either case, one vertex of C_i would also be in T_0 and, consequently, in X_i , which would be a contradiction. Thus, since the graph is bipartite, there are two vertices y_i and y'_i in $V(C_i)$ that are at distance 2 of each other. Let $S'_{i+1} = S_i \cup \{y_i\}$. Let us prove that S'_{i+1} infects y'_i at time ≥ 4 . Let z_i and z'_i be the neighbors of y_i and y'_i outside of C_i respectively. Also, let v be any vertex in $N(y_i) \cap V(C_i)$ and v' his neighbor outside C_i . We have that, since G is bipartite, y_i cannot be neighbor of v' and, since v' is infected by S_i at time ≥ 2 then, by Lemma 3.3, S'_{i+1} infects v' at time ≥ 2 . Also, since all vertices in $N(v) \cap V(C_i)$ are at distance ≥ 2 of y_i , except y_i itself, by Lemma 3.3, they all are infected at time ≥ 2 by S'_{i+1} . Thus, S'_{i+1} infects v at time ≥ 3 . Therefore, since all neighbors of y'_i in $V(C_i)$ are either at distance ≥ 3 of y_i , and then, by Lemma 3.3, are infected by S'_{i+1} at time ≥ 3 , or at distance 1 of y_i , which are also infected at time ≥ 3 by S'_{i+1} , we have that S'_{i+1} infects at most one neighbor of y'_i at time ≤ 2 , which is z'_i . Since all vertices in C_i are infected by S'_{i+1} , then y'_i is infected by S'_{i+1} at time ≥ 4 . In Figure 4, we can see an example of how the infection times of the vertices in C_i will be like before and after we add y_i to S_i .

If y'_i is in $N_2(x)$ (resp. $N(x)$ or $N_3(x)$), let $S_{i+1} = S'_{i+1} \cup ((Y_i - V(C_i)) \cap N_2(x))$ (resp. $S_{i+1} = S'_{i+1} \cup ((Y_i - V(C_i)) \cap (N(x) \cup N_3(x)))$). Since G is bipartite, then each vertex in $S_{i+1} - S'_{i+1}$ is at distance ≥ 3 of



Figure 4: Vertices of the component C_i before and after the addition of y_i to S_i .

all the neighbors of y'_i in C_i and, therefore, since all neighbors of y'_i in C_i are infected at time ≥ 3 by S'_{i+1} , applying Lemma 3.3 once for each vertex in $S_{i+1} - S'_{i+1}$, we have that S_{i+1} infects all neighbors of y'_i in C_i at time ≥ 3 . Thus, since y'_i has only one neighbor outside C_i , then S_{i+1} infects y'_i at time ≥ 4 . Also, since there is at least one vertex in $N_2(x)$ and one vertex in either $N(x)$ or $N_3(x)$ in each connected component of $G[Y_i]$, because each connected component of $G[Y_i]$ has at least three vertices, then all connected components of $G[Y_i]$ are infected by S_{i+1} and, thus, S_{i+1} is a hull set that infects G at time ≥ 4 . At this point, we stop the construction and we have the constructed set S is a hull set such that $t(G, S) \geq 4$. \blacksquare

Given a vertex u , a set $T_0 \in \Gamma_0^u$ and a set of vertices F , such that $T_0 \cup F \cup N_{\geq 3}(u)$ infects some vertex x at time 4, the construction to build a hull set S such that $t(G, S) \geq 4$ that is described in Lemma 6.5 can be implemented to run in time $O(mn^2)$. This is because, since G is connected, the sets $N(u)$, $N_2(u)$, $N_3(u)$ and $N_{\geq 4}(u)$ can be computed in time $O(m)$. Also, for each step i , finding one vertex, if there is any, belonging to the set $N_2(u) \cap Y$ and the set $N_3(u) \cap Y$ that has at least one neighbor that is either not infected or infected at time ≤ 1 by the current set S can be done in time $O(n)$ and updating the set Y can be done in time $O(mn)$. Finally, the last step can be done in time $O(m)$.

Now, let us prove the main theorem of the section.

Theorem 6.1. *Deciding whether $t(G) \geq 4$ is $O(m^2n^9)$ -time solvable in bipartite graphs.*

PROOF. Consider the following algorithm:

Algorithm 3: Algorithm that solves the Maximum Percolation Problem for $k = 4$ in bipartite graphs

Input: A bipartite connected graph G .

Output: **Yes**, if $t(G) \geq 4$. **No**, otherwise.

forall $u \in V(G)$ **do**

forall $T_0 \in \Gamma_0^u$ **do**

forall $F \subseteq V(G)$, where $|F| \leq 8$, **do**

if $F \cup T_0 \cup N_{\geq 4}(u)$ infects any vertex at time 4 **then**

return Yes

return No

The Algorithm 3 outputs **Yes** if and only if, given a bipartite graph G , there is a vertex u , a set $T_0 \in \Gamma_0^u$, and a set $F \subseteq V(G)$, where $|F| \leq 8$, such that $F \cup T_0 \cup N_{\geq 4}(u)$ infects some vertex at time 4. Thus, by Lemma 6.5, the Algorithm 3 outputs **Yes** if and only if $t(G) \geq 4$. Furthermore, we have that, given a vertex u , by Lemma 6.4, there is $O(m)$ sets in Γ_0^u and each one can be computed at time $O(n^2)$. Additionally, given a set F , to test whether the set $F \cup T_0 \cup N_{\geq 4}(u)$ infects some vertex at time 4, it is necessary $O(m)$ -time. Thus, the Algorithm 3 runs at time $O(n \cdot (m \cdot (m + n^8 \cdot m))) = O(m^2n^9)$. Therefore, the Algorithm 3 decides whether $t(G) \geq 4$, for any bipartite connected graph G , in $O(m^2n^9)$ time. \blacksquare

7. Technical lemmas

7.1. Proof of Claim 5.4.1

Claim 5.4.1. *Let G be a graph and $u \in V(G)$. If G has a hull set which infects u at time 3, then G has a set F with at most 4 vertices and a hull set S such that $F \subseteq S$ and $t(G, F, u) = t(G, S, u) = 3$.*

PROOF. First, let us start by defining the set $C_{H,v,t}$. For any vertex v infected by some hull set H at time t , let us define $C_{H,v,t}$ to be any minimal subset of H such that the set $C_{H,v,t}$ infects v at time t . In general, if v is infected at time t by H , then $|C_{H,v,t}| \leq 2^t$ because every vertex needs only two infected neighbors to be infected at any time ≥ 1 . Additionally, if v is infected at time $t \geq 1$ and has a neighbor vertex infected at time 0 by H , since v must have a neighbor v' infected at time $t-1$, then, we have that $|C_{H,v,t}| \leq |C_{H,v',t-1}| + 1 \leq 2^{t-1} + 1$. Particularly, if v is infected at time 2 and has a neighbor infected at time 0 by H , then $|C_{H,v,t}| \leq 3$.

We will start the proof by demonstrating the following fact:

Fact 1. *There is a hull set $S' \supseteq S$ and a vertex $v \in N(u)$ infected at time 2 by S , such that u is also infected at time 3 by S' , v is also infected at time 2 by S' and v has a neighbor in S' .*

PROOF. Let D be the set of vertices in $N(u)$ that are infected at time 2 by S . If there is a vertex $v \in D$ that has a neighbor in S , then just let $S' = S$ and we are done.

If not, then no vertex in D has a neighbor in S . Let v be any vertex in D , and w be a vertex in $N(v) \cap N_2(u)$ that is infected at time 1 by S . Let $S' = S \cup \{w\}$. We have that:

- For every vertex z in D , z does not have neighbors in S . Thus, z will have, at most, one neighbor in S' because $S' = S \cup \{w\}$. Therefore, z is infected at time ≥ 2 by S' .
- For every vertex z adjacent to u and infected at time t by S , where $t \geq 3$, we have that z cannot be adjacent to w and another vertex in S because, otherwise, it would be infected at time ≤ 2 by S . Therefore, z has at most one neighbor in S' and, hence, is infected by S' at time ≥ 2 .

Thus, we have that $t(G, S', u) \geq t(G, S, u)$, and, since $S \subseteq S'$, we also have that, by Lemma 3.4, $t(G, S', u) \leq t(G, S, u)$. Therefore, $t(G, S', u) = t(G, S, u) = 3$. Additionally, we have that the vertex v , which is infected at time 2 by S' , as we have shown before, has a neighbor in S' . Since $S \subseteq S'$, then we have that S' is the hull set we are looking for. \blacksquare

Thus, let S' be the hull set as described in the Fact 1. Since v has one neighbor in S' , we have that $|C_{S',v,2}| \leq 3$.

Suppose that there is a vertex w in $N(u) \cap S'$, let $F = C_{S',v,2} \cup \{w\}$ and $R = S'$. Since $t(G, F, u) \leq 3$, $|F| \leq 4$, $F \subseteq R$ and $t(G, R, u) = 3$, then F and R are the sets that we are looking for.

Now, suppose that there is a vertex w in $N(u)$ such that $t(G, S', w) = 1$. If w is adjacent to some vertex $z \in N(u)$ such that z has a neighbor s in S' , then let $F = C_{S',w,1} \cup \{s\}$ and $R = S'$. Since $t(G, F, s) = 0$ and $t(G, F, w) = 1$ then $t(G, F, z) \leq 2$ and, therefore, $t(G, F, u) \leq 3$. Since $t(G, F, u) \leq 3$, $|F| \leq 4$, $F \subseteq R$ and $t(G, R, u) = 3$, then F and R are the sets that we are looking for. However, if w has no neighbor in $N(u)$ that has a neighbor in S' , let $F = C_{S',v,2} \cup \{w\}$ and $R = S' \cup \{w\}$. We have that $t(G, F, u) \leq 3$ because $t(G, F, v) \leq 2$ and $t(G, F, w) = 0$. Additionally, since all vertices in $N(u) - \{w\}$ are infected by S' at time ≥ 2 and all vertices in $N(u)$ that are adjacent to w are not adjacent to any vertex in S' , then all vertices in $N(u) - \{w\}$ have at most one neighbor in R . Thus, all vertices in $N(u) - \{w\}$ are infected at time ≥ 2 . Therefore, we have that $t(G, R, u) \geq 3$, and, since $t(G, F, u) \leq 3$, $|F| \leq 4$, $F \subseteq R$ and $t(G, R, u) \geq 3$, then F and R are the sets that we are looking for.

Henceforth, we will consider only the cases where all vertices in $N(u)$ are infected at time ≥ 2 by S' .

Suppose that there is a vertex $v' \in N(u) - \{v\}$ that has no neighbors in $N(u)$ that has some neighbor in S' . In this case, let $F = C_{S',v,2} \cup \{v'\}$ and $R = S' \cup \{v'\}$. We have that $t(G, F, u) \leq 3$ because $t(G, F, v) \leq 2$ and $t(G, F, v') = 0$. Additionally, since all vertices in $N(u) - \{v'\}$ are infected by S' at time ≥ 2 and all vertices in $N(u)$ that are adjacent to v' are not adjacent to any vertex in S' , then all vertices in $N(u) - \{v'\}$ have at most one neighbor in R . Since all vertices in $N(u)$ are infected by S' at time ≥ 2 , then all vertices in $N(u) - \{v'\}$ are infected by R at time ≥ 2 . Therefore, we have that $t(G, R, u) \geq 3$, and, since $t(G, F, u) \leq 3$, $|F| \leq 4$, $F \subseteq R$ and $t(G, R, u) \geq 3$, then F and R are the sets that we are looking for.

From now on, suppose that every vertex in $N(u) - \{v\}$ has a neighbor vertex in $N(u)$ that has a neighbor in S' . Let us show the following fact:

Fact 2. *There is a hull set $S'' \supseteq S'$ that infects u at time 3 and there is a vertex $v' \in N(u) - \{v\}$ that is infected at time 2 by S' such that v' is infected at time 2 by S'' and has a neighbor w in S'' .*

PROOF. This proof is similar to the proof of the Fact 1. If there is a vertex $v' \in N(u) - \{v\}$ that is infected at time 2 by S' and that has a neighbor in S' , then, letting $S'' = S'$, S'' is the set we are looking for. Then, let us assume that there is no vertex in $N(u) - \{v\}$ that is infected at time 2 by S' and that has a neighbor in S' . Then, since all vertices in $N(u)$ are infected by S' at time ≥ 2 and $t(G, S', u) = 3$, there must be at least one vertex in $N(u) - \{v\}$ infected at time 2 by S' . Let v' be such vertex. Let $S'' = S' \cup \{w\}$, where w is in $N(v')$ and is infected at time 1 by S' . We have that $w \in N_2(u)$ because, otherwise, there would be a vertex infected by S' at time 1 in $N(u)$. Let D be the set of vertices in $N(u)$ that are infected at time 2 by S' . We have that:

- For every vertex z in $D - \{v\}$, z does not have neighbors in S' . Thus, z will have at most one neighbor in S'' , because $S'' = S' \cup \{w\}$, and, hence, $t(G, S'', z) \geq 2$.
- For every vertex z infected at time t , where $t \geq 3$, by S' , we have that z cannot be adjacent to w and another vertex in S' because, otherwise, it would be infected at time ≤ 2 by S' . Therefore, z has at most one neighbor in S'' and, hence, is infected by S'' at time ≥ 2 .

Tus, we have that v is the only vertex that can be infected at time ≤ 1 by S'' . Also, since all vertices in $N(u) - \{v\}$ are infected at time ≥ 2 by S'' , we have that $t(G, S'', u) \geq 3$. Since $S' \subseteq S''$, then, by Lemma 3.4, we also have that $t(G, S'', u) \leq 3$ and, thus, $t(G, S'', u) = 3$. Also, by Lemma 3.4, $t(G, S'', v') = 2$. Then, since $t(G, S'', w) = 0$, S'' is the set that we are looking for. ■

Thus, suppose that v is infected at time 1 by S'' . Since v is infected at time 2 by S' and 1 by S'' , then v must be adjacent to w and to some other vertex r , which is in S' and, therefore, it is also in S'' . Let x be a vertex in $N(v')$ that is infected at time 1 by S'' . Let $F = \{r, w\} \cup C_{S'', x, 1}$ and $R = S''$. Since $t(G, F, w) = 0$ and $t(G, F, x) = 1$, then $t(G, F, v') = 2$. Hence, $t(G, F, u) \leq 3$ because $t(G, F, v') \leq 2$ and, since $t(G, F, r) = 0$ and $t(G, F, w) = 0$, $t(G, F, v) = 1$. Thus, since $t(G, F, u) \leq 3$, $|F| \leq 4$, $F \subseteq R$ and $t(G, R, u) = 3$, then F and R are the sets that we are looking for.

Thus, henceforth, suppose that v is not infected at time ≤ 1 by S'' . Since, in the proof of the Fact 2, we showed that v was the only vertex that could be infected by S'' at time ≤ 1 , then, there is no vertex in $N(u)$ infected at time ≤ 1 by S'' . Let x be a vertex in $N_2(u) \cap N(v')$ that is infected at time 1 by S'' .

Also, if there is a vertex $z \in N(u) - \{v'\}$ that is adjacent to x and to some vertex in $q \in S''$, we have that z is infected at ≤ 2 by S'' . Since, we assumed previously that there is no vertex in $N(u)$ that is infected at time ≤ 1 by S'' , we have that z , in fact, is infected at time 2 by S'' . Let $F = \{q, w\} \cup C_{S'', x, 1}$ and $R = S''$. We have that $t(G, F, z) \leq 2$ because $t(G, F, q) = 0$ and $t(G, F, x) \leq 1$. Also, $t(G, F, v') \leq 2$ because $t(G, F, w) = 0$ and $t(G, F, x) \leq 1$. Therefore, $t(G, F, u) \leq 3$. Since, we have that $q, w \in S''$, then $F \subseteq R$, and, therefore, since $t(G, F, u) \leq 3$, $|F| \leq 4$, $F \subseteq R$ and $t(G, R, u) = 3$, then F and R are the sets that we are looking for.

If, however, there is no vertex in $N(u) - \{v'\}$ that is adjacent to x and to some vertex in S'' , let $R = S'' \cup \{x\}$. Since there is no vertex in $N(u) \cap S''$ and there is no vertex in $N(u)$, except v' , adjacent to two vertices in $S'' \cup \{x\}$, then all vertices in $N(u) - \{v'\}$ are infected by R at time ≥ 2 . Thus, $t(G, R, v') = 1$ and $t(G, R, u) \geq 3$. Let $q \in N(u)$ be the neighbor of v' that has a neighbor $c \in S'$. Since $S' \subseteq S'' \subseteq R$, then $c \in R$. Also, let $F = C_{R, v', 1} \cup \{c\}$. We have that $t(G, R, q) \leq 2$ because $t(G, F, v') = 1$ and $t(G, F, c) = 0$ and, therefore, since $t(G, R, q) \leq 2$ and $t(G, F, v') = 1$, $t(G, F, u) \leq 3$. Since $t(G, F, u) \leq 3$, $|F| \leq 4$, $F \subseteq R$ and $t(G, R, u) \geq 3$, then F and R are the sets that we are looking for. ■

7.2. Proof of Claim 6.5.1

Claim 6.5.1. *Let G be a bipartite graph and $u, x \in V(G)$. If G has a hull set which contains a set $T_0 \in \Gamma_0^u$ and infects x at time 4, then there exists a set F with at most 8 vertices and a hull set S such that $F \subseteq S$ and $t(G, F, x) = t(G, S, x) = 4$.*

PROOF. First, let us start by defining the set $C_{H, v, t}$. For any vertex v infected by some hull set H at time t , let us define $C_{H, v, t}$ to be any minimal subset of H such that the set $C_{H, v, t}$ infects v at time t . In general, if v is infected at time t by H , then $|C_{H, v, t}| \leq 2^t$ because every vertex needs only two infected neighbors to be infected at any time ≥ 1 . Additionally, if v is infected at time $t \geq 1$ and has a neighbor vertex infected at time 0 by H , since v must have a neighbor v' infected at time $t-1$, then, we have that $|C_{H, v, t}| \leq |C_{H, v', t-1}| + 1 \leq 2^{t-1} + 1$.

Let us begin the proof by proving the following fact:

Fact 1. *There is a hull set $Z \supseteq S$ that infects x at time 4 and some vertex $v \in N(x)$ at time 3 such that $|C_{Z, v, 3}| \leq 4$.*

PROOF. If there is some vertex $v \in N(x)$ infected at time 3 by S such that $|C_{S, v, 3}| \leq 4$, let $Z = S$ and we are done. So, assume that that is not the case.

Let us prove that there is a hull set $Z \supseteq S$ that infects x at time 4 and some vertex $v \in N(x)$ at time 3 such that v has a neighbor in Z and another neighbor s that is infected at time 2 and that has also a neighbor in Z . This should be enough because, if s has a neighbor in Z , then $|C_{Z, s, 2}| \leq 3$ and, if v also has a neighbor in Z , then $|C_{Z, v, 3}| \leq |C_{Z, s, 2}| + 1 \leq 3 + 1 = 4$.

Since there is no vertex $v \in N(x)$ infected at time 3 by S such that $|C_{S, v, 3}| \leq 4$, then there is no vertex $v \in N(x)$ that is infected at time 3 by S such that v has a neighbor in S and another neighbor s that is infected at time 2 and has a neighbor in S .

So, if there is a vertex in $N(x)$ infected at time 3 by S that has a neighbor in S , let v be that vertex and let $P = S$. If not, let v be any vertex in $N(x)$ that is infected at time 3 by S , has a neighbor r infected at time 1 by S and has another neighbor s infected at time 2 by S that has a neighbor in S . If there is no such vertex v , let v be any vertex in $N(x)$ that is infected at time 3 by S and has a neighbor r infected at time 1 by S , or, if there is no such vertex v , let v be any vertex in $N(x)$ that is infected at time 3 by S and let r be any vertex adjacent to v infected at time 2 by S . Let $P = S \cup \{r\}$.

We have that, since G is bipartite, all vertices infected at time ≥ 2 by S that are adjacent to any vertex in $N(x)$ are at distance ≥ 2 of r , except for r itself, and, therefore, by Lemma 3.3, all vertices infected at time ≥ 2 by S that are adjacent to any vertex in $N(x)$ are infected at time ≥ 2 by P .

Additionally, if r is infected at time 2 by S , by the choice of r , there is no vertex in $N(x)$ infected at time 3 by S that has one neighbor infected at time ≤ 1 by S and, since all vertices infected at time ≥ 4 by S cannot be adjacent to r and another vertex infected at time ≤ 1 by S , then, if r is infected at time 2 by S , all vertices in $N(x)$ that are infected at time ≥ 3 by S have at most one neighbor infected at time ≤ 1 by P .

If r is infected at time 1 by S , then all vertices in $N(x)$ infected at time ≥ 3 by S cannot be adjacent to r and another vertex infected at time ≤ 1 by S because, otherwise, they would be infected at time ≤ 2 by S . Then, if r is infected at time 1 by S , all vertices in $N(x)$ that are infected at time ≥ 3 by S have at most one neighbor infected at time ≤ 1 by P .

Thus, whichever the infection time of r by S , we have that all vertices in $N(x)$ that are infected at time ≥ 3 by S have at most one neighbor infected at time ≤ 1 by P . Thus, all vertices in $N(x)$ that are infected at time ≥ 3 by S are infected at time ≥ 3 by P . Hence, v is infected at time ≥ 3 and x at time ≥ 4 by P . Thus, by Lemma 3.4, v is infected at time 3 and x at time 4 by P .

Thus, there is a vertex v in $N(x)$ that is infected at time 3 by P such that v has a neighbor that is infected at time 0 by the hull set P and x is infected at time 4 by P . At this point, if there is some vertex v' in $N(x)$ such that P infects v' at time 3 and $|C_{P,v',3}| \leq 4$, let $Z = P$ and we are done. If not, henceforth, assume that there is not such vertex v' .

Note that if v has a neighbor r infected at time 1 by S and has another neighbor infected at time 2 by S that has a neighbor in S , then $|C_{P,v,3}| \leq 4$. Thus, since there is no vertex v' in $N(x)$ such that P infects v' at time 3 and $|C_{P,v',3}| \leq 4$, and all vertices infected at time ≥ 2 by S and adjacent to some vertex in $N(x)$ are infected at time ≥ 2 by P , then there is no vertex in $N(x)$ infected at time 3 by P that has a neighbor infected at time 1 by P and has another neighbor infected at time 2 by P that has a neighbor in P .

Thus, we have that all vertices in $N(x)$ infected at time 3, including v , that are neighbor of some vertex in P do not have a neighbor infected at time 2 that also has a neighbor in P . Then, let s be any vertex adjacent to v that is infected at time 2 by P . Since v has a neighbor in P , we have that s does not have a neighbor in P . Let z be any vertex adjacent to s that is infected at time 1 by P . Let $Z = P \cup \{z\}$. Let us prove that Z infects x at time 4, v at time 3 and s at time 2. Using the Lemma 3.4, we have that is enough to prove that Z infects x at time ≥ 4 , v at time ≥ 3 and s at time ≥ 2 . First, since s has no neighbor in P , then it has at most one neighbor in Z . Therefore s is infected at time ≥ 2 by Z . Now, assume, by contradiction, that Z infects some vertex $q \in N(x)$ that is infected at time ≥ 3 by P at time ≤ 2 . Since $q \neq z$ and P infects q at time 3 and Z infects q at time ≤ 2 , by Lemma 3.3, the vertices z and q are at distance either 1 or 2 of each other, but, since G is bipartite, z and q are at distance 2 of each other. Therefore, since q cannot have a neighbor infected at time 1 by P and another neighbor q' that is infected at time ≥ 2 by P that has a neighbor in P , we have that only two cases can happen.

In the first case, the vertex q has a neighbor in Z and another neighbor q' that is infected at time ≥ 2 by P and at time 1 by Z . In this case, we have that z is adjacent to q' and, since $Z = P \cup \{z\}$, q' must have a neighbor in P . Since z and q are at distance 2 of each other, then the vertex adjacent to q in Z cannot be z and, therefore, q has a neighbor in P . Thus, since z is infected at time 1 by P and q and q' have a neighbor in P , then, in fact, q' is infected at time 2 and q is infected at time 3 by P . Thus, since q' has a neighbor in P , then $|C_{P,q',2}| \leq 3$. Also, since q has a neighbor in P , we have that $|C_{P,q,3}| \leq |C_{P,q',2}| + 1 \leq 3 + 1 = 4$, which is a contradiction because we assumed that there is no vertex v in $N(x)$ that is infected at time 3 by P such that $|C_{P,v,3}| \leq 4$.

In the second, case we have that q has two neighbors q_1 and q_2 that are infected at time ≥ 2 by P and at time 1 by Z . In this case, we have that z is adjacent to both q_1 and q_2 and, also, we have that both q_1 and q_2 have a neighbor in Z that is different from z , implying that both q_1 and q_2 have a neighbor in P . Thus, since q_1 and q_2 have a neighbor in P and z is infected at time 1 by P , then q_1 and q_2 are infected at time 2 by P and, therefore, q is infected at time 3 by P . So, since z is adjacent to both q_1 and q_2 , z is infected by P at time 1 and both q_1 and q_2 , which are infected at time 2 by P , have a neighbor in P , then $|C_{P,q,3}| \leq |C_{P,q_1,2}| + |C_{P,q_2,2}| \leq |C_{P,z,1}| + 1 + 1 = 4$, which is a contradiction because we assumed that there is no vertex v in $N(x)$ infected at time 3 by P such that $|C_{P,v,3}| \leq 4$.

Thus, we have that Z infects at time ≥ 3 all vertices in $N(x)$ that are infected at time ≥ 3 by P . Hence, Z infects v at time ≥ 3 and x at time ≥ 4 . Also, since v and s have a neighbor in Z , then Z is the set we are

looking for. ■

Thus, there is a hull set Z that is superset of S that infects x at time 4 and some vertex v in $N(x)$ at time 3 such that $|C_{Z,v,3}| \leq 4$. Thus, if there is a vertex v' in $N(x)$ that is infected at time $t \leq 2$ by Z , let $F = C_{Z,v,3} \cup C_{Z,v',t}$. Thus, we have that $|F| \leq |C_{Z,v,3}| + |C_{Z,v',t}| \leq 4 + 2^t \leq 8$. Since F infects v at time ≤ 3 and v' at time ≤ 2 , then F infects x at time ≤ 4 . Additionally, since $T_0 \cup N_{\geq 4}(u) \subseteq S \subseteq Z$ and $F \subseteq Z$, then $T_0 \cup N_{\geq 4}(u) \cup F \subseteq Z$. Since x is infected by Z at time 4 and by F at time ≤ 4 , $|F| \leq 8$, and $T_0 \cup N_{\geq 4}(u) \cup F \subseteq Z$, letting $R = Z$, R and F are the sets we are looking for.

So, henceforth, assume that all vertices in $N(x)$ are infected at time ≥ 3 by Z . Then, in this case, let us prove the following fact:

Fact 2. *There is a hull set $R \supseteq Z$ and a vertex $v' \neq v$ in $N(x)$ that is infected at time 3 by Z such that R infects x at time 4, infects v at time $t \leq 3$, infects v' at time 3, $|C_{R,v,t}| \leq 4$ and $|C_{R,v',3}| \leq 4$.*

PROOF. To prove what we want, it is enough to prove that there is a hull set R that infects x at time 4 and some vertex $v' \in N(x) - \{v\}$ at time 3 such that v' has a neighbor in R and another neighbor s that is infected by R at time 2 and has a neighbor in R . This is because, since v' and s have a neighbor in R then $|C_{R,s,2}| \leq 3$ and $|C_{R,v',3}| \leq |C_{R,s,2}| + 1 \leq 4$. Also, since $Z \subseteq R$ and Z infects v at time 3, then, by Lemma 3.4, R infects v at time $t \leq 3$ and, since v has a neighbor in Z , and consequently in R , then $|C_{R,v,t}| \leq 4$.

If there is a vertex $v' \in N(x) - \{v\}$ that is infected at time 3 by Z such that $|C_{Z,v',3}| \leq 4$, let $R = Z$ and we are done. If not, henceforth, assume that there is not such vertex v' .

So, if there is a vertex in $N(x) - \{v\}$ infected at time 3 by Z that has a neighbor in Z , let v' be that vertex and let $P = Z$. If not, let v' be any vertex in $N(x) - \{v\}$ that is infected at time 3 by Z , has a neighbor r infected at time 1 by Z and has another neighbor s infected at time 2 by Z that has a neighbor in Z . If there is no such vertex v' , let v' be any vertex in $N(x) - \{v\}$ that is infected at time 3 by Z and has a neighbor r infected at time 1 by Z , or, if there is no such vertex v' , let v' be any vertex in $N(x) - \{v\}$ that is infected at time 3 by Z and let r be any vertex adjacent to v' infected at time 2 by Z . Let $P = Z \cup \{r\}$.

We have that, since G is bipartite, all vertices infected at time ≥ 2 by Z that are adjacent to any vertex in $N(x)$ are at distance ≥ 2 of r , except for r itself, and, therefore, by Lemma 3.3, all vertices infected at time ≥ 2 by Z that are adjacent to any vertex in $N(x) - \{v\}$ are infected at time ≥ 2 by P .

Additionally, if r is infected at time 2 by Z , by the choice of r , there is no vertex in $N(x) - \{v\}$ infected at time 3 by Z that has one neighbor infected at time ≤ 1 by Z and, since all vertices infected at time ≥ 4 by Z cannot be adjacent to r and another vertex infected at time ≤ 1 by Z , then, if r is infected at time 2 by Z , all vertices in $N(x) - \{v\}$ that are infected at time ≥ 3 by Z have at most one neighbor infected at time ≤ 1 by P .

If r is infected at time 1 by Z , then all vertices in $N(x) - \{v\}$ infected at time ≥ 3 by Z cannot be adjacent to r and another vertex infected at time ≤ 1 by Z because, otherwise, they would be infected at time ≤ 2 by Z . Then, if r is infected at time 1 by Z , all vertices in $N(x) - \{v\}$ that are infected at time ≥ 3 by Z have at most one neighbor infected at time ≤ 1 by P .

Thus, whichever the infection time of r by Z , we have that all vertices in $N(x) - \{v\}$ that are infected at time ≥ 3 by Z have at most one neighbor infected at time ≤ 1 by P . Thus, all vertices in $N(x) - \{v\}$ that are infected at time ≥ 3 by Z , which is all vertices in $N(x) - \{v\}$, are infected at time ≥ 3 by P . Hence, v' is infected at time ≥ 3 and x at time ≥ 4 by P . Thus, by Lemma 3.4, v is infected at time 3 and x at time 4 by P .

Thus, there is a vertex v' in $N(x) - \{v\}$ that is infected at time 3 by P such that v' has a neighbor in P and x is infected at time 4 by P . At this point, if there is some vertex w in $N(x) - \{v\}$ such that P infects w at time 3 and $|C_{P,w,3}| \leq 4$, let $R = P$ and we are done. If not, henceforth, assume that there is not such vertex w .

Note that if v' has a neighbor r infected at time 1 by Z and has another neighbor infected at time 2 by Z that has a neighbor in Z , then $|C_{P,v',3}| \leq 4$. Thus, since there is no vertex w in $N(x) - \{v\}$ such that P infects w at time 3 and $|C_{P,w,3}| \leq 4$, and all vertices infected at time ≥ 2 by Z and adjacent to some vertex in $N(x) - \{v\}$ are infected at time ≥ 2 by P , then there is no vertex in $N(x) - \{v\}$ infected at time 3 by P that has a neighbor infected at time 1 by P and has another neighbor infected at time 2 by P that has a neighbor in P .

Thus, we have that all vertices in $N(x) - \{v\}$ infected at time 3, including v' , that are neighbor of some vertex in P do not have a neighbor infected at time 2 that also has a neighbor in P . Then, let s be any vertex adjacent to v' that is infected at time 2 by P . Since v' has a neighbor in P , we have that s does not have a neighbor in P . Let z be any vertex adjacent to s that is infected at time 1 by P . Let $R = P \cup \{z\}$. Let us prove that R infects x at time 4, v' at time 3 and s at time 2. Using the Lemma 3.4, we have that is enough to prove that R infects x at time ≥ 4 , v' at time ≥ 3 and s at time ≥ 2 . First, since s has no neighbor in P , then it has at most one neighbor in R . Therefore s is infected at time ≥ 2 by R . Now, assume, by contradiction, that R infects some vertex $q \in N(x) - \{v\}$ that is infected at time ≥ 3 by P at time ≤ 2 . Since $q \neq z$ and P

infects q at time 3 and R infects q at time ≤ 2 , by Lemma 3.3, the vertices z and q are at distance either 1 or 2 of each other, but, since G is bipartite, z and q are at distance 2 of each other. Therefore, since q cannot have a neighbor infected at time 1 by P and another neighbor q' that is infected at time ≥ 2 by P that has a neighbor in P , we have that only two cases can happen.

In the first case, the vertex q has a neighbor in R and another neighbor q' that is infected at time ≥ 2 by P and at time 1 by R . In this case, we would have that z is adjacent to q' and, since $R = P \cup \{z\}$, q' must have a neighbor in P . Since z and q are at distance 2 of each other, then the vertex adjacent to q in R cannot be z and, therefore, q has a neighbor in P . Thus, since z is infected at time 1 by P and q and q' have a neighbor in P , then, in fact, q' is infected at time 2 and q is infected at time 3 by P . Thus, since q' has a neighbor in P , then $|C_{P,q',2}| \leq 3$. Also, since q has a neighbor in P , we have that $|C_{P,q,3}| \leq |C_{P,q',2}| + 1 \leq 3 + 1 = 4$, which is a contradiction because we assumed that there is no vertex w in $N(x) - \{v\}$ that is infected at time 3 by P such that $|C_{P,w,3}| \leq 4$.

In the second case, we have that q has two neighbors q_1 and q_2 that are infected at time ≥ 2 by P and at time 1 by R . In this case, we have that z is adjacent to both q_1 and q_2 and, also, we have that both q_1 and q_2 have a neighbor in R that is different from z , implying that both q_1 and q_2 have a neighbor in P . Thus, since q_1 and q_2 have a neighbor in P and z is infected at time 1 by P , then q_1 and q_2 are infected at time 2 by P and, therefore, q is infected at time 3 by P . So, since z is adjacent to both q_1 and q_2 , z is infected by P at time 1 and both q_1 and q_2 , which are infected at time 2 by P , have a neighbor in P , then $|C_{P,q,3}| \leq |C_{P,q_1,2}| + |C_{P,q_2,2}| \leq |C_{P,z,1}| + 1 + 1 = 4$, which is a contradiction because we assumed that there is no vertex w in $N(x) - \{v\}$ infected at time 3 by P such that $|C_{P,w,3}| \leq 4$.

Thus, we have that R infects at time ≥ 3 all vertices in $N(x) - \{v\}$ that are infected at time ≥ 3 by P , which is all vertices in $N(x) - \{v\}$. Therefore, R infects v' at time ≥ 3 and x at time ≥ 4 . Also, since v' and s have a neighbor in R , then R is the set we are looking for. ■

Thus, there is a hull set R that is superset of Z that infects x at time 4, some vertex $v \in N(x)$ at time $t \leq 3$ and some other vertex $v' \in N(x)$ at time 3 such that $|C_{R,v,t}| \leq 4$ and $|C_{R,v',3}| \leq 4$. Let $F = C_{R,v,t} \cup C_{R,v',3}$. Thus, we have that $|F| \leq |C_{R,v,t}| + |C_{R,v',3}| \leq 4 + 4 = 8$. Since F infects v at time ≤ 3 and v' at time ≤ 3 , then F infects x at time ≤ 4 . Additionally, since $T_0 \cup N_{\geq 4}(u) \subseteq S \subseteq Z \subseteq R$ and $F \subseteq R$, then $T_0 \cup N_{\geq 4}(u) \cup F \subseteq R$. So, since x is infected by R at time 4 and by F at time ≤ 4 , $|F| \leq 8$, and $T_0 \cup N_{\geq 4}(u) \cup F \subseteq R$, then R and F are the sets we are looking for. ■

8. Final Remarks

In this paper, we showed the Percolation Time Problem is polynomial for a fixed $k = 3$ by finding polynomially computable characterization for graphs that have the percolation time ≥ 3 . Also, when the problem is restricted only to bipartite graphs, we showed that the Percolation Time Problem for a fixed $k \geq 5$ is NP-Complete. However, we found polynomially computable characterizations for graphs that have the percolation time ≥ 3 and ≥ 4 . Regarding the polynomial cases, we derived from the characterizations algorithms that run in times $O(mn^5)$, $O(mn^3)$ and $O(m^2n^9)$ that solves the Percolation Time Problem when, respectively, the parameter k is fixed in 3 and, in bipartite graphs, when k is fixed in 3 and 4.

The results in [7] along with the results in this work set the threshold for the fixed parameter k in which the Percolation Time Problem ceases to be polynomial and becomes NP-complete.

We are hoping that these results, the results in [7] regarding NP-completeness of the Percolation Time Problem for planar graphs and future results regarding the NP-completeness of the Percolation Time Problem in restricted degree graphs shed some light in the following question: Can the Percolation Time Problem in subgraphs and induced subgraphs of d -dimensional grids be solved in polynomial time or this problem in these types of graphs is NP-Complete?

Other interesting questions arises from this work, such as, can the complexity of the algorithms, which, in this case, is directly related to the maximum size of the sets of vertices that initially infects some vertex at time k , be improved? Also, is there a relation between the P_3 -Carathodory number [31] and the maximum size of the sets of vertices that initially infects some vertex at time k ?

References

- [1] Amini, H.: Bootstrap percolation in living neural networks. J. Stat. Phys. 141, no. 3, 459–475 (2010)
- [2] Balogh, J., Bollobás, B.: Bootstrap percolation on the hypercube. Probab. Theory Related Fields 134, no. 4, 624–648 (2006)
- [3] Balogh, J., Bollobás, B., Duminil-Copin, H., Morris, R.: The sharp threshold for bootstrap percolation in all dimensions. Trans. Amer. Math. Soc. 364, no. 5, 2667–2701 (2012)

- [4] Balogh, J., Pete, G.: Random disease on the square grid. *Random Structures and Algorithms* 13, 409–422 (1998)
- [5] Balogh, J., Bollobás, B., Morris, R.: Bootstrap percolation in three dimensions. *Ann. Probab.* 37, no. 4, 1329–1380 (2009)
- [6] Balogh, J., Bollobás, B., Morris, R.: Bootstrap percolation in high dimensions. *Combin. Probab. Comput.* 19, no. 5–6, 643–692 (2010)
- [7] Benevides, F., Campos, V., Dourado, M.C., Sampaio, R.M., Silva, A.: The maximum time of 2-neighbor bootstrap percolation: algorithmic aspects. *European Journal of Combinatorics* 48, 88–99 (2015)
- [8] Benevides, F., Przykucki, M.: Maximum percolation time in two-dimensional bootstrap percolation. *SIAM Journal on Discrete Mathematics* 29, 224–251 (2015).
- [9] Benevides, F., Przykucki, M.: On slowly percolating sets of minimal size in bootstrap percolation. *The Electronic Journal of Combinatorics* 20, no. 2, 46pp (2013)
- [10] B.Bollobás and O. Riordan: *Percolation*. Cambridge Univ. Press (2006)
- [11] Bollobás, B., Smith, P.J., Uzzell, A.J.: The time of bootstrap percolation with dense initial sets for all thresholds. *Random Structures & Algorithms* 47, no. 1, 1–29 (2015).
- [12] Calder, J.: Some elementary properties of interval convexities. *J. London Math. Soc.* 3, 422–428 (1971)
- [13] Centeno, C., Dourado, M.C., Penso, L., Rautenbach, D., Szwarcfiter, J.L.: Irreversible conversion of graphs. *Theoretical Computer Science* 412, 3693–3700 (2011)
- [14] G. Chae, E. M. Palmer, and W. Siu: Geodetic number of random graphs of diameter 2. *The Australasian Journal of Combinatorics* 26, 11–20 (2002)
- [15] Chalupa, J., Leath, P.L., Reich, G.R.: Bootstrap percolation on a Bethe lattice. *J. Phys. C* 12, no. 1, 31–35 (1979)
- [16] Chen, N.: On the Approximability of Influence in Social Networks. *SIAM J. Discrete Math.* 23, no. 3, 1400–1415 (2009)
- [17] Dahlhaus, E., Johnson, D., Papadimitriou, C., Seymour, E., Yannakakis, M.: The Complexity of multiterminal cuts. *SIAM J. Comp.* 23, no. 4, 864–894 (1994)
- [18] Dreyer, P.A., Roberts, F.S.: Irreversible k-threshold processes: Graph-theoretical threshold models of the spread of disease and of opinion. *Discrete Appl. Math.* 157, no. 7, 1615–1627 (2009)
- [19] Duchet, P.: Convex sets in graphs, II. Minimal path convexity. *J. Comb. Theory B* 44, 307–316 (1988)
- [20] Erdős, P., Fried, E., Hajnal, A., Milner, E.C.: Some remarks on simple tournaments. *Algebra Univers.* 2, 238–245 (1972)
- [21] Farber, M., Jamison, R.E.: Convexity in graphs and hypergraphs. *SIAM J. Algebraic Discrete Methods* 7, 433–444 (1986)
- [22] Fey, A., Levine, L., Peres, Y.: Growth rates and explosions in sandpiles. *J. Stat. Phys.* 138, 143–159 (2010)
- [23] Harary, F., Nieminen, J.: Convexity in graphs. *Journal of Differential Geometry* 16, 185–190 (1981)
- [24] Holroyd, A.E.: Sharp metastability threshold for two-dimensional bootstrap percolation. *Probab. Theory Related Fields* 125, no. 2, 195–224 (2003)
- [25] Levi, F.W.: On Helly’s theorem and the axioms of convexity. *J. Indian Math. Soc.* 15, 65–76 (1951)
- [26] Lichtenstein, D.: Planar Formulae and Their Uses. *SIAM J. Comput.* 11, 329–343 (1982)
- [27] Morris, R.: Minimal percolating sets in bootstrap percolation. *The Electronic Journal of Combinatorics* 16, no. 1, 20pp (2009)
- [28] Przykucki, M.: Maximal Percolation Time in Hypercubes Under 2-Bootstrap Percolation. *The Electronic Journal of Combinatorics* 19, no. 2, 41pp (2012)
- [29] Riedl, E.: Largest minimal percolating sets in hypercubes under 2-bootstrap percolation. *The Electronic Journal of Combinatorics* 17, no. 1, 13pp (2010)

- [30] Van de Vel, M.L.J.: Theory of Convex Structures. North-Holland, Amsterdam (1993)
- [31] Barbosa, R.M., Coelho, E.M.M., Dourado, M.C., Rautenbach, D., Szwarcfiter, J.L.: On the Carathodory number for the convexity of paths of order three. SIAM J. Discrete Math. 26, 929–939 (2012)

APÊNDICE B – PERCOLATION TIME PROBLEM IN GRID GRAPHS AND SOME PARAMETERIZED RESULTS

Neste apêndice, as provas completas em inglês dos resultados submetidos e aceitos no congresso *41th International Workshop on Graph-Theoretic Concepts in Computer Science* (WG-2015) em 2015 cujos anais foram publicados no *Lecture Notes in Computer Science* [88] são apresentadas. Tais resultados também foram submetidos ao periódico *Discrete Applied Mathematics* e estão todos também relacionados ao problema TEMPO MÁXIMO DE INFECÇÃO NA CONVEXIDADE P_3 . Eles são:

- A **NP**-completude para grafos grade de grau máximo três;
- A **NP**-completude para $k = n - n^\varepsilon + 4$, onde $0 < \varepsilon \leq 1$;
- A **NP**-completude para grafos de grau máximo fixo pelo menos quatro e $k = \Theta(\log n)$;
- A **W[1]**-dificuldade no parâmetro $tw(G)$;
- Um algoritmo polinomial para grafos grade sólidos de grau máximo três;
- Um algoritmo polinomial para o tempo $k = n - c$, onde c é constante;
- Um algoritmo FPT no parâmetro $\Delta(G) + k$; e
- Um algoritmo FPT no parâmetro $tw(G) + k$.

The maximum time of 2-neighbour bootstrap percolation in grid graphs and some parameterized results

Thiago Marcilon^a, Rudini Sampaio^a

^a*Department of Computer Science, ParGO Research Group, Federal University of Ceará, Brazil.*

Abstract

In 2-neighborhood bootstrap percolation on a graph G , an infection spreads according to the following deterministic rule: infected vertices of G remain infected forever and in consecutive rounds healthy vertices with at least two already infected neighbors become infected. Percolation occurs if eventually every vertex is infected. The maximum time $t(G)$ is the maximum number of rounds needed to eventually infect the entire vertex set. In 2013, it was proved by Benevides et al [3] that $t(G)$ is NP-hard for planar graphs and that deciding whether $t(G) \geq k$ is polynomial time solvable for $k \leq 2$, but is NP-complete for $k \geq 4$. They left two open problems which are the complexities for $k = 3$ and for planar bipartite graphs. In 2014, we solved the first problem[17]. In this paper, we solve the second one by proving that $t(G)$ is NP-complete even in grid graphs with maximum degree 3. We also prove that $t(G)$ is polynomial time solvable for solid grid graphs with maximum degree 3. Moreover, we prove that the percolation time problem is W[1]-hard on the treewidth of the graph, but it is fixed parameter tractable with parameters $\text{treewidth}+k$ and $\text{maxdegree}+k$.

Keywords: 2-neighbor bootstrap percolation, maximum percolation time, grid graph, fixed parameter tractability, treewidth

1. Introduction

We consider a problem in which an infection spreads over the vertices of a connected simple graph G following a deterministic spreading rule in such a way that an infected vertex will remain infected forever. Given a set $S \subseteq V(G)$ of initially infected vertices, we build a sequence S_0, S_1, S_2, \dots in which $S_0 = S$ and S_{i+1} is obtained from S_i using such spreading rule.

Under r -neighbor bootstrap percolation on a graph G , the spreading rule is a threshold rule in which S_{i+1} is obtained from S_i by adding to it the vertices of G that have at least r neighbors in S_i . We say that a set S infects a vertex v at time i if $v \in S_i \setminus S_{i-1}$. Let, for any set of vertices S and vertex v of G , $t_r(G, S, v)$ be the minimum t such that v belongs to S_t or, if there is no t such that v belongs to S_t , then $t_r(G, S, v) = \infty$. Also, we say that a set S_0 infects G , or that S_0 is a percolating set of G , if eventually every vertex of G becomes infected, that is, there exists a t such that $S_t = V(G)$. If S is a percolating set of G , then we define $t_r(G, S)$ as the minimum t such that $S_t = V(G)$. Also, define the *percolation time of G* as $t_r(G) = \max\{t_r(G, S) : S \text{ is a percolating set of } G\}$. In this paper, we shall focus on the case where $r = 2$ and in such case we omit the subscript of the notations $t_r(G, S)$ and $t_r(G)$. Also, from the notation $t(G, S)$ and $t(G, S, v)$, when the parameter G is clear from context, it will be omitted.

Several papers investigated extremal questions regarding bootstrap percolation. For example, what is the smallest or largest size of a percolating set with a given property? Morris [18] and Riedl [21] studied the maximum size of minimal percolating sets on the square grid $[n]^2$ and the hypercube $\{0, 1\}^d$, respectively, answering a question posed by Bollobás. However, the problem of finding the smallest percolating set is NP-hard even on subgraphs of the square grid [2] and it is APX-hard even for bipartite graphs with maximum degree four [8]. Moreover, it is hard [7] to approximate within a ratio $O(2^{\log^{1-\varepsilon} n})$, for any $\varepsilon > 0$, unless $NP \subseteq DTIME(n^{\text{polylog}(n)})$.

Another type of question is: what is the minimum or maximum time that percolation can take, given that S_0 satisfies certain properties? Recently, Przykucki [20] determined the precise value of the maximum percolation

Email addresses: thiagomarcilon@gmail.com (Thiago Marcilon), rudini@lia.ufc.br (Rudini Sampaio)

time on the hypercube $2^{[n]}$ as a function of n , and Benevides and Przykucki [4, 5] have similar results for the square grid $[n]^2$, also answering a question posed by Bollobás. In particular, they have a polynomial time dynamic programming algorithm to compute the maximum percolation time on rectangular grids [4].

Here, we consider the decision version of the Percolation Time Problem, as stated below.

PERCOLATION TIME

Input: A graph G and an integer k .

Question: Is $t(G) \geq k$?

In 2013, Benevides et.al. [3], among other results, proved that the Percolation Time Problem is polynomial time solvable for $k \leq 2$, but is NP-complete for $k \geq 4$ and, when restricted to bipartite graphs, it is NP-complete for $k \geq 7$. Moreover, it was proved that the Percolation Time Problem is NP-complete for planar graphs. They left three open questions about the complexity for $k = 3$ in general graphs, the complexity for $3 \leq k \leq 6$ in bipartite graphs and the complexity for planar bipartite graphs.

In 2014, the first and the second questions were solved [17]: it was proved that the Percolation Time Problem is $O(mn^5)$ -time solvable for $k = 3$ in general graphs and, when restricted to bipartite graphs, it is $O(mn^3)$ -time solvable for $k = 3$, it is $O(m^2n^9)$ -time solvable for $k = 4$ and it is NP-complete for $k \geq 5$.

In this paper, we solve the third question of [3]. We prove that the Percolation Time Problem is NP-complete for planar bipartite graphs. In fact, we prove a stronger result: the NP-completeness for grid graphs, which are induced subgraphs of grids, with maximum degree 3.

There are NP-hard problems in grid graphs which are polynomial time solvable for solid grid graphs. For example, the Hamiltonian cycle problem is NP-complete for grid graphs [14], but it is polynomial time solvable for solid grid graphs [24]. Motivated by the work of [4] for rectangular grids, we obtain in this paper a polynomial time algorithm for solid grid graphs with maximum degree 3.

Finally, we prove several complexity results for $t(G)$ in graphs with bounded maximum degree and bounded treewidth, some of which implies fixed parameter tractable algorithms for the Percolation Time Problem. We prove that the percolation time problem is W[1]-hard on the treewidth of the graph, but it is fixed parameter tractable with parameters treewidth+ k and maxdegree+ k .

2. Percolation Time Problem in grid graphs with $\Delta = 3$

In this section, we prove that the Percolation Time Problem is NP-complete in grid graphs with maximum degree $\Delta = 3$. We also show that, when the graph is a grid graph with $\Delta = 3$ and $k = O(\log n)$, the Percolation Time Problem can be solved in polynomial time. But, first, let us define a S -infection path and, then, prove two lemmas that will be useful in the proofs.

Let S be a percolating set. A path $P = v_0, v_1, \dots, v_n$ is a S -infection path if, for every $0 \leq i \leq n - 1$, $t(S, v_i) < t(S, v_{i+1})$. Notice that, if $t(S, v) = k$, then there is a S -infection path $v_0, v_1, \dots, v_k = v$, where $t(S, v_i) = i$ for each $0 \leq i \leq k$.

Roughly speaking, the next lemma shows that, if the S -infection time of a vertex v is decreased by the inclusion of a vertex v' to S , then there is an infection path starting at v' and ending at v .

Lemma 1. *Let S be a subset of $V(G)$, v, v' be vertices of G and $S' = S \cup \{v'\}$. If $t(S', v) < t(S, v)$, then there is a S' -infection path starting in v' and ending in v .*

Proof. Notice that, since $t(S', v) < t(S, v)$, then $t(S', v) < \infty$, i.e., S' infects v . Also, $t(S', v)$ cannot be 0; otherwise, $v \in S'$ and then $S = S'$ and $t(S, v) = t(S', v)$, a contradiction. Thus $t(S', v) \geq 1$.

Let us prove by induction on $t(S', v)$ that there is a S' -infection path starting in v' and ending in v . For $t(S', v) = 1$, we have that v must be neighbor of two vertices in S' , where one of these two vertices must be v' because, otherwise, $t(S, v) = 1 = t(S', v)$. Thus, the path v', v is a S' -infection path.

Now, suppose that the theorem holds for all values less than k . Let us prove that the theorem still holds if $t(S', v) = k$. We have that v must have a neighbor u such that $t(S', u) < t(S, u)$ and $t(S', u) < k$ because, otherwise, we would have that $t(S', v) = t(S, v)$. Thus, by our inductive hypothesis, there is a S' -infection path from some vertex v' to u and, since $t(S', u) < t(S', v) = k$, there is a S' -infection path from v' to v . \square

The next lemma, which is valid for every graph with maximum degree 3, is the main technical lemma of this section.

Lemma 2. *Let G be a connected graph with $\Delta(G) = 3$ and k a non-negative integer. Then, $t(G) \geq k$ if and only if G has an induced path P where either all vertices of $V(P)$ have degree 3 and $|E(P)| \geq 2k - 2$ or all vertices of $V(P)$ have degree 3, except for one of its extremities, which has degree 2, and $|E(P)| \geq k - 1$.*

Proof. First, suppose that $t(G) \geq k$. Let us prove that G has an induced path P where either all vertices in $V(P)$ have degree 3 and $|E(P)| \geq 2k - 2$ or all vertices in $V(P)$ have degree 3, except for one of its extremities, which has degree 2, and $|E(P)| \geq k - 1$.

Since $t(G) \geq k$, there is a percolating set S such that $t(S) \geq k$. Let $t = t(S)$ and let v be a vertex that is infected by S at time t . Note that v cannot have degree 1, because otherwise $v \in S$, a contradiction. So, let us divide the proof in 2 cases.

The first case occurs when v has degree 2. In this case, let $P = v_1, \dots, v_{t-1}, v_t = v$ be a S -infection path where each v_i is infected at time i by S . Thus, we have that all vertices $v_1, v_2, \dots, v_{t-2}, v_{t-1}$ have degree three because each vertex v_i , for $1 \leq i \leq t-1$, must have two neighbors infected by S at time $\leq i-1$ and, additionally, v_i also has v_{i+1} , which is the next vertex in P , as its neighbor. Thus, since $\Delta = 3$, each vertex $v_i \in V(P)$, for $1 \leq i \leq t-1$, has exactly one neighbor infected at time $i+1$ by S , which is the vertex v_{i+1} , and has no neighbor infected at time $\geq i+2$ by S , which implies that no two non-consecutive vertices in P are neighbors. Therefore, we have that P is an induced path in G such that $|E(P)| = t - 1 \geq k - 1$.

The second case occurs when v has degree 3. Thus, since there is no vertex infected by S at time greater than the time of v , we have that v has one neighbor infected by S at time $\leq t-1$, another neighbor infected by S at time $t-1$ and yet another neighbor infected by S at time either $t-1$ or t . Let t' be the infection time of the neighbor of v' that is infected by S at the greatest time among the neighbors of v , which may be t or $t-1$. Let $P_1 = v_1, \dots, v_{t-1}, v_t = v$ and $P_2 = v'_1, \dots, v'_{t'-1}, v'_{t'}$ be two S -infection paths where each v_i and v'_i is infected at time i by S and $v'_{t'}$ is the neighbor of v that is infected by S at time t' . By the same arguments used in the first case, we have that both P_1 and P_2 are induced paths in G . Additionally, no vertex of P_1 is adjacent to a vertex of P_2 , except the vertex v that is adjacent to $v'_{t'}$ because, otherwise, we would have either that $v_{t-1} = v'_{t'}$ or that $v_t = v'_{t'}$. Thus, let P be the path $v_1, \dots, v_{t-1}, v, v'_{t'}, \dots, v'_2, v'_1$, which is the resulting path from the union of the induced paths P_1 and P_2 through the edge between the vertices v and $v'_{t'}$. We have that P is an induced path in G because there is no edge between two non-consecutive vertices of P . We also have that all vertices in P have degree 3 and $|E(P)| = t + t' - 1 \geq t + (t-1) - 1 \geq 2k - 2$.

Now, suppose that G has an induced path P where either all of its vertices has degree 3 and $|E(P)| \geq 2k - 2$ or all of its vertices has degree 3, except for one of its extremities, which has degree 2, and $|E(P)| \geq k - 1$. Let us prove that $t(G) \geq k$.

Suppose that G has an induced path $P = v_1, v_2, \dots, v_t$, for $t \geq k$, where all of its vertices have degree 3, except for v_t , which has degree 2. Since each vertex v_i has degree 3, except v_t , which has degree 2, and P is an induced path, then each vertex v_i has exactly one neighbor outside of P , except for v_1 , which has two neighbors outside of P . Let S' be the set of neighbors of each v_i that is not in $V(P)$. It is easy to see that S' infects all vertices in P . Since v_t has only two neighbors and one of them is in S' , then $t(S', v_t) = t(S', v_{t-1}) + 1$. Since v_{t-1} has 3 neighbors, where one of them is in S' and the other is v_t , which is infected by S' after v_{t-1} , then $t(S', v_{t-1}) = t(S', v_{t-2}) + 1$. Therefore, by this same argument, we have that, for all $2 \leq i \leq t$, we have that $t(S', v_i) = t(S', v_{i-1}) + 1$. Since v_1 has two neighbors in S' , then v_1 is infected by S' at time 1 and, hence, for all $1 \leq i \leq t$, $t(S', v_i) = i$. Thus, there is a set S' that infects v_t at time t .

Now, suppose that G has an induced path $P = v_1, v_2, \dots, v_{2t-2}, v_{2t-1}$, where all of its vertices have degree 3. Since each vertex v_i has degree 3 and P is an induced path, then each vertex v_i has exactly one neighbor outside of P , except for both v_1 and v_{2t-1} , which have two neighbors outside of P . Let S' be the set of neighbors of each v_i that is not in $V(P)$. Again, it is easy to see that S' infects all vertices in P . Also, similarly to the prior case, it is not hard to see that all vertices at distance d of v_t are infected by S' at time $t - d$, i.e., S' infects a vertex v_i at time $t - |t - i|$.

Thus, in both cases, we have that S' infects each vertex v_i in P at time $t - |t - i|$ (in the first case, since $1 \leq i \leq t$, we have that $i = t - |t - i|$). Let Y be the set of vertices that are neither in $V(P)$ nor in the neighborhood of any vertex in $V(P)$. Let $S = S' \cup Y$. We have that S is a percolating set because all vertices in $V(G)$ are either in $V(P)$ or in the neighborhood of some vertex in $V(P)$ or in Y , and S' infects all vertices that are either in $V(P)$ or in the neighborhood of some vertex in $V(P)$. Also, since all neighbors of the vertices in $V(P)$ are in S' , S cannot possible

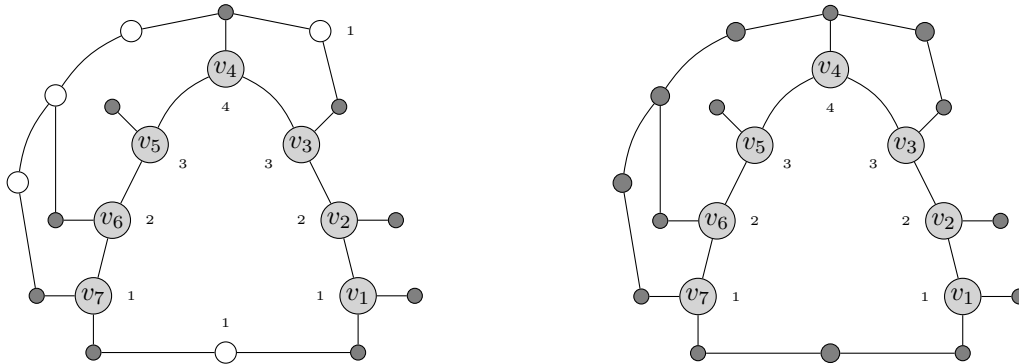


Figure 1: A graph with $\Delta = 3$ infected by the set S' to the left and by the percolating set S to the right.

infect the vertices in $V(P)$ in a different time than S' , as exemplified in Figure 1. Thus, since S is a percolating set that infects v_t at time t , we have that $t(S) \geq t \geq k$ and, hence, $t(G) \geq k$. \square

Before proving the NP-completeness result of this section, we use Lemma 2 to show that the Percolation Time Problem is polynomial time solvable for $k = O(\log n)$ when the graph has maximum degree 3.

Theorem 1. *If G is a graph with maximum degree 3, then deciding whether $t(G) \geq k$ can be done in time $O(n^{2c+3})$ for $k \leq c \cdot \log_2 n$ and $c > 0$.*

Proof. We can decide whether $t(G) \geq k$ by making use of a modified version of the depth-first search. This version of the depth-first search with maximum search depth ℓ traverses all paths with $\ell + 1$ vertices starting from some vertex v . For each $v \in V(G)$, we will run this version of the depth-first search starting in v . If $d(v) = 2$, we run the modified depth-first search with maximum search depth $k - 1$. If $d(v) = 3$, we run the modified depth-first search with maximum search depth $2k - 2$. If there is a vertex v such that the depth-first search that starts in v finds a path that is an induced path, reaches the maximum depth and passes only by vertices of degree 3, except maybe for v , then, by Lemma 2, $t(G) \geq k$. Otherwise, $t(G) < k$.

Now, let us show that this algorithm runs in polynomial time. For each vertex v in G , there are at most $3 \cdot 2^{\ell-2}$ paths of length ℓ in G that starts in v , for any ℓ . In this case, since $\ell \leq 2k - 2$, there are at most $3 \cdot 2^{2k-2} = 3n^{2c}/4$ paths of length ℓ in G that starts in v . Therefore, since we have n vertices and we take time $O(n^2)$ to obtain each path, the algorithm runs in time $O(n^{2c+3})$. \square

Thus, if $k = O(\log n)$, we can find whether $t(G) \geq k$ in polynomial time for every graph G with $\Delta(G) = 3$. However, the following theorem states that the Percolation Time Problem is NP-complete, even when G is restricted to be a grid graph with $\Delta = 3$.

Theorem 2. *Deciding whether $t(G) \geq k$ is NP-complete when the input G is restricted to be a grid graph with $\Delta(G) \leq 3$.*

Proof. Clearly, the problem is in NP. To prove that the problem is also NP-hard, we obtained a reduction from the Longest Path problem with input restricted to be grid graphs with maximum degree 3. The Longest Path problem with input restricted to be grid graphs with maximum degree 3 is a NP-complete problem because the Hamiltonian Path Problem with input restricted to be grid graphs with maximum degree 3 is also NP-complete [19] and there is a trivial reduction from the Hamiltonian Path Problem to the Longest Path problem that does not change the input graph: G has an Hamiltonian Path if and only if G has a path greater or equal to $n - 1$.

Consider the following reduction from the Longest Path Problem's instance (G, k) where G is restricted to be a grid graph with maximum degree 3 to the Percolation Time Problem's instance $(G', 3k + 2)$ where G' is also a grid graph with maximum degree 3: Multiply the scale of the grid G by three. Each edge in G becomes a path in G' with 4 vertices where the vertices at the extremities are vertices that were originally in G . Let us call an *original vertex* the vertices in G' that were originally in G . After that, for each original vertex v , if $d(v) < 3$, add to G' $3 - d(v)$ vertices in any free position in the grid adjacent to v and link them to v . Thus, after we do that, each original vertex has degree 3 in G' . Henceforth, if a vertex in G' is not an original vertex at this point, then

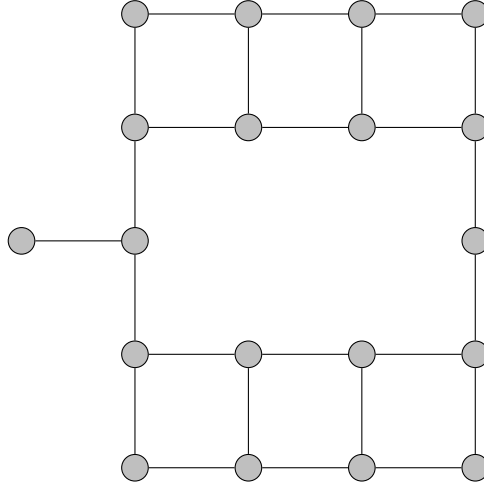


Figure 2: Grid graph with $\Delta = 3$

we will call it an *auxiliary vertex*. Note that each auxiliary vertex is adjacent to exactly one original vertex and one auxiliary vertex, and each original vertex is adjacent to 3 auxiliary vertices.

After that, for each auxiliary vertex v , add a new vertex adjacent to v in the following manner: if the original neighbor of v is located above it, add a vertex adjacent to v at its right position, if there is not one there already, and link it to v . If the original neighbor of v is located below it, add a vertex adjacent to v at its left position, if there is not one there already, and link it to v . If the original neighbor of v is located at its left position, add a vertex adjacent to v at the position below it, if there is not one there already, and link it to v . If the original neighbor of v is located at its right position, add a vertex adjacent to v at the position above it, if there is not one there already, and link it to v . The Figure 3 show how a 4x4 block will look like in G' before and after we add these vertices.

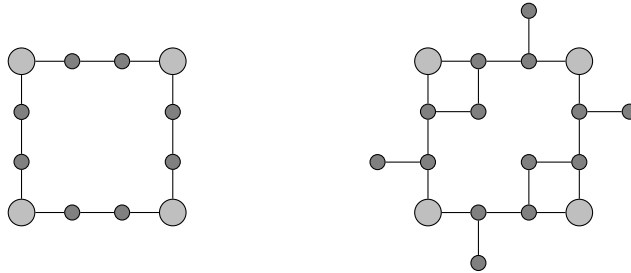


Figure 3: 4x4 block before and after addition of the auxiliary vertices' neighbors

Then, for each auxiliary vertex v , if $d(v) = 2$, add a new vertex adjacent to v in the following position: if the original neighbor of v is at the left position of v , add a vertex adjacent to v at its right position. If the original neighbor of v is at the right position of v , add a vertex adjacent to v at its left position. If the original neighbor of v is below v , add a vertex adjacent to v above v . If the original neighbor of v is above v , add a vertex adjacent to v below v .

Thus, the construction of G' is finished. Figure 4 shows the reduction applied to the grid graph of the Figure 2. Since G is a grid graph and, every time an original vertex and an auxiliary vertex are in adjacent positions in the grid, they are linked, then G' is a grid graph.

By our construction, all original and auxiliary vertices have degree 3 and they are the only vertices that have degree 3. Let us call *corner vertex* all the vertices that have degree 2 in G' . Note that, for each corner vertex, there is exactly one original vertex at distance 2 of it, and, for each original vertex, there is exactly one corner vertex at distance 2 of it. This happens because each original vertex has degree exactly three, so we cannot have two corner vertices at distance 2 of an original vertex. Let f be the bijective function that maps each original vertex to the corner vertex that is at distance 2 of it. Let P be a path in G' that only passes by original and auxiliary vertices and starts with an original vertex. We have that P has length multiple of 3 if and only if it ends in an original

vertex. This is because, for each three consecutive vertices in a path that only passes by original and auxiliary vertices, two are auxiliary vertices and one is an original vertex.

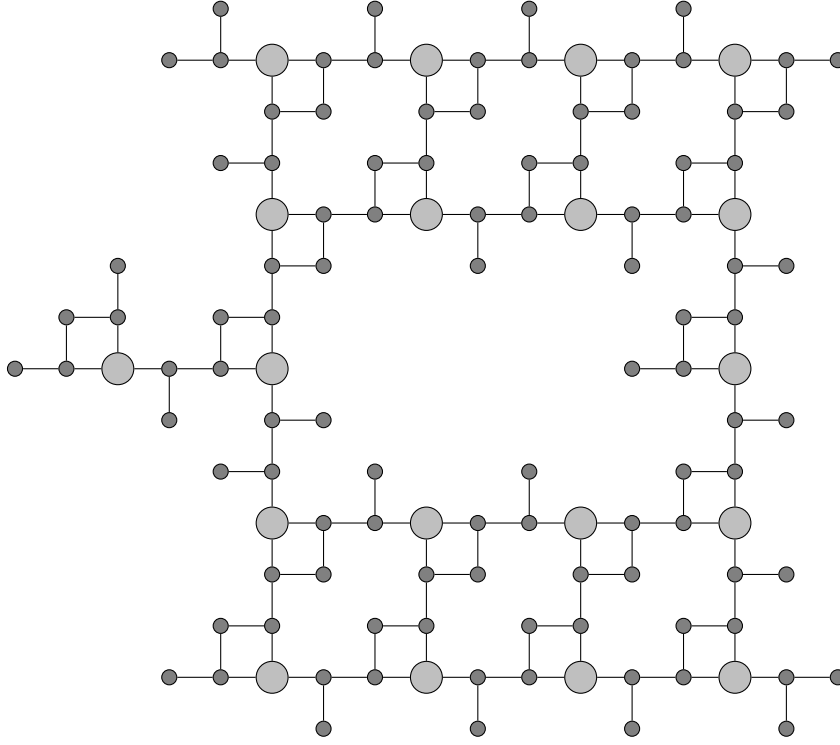


Figure 4: Grid graph resulting from the reduction applied to the grid graph of the Figure 2.

Now, let us prove that G has a path of length $\geq k$ if and only if $t(G') \geq 3k + 2$.

Suppose that G is a grid graph with maximum degree 3 that has a path $P_G = v_1, v_2, \dots, v_{k'}$ of length $k' \geq k$. Let us prove that $t(G') \geq 3k + 2$. Let P be a path in G' such that $P = v_1, a_{1,2}^1, a_{1,2}^2, v_2, a_{2,3}^1, a_{2,3}^2, v_3, \dots, v_{k'}$, where $a_{i,j}^1$ and $a_{i,j}^2$ are the auxiliary vertices resulting from the two subdivisions of the edge v_i, v_j in G . Intuitively, P passes by the same original vertices in G' as P_G does in G and, in general, follows the same direction. Note that P is an induced path whose length is greater or equal to $3k$ and passes by only original and auxiliary vertices. Also, when an auxiliary vertex is in $V(P)$, its auxiliary neighbor is also in $V(P)$.

Let v and v' be the two extremities of P and $f(v') = q'$. Since v is an original vertex, then v has at least one auxiliary neighbor that is not in $V(P)$. Let w be one such auxiliary neighbor of v . All neighbors of w , except v , are not in $V(P)$ because P starts and ends in an original vertex and, if there were a neighbor of w in $V(P)$, this neighbor would have to be the other extremity of P . Let r be the vertex auxiliary neighbor of v' that is in P and let P' be the induced path that we obtain from P by adding w , by removing v' and by adding all vertices in any smallest path between r and q' , excluding r , that has only vertices not adjacent to w and passes only by original and auxiliary vertices, with the exception of q' itself. We have that P' is an induced path because the path $Q = w, v, \dots, r$ that is the path P when we remove v' and add w , is an induced path, the smallest path Q' between r and q' that we appended to Q by r to form P' is also an induced path, and its internal vertices have no neighbors in $V(Q)$, except for r .

Thus, since $|E(Q')| \geq 1$, we have that P' is an induced path with length greater or equal to $3k + 1$ where all of its vertices have degree 3, except for q' , which has degree 2. Therefore, by Lemma 2, we have that $t(G') \geq 3k + 2$.

Now, suppose that G is a grid graph with maximum degree 3 such that, when we apply the reduction to G to create G' , we have that $t(G') \geq 3k + 2$. Let us prove that G has a path of length $\geq k$. Since $t(G') \geq 3k + 2$, applying the Lemma 2, we have that G' has an induced path P where either all vertices in $V(P)$ have degree 3 and $|E(P)| \geq 6k + 2$ or all vertices in $V(P)$ have degree 3, except for one of its extremities, which has degree 2, and $|E(P)| \geq 3k + 1$.

Firstly, suppose that G' has an induced path P where all vertices in $V(P)$ have degree 3 and $|E(P)| \geq 6k + 2$. Since, the only vertices that have degree 3 are the original and auxiliary vertices and for each three consecutive

vertices in P there is one original vertex and two auxiliary vertices, it is easy to see that P has at least $k+1$ original vertices and, thus, there is a path in G of length at least k .

Finally, suppose that G' has an induced path P where all vertices in $V(P)$ have degree 3, except for one of its extremities, which has degree 2, and $|E(P)| \geq 3k+1$. It is enough to analyze the case $|E(P)| = 3k+1$ because, if $|E(P)| > 3k+1$, any subpath of P of length $3k+1$ that starts at the extremity of P that has degree 2 is an induced path where all of its vertices have degree 3, except for one of its extremities, which has degree 2, and has length $3k+1$. So, let us say that P starts in the vertex that has degree 2. Since the only vertices that have degree 2 are corner vertices, then P starts with a corner vertex. Let q be that corner vertex, let $q' = f^{-1}(q)$ and let v be the other extremity of P .

Suppose that P passes by q' . Since P is an induced path, then q' is the third vertex of P . Since q and q' are at distance 2 of each other and $|E(P)| = 3k+1$, then v is an auxiliary vertex which its neighbor that is an original vertex, say v' , is not in P . Let us append v' to P and remove all vertices between q and q' , including q and excluding q' . So, since P starts at q' , an original vertex, ends in v' , another original vertex, and has length $3k$, then there is a path in G of length greater or equal to k .

Now, suppose that P does not pass by q' . Since $|E(P)| = 3k+1$, then v is an auxiliary vertex which its neighbor that is an original vertex, say v' , is in P . Let us remove q , appending q' in its place, and v from P . Thus, since P starts at q' , an original vertex, ends in v' , another original vertex, and has length $3k$, then there is a path in G of length greater or equal to k . \square

3. Percolation Time Problem in solid grid graphs with $\Delta = 3$

A solid grid graph is a grid graph in which all of its bounded faces have area one. There are NP-hard problems in grid graphs that are polynomial time solvable for solid grid graphs. For example, since 1982 it is known that the hamiltonian cycle problem is NP-hard for grid graphs [14], but, in 1997, it was proved that it is polynomial time solvable for solid grid graphs [24]. Motivated by the work of [4] on the maximum percolation time for rectangular grids, we obtain in this section a polynomial time algorithm for solid grid graphs with maximum degree 3. However, the Percolation Time Problem for solid grid graphs with maximum degree 4 is still open. Before we demonstrate the main theorem of this section, let us present some definitions and prove a technical lemma.

Let a ladder L_k , for some integer $k > 0$, be a grid with dimensions $2 \times k$. Let the extremities of a ladder be the four vertices that have only two neighbors in the ladder and let all the other vertices of the ladder be its internal vertices.

Let $d(0, k)$ be the longest induced path in a L_k between the two extremities that are at distance $k-1$ of each other and let $d(1, k)$ be the longest induced path in a L_k between the two extremities that are at distance k .

Lemma 3. *For any $k > 0$, $d(0, k) = k + 2 \cdot \lfloor (k+1)/4 \rfloor - 1$ and $d(1, k) = k + 2 \cdot \lfloor (k-1)/4 \rfloor$.*

Proof. We can manually verify that $d(0, 1) = 0$, $d(0, 2) = 1$, $d(1, 1) = 1$ and $d(1, 2) = 2$. Let us prove that, for every $k \geq 3$ and $m \in \{0, 1\}$, we have that

$$d(m, k) = \max(d(m, k-1) + 1, d(1-m, k-2) + 3). \quad (1)$$

Consider the Figure 5. Any induced path between v and u passes by either x or y . Also, every induced path P between v and y is such that $V(P) \cap \{u, w, x, z\} = \emptyset$, and every induced path P between v and x is such that $V(P) \cap \{u, w\} = \emptyset$. Because of that, we have that, for any induced path between v and y , there is an induced path between y and u such that when we join the two paths, we have an induced path between v and u that passes by y and, for any induced path between v and x , there is an induced path between x and u such that when we join the two paths, we have an induced path between v and u that passes by x .

Let P be a maximum induced path between v and u and let $t \in V(P) \cap \{x, y\}$. We have that the induced subpath of P from v to t is a maximum induced path between v and t . This is because, if the subpath of P from v to t is not a maximum induced path between v and t , then we could replace this subpath with any maximum induced path between v and t and join it with an induced path between t and u . Since every induced path between t and u has the same length, the resulting induced path from v to u has length greater than the length of P , which is a contradiction because P is a maximum induced path between v and u .

Thus, since the length of any induced path between x and u is 1 and y and u is 3, we have the Equation 1, for $m = 0$. We can use the same argument, only changing the vertex u for the vertex w , the vertex x for z , the vertex z for x and the vertex y for q , to prove that the Equation 1 also holds when $m = 1$.

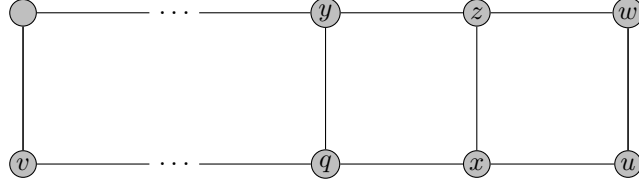


Figure 5

Now, let us prove using induction on k that $d(0, k) = k + 2 \cdot \lfloor (k + 1)/4 \rfloor - 1$ and $d(1, k) = k + 2 \cdot \lfloor (k - 1)/4 \rfloor$.

For $k = 1$, we have that $d(0, 1) = 0 = 1 + 2 \cdot \lfloor (1 + 1)/4 \rfloor - 1$ and that $d(1, 1) = 1 = 1 + 2 \cdot \lfloor (1 - 1)/4 \rfloor$. For $k = 2$, we have that $d(0, 2) = 1 = 2 + 2 \cdot \lfloor (2 + 1)/4 \rfloor - 1$ and that $d(1, 2) = 2 = 2 + 2 \cdot \lfloor (2 - 1)/4 \rfloor$.

Now, for some $k \geq 2$, let us assume that, for all $i \leq k$, $d(0, i) = i + 2 \cdot \lfloor (i + 1)/4 \rfloor - 1$ and $d(1, i) = i + 2 \cdot \lfloor (i - 1)/4 \rfloor$. Thus, we have that

$$d(0, k + 1) = \max(d(0, k) + 1, d(1, k - 1) + 3) = \max(k + 2 \cdot \lfloor (k + 1)/4 \rfloor, (k + 2) + 2 \cdot \lfloor (k - 2)/4 \rfloor).$$

Since $k + 2 \cdot \lfloor (k + 1)/4 \rfloor \leq k + 2 \cdot \lfloor (k - 2)/4 + 1 \rfloor \leq (k + 2) + 2 \cdot \lfloor (k - 2)/4 \rfloor$, we have that $d(0, k + 1) = (k + 2) + 2 \cdot \lfloor (k - 2)/4 \rfloor = (k + 2) + 2 \cdot \lfloor (k + 2)/4 - 1 \rfloor = (k + 1) + 2 \cdot \lfloor (k + 2)/4 \rfloor - 1$, which is what we wanted.

On the other hand, we have that $d(1, k + 1) = \max(d(1, k) + 1, d(0, k - 1) + 3) = \max(k + 2 \cdot \lfloor (k - 1)/4 \rfloor + 1, k + 2 \cdot \lfloor k/4 \rfloor + 1) = (k + 1) + 2 \cdot \lfloor k/4 \rfloor = (k + 1) + 2 \cdot \lfloor ((k + 1) - 1)/4 \rfloor$, which is what we wanted. □

Theorem 3. For any solid grid graph G with $\Delta = 3$, $t(G)$ can be found in $O(n^2)$ time.

Proof. If a solid grid graph has $\Delta = 3$, then, since it is $K_{1,4}$ -free, it can be decomposed in ladders and trees of maximum degree 3, whose leaves can be the extremities of some ladder. In Figure 6, there is an example of solid grid graph with $\Delta = 3$.

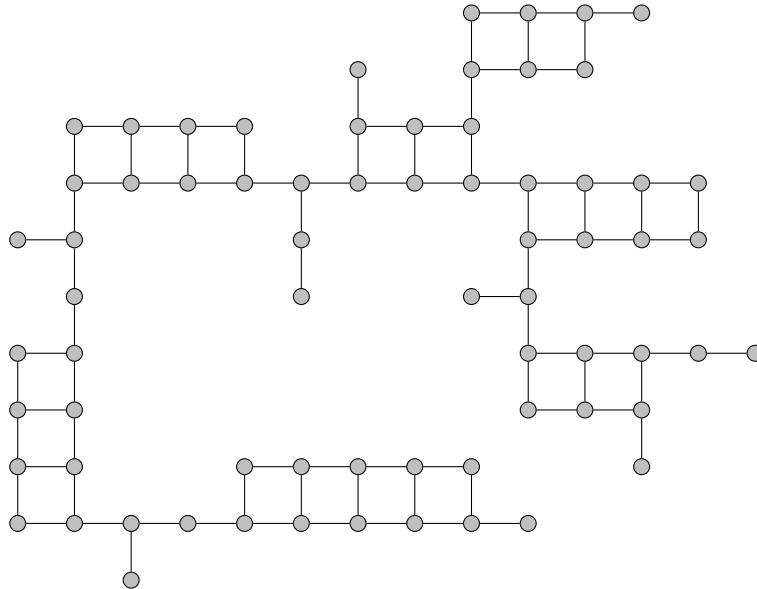


Figure 6: A solid grid graph with maximum degree 3.

To find the percolation time of G , according to Lemma 2, it is enough to find both the longest induced path that starts with a degree 2 vertex and, then, passes only by vertices with degree 3, and the longest induced path

that passes only by vertices with degree 3. Thus, since all bounded faces of G are squares with area one and G is composed only by trees of maximum degree 3 and ladders, the only difficulty to calculate $t(G)$ is in finding the longest induced paths, whose all internal vertices have degree 3, between any two extremities of a ladder.

However, using Lemma 3, we can calculate in the size of such longest induced paths of a ladder. Suppose we want to calculate the length of a longest induced path P , whose all internal vertices have degree 3, between two extremities x and y of a L_k . Let t be the number of vertices among the other two extremities that have degree 2 and let $m = 0$, if x is at distance $k - 1$ of y , or $m = 1$, if x is at distance k of y . Also, let x_1 and y_1 be the neighbors of x and y , respectively, that are internal vertices of the ladder and x_2 and y_2 be the neighbors of x and y , respectively, that are extremities of the ladder.

Therefore, we can calculate the longest induced path P between two extremities x and y in a L_k , whose all internal vertices have degree 3, in the following manner: if $k = 1$, then $|E(P)| = 1$, if $m = 1$, and $|E(P)| = 0$, if $m = 0$. If $k \geq 2$ and $t = 0$, then $|E(P)| = d(m, k)$. If $k \geq 2$ and $t = 1$, then either x_2 or y_2 has degree 2, which implies that every induced path P between x and y that passes only by vertices with degree 3, passes by x_1 , if x_2 has degree 2, or by y_1 , if y_2 has degree 2. Therefore, if $k \geq 2$ and $t = 1$, $|E(P)| = d(m, k - 1) + 1$. Finally, if $t = 2$ and $k \geq 3$, both x_1 and y_1 have degree 2, which implies that every induced path P between x and y that passes only by vertices with degree 3, passes by x_2 and y_2 and, hence, we have that $|E(P)| = d(m, k - 2) + 2$. Finally, when we have $t = 2$ and $k = 2$, there is no induced path, whose all internal vertices have degree 3, between x and y .

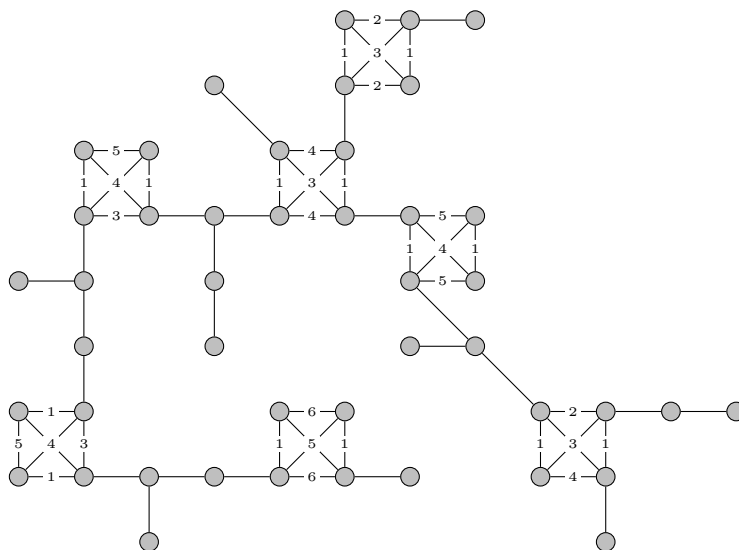


Figure 7: The resulting graph of the transformation applied to the graph in the Figure 6.

So, first, we will transform G in a weighted graph G' where G' is the same graph as G only with all the ladders replaced by weighted K_4 's, where the weight of an edge between two vertices in a K_4 represents the length of a longest induced path, whose internal vertices all have degree 3, between the corresponding extremities of the ladders in G . The weight of all the other edges is 1. The Figure 7 represents the transformation applied in the graph of the Figure 6. Note that there is exactly one induced path between any two vertices in G' , which length is equal to the longest induced path between the same two vertices in G . It is not hard to see that this transformation from G to G' can be done in linear time.

In Algorithm 1, the function `LongestInducedPathFrom` receives as parameters a graph G' and a vertex u such that $d_G(u) \geq 2$. At first, it does a depth-first search to find the path $P = u, v_1, v_2, \dots, v_k$ in G' that has the greatest $L(P)$ such that, for all $1 \leq i \leq k - 1$, $d_G(v_i) = 3$ and the subpath $u, v_1, v_2, \dots, v_{k-1}$ is an induced path, where, if P is an induced path and either $d_G(u) = 3$ and $d_G(v_k) \geq 2$ or $d_G(u) = 2$ and $d_G(v_k) = 3$, $L(P) = |E(P)|$, otherwise, $L(P) = |E(P)| - 1$. Then it returns $L(P)$.

This is necessary because a longest induced path in G can end in a vertex v internal to a ladder, but internal vertices of every ladder in G are not represented by a vertex in G' . However, if that happens, v must be adjacent to some vertex that has degree 2 and that is at the extremity of a ladder.

Since there is only one induced path between any two vertices in G' , we have that the recursive function

Algorithm 1: Algorithm that finds $t(G)$ for any solid grid graph G with $\Delta = 3$

```

Algorithm MaximumTimeSolidGrid $\Delta$ 3( $G$ )
   $G' = \text{Transform}(G)$ 
   $\text{maxPercTime} = 0$ 
  forall  $u \in V(G')$  such that  $d_G(u) \geq 2$  do
    if  $d_G(u) = 2$  then
       $\text{percTimeU} = \text{LongestInducedPathFrom}(G', u) + 1$ 
    else
       $\text{percTimeU} = \lfloor (\text{LongestInducedPathFrom}(G', u) + 2) / 2 \rfloor$ 
    if  $\text{maxPercTime} < \text{percTimeU}$  then
       $\text{maxPercTime} = \text{percTimeU}$ 
  return  $\text{maxPercTime}$ 

```

`LongestInducedPathFrom` takes the same time as any Depth-First Search algorithm. Thus, since $m = O(n)$, the Function `LongestInducedPathFrom` takes $O(n)$ time. Therefore, the Algorithm 1 takes $O(n^2)$ time. \square

4. Percolation Time Problem for time $n - k$

We first prove the following easy fact:

Lemma 4. *For any constant k , we can decide whether $t(G) \geq n - k$ in $O(m \cdot n^{k+1})$ time.*

Proof. If $t(G) \geq n - k$, then, in an infection process with time $n - k$, there is at least a vertex with time 1, a vertex with time 2, and so on, until a vertex with time $n - k$. Then there are at most k vertices with time 0. Since there are $O(n^k)$ vertex subsets of size at most k and the infection process of a vertex subset can be done in time $O(mn)$, then we can decide if $t(G) \geq n - k$ in time $O(m \cdot n^{k+1})$. \square

In 2013, Benevides *et al.* [3] proved that Percolation Time Problem is NP-complete for any constant $k \geq 4$. In this section, we generalize this result. In Theorem 4, we prove that there is no polynomial time algorithm for the Percolation Time Problem even when $k = n - n^\epsilon + 4$, where $n = |V(G)|$, unless $P = NP$.

Theorem 4. *Deciding whether $t(G) \geq k$ is NP-Complete for $k = n - n^\epsilon + 4$ for any fixed $0 < \epsilon \leq 1$.*

Proof. Let us prove that the Percolation Time Problem is NP-Complete by showing a polynomial reduction from the problem **3-SAT**. Given M clauses $\mathcal{C} = \{C_1, \dots, C_M\}$ on N variables $X = \{x_1, \dots, x_N\}$ of an instance of **3-SAT**, let us denote the three literals of C_i by $\ell_{i,1}$, $\ell_{i,2}$ and $\ell_{i,3}$.

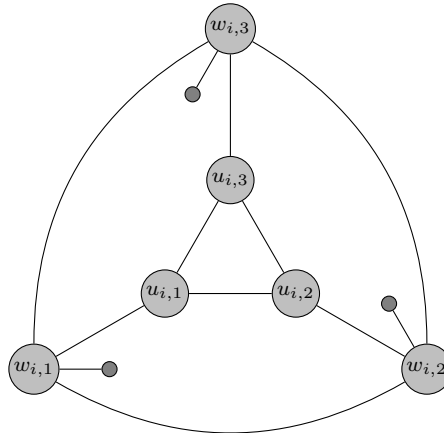


Figure 8: Gadget for clause C_i .

The reduction is as follows: For each clause C_i of \mathcal{C} , add to G a gadget like the one in Figure 8. Let $U_i = \{u_{i,1}, u_{i,2}, u_{i,3}\}$, $U = \bigcup_{1 \leq i \leq M} U_i$ and $W = \bigcup_{1 \leq i \leq M} \{w_{i,1}, w_{i,2}, w_{i,3}\}$. Then, for each pair of literals $\ell_{i,a}, \ell_{j,b}$ such that one is the negation of the other, add a vertex $y_{(i,a),(j,b)}$ adjacent to $w_{i,a}$ and $w_{j,b}$. Let Y be the set of all vertices created this way. Until now, we added $9M + y$ vertices, where $y = |Y|$.

Let $c' = 9M + y + 4$, $t = c'^{1/\epsilon}$ and $c = \lceil t \rceil - c'$. Note that, since $t \geq t^\epsilon = c'$, we have that c is a non-negative integer. Finally, add a P_{c+1} such that z and z' are its extremities, add three vertices u, v, w , add an edge between u and v , u and w , u and every vertex of the path P_{c+1} and between z and every vertex in Y . Note that the number of vertices of the graph is $n = c' + c$. Let T_0 be the set of all vertices with degree one.

Since $n = \lceil t \rceil = \lceil c'^{1/\epsilon} \rceil$ and ϵ is constant, the number of vertices of G is a polynomial in N and M . Thus, the construction of G can be done in polynomial time.

Now, let us show that \mathcal{C} is satisfiable if and only if G contains a hull set that infects all vertices in G at time at least $4 + c$.

Suppose that \mathcal{C} has a truth assignment. For each clause C_i , let K_i denote the subset of $\{1, 2, 3\}$ such that $\ell_{i,k}$ is true for all $x \in K_i$. Let $S = \{v, w\} \cup \{u_{i,x} : x \in K_i, 1 \leq i \leq M\} \cup T_0$. We can see in Figure 8 that all vertices in the clause gadgets are infected by S at time at most 3. Also, for every $1 \leq i \leq M$, we have that S infects $w_{i,x}$, for any $x \in K_i$, at time 1 and infects $w_{i,x'}$, for any $x' \notin K_i$, at time 2. Since we used a truth assignment, we have that all vertices of Y are infected at time exactly 3 because any vertex $y \in Y$ is adjacent to exactly one vertex in W infected at time 1 and another vertex in W infected at time 2. Thus, the vertex z' is infected at time $4 + c$ and, therefore, $t(G) \geq 4 + c$.

Now, suppose that the instance \mathcal{C} is not satisfiable. Let G be the graph resulting from the reduction applied to \mathcal{C} . Let us prove that $t(G) < 4 + c$. Let S be any hull set of G . Since S is a hull set, then it must have at least one vertex in each set U_i , because each vertex in U_i have only one neighbor outside U_i , and all vertices of degree one, for the same reason. Thus, since \mathcal{C} is not satisfiable, then there is a vertex $y \in Y$ such that S infects y at time ≤ 2 . This is because, if S infects all vertices in Y at time 3, then we can find an assignment to the variables of \mathcal{C} that satisfies \mathcal{C} . Consider the following assignment: If a vertex $u_{i,j}$ is in S and $\ell_{i,j}$ is a positive literal then assign true to the variable that it represents and, hence, if a vertex $u_{i,j}$ is in S and $\ell_{i,j}$ is a negative literal then assign false to the variable that it represents. After that, if there is a variable that does not yet have a value, assign true to that variable. This assignment satisfies \mathcal{C} because since S has at least one vertex in each set U_i , then each clause will have at least one literal that evaluates to true. Furthermore, since all vertices in Y are infected at time 3, then we have that two vertices in U that represent the same variable but negate each other, cannot be in S simultaneously and, therefore, we do not risk to assign different values to the same variable.

Thus, we have that z is infected by S at time ≤ 3 and z' at time $< 4 + c$. Since any hull set S infects all vertices in $Y \cup U$ at time at most 3, then z is the only possible vertex that can be infected at time at least 4 and, hence, z' is the only possible vertex that can be infected at time at least $4 + c$ by any hull set. Therefore, all hull sets infect G at time $< 4 + c$, which implies that $t(G) < 4 + c$.

Therefore, we have that \mathcal{C} is satisfiable if and only if G contains a hull set that infects all vertices in G at time at least $4 + c$. However, since $n = c + c'$ and $c = \lceil t \rceil - c'$, we have that $c = n - c' = n - t^\epsilon$ and $n = \lceil t \rceil$. Hence, we have that $c - 1 < n - n^\epsilon \leq c$ since $n - t^\epsilon - 1 < n - n^\epsilon = n - \lceil t \rceil^\epsilon \leq n - t^\epsilon$. This implies that \mathcal{C} is satisfiable if and only if $t(G) \geq n - n^\epsilon + 4$. \square

5. Percolation Time Problem in graphs with bounded maximum degree

In Section 2 (Theorem 1), we proved that the Percolation Time Problem is polynomial time solvable in grid graphs with $\Delta(G) \leq 3$ for $k = O(\log n)$. In this section, we prove on Theorem 5 that this not happen for general graphs with fixed maximum degree $\Delta \geq 4$, unless $P = NP$. However, if the percolation time k is also fixed, then we prove in Theorem 6 that the Percolation Time Problem is solvable in quadratic time (in other words, it is fixed parameter tractable on $\Delta(G) + k$).

Given $r > 0$, let CT_r be the caterpillar graph with $2r$ vertices obtained from a path u_1, \dots, u_r joining to each u_i a vertex v_i of degree 1. We say that u_1 and u_r are the ends of CT_r .

Theorem 5. *Let $\Delta \geq 4$ be fixed. Deciding whether $t(G) \geq k$ is NP-complete for graphs with bounded maximum degree Δ and $k = \Theta(\log n)$.*

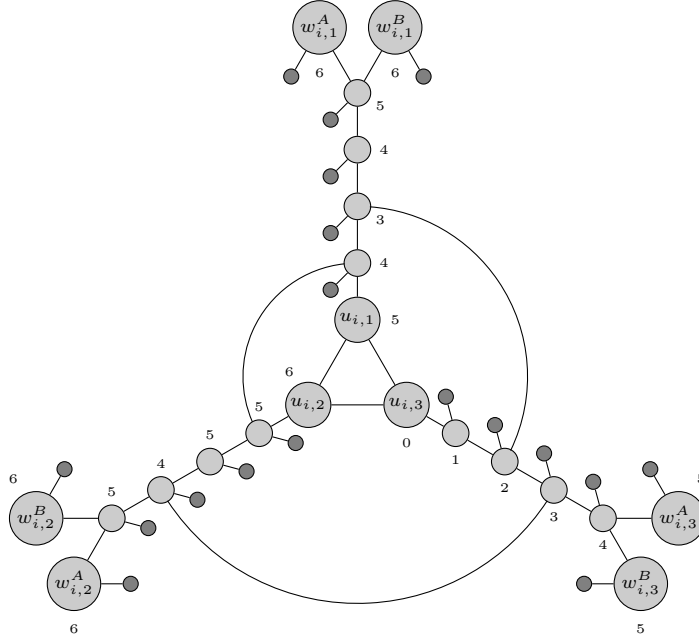


Figure 9: Gadget with infection times for each clause C_i .

Proof. Since $k = \Theta(\log n)$, we can assume that $k = c \log_{\Delta-2}(n)$ for some constant $c > 0$. We obtain a reduction from the variation of the **SAT** problem where each clause has exactly three literals, each variable appears in at most four clauses [23].

Given M clauses $\mathcal{C} = \{C_1, \dots, C_M\}$ on N variables $X = \{x_1, \dots, x_N\}$ as an instance of **SAT**, we denote the three literals of C_i by $\ell_{i,1}$, $\ell_{i,2}$ and $\ell_{i,3}$. We can assume that $N \geq c^2$. Note, since any variable can only appear in at most 4 clauses, that $N/3 \leq M \leq 4N/3$. So, first, let us show how to construct a graph G with maximum degree Δ . For each clause C_i of \mathcal{C} , add to G a gadget as the one in Figure 9. Then, for each pair of literals $\ell_{i,a}$, $\ell_{j,b}$ such that one is the negation of the other, add a vertex $y_{(i,a),(j,b)}$, link it to either $w_{i,a}^A$ or $w_{i,a}^B$ and link it to either $w_{j,b}^A$ or $w_{j,b}^B$, but always respecting the restriction of degree at most 4 for the vertices $w_{i,a}^A$, $w_{i,a}^B$, $w_{j,b}^A$ and $w_{j,b}^B$. Since each variable can appear in at most 4 clauses, it is always possible to do that. Let Y be the set of all vertices $y_{(i,a),(j,b)}$ created this way. Notice that $y = |Y| \leq 4N$.

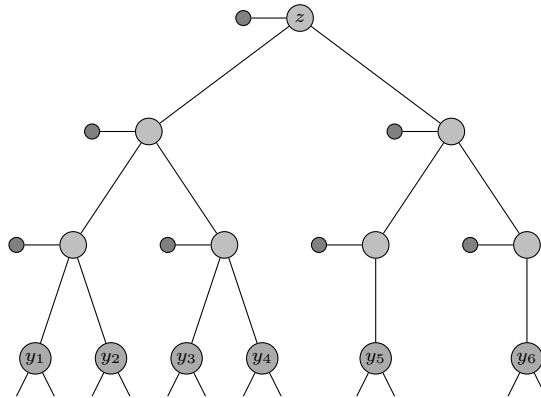


Figure 10: Example of the full $(\Delta - 2)$ -ary tree added to G , for $\Delta = 4$ and $y = 6$.

Then, add the maximum full $(\Delta - 2)$ -ary tree T with root z such that the number y' of leaves is less than y and, then, add a new vertex of degree one adjacent to each vertex of the tree. Let $t = 2|V(T)|$ be the number of vertices introduced. After that, link every vertex of Y to exactly one leaf of T in such a way that each leaf of T has at least one and at most $\Delta - 2$ neighbors in Y . Thus, each vertex of T has degree Δ in G , except for the leaves, which have degree at most Δ , and z , which has degree $\Delta - 1$. Figure 10 shows an example for $\Delta = 4$ and $y = 6$. Notice that $\text{height}(T) = \lceil \log_{\Delta-2} y \rceil - 1$. Moreover, from the sum of the geometric progression,

$$t = 2 \cdot \frac{(\Delta-2)y'-1}{(\Delta-2)^{-1}} \leq 2 \cdot \frac{(\Delta-2)y-1}{\Delta-3} \leq 16N.$$

Let $r = \lceil \log_{\Delta-2}(4N) \rceil - \lceil \log_{\Delta-2} y \rceil \geq 0$. If $r = 0$, let $q = z$. If $r > 0$, add the caterpillar graph CT_r , link one end to z and let q be the other end.

Let n' be the number of vertices added so far, that is, $n' = 39M + y + t + 2r \leq 80N$. Let $c' = \min\{c, 1\}$. Let $\alpha = (\Delta - 2)^{-7} \leq 1/128$. Finally, add $n'' = \lceil (4N/\alpha)^{1/c'} \rceil - n' > 900N$ isolated vertices. By our construction, we have that G is a graph with exactly $n = n' + n'' = \lceil (4N/\alpha)^{1/c'} \rceil$ vertices in which every vertex has degree at most Δ .

If $c \leq 1$, let $q' = q$ and $r' = 0$. If $c > 1$, let $r' = \lceil (c-1) \log_{\Delta-2}(n) \rceil$ and, using the recently introduced isolated vertices, build the caterpillar graph $CT_{r'}$, link one end to q and let q' be the other end. Notice that $2r' \leq n''$, since $n'' \geq n/2 = \sqrt{n}\sqrt{n}/2 \geq 8\sqrt{N} \log_2 n \geq 8c \log_{\Delta-2} n \geq 2r'$.

Notice that any percolating set must contain a vertex of $\{u_{i,1}, u_{i,2}, u_{i,3}\}$ for each clause C_i of \mathcal{C} and all vertices that have degree 1. Thus, following similar arguments in the proof of Theorem 4, it is possible to prove that the maximum percolation time of the vertex z is $\text{height}(T) + 8 = \lceil \log_{\Delta-2} y \rceil + 7$ if and only if \mathcal{C} is satisfiable, which implies that the maximum percolation time of the vertex q' is $\lceil \log_{\Delta-2}(4N) \rceil + 7 + r'$ if and only if \mathcal{C} is satisfiable. Since $4N = \alpha n^{c'}$, we have that \mathcal{C} is satisfiable if and only if $t(G) \geq c' \log_{\Delta-2}(n) + r'$. Thus, \mathcal{C} is satisfiable if and only if $t(G) \geq c \log_{\Delta-2}(n)$. \square

We already saw that the Percolation Time Problem is NP-hard for graphs with maximum degree $\Delta(G) \geq 3$. In this section, we prove that the Percolation Time Problem is FPT on $\Delta(G) + k$. In fact, we prove a stronger result:

Theorem 6. *Percolation Time Problem is fixed parameter tractable with parameter $\Delta(G) + k$. Moreover, for fixed Δ , the Percolation Time Problem is polynomial time solvable in graphs with bounded maximum degree Δ for $k = \log_{\Delta} O(\log n)$, if $\Delta \geq 4$, and for $k = O(\log n)$, if $\Delta = 3$.*

Proof. Let $\Delta = \Delta(G)$ and let $u \in V(G)$. Then $|N_{\leq k}(u)| \leq \Delta^k$ and, consequently, the power set $2^{N_{\leq k}(u)}$ has $2^{|N_{\leq k}(u)|} \leq 2^{\Delta^k}$ sets. We claim that $t(G) \geq k$ if and only if there is a vertex u and a percolating set $S \supseteq N_{\geq k}(u)$ such that $t(G, S, u) = k$.

If $t(G) \geq k$, then there is a percolating set S' that infects some vertex u at time k . In [17], it was proved that, given a graph G , a set $Q \subseteq V(G)$ and a vertex $z \in V(G) \setminus S$, if $t(G, Q, w) \geq k$, then $t(G, Q \cup \{z\}, w) \geq k$, for any k and any $w \in N_{\geq k}(z)$. Then, applying this result once for each vertex in $N_{\geq k}(u)$, the percolating set $S = S' \cup N_{\geq k}(u)$ infects u also at time k .

On the other hand, if there is a percolating set $S \supseteq N_{\geq k}(u)$ such that $t(G, S, u) = k$, for some vertex u , then, trivially, $t(G) \geq k$. Then the claim is true.

Therefore, since for each vertex u and set $S' \subseteq N_{\leq k-1}(u)$, it takes $O(km)$ time to know whether the set $S' \cup N_{\geq k}(u)$ infects u at time k , this equivalence gives us an algorithm that decides whether $t(G) \geq k$ in time $n \cdot O(m + km \cdot 2^{\Delta^k}) = O(2^{\Delta^k} k \Delta \cdot n^2)$, since $m = O(\Delta n)$. Notice that, if $k = \log_{\Delta} O(\log n)$, then the time is polynomial in n . Moreover, if $\Delta = 3$, by Theorem 1, we are done. \square

6. Fixed Parameter Tractability on the treewidth

A *parameterized problem* is a pair (P, k) , where P is a decision problem and $k = k(I)$ is a parameter on the instances I of P . We say that a parameterized problem (P, k) is *fixed parameter tractable* (or just *FPT*) if there exists an algorithm (called *fpt-algorithm*) that solves P in time $f(k(I)) \cdot |I|^{O(1)}$, for any instance I of P , where f is an arbitrary function depending only on the parameter $k(\cdot)$.

A *fpt-reduction* from a parameterized problem (P_1, k_1) to (P_2, k_2) is a pair (f, g) of functions such that, for each instance I of P_1 , (a) $f(I)$ is an instance of P_2 , (b) f is computable by an fpt-algorithm on the parameter k_1 and (c) g is a computable function such that $k_2(f(I)) \leq g(k_1(I))$. With this, if (P_2, k_2) is FPT, then (P_1, k_1) is also FPT.

We say that a parameterized problem (P, k) is *W[1]* if there is a fpt-reduction from (P, k) to the Clique decision problem (parameterized by the size of the clique). We say that (P, k) is *W[1]-hard* if there is a fpt-reduction from the Clique decision problem (parameterized by the size of the clique) to (P, k) . We say that (P, k) is *W[1]-complete* if it is *W[1]* and *W[1]-hard*.

The Exponential Time Hypothesis (ETH) is a conjecture of Impagliazzo et al. [13] which states that there is a constant $\delta_3 > 0$ such that, for any polynomial $p(\cdot)$ and constant $c < \delta_3$, the problem **3-SAT** cannot be decided in $O(2^{cN} p(N))$ time, where N is the number of variables of the instance of **3-SAT**.

If we assume the ETH, we can come to several conclusions about the time lower bounds of algorithms that can solve NP-hard and W[1]-hard and even FPT problems.

In this section, we obtain some results regarding the fixed parameter tractability of the Percolation Time Problem when the parameter is the treewidth $tw(G)$ of the graph. We prove on Theorem 8 that the Percolation Time Problem is FPT on $tw(G) + k$ (that is, to decide if the time is at most k). In fact, we prove a stronger (but technical) result: $t(G)$ can be computed in polynomial time for fixed $tw(G)$ and $t(G) \geq k$ can be decided in linear time for fixed $tw(G) + k$. And what happens when $tw(G)$ is fixed, but k is not fixed? The decision problem is FPT on the treewidth? We prove in Theorem 7 that the Percolation Time Problem is W[1]-hard when parameterized by the treewidth. Furthermore, in Corollary 1, we give a lower bound for the time that the Percolation Time Problem can be solved basing on the ETH.

Let us present some definitions. A *tree decomposition* [22] of a graph G is a tuple $(\mathcal{T}, \mathcal{B})$, where \mathcal{T} is a tree, \mathcal{B} contains a subset $B_t \subseteq V(G)$ for each node $t \in \mathcal{T}$ and:

- (i) for each vertex u of G , there exists some $B_t \in \mathcal{B}$ containing u ;
- (ii) for each edge uv of G , there exists some $B_t \in \mathcal{B}$ containing u and v ; and
- (iii) if B_i and B_j both contain a vertex v , then, for every node t of the tree in the (unique) path between i and j , B_t also contains v .

Observe from (iii) that $\mathcal{T}[\{t : v \in B_t\}]$ is connected, for all $v \in V(G)$. That is, the nodes associated with vertex v form a connected component of \mathcal{T} . In other words, if t is on the unique path of \mathcal{T} between i to j , then $B_i \cap B_j \subseteq B_t$. The *width of $(\mathcal{T}, \mathcal{B})$* is $\max\{|B_t| - 1 : t \in \mathcal{T}, B_t \in \mathcal{B}\}$ and the *treewidth $tw(G)$* of G is the minimum width over all tree decompositions of G [22]. The proof of the next theorem uses some technicalities inspired by [10].

Theorem 7. *The Percolation Time Problem is W[1]-hard with parameter $tw(G)$.*

Proof. Let us prove that the percolation time problem is W[1]-hard when parametrized by *treewidth*. We found a reduction from the Multicolored Clique Problem, which is W[1]-hard when parametrized by the number of colors k [9]. Given a graph G , an integer k and a partition V_1, V_2, \dots, V_k of $V(G)$, which represent the colors of the vertices, the Multicolored Clique Problem asks for a k -clique containing exactly one vertex from each partition V_i .

Let us show a parameterized reduction from an instance (G, k, C) of Multicolored Clique Problem to an instance (G', k') of Percolation Time Problem, where the treewidth of G' is bounded by $2k + 3$.

To save some space in the figures, every time we have a path where every vertex of this path has one pendant neighbor, we will replace it by a weighted edge, like in Figure 11. So, if the vertex i is infected at time t by some set, then, if the infection follows the path v_1, v_2, \dots, v_{n-1} , the vertex v_{n-1} will be infected at time $t + (n - 1)$ by the same set.

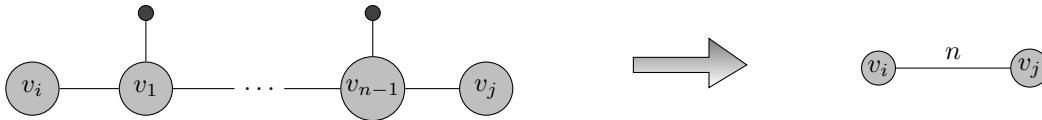


Figure 11: A weighted edge on the right and the real structure on the left

Let us describe how to construct the graph G' . Let $n = |V(G)|$. First, assign the labels v_1, v_2, \dots, v_n arbitrarily to the vertices in G and, for $1 \leq i \leq n$, let $M_i^a = 2n + 2i - 2$ and $M_i^b = 2n - 2i + 2$. For $1 \leq x \leq k$ we are going to add the gadget Q^x in the Figure 12, where the vertices v_i, v_j, \dots are the vertices in the partition V_x in the original graph G . Let $T = \bigcup_{1 \leq i \leq k} \{a^i, b^i\}$. Also, let V^x be the set of all vertices in $\{v_1, v_2, \dots, v_n\} \cap V(Q^x)$. Note that, when we use V_x , we are talking about the vertices in the original graph G and, when we use V^x , we are talking about the vertices in G' .

For each pair of vertices $v_i \in V_x$ and $v_j \in V_y$, for $x \neq y$, such that $(v_i, v_j) \notin E(G)$, we add a gadget such as the one in the Figure 13. Also, we add six vertices $a_i^j, b_i^j, a_j^i, b_j^i$ and link a_i^j to a^x , b_i^j to b^x , a_j^i to a^y and b_j^i to b^y . After

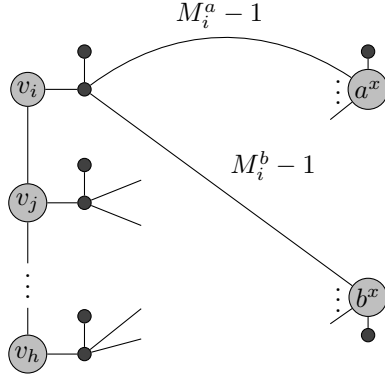


Figure 12: Gadget Q^x we put k times where each $v_i \in V^x$ represents the vertex $v_i \in V_x$ of the original graph G .

that, link the vertex vo_i^j to a_i^j and to b_i^j with edges with weights M_i^a and M_i^b , respectively, and link the vertex vo_j^i to a_j^i and to b_j^i with edges with weights M_j^a and M_j^b , respectively. Let the set $VC_i^j = VC_j^i = \{vc_i^j, vc_j^i\}$.

In the Figure 14, there is an example where $v_i, v_s \in V^x$ and $(v_i, v_j), (v_s, v_r) \notin E(G)$.

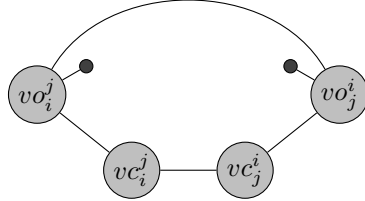


Figure 13: Gadget we put between the Graph Gadgets whenever there is no edge between the vertices v_i and v_j .

Let F be how many vertices we added until this point, including the vertices hidden by the weighted edges. We continue the construction of G' by adding, for each vertex a_i^j and b_i^j , a vertex l_i^j and link both a_i^j and b_i^j to it with edges with weights $F - M_i^a - 2$ and $F - M_i^b - 2$, respectively. Let L be the set of all vertices l_i^j added in that manner. Then, add a vertex p , add another vertex and link it to p and also link all vertices in L to p . Let D be how many vertices we added until this point. Finally, add a vertex z , add another vertex and link it to z and link p to z with an edge with weight D .

In the Figure 15 there is an example for this step of the construction. Each step of our construction is linear on the number of vertices added in that step. Since $|\bigcup_{1 \leq x \leq k} V(Q^x)| = O(kn^2)$, $F = O(kn^2 + n^3)$, $D = O(n^2 \cdot (n^3 + kn^2))$ and $V(G') \leq 3D + 2 = O(n^5 + kn^4)$, we have that G' can be constructed in $O(n^5 + kn^4)$ time.

Let P be the set of all vertices with degree one. Let Q be either V^x , for some x , or VC_i^j , for some i and j . We have that every vertex in Q has at most one neighbor outside of Q . This implies that the set $V(G') \setminus Q$ infects

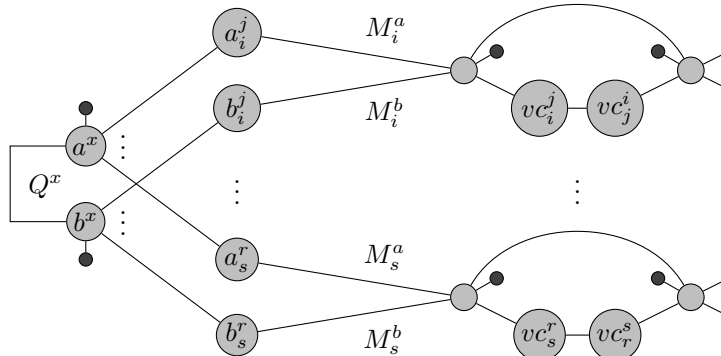


Figure 14: Example of Choice Gadgets that link the Gadget Q^x to other gadgets.

no vertex in Q . Then, any percolating set of G' must contain at least one vertex in Q , that is, G' must contain at least one vertex in each set V^x and each set VC_i^j . Also, any percolating set S of G' must contain P , because, otherwise, S could never infect the vertices in $P \setminus S$ in subsequent rounds, since the vertices in P have degree one and, in order to be infected, a vertex is required to have at least two infected neighbors.

Furthermore, let $S = S' \cup P$, where S' is any set that contains exactly one vertex of each V^x and each set VC_i^j set and only those vertices. Let us prove that S is a percolating set of G' . Let $v^x \in V^x \cap S$ and $vc_{i,j} \in VC_i^j \cap S$. Initially, we have that $\{v^x\} \cup P$ infects all vertices in $V(Q^x)$ and $\{vc_{i,j}\} \cup P$ infects all vertices in $\{vc_i^j, vc_j^i, vo_i^j, vo_j^i\}$. Therefore, we have that the set $\{a^x, vo_i^j\}$ infects a_i^j and the set $\{b^x, vo_i^j\}$ infects b_i^j , for every $v_i \in V^x$ and every $v_j \notin N(v_i) \cup V^x$. In its turn, each set $\{a_i^j, b_i^j\} \cup P$ infects the vertex l_i^j and all the internal vertices of the path of size $F - M_i^a - 2$ between a_i^j and l_i^j , and all the internal vertices of the path of size $F - M_i^b - 2$ between b_i^j and l_i^j . Hence, the set L is infected by S . Finally, the set $L \cup P$ infects p and the set $\{p\} \cup P$ infects z and the internal vertices of the path of size D between p and z . Thus, S infects the whole graph G' .

Therefore, we conclude that it is necessary and sufficient to choose exactly one vertex of each set V^x and each set VC_i^j to be part of a set S in order for $S \cup P$ to infect the whole graph G' .

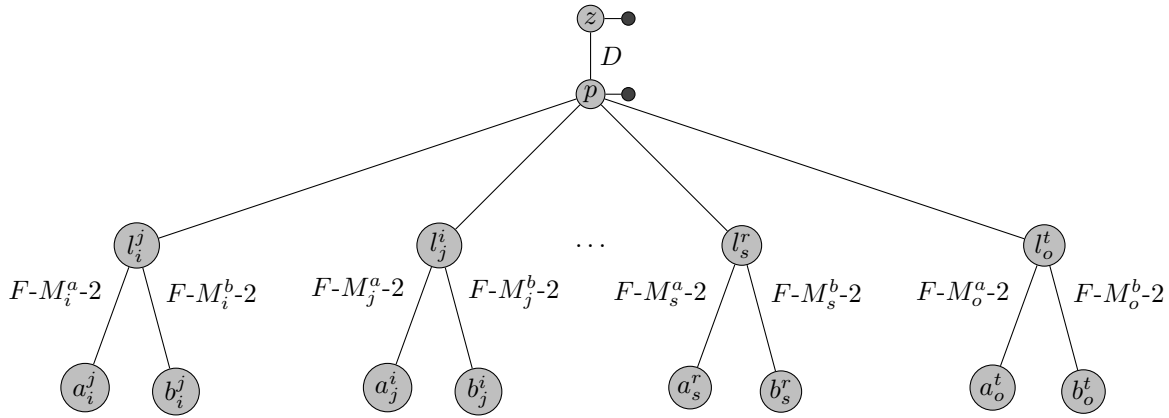


Figure 15: Gadget to link all vertices in L

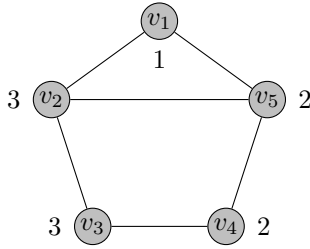


Figure 16: Graph G

In Figure 16 there is an example of Instance for the Multicolor Clique Problem for 3 partitions. The Figures 17 and 18 are the graph G' , which is the result of our construction applied to the graph in the Figure 16.

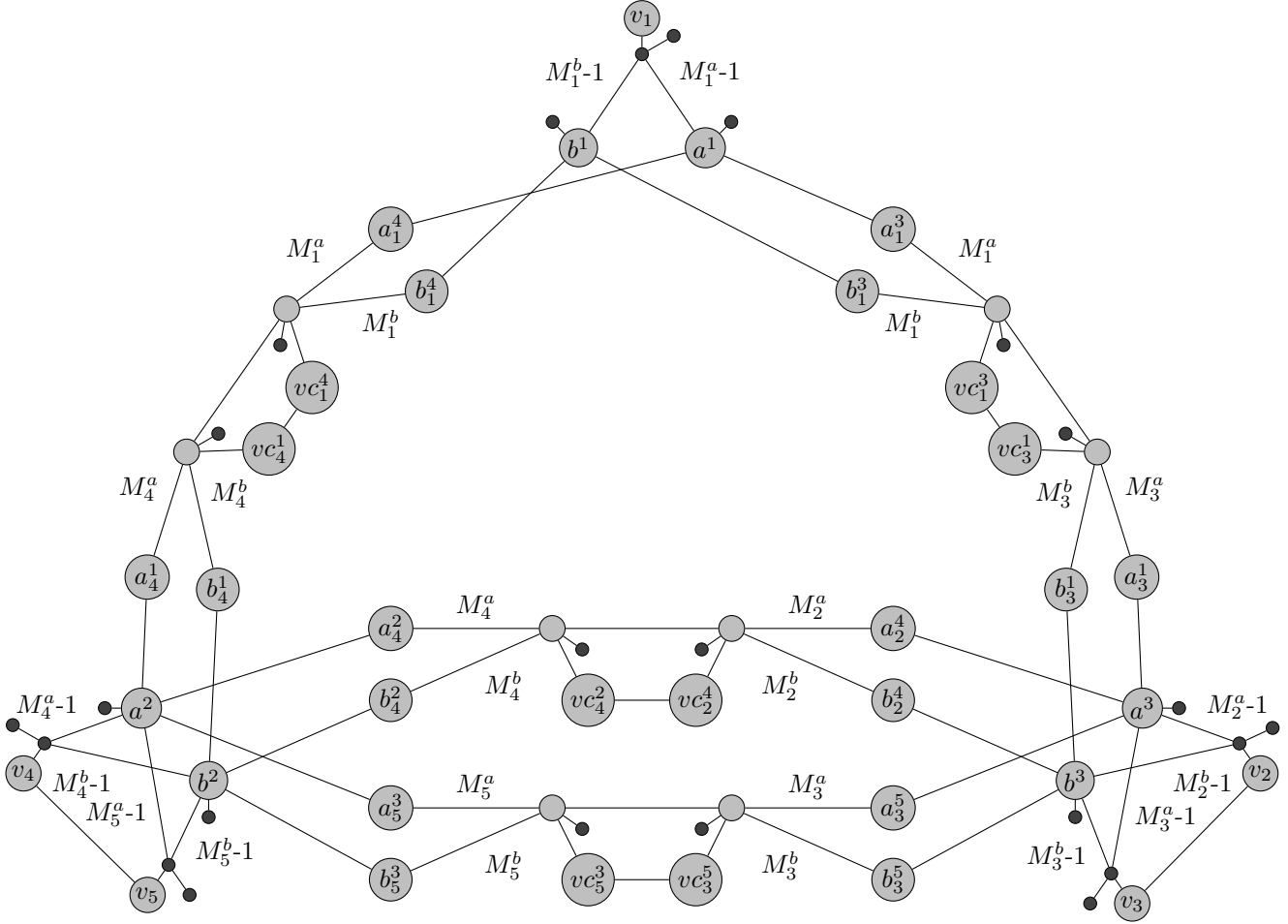


Figure 17: Part one of the graph G' resulting from the reduction applied to graph G of the Figure 16 regarding the 3-Multicolor Clique Problem

let $S = T \cup \{p\}$, which is a separator of G' . Besides isolated vertices, the components of $G' - S$ are one of the three types in the Figure 19. Thus, since the components of $G' - S$ are isolated vertices, trees or cacti, which have treewidth 0, 1 and at most 2 respectively, we conclude that $tw(G') \leq |S| + 3 - 1 = 2k + 3$.

Now, let us prove that there is a k -clique in G such that all vertices are in different partitions if and only if $t(G') \geq D + F + 1$.

First, assume that there is a k -clique in G such that all vertices are in different partitions. Let us build a percolating set S such that $t(S) \geq D + F + 1$.

Let $C = \{v_{c_1}, v_{c_2}, \dots, v_{c_k}\}$ be the set of vertices in this k -clique. Initially, set $S = C$. For each set VC_i^j such that, for some $1 \leq s \leq k$, either $i = c_s$ or $j = c_s$, add to S the vertex vc_i^j , if $j = c_s$, or add vc_j^i to S , if $i = c_s$. Since C induce a k -clique on G , we do not have a set VC_i^j such that, for some $1 \leq s, r \leq k$, $i = c_s$ and $j = c_r$. Finally, for each set VC_i^j such that, for all $1 \leq s \leq k$, neither $i = c_s$ nor $j = c_s$, add, arbitrarily, either vc_i^j or vc_j^i to S .

We have that S infects each vertex a^x at time $M_{v_{c_x}}^a$ and each vertex b^x at time $M_{v_{c_x}}^b$. Since, for all $1 \leq i \neq j \leq n$, either $M_i^a \geq M_j^a + 2$ and $M_i^b \leq M_j^b - 2$ or $M_i^a \leq M_j^a - 2$ and $M_i^b \geq M_j^b + 2$, then for all vertices a_i^j and b_i^j , either a_i^j is infected by S at time at least $M_i^a + 2$ or b_i^j is infected by S at time at least $M_i^b + 2$ and, thus, all vertices $l_i^j \in L$ are infected by S at time at least F . Then, we can conclude that p is infected by S at time at least $F + 1$ and z is infected by S at time at least $D + F + 1$. Therefore, we have that $t(G') \geq D + F + 1$.

Now, assume that there is no k -clique in G such that all vertices are in different partitions.

Since is necessary and sufficient to choose one vertex, and only one, from each set V^x and each set VC_i^j so that the resulting set infects the whole graph G' , we can focus on proving that the percolating sets built in that way infect G' at time less than $D + F + 1$. Let SS be the family of such percolating sets.

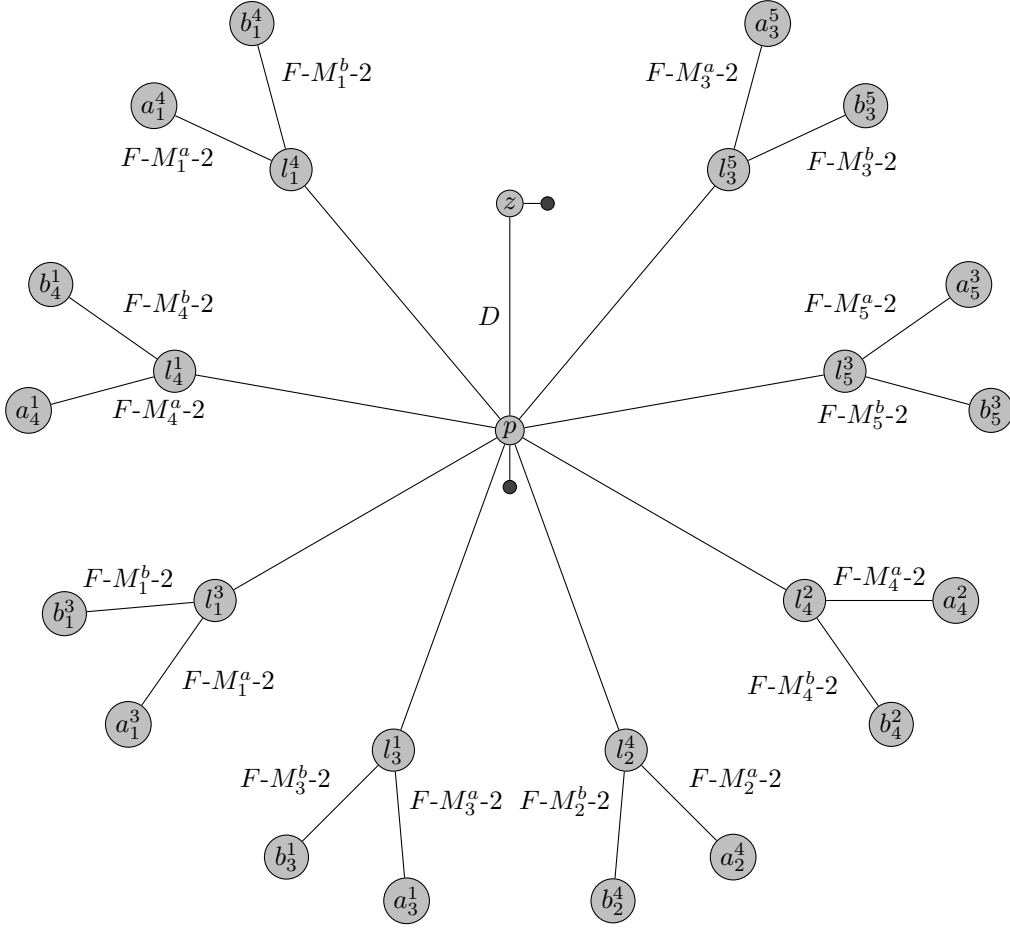


Figure 18: Part two of the graph G' resulting from the reduction applied to graph G of the Figure 16 regarding the 3-Multicolor Clique Problem

By our construction and the values of D and F , we have that only z can possibly be infected at time at least $D + F + 1$ by some percolating set in SS because z is the only vertex in G' such that there is a simple path from some vertex in S to z with length greater than $D + F$, for any $S \in SS$. Let S be an arbitrary percolating set in SS . Since there is no k -clique in G such that all vertices are in different partitions, there will be vertices a^x, b^x and vo_i^j such that S infects a^x at time M_i^a , b^x at time M_i^b and vo_i^j at time 1. Thus, we have that there will be vertices a_i^j and b_i^j that are infected by S at times $M_i^a + 1$ and $M_i^b + 1$, respectively. Then, we have that there will be some vertex l_i^j infected by S at time $F - 1$. We can, then, conclude that p is infected by S at time at most F and z at time at most $D + F$. Since z is the only vertex that could be possibly infected at time at least $D + F + 1$ by S , we have that $t(S) < D + F + 1$ and, since S is an arbitrary set of SS , therefore, $t(G) < D + F + 1$. \square

It is known that the Multicolored Clique Problem parameterized by k cannot be solved in $O(f(k) \cdot n^{o(k)})$ [16], where n is the number of vertices of the input graph, for any computable function f , time unless the ETH fails. Thus, as a result of the last theorem, since $tw(G') \leq 2k + 3$, where G' is the graph resulting from the $W[1]$ -hardness construction, we have the following corollary.

Corollary 1. *Percolation Time Problem cannot be decided in time $O(f(tw(G))n^{o(n)})$, where $n = |V(G)|$, for any computable function f , unless ETH fails.*

The next theorem proves that the Percolation Time Problem is polynomial time solvable for fixed $tw(G)$ and it is linear time solvable for fixed $tw(G) + k$.

Theorem 8. *Let w be an integer and let G be a graph with treewidth $tw(G) = w$. Then $t(G)$ can be computed in time $O(50^{w+1}n^{w+2})$ and we can decide whether $t(G) \geq k$ in time $O((50k)^{w+1}n)$.*

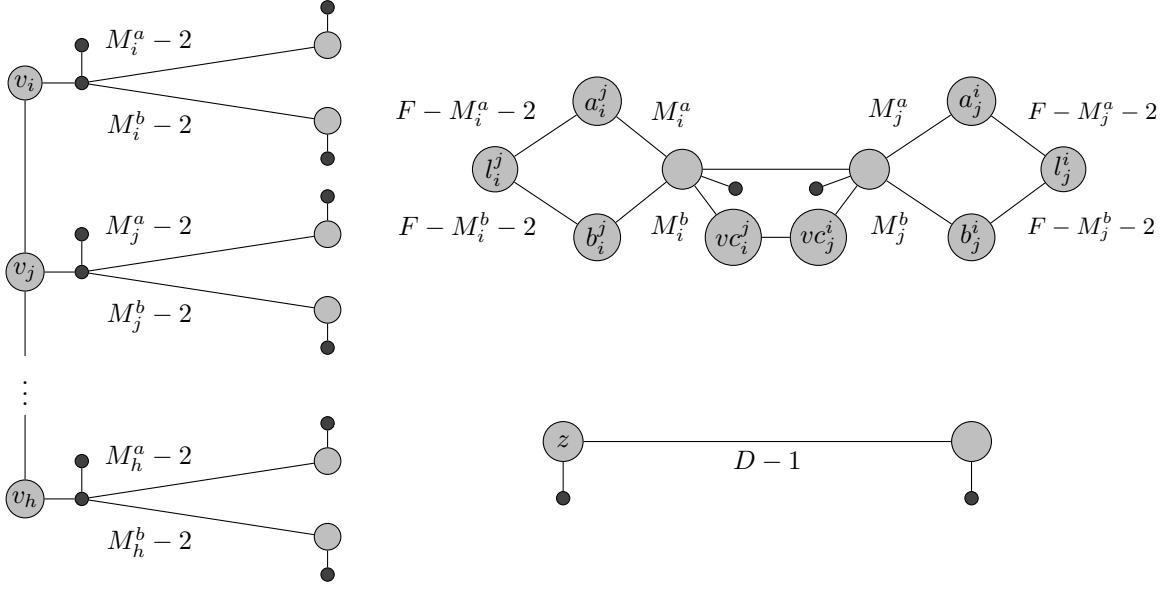


Figure 19: The three types of components of $G' - S$.

In order to prove this theorem, we have to give some definitions first. Let $(\mathcal{T}, \mathcal{B})$ be a tree decomposition of a graph G and consider the tree \mathcal{T} to be rooted in r . We say that $(\mathcal{T}, \mathcal{B})$ is a *nice tree decomposition* [15] if each node t of \mathcal{T} is either a leaf, or t has exactly two children t_1 and t_2 with $B_t = B_{t_1} = B_{t_2}$ (called *join node*), or t has exactly one child t' and either $B_t = B_{t'} \setminus \{x\}$ (called *forget node*) or $B_t = B_{t'} \cup \{x\}$ (called *introduce node*), for some $x \in V(G)$. It is known that, for a fixed k , we can determine if the treewidth of a given graph G is at most k , and if so, find a nice tree decomposition of G with $O(n)$ nodes and width at most w in linear time [6]. Given a node t of \mathcal{T} , we denote by G_t the subgraph of G induced by $\bigcup_{t' \in V(\mathcal{T}_t)} B_{t'}$, where \mathcal{T}_t is the subtree of \mathcal{T} rooted at t .

For each node $t \in \mathcal{T}$, we will compute a table W_t with integer values for every pair (p, f) , where p is a function $p : B_t \rightarrow \{0, 1, \dots, n-1\}$ that is an assignment of times for the vertices in B_t and f is a function $f : B_t \rightarrow \{z, o_1, o_2, t_1, t_2\}$ which associate a status to each vertex in B_t . Let $v \in B_t$:

- $f(v) = z$ indicates that v does not have any neighbor in G_t infected at time less than $p(v)$;
- $f(v) = o_1$ indicates that v has exactly one neighbor in G_t infected at time less than $p(v)$ and this neighbor is infected at time less than $p(v) - 1$;
- $f(v) = o_2$ indicates that v has exactly one neighbor in G_t infected at time less than $p(v)$ and this neighbor is infected at time equal to $p(v) - 1$;
- $f(v) = t_1$ indicates that v has two or more neighbors in G_t infected at time less than $p(v)$ and exactly one of them is infected at time less than $p(v) - 1$;
- $f(v) = t_2$ indicates that v has two or more neighbors in G_t infected at time less than $p(v)$ and all of them are infected at time equal to $p(v) - 1$.

Since $|B_t| \leq w + 1$, we have that the number of functions p and f are bounded by n^{w+1} and 5^{w+1} , respectively.

Let f be a function $V(G) \rightarrow \{0, 1, \dots, n-1\}$. We say that f is a *infection time function* of G if there is a percolating set S such that $t(S, v) = f(v)$ for all $v \in V(G)$. We say that f is a *quasi-infection time function* of G if, for each $v \in V(G)$, there is at most one vertex $u \in N(v)$ such that $f(u) < f(v) - 1$.

Let $W_t(p, f) = \max_g \max_{v \in V(G_t)} g(v)$ where g iterates over all quasi-infection time functions of G_t such that: each vertex $v \in V(G_t) \setminus B_t$, where $g(v) > 0$, has at least two neighbors u_1 and u_2 such that $g(u_1) < g(v)$ and $g(u_2) < g(v)$; p is g restricted to B_t and g does not contradict f regarding the vertices in B_t . We say that g contradicts f if, for some $v \in B_t$, $f(v)$ does not correspond to the reality of g on v and its neighbors in G_t . Set $W_t(p, f) = -1$ if there is no quasi-infection time function g satisfying these conditions.

Our algorithm computes $W_t(p, f)$ for every $t \in T$ and every pair (p, f) of functions. Clearly, $t(G)$ is equal to the highest $W_r(p, f)$ (recall that r is the root of \mathcal{T}) such that, for all vertices $v \in B_r$, either $f(v) = t_1$ or $f(v) = t_2$ or $p(v) = 0$.

For some indexes $h = (p, f)$, we can conclude directly that $W_t(h) = -1$ for some $t \in T$. We will say that such indexes are invalid. We are only interested in valid indexes, defined as below. Let $t \in T$ and $h = (p, f)$ be an index of W_t . We say that h is valid if and only if, for each vertex $v \in B_t$, we have that:

- If $f(v) = z$ then $p(z) \geq p(v)$ for all $z \in N(v) \cap B_t$;
- If $f(v) = o_1$ then there is at most one vertex $z \in N(v) \cap B_t$ such that $p(z) < p(v)$ and, if there is one, we have that $p(z) < p(v) - 1$;
- If $f(v) = o_2$ then there is at most one vertex $z \in N(v) \cap B_t$ such that $p(z) < p(v)$ and, if there is one, we have that $p(z) = p(v) - 1$;
- If $f(v) = t_1$ then there is at most one vertex $z \in N(v) \cap B_t$ such that $p(z) < p(v) - 1$;
- If $f(v) = t_2$ then all vertices $z \in N(v) \cap B_t$ where $p(z) < p(v)$, are such that $p(z) = p(v) - 1$.

Clearly, if $h = (p, f)$ is an invalid index then $W_t(h) = -1$ because any extension of p for G_t would contradict f . Now, we will determine the value of $W_t(h)$ depending on the type of the node t .

Lemma 5. *If t is a leaf node and $h = (p, f)$ is an index of W_t , then we can compute $W_t(h)$ in $O(w^2)$ time.*

Proof. We have that $W_t(h) = \max_{v \in B_t} p(v)$ if and only if h is valid, which can be checked in $O(w^2)$ time. \square

Lemma 6. *Let t be a forget node with child t' such that $B_t = B_{t'} \setminus \{v\}$. Let $h = (p, f)$ be a valid index of W_t . Then $W_t(h) = \max_{h'} W_{t'}(h')$ where $h' = (p', f')$ iterates over all p' and f' such that p' and f' are extensions of, respectively, p and f for $B_{t'}$ and either $f'(v) = t_1$ or $f'(v) = t_2$, or $p'(v) = 0$.*

Proof. First suppose that $\max_{h'} W_{t'}(h') = -1$, i.e., for each valid index $h' = (p', f')$ as stated in the Lemma, $W_{t'}(h') = -1$. Then, since $G_t = G_{t'}$, $B_t \cup \{v\} = B_{t'}$ and each p' and f' are extensions of, respectively, p and f for $B_{t'}$, we have that there is no quasi-infection time function g of G_t such that each vertex $u \in V(G_t) \setminus B_t$, where $g(u) > 0$, has at least two neighbors u_1 and u_2 such that $g(u_1) < g(u)$ and $g(u_2) < g(u)$; p is g restricted to B_t and g does not contradict f regarding the vertices in B_t . Therefore, $W_t(h) = -1 = \max_{h'} W_{t'}(h')$.

Now, suppose that $\max_{h'} W_{t'}(h') > -1$. Let us prove that $W_t(h) = \max_{h'} W_{t'}(h')$.

First, we are going to prove that $W_t(h) \geq \max_{h'} W_{t'}(h')$. Let $i = (x, y)$ be an index that realizes the previous maximum and let g be the quasi-infection time function of $G_{t'}$ such that $W_{t'}(i) = \max_{r \in V(G_{t'})} g(r)$ and each vertex $r \in V(G_t) \setminus B_t$, where $g(r) > 0$, has at least two neighbors u_1 and u_2 such that $g(u_1) < g(r)$ and $g(u_2) < g(r)$; x is g restricted to B_t and g does not contradict y regarding the vertices in B_t .

Since g does not contradict y and $G_t = G_{t'}$, we have that each vertex $r \in \{v\} \cup (V(G_{t'}) \setminus B_{t'}) = V(G_t) \setminus B_t$ either has at least two neighbors z_1 and z_2 such that $g(z_1) < g(r)$ and $g(z_2) < g(r)$ or $g(r) = 0$.

Also, since x is an extension of p for $B_{t'}$ and x is g restricted to $B_{t'}$ and $B_t \subseteq B_{t'}$, we have that p is g restricted to B_t .

Additionally, since g does not contradict y regarding the vertices in $B_{t'}$ and y is an extension of f for $B_{t'}$, g does not contradict f regarding the vertices in B_t .

Therefore, we have that $W_t(h) = \max_{g'} \max_{r \in V(G_t)} g'(r) \geq \max_{r \in V(G_t)} g(r) = W_{t'}(i) = \max_{h'} W_{t'}(h')$.

Now, let us prove that $W_t(h) \leq \max_{h'} W_{t'}(h')$. We have that $W_t(h) = \max'_g \max_{v \in V(G_t)} g'(v)$. Let g be a quasi-infection time function of G_t that realizes this maximum. Let p' be g restricted to $B_{t'}$, which is an extension of p for $B_{t'}$, and f' be set accordingly to g , which is an extension of f for $B_{t'}$.

Since $v \in G_t \setminus B_t$ and f' was set accordingly to g , we have that either $f'(v) = t_1$ or $f'(v) = t_2$ or $p'(v) = 0$.

Thus, g is also a quasi-time infection function of $G_{t'}$ that realizes $\max_{g'} \max_{r \in V(G_{t'})} g'(r)$ and, since also each vertex $r \in V(G_{t'}) \setminus B_{t'}$, where $p'(r) > 0$, has at least two neighbors z_1 and z_2 such that $g(z_1) < g(r)$ and $g(z_2) < g(r)$; p' is g restricted to $B_{t'}$ and g does not contradict f' regarding the vertices in $B_{t'}$, then $W_{t'}(p', f') = \max_{r \in V(G_t)} g(r)$.

Therefore, we have that $W_t(h) = \max_{r \in V(G_t)} g(r) = W_{t'}(p', f') \leq \max_{h'} W_{t'}(h')$. \square

Lemma 7. *Let t be a introduce node with child t' such that $B_t = B_{t'} \cup \{v\}$. Let $h = (p, f)$ be a valid index of W_t . Also, let p' be p restricted to $B_{t'}$. Then, if $\max_{f'}(W_{t'}(p', f')) > -1$, $W_t(h) = \max(p(v), \max_{f'}(W_{t'}(p', f')))$, where f' iterates over all functions $f' : B_{t'} \rightarrow \{z, o_1, o_2, t_1, t_2\}$ such that, for all $r \in B_{t'} \cap N(v)$ where $p(r) > p(v)$, we have:*

1. If $p(v) < p(r) - 1$ and $f(r) = o_1$ then $f'(r) = z$;
2. If $p(v) = p(r) - 1$ and $f(r) = o_2$ then $f'(r) = z$;
3. If $p(v) = p(r) - 1$ and $f(r) = t_1$ then either $f'(r) = o_1$ or $f'(r) = t_1$;
4. If $p(v) < p(r) - 1$ and $f(r) = t_1$ then either $f'(r) = o_2$ or $f'(r) = t_2$;
5. If $p(v) = p(r) - 1$ and $f(r) = t_2$ then either $f'(r) = o_2$ or $f'(r) = t_2$;

And for all other vertices r , $f'(r) = f(r)$. If, for all such f' , $\max_{f'}(W_{t'}(p', f')) = -1$, then $W_t(h) = -1$.

Proof. First suppose that $\max_{f'} W_{t'}(p', f') = -1$, then we have that there is no quasi-infection time function g of $G_{t'}$ that extends p' and respects some f' where f' is as in the Lemma.

Since $V(G_t) = V(G_{t'}) \cup \{v\}$, $B_t = B_{t'} \cup \{v\}$ and p' is p restricted to $B_{t'}$, by a simple case analysis on f' , we can conclude that there is no quasi-infection time function g of G_t that extends p and does not contradict f . Therefore, $W_t(h) = -1$.

Now, suppose that $\max_{f'} W_{t'}(p', f') > -1$. Let us prove that $W_t(h) = \max(p(v), \max_{f'}(W_{t'}(p', f')))$.

First, we are going to prove that $W_t(h) \geq \max(p(v), \max_{f'}(W_{t'}(p', f')))$. Let $y : B_{t'} \rightarrow \{z, o_1, o_2, t_1, t_2\}$ be a function that realizes the previous maximum ($\max_{f'}(W_{t'}(p', f'))$) and let g' be the quasi-infection time function of $G_{t'}$ such that $W_{t'}(p', y) = \max_{r \in V(G_{t'})} g'(r)$ where each vertex $r \in V(G_{t'}) \setminus B_{t'}$, where $g'(r) > 0$, has at least two neighbors z_1 and z_2 such that $g'(z_1) < g'(r)$ and $g'(z_2) < g'(r)$; p' is g' restricted to $B_{t'}$ and g' does not contradict y regarding the vertices in $B_{t'}$.

Let g be an extension of g' for B_t such that $g(v) = p(v)$. So, we have that g is a quasi-infection time function of G_t where each vertex $r \in V(G_t) \setminus B_t$, where $g(r) > 0$, has at least two neighbors z_1 and z_2 such that $g(z_1) < g(r)$ and $g(z_2) < g(r)$; p is g restricted to B_t and, by a case analysis on y , g does not contradict f regarding the vertices in B_t .

Therefore, we have that $W_t(h) = \max_{g'} \max_{r \in V(G_t)} g'(r) \geq \max_{r \in V(G_t)} g(r) = \max(p(v), W_{t'}(p', y)) = \max(p(v), \max_{f'} W_{t'}(p', f'))$.

Now, let us prove that $W_t(h) \leq \max(p(v), \max_{f'} W_{t'}(p', f'))$. We have that $W_t(h) = \max_g \max_{r \in V(G_t)} g'(r)$. Let g be a quasi-infection time function of G_t that realizes this maximum. Let p' be g restricted to $B_{t'}$, which implies that p' is p restricted to $B_{t'}$, and $y : B_{t'} \rightarrow \{z, o_1, o_2, t_1, t_2\}$ be set accordingly to g restricted to $B_{t'}$, which, by a simple case analysis on f and p , will fall in one of the cases in the Lemma.

Thus, at least one of the two cases occur:

1. $\max_{r \in V(G_t)} g(r) = g(v)$
2. $\max_{r \in V(G_t)} g(r) = \max_{r \in V(G_{t'})} g(r)$.

If $g(v) = \max_{r \in V(G_t)} g(r)$ then we have that $W_t(h) = \max_{r \in V(G_t)} g(r) = g(v) = p(v) \leq \max(p(v), \max_{f'} W_{t'}(p', f'))$.

On the other hand, if $\max_{r \in V(G_t)} g(r) = \max_{r \in V(G_{t'})} g(r)$, then g is also a quasi-time infection function of $G_{t'}$ that realizes $\max_{g'} \max_{r \in V(G_{t'})} g'(r)$ and, since also each vertex $r \in V(G_{t'}) \setminus B_{t'}$, where $g(r) > 0$, has at least two neighbors z_1 and z_2 such that $g(z_1) < g(r)$ and $g(z_2) < g(r)$; p' is g restricted to $B_{t'}$ and g does not contradict y regarding the vertices in $B_{t'}$, then $W_{t'}(p', y) = \max_{r \in V(G_{t'})} g(r)$ and, therefore, we have that $W_t(h) = \max_{r \in V(G_t)} g(r) = W_{t'}(p', y) \leq \max_{f'} W_{t'}(p', f') \leq \max(p(v), \max_{f'} W_{t'}(p', f'))$. \square

Lemma 8. *Let t be a join node with children t_1 and t_2 and let $h = (p, f)$ be a valid index of W_t . If there is some pair (f_1, f_2) such that $\min(W_{t_1}(p, f_1), W_{t_2}(p, f_2)) > -1$, then $W_t(h) = \max_{(f_1, f_2)} \max(W_{t_1}(p, f_1), W_{t_2}(p, f_2))$ where the pair (f_1, f_2) iterates over all pair of functions $f_1 : B_{t_1} \rightarrow \{z, o_1, o_2, t_1, t_2\}$ and $f_2 : B_{t_2} \rightarrow \{z, o_1, o_2, t_1, t_2\}$ such that, for each $r \in B_t$, there are $i \neq j \in \{1, 2\}$ where:*

1. If $f(r) = z$ then $f_i(r) = z$ and $f_j(r) = z$;
2. If $f(r) = o_1$ then $f_i(r) = z$ and $f_j(r) = o_1$;
3. If $f(r) = o_2$ then $f_i(r) = z$ and $f_j(r) = o_2$;
4. If $f(r) = t_1$ then either:
 - (a) $f_i(r) = z$ and $f_j(r) = t_1$; or

- (b) $f_i(r) = o_1$ and $f_j(r) = o_2$; or
- (c) $f_i(r) = o_1$ and $f_j(r) = t_2$; or
- (d) $f_i(r) = o_2$ and $f_j(r) = t_1$; or
- (e) $f_i(r) = t_1$ and $f_j(r) = t_2$.

5. If $f(r) = t_2$ then either:

- (a) $f_i(r) = z$ and $f_j(r) = t_2$; or
- (b) $f_i(r) = o_2$ and $f_j(r) = o_2$; or
- (c) $f_i(r) = o_2$ and $f_j(r) = t_2$; or
- (d) $f_i(r) = t_2$ and $f_j(r) = t_2$.

If there is no pair (f_1, f_2) such that $\min(W_{t_1}(p, f_1), W_{t_2}(p, f_2)) > -1$, then $W_t(h) = -1$

Proof. First, suppose that there is no pair (f_1, f_2) such that $\min(W_{t_1}(p, f_1), W_{t_2}(p, f_2)) > -1$ where f_1 and f_2 are as stated in the Lemma. Thus, we have that, for all pairs (f_1, f_2) as stated in the Lemma, there is a $i \in \{1, 2\}$, such that there is no quasi-infection time function g_i of G_{t_i} such that each vertex $v \in V(G_{t_i}) \setminus B_{t_i}$, where $g_i(v) > 0$, has at least two neighbors z_1 and z_2 such that $g_i(z_1) < g_i(v)$ and $g_i(z_2) < g_i(v)$; p is g_i restricted to B_{t_i} and g_i does not contradict f_i regarding the vertices in B_{t_i} .

Suppose, by contradiction, that there is a quasi-infection time function g of G_t such that each vertex $v \in V(G_t) \setminus B_t$, where $g(v) > 0$, has at least two neighbors z_1 and z_2 such that $g(z_1) < g(v)$ and $g(z_2) < g(v)$; p is g restricted to B_t and, for each vertex $v \in B_t$, g does not contradict f regarding the vertices in B_t . Letting g_1 be g restricted to G_{t_1} and g_2 be g restricted to G_{t_2} and (f_1, f_2) be as stated in the Lemma, basing on f , we have that for all $i \in \{1, 2\}$, g_i is a quasi-infection time function of G_{t_i} such that each vertex $v \in V(G_{t_i}) \setminus B_{t_i}$, where $g_i(v) > 0$, has at least two neighbors z_1 and z_2 such that $g_i(z_1) < g_i(v)$ and $g_i(z_2) < g_i(v)$; p is g_i restricted to B_{t_i} and g_i does not contradict f_i regarding the vertices in B_{t_i} . However, this contradicts the former paragraph, and, hence, we have that there is no quasi-infection time function g of G_t such that each vertex $v \in V(G_t) \setminus B_t$, where $g(v) > 0$, has at least two neighbors z_1 and z_2 such that $g(z_1) < g(v)$ and $g(z_2) < g(v)$; p is g restricted to B_t and, for each vertex $v \in B_t$, g does not contradict f regarding the vertices in B_t . Therefore, $W_t(h) = -1$.

Now, suppose that there is some pair (f_1, f_2) such that $\min(W_{t_1}(p, f_1), W_{t_2}(p, f_2)) > -1$.

First, let us prove that $W_t(h) \geq \max_{(f_1, f_2)} \max(W_{t_1}(p, f_1), W_{t_2}(p, f_2))$. Let (y_1, y_2) be a pair of functions that realizes the previous maximum ($\max_{(f_1, f_2)} \max(W_{t_1}(p, f_1), W_{t_2}(p, f_2))$). Also, for all $i \in \{1, 2\}$, let g_i be the quasi-infection time function of G_{t_i} such that $W_{t_i}(p, y_i) = \max_{r \in V(G_{t_i})} g_i(r)$ where each vertex $r \in V(G_{t_i}) \setminus B_{t_i}$, where $g_i(r) > 0$, has at least two neighbors z_1 and z_2 such that $g_i(z_1) < g_i(r)$ and $g_i(z_2) < g_i(r)$; p is g_i restricted to B_{t_i} and g_i does not contradict y_i regarding the vertices in B_{t_i} .

Let g be a quasi-infection time function of G_t where $g(v) = g_1(v)$, if $v \in V(G_{t_1})$, and $g(v) = g_2(v)$, if $v \in V(G_{t_2})$. Note that, since, for all $v \in V(G_{t_1}) \cap V(G_{t_2}) = B_t$, $g_1(v) = g_2(v) = p(v)$, then g is well defined.

We have that g is a quasi-infection time function of G_t such that each vertex $r \in V(G_t) \setminus B_t$, where $g(r) > 0$, has at least two neighbors z_1 and z_2 such that $g(z_1) < g(r)$ and $g(z_2) < g(r)$; p is g restricted to B_t and, by a simple case analysis on y_1 and y_2 , we have that g does not contradict f regarding the vertices in B_t .

Therefore, $W_t(h) = \max_{g'} \max_{r \in V(G_t)} g'(r) \geq \max_{r \in V(G_t)} g(r) = \max(W_{t_1}(p, y_1), W_{t_2}(p, y_2)) = \max_{(f_1, f_2)} \max(W_{t_1}(p, f_1), W_{t_2}(p, f_2))$.

Now, let us prove that $W_t(h) \leq \max_{(f_1, f_2)} \max(W_{t_1}(p, f_1), W_{t_2}(p, f_2))$. We have that $W_t(h) = \max'_g \max_{r \in V(G_t)} g'(r)$. Let g be a quasi-infection time function of G_t that realizes this maximum. Let g_1 be g restricted to G_{t_1} and g_2 be g restricted to G_{t_2} . Also, let $y_1 : B_{t_1} \rightarrow \{z, o_1, o_2, t_1, t_2\}$ be set accordingly to g_1 restricted to the graph G_{t_1} and $y_2 : B_{t_2} \rightarrow \{z, o_1, o_2, t_1, t_2\}$ be set accordingly to g_2 restricted to the graph G_{t_2} . By case analysis on f , it is easy to see that y_1 and y_2 will fall in one of the cases in the Lemma.

Thus, we have that, for all $i \in \{1, 2\}$, g_i is a quasi-infection time functions of G_{t_i} such that each vertex $v \in V(G_{t_i}) \setminus B_{t_i}$, where $g_i(v) > 0$, has at least two neighbors z_1 and z_2 such that $g_i(z_1) < g_i(v)$ and $g_i(z_2) < g_i(v)$; p is g_i restricted to B_{t_i} and g_i does not contradict y_i regarding the vertices in B_{t_i} .

Additionally, note that $\max_{r \in V(G_t)} g(r) = \max_{r \in V(G_{t_1})} g_1(r)$ or $\max_{r \in V(G_t)} g(r) = \max_{r \in V(G_{t_2})} g_2(r)$ or possibly both.

Therefore, we have that $W_t(h) = \max_{r \in V(G_t)} g(r) = \max(\max_{r \in V(G_{t_1})} g_1(r), \max_{r \in V(G_{t_2})} g_2(r)) \leq \max(W_{t_1}(p, y_1), W_{t_2}(p, y_2)) \leq \max_{(f_1, f_2)} \max(W_{t_1}(p, f_1), W_{t_2}(p, f_2))$. \square

Proof of Theorem 8. For each node t , we have that W_t has $5^{w+1} \cdot n^{w+1}$ indexes. Each index can be computed by checking, in a join node, $O(10^{w+1})$ indexes of its children, in a introduce node, $O(1)$ indexes of its child, and, in a forget node, $O(n)$ indexes of its child. However, one can reduce the time needed to compute the entire table of the forget node simple by computing its table while computing the table of its child, i.e., if t is the forget node and t' is its child, we can initialize $W_t(h)$ with -1 for every index h of W_t and every time we compute the value $W_{t'}(p, f)$, letting p' and f' be, respectively, p and f restricted to B_t , if $W_{t'}(p, f) > W_t(p', f')$ and either $f(v) = t_1$ or $f(v) = t_2$ or $p(v) = 0$, then we update the value of $W_t(p', f')$ with the value $W_{t'}(p, f)$. This would add only $O(1)$ time to compute each index of its child's table. Thus, we can find $t(G)$ in $O(n \cdot 10^{w+1} \cdot (5^{w+1} \cdot n^{w+1})) = O(50^{w+1} \cdot n^{w+2})$ time.

In a very similar way, considering that we only have to check the indexes where p is a function $p : B_t \rightarrow \{0, 1, \dots, k\}$, we can show that, for each node t , we have that W_t has $5^{w+1} \cdot k^{w+1}$ indexes and each index can be checked worst case in $O(10^{w+1})$, which happens when t is a join node, and then we can find $t(G)$ in $O(n \cdot 10^{w+1} \cdot (5^{w+1} \cdot k^{w+1})) = O((50k)^{w+1}n)$ time. \square

7. Acknowledgments

The statements of some of the results of this paper appeared in WG-2015 (Workshop on Graph-Theoretic Concepts in Computer Science). This research was partially supported by CNPq (Universal Proc. 478744/2013-7) and FAPESP (Proc. 2013/03447-6).

- [1] H. Amini, *Bootstrap percolation in living neural networks*. J. Stat. Phys. **141** no. 3 (2010), 459–475.
- [2] R. Araújo, R. Sampaio, V. Santos, J. Szwarcfiter, *The convexity of induced paths of order three and applications: complexity aspects*. Discrete Applied Mathematics (2016), to appear.
- [3] F. Benevides, V. Campos, M. Dourado, R. Sampaio, A. Silva, *The maximum time of 2-neighbour bootstrap percolation: algorithmic aspects*. In: The Seventh European Conference on Combinatorics, Graph Theory and Applications, Ser. CRM **16**, pp 135–139. Scuola Normale Superiore (2013).
- [4] F. Benevides, M. Przykucki, *Maximum percolation time in two-dimensional Bootstrap Percolation*. SIAM Journal on Discrete Mathematics, **29** no. 1 (2015), pp. 224–251.
- [5] F. Benevides, M. Przykucki, *On slowly percolating sets of minimal size in bootstrap percolation*. The Electronic Journal of Combinatorics **20** no. 2 (2013), P46.
- [6] H. Bodlaender, *A linear time algorithm for finding tree-decompositions of small treewidth*. SIAM J. Comput. **25** (1996), 1305–1317.
- [7] N. Chen, *On the Approximability of Influence in Social Networks*. SIAM J. Discrete Math. **23** no. 3 (2009), 1400–1415.
- [8] E. Coelho, M. Dourado, R. Sampaio, *Inapproximability Results for Graph Convexity Parameters*. Workshop on Approximation and Online Algorithms (WAOA-2013), Lecture Notes in Computer Science **8447** (2014), 97–107.
- [9] M. Fellows, D. Hermelin, F. Rosamond, S. Vialette, *On the parameterized complexity of multiple-interval graph problems*. Theoretical Computer Science **410** no. 1 (2009), 53–61.
- [10] M. Fellows, F. Fomin, D. Lokshtanov, F. Rosamond, S. Saurabh, S. Szeider, C. Thomassen, *On the complexity of some colorful problems parameterized by treewidth*. Information and Computation **209** no.2 (2011), 143–153.
- [11] A. Fey, L. Levine, Y. Peres, *Growth rates and explosions in sandpiles*. J. Stat. Phys. **138** (2010), 143–159.
- [12] J. Flum, M. Grohe, *Parameterized Complexity Theory*. Springer, 2010.
- [13] R. Impagliazzo, R. Paturi, *On the complexity of k -sat*. Journal of Computer and System Sciences **62** (2001), 367–375.
- [14] A. Itai, C. Papadimitriou, J. Szwarcfiter, *Paths in grid graphs*. SIAM Journal on Computing **11** (1982), 676–686.

- [15] T. Kloks, *Treewidth: computation and approximation*. Lecture Notes in Computer Science **842** (1991).
- [16] D. Lokshtanov, D. Marx, S. Saurabh, *Lower bounds based on the Exponential Time Hypothesis*. Bulletin of the EATCS **105** (2011), 41–72.
- [17] T. Marcilon, S. Nascimento, R. Sampaio, *The maximum time of 2-neighbour bootstrap percolation: complexity results*. Proceedings of the 40th Workshop on Graph-Theoretic Concepts in Computer Science (WG), 2014, Orleans, France. Lecture Notes in Computer Science **8747** (2014), 372–383.
- [18] R. Morris, *Minimal percolating sets in bootstrap percolation*. The Electronic Journal of Combinatorics **16** no. 1 (2009), 20pp.
- [19] C. Papadimitriou, U. Vazirani, *On two geometric problems related to the travelling salesman problem*. Journal of Algorithms **5** no. 2 (1984), 231–246.
- [20] M. Przykucki, *Maximal Percolation Time in Hypercubes Under 2-Bootstrap Percolation*. The Electronic Journal of Combinatorics **19** no. 2 (2012), 41pp.
- [21] E. Riedl, *Largest minimal percolating sets in hypercubes under 2-bootstrap percolation*. The Electronic Journal of Combinatorics **17** no. 1 (2010), 13pp.
- [22] N. Robertson, P. Seymour, *Graph minors III: Planar tree-width*. Journal of Combinatorial Theory Series B **36** no. 1 (1984), 49–64.
- [23] C. Tovey, *A simplified NP-complete satisfiability problem*. Discrete Applied Mathematics **8** no. 1 (1984), 85–89.
- [24] C. Umans, W. Lenhart, *Hamiltonian cycles in solid grid graphs*. Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society (1997), 496–505.

APÊNDICE C – ON THE PARAMETERIZED COMPLEXITY OF THE HULL NUMBER PROBLEM

Neste apêndice, nós apresentamos as provas na íntegra em inglês de todos os resultados do Capítulo 5. Tais resultados estão em fase de elaboração e estão todos relacionados ao problema NÚMERO DE ENVOLTÓRIA. Eles são:

- A $\mathbf{W}[1]$ -dificuldade na convexidade P_3 para o parâmetro k em grafos livres de K_3 de grau máximo seis;
- A $\mathbf{W}[1]$ -dificuldade na convexidade geodésica para o parâmetro k em grafos com diâmetro dois;
- A $\mathbf{W}[1]$ -dificuldade na convexidade geodésica para o parâmetro $tw(G) + k$ e
- Um algoritmo XP para o problema na convexidade geodésica para o parâmetro $tw(G)$.

On the parameterized complexity of the hull number problem[☆]

J. Araujo^a, M. Kanté^b, T. Marcilon^c, R. Sampaio^c

^aDepartment of Mathematics, ParGO Research Group, Federal University of Ceará, Brazil.

^bAlgorithmes, Graphes et Complexité Team, LIMOS, Université Blaise Pascal, France.

^cDepartment of Computer Science, ParGO Research Group, Federal University of Ceará, Brazil.

Abstract

Given a simple undirected graph $G = (V, E)$ and $u, v \in V(G)$, let $I_g[u, v]$ be the set of vertices that belong to some shortest u, v -path (a.k.a. u, v -geodesic). Given $S \subseteq V(G)$, define $I_g[S] = \bigcup_{u, v \in S} I_g[u, v]$. A set $S \subseteq V(G)$ is *convex* in the geodesic convexity if $I_g[S] = S$. The *convex hull* of $S \subseteq V(G)$, denoted by $H_g[S]$, is the convex set of minimum cardinality that contains S . A set $S \subseteq V(G)$ is a *hull set* if $H_g[S] = V(G)$. The *geodetic hull number* of G is the cardinality of a hull set of minimum size and it is denoted $hn_g(G)$.

Define as $I_{P_3}[u, v]$ the set of all vertices that belong to some u, v -path on three vertices. Then, one can analogously define $I_{P_3}[S]$, convex set, convex hull, hull set and hull number $hn_{P_3}(G)$ in the P_3 -convexity.

In this work, we first prove that determining whether $hn_{P_3}(G) \leq k$ is $W[1]$ -hard, when k is the parameter, even if G is triangle-free. A straightforward consequence is that determining whether $hn_g(G) \leq k$ is $W[1]$ -hard, when k is the parameter. This answers a question raised by [Araujo et al., 2016].

We go further on the study of the parameterized complexity of these problems and we show that determining whether $hn_g(G) \leq k$ is also $W[1]$ -hard when parameterized by $tw(G) + k$, where $tw(G)$ is the treewidth of G .

Finally, we prove that if we only consider $tw(G)$ as parameter, then the problem of determining whether $hn_g(G) \leq k$ belongs to XP, for a given $k \in \mathbb{N}$.

Keywords: Graph Convexity, Geodetic Convexity, P_3 -convexity, Hull Number, Parameterized Complexity, Treewidth

1. Introduction

All the graphs in this work are simple and undirected. For basic notions in Graph Theory and Computational Complexity, including Parameterized Complexity, the reader is referred to [20, 26].

A finite *convexity space* [30, 40] is a pair (V, \mathcal{C}) consisting of a finite ground set V and a set \mathcal{C} of subsets of V satisfying $\emptyset, V \in \mathcal{C}$ and if $C_1, C_2 \in \mathcal{C}$, then $C_1 \cap C_2 \in \mathcal{C}$. The members of \mathcal{C} are called \mathcal{C} -*convex sets* and the *convex hull* of a set S is the minimum convex set $H[S] \in \mathcal{C}$ containing S . We say that a set S is a *hull set* if its convex hull is V . The hull number is the cardinality of a minimum size hull set. These notions generalize the respective ones in Euclidean geometry [39].

A convexity space (V, \mathcal{C}) is an *interval convexity* [10] if there is a so-called *interval function* $I : \binom{V}{2} \rightarrow 2^V$ such that a subset C of V belongs to \mathcal{C} if and only if $I(\{x, y\}) \subseteq C$, for every two distinct elements x and y of C . We usually denote $I(\{x, y\})$ by $I[x, y]$.

For every $S \subseteq V$, let $I[S] = \bigcup_{x, y \in S} I[x, y]$, $I^0[S] = S$ and $I^k[S] = I[I^{k-1}[S]]$. In interval convexities, the convex hull of a set S can be computed by exhaustively applying the corresponding interval function until obtaining a convex set and a set S is *convex* if $I[S] = S$.

For a given graph $G = (V, E)$, one can use $V(G)$ as ground set and a family of its subsets to obtain a graph convexity. Notions related to graph convexities have been studied since the 70's [22, 34].

The most studied graph convexities defined by interval functions are those in which $I[x, y]$ is the union of paths between x and y with some particular property. Some common examples are the P_3 -convexity [22], geodetic convexity [24] and monophonic convexity [21].

In this work, we focus on the P_3 and the geodetic convexity. Thus, let us formalize the parameters we study here. Let us start by the geodesic convexity.

Let $G = (V, E)$ be a graph. For every $u, v \in V(G)$, we define $I_g[u, v]$ as the set of vertices that belong to some shortest u, v -path (a.k.a. u, v -geodesic). For every $S \subseteq V(G)$, we say that $I_g[S] = \bigcup_{u, v \in S} I_g[u, v]$. We

[☆]This work was partially supported by ...

Email addresses: julio@mat.ufc.br (J. Araujo), kante@isima.fr (M. Kanté), thiagomarcilon@gmail.com (T. Marcilon), rudini@lia.ufc.br (R. Sampaio)

denote by $H_g[S]$ the convex hull of $S \subseteq V(G)$ in the geodesic convexity. The *geodetic hull number* of G is the cardinality of a hull set of minimum size and it is denoted $hn_g(G)$.

Now, let us formalize the P_3 convexity. Let $G = (V, E)$ be a graph. For every $u, v \in V(G)$, let $I_{P_3}[u, v]$ denote the set of vertices that are in some shortest u, v -path (a.k.a. u, v -geodesic). For every $S \subseteq V(G)$, let $I_{P_3}[S] = \bigcup_{u, v \in S} I_{P_3}[u, v]$. We denote by $H_{P_3}[S]$ the convex hull of $S \subseteq V(G)$ in the P_3 convexity. The P_3 *hull number* of G is the cardinality of a minimum hull set and it is denoted $hn_{P_3}(G)$.

1.1. State of Art

In the literature, there are several works regarding the hull number in the P_3 and the geodesic convexity. Many of these works focus on structural results, usually providing lower and upper bounds for these graph parameters [7, 6, 23, 16]. In the sequel, we list some computational complexity results.

Regarding the geodesic convexity, in 2009, Dourado *et al.* [15] proved that determining whether $hn_g(G) \leq k$ is an NP-complete problem, given an arbitrary graph G and $k \in \mathbb{N}$. On the other hand, they presented polynomial-time algorithms to compute $hn_g(G)$ whenever G is an unit-interval graph, or a cograph or a split graph. Later, in 2013, Araujo *et al.* [2] strengthened the proof in [15] by proving that determining whether $hn_g(G) \leq k$ is an NP-complete problem, given a *bipartite* graph G and $k \in \mathbb{N}$. They also showed that computing $hn_g(G)$ can be done in polynomial time whenever G is a complement of a bipartite graph, a cactus, or a $(q, q-4)$ -graph G . In particular, the algorithm they present for $(q, q-4)$ -graphs is a fixed parameter tractable (FPT) algorithm for the problem parameterized by $q(G)$, which was introduced in [5] and is the minimum q such that G is a $(q, q-4)$ graph. In 2015, Coelho *et al.* [14] showed that the problem is APX-hard. In 2016, Araujo *et al.* [3] showed a polynomial-time algorithm that solves the problem for any $\{P_5, K_3\}$ -free graph and a fixed parameter tractable algorithm that solves the problem parameterized by the neighborhood diversity [29] of the input graph. Also in 2016, Albenque and Knauer [1] proved that the problem is NP-complete even on partial cubes. Yet in 2016, Dourado, Penso and Rautenbach [17] showed that the problem is NP-complete even on P_9 -free graphs. They also proved that the problem is polynomial on $\{\text{claw}, P_5\}$ -free graphs, on C_3 -free graphs where any six vertices induces at most one P_5 and on $\{C_3, C_4, \dots, C_{k-2}, P_k\}$ -free graphs for any integer k .

Regarding the P_3 convexity, in 2009, Chen [13] showed that the problem is NP-complete even on bipartite graphs with fixed maximum degree at least eight and hard to approximate within a ratio $O(2^{\log^{1-\varepsilon} n})$, for any $\varepsilon > 0$, unless some unexpected complexity-theoretic collapse occurs. In 2011, Ben-Zwi *et al.* in [8] showed an FPT-algorithm for the problem parameterized by the treewidth $tw(G)$ of the input graph G . In 2013, Nichterlein *et al.* [35] showed that determining whether $hn_{P_3}(G) \leq k$ is W[2]-hard when the parameter is k even in bipartite graphs with diameter four. They also showed FPT algorithms for the TARGET SELECTION PROBLEM, which is a more general problem, when the parameter is the vertex cover number, feedback edge set number, cluster edge deletion number and cluster editing number. In 2015, in [14], Coelho *et al.* showed that the problem is also APX-hard. Also in 2015, Campos *et al.* [11] showed that there is an FPT-algorithm for computing $hn_{P_3}(G)$, when the parameter is $q(G)$. Yet in 2015, Penso *et al.* [36] showed that the problem is NP-complete in graphs with maximum degree three and planar graphs with maximum degree four. They also presented polynomial algorithms for the problem when the graph is cubic, or the graph has bounded feedback vertex set number, or the minimum degree of the graph is n/c , where n is the number of vertices of the graph and c is a constant.

1.2. Our Contributions

Although it was already known that determining whether $hn_{P_3}(G) \leq k$ is FPT when the parameter is $tw(G)$ or $q(G)$, it is still open the parameterized complexity for the natural parameter k , the size of the solution. We first prove that:

Theorem 1. *Determining whether $hn_{P_3}(G) \leq k$ is W[1]-hard, when k is the parameter, even if G is triangle-free with maximum degree six.*

Based on this result, we can extend it for the geodesic convexity.

Theorem 2. *Determining whether $hn_g(G) \leq k$ is W[1]-hard, when k is the parameter, even if G has diameter two.*

Since Ben-Zwi *et al.* [8] obtained an FPT algorithm to determine whether $hn_{P_3}(G) \leq k$ when parameterized by the treewidth of G , we investigate this problem for the geodesic convexity and we obtain the following result. For some background on the parameter treewidth $tw(G)$ we refer the reader to [37, 38].

Theorem 3. *Determining whether $hn_g(G) \leq k$ is W[1]-hard, when $tw(G) + k$ is the parameter.*

Despite the W[1]-hardness of the geodetic hull number parameterized by $tw(G) + k$, we obtain an XP algorithm.

Theorem 4. *The geodesic hull number of a graph can be computed in $O((tw + 1)^{tw+6} n^{2tw+5})$ time.*

2. W[1]-hardness of the P_3 and Geodetic Hull Number Problem

In this section, we are going to prove that the Geodetic Hull Number Problem is W[1]-hard when parameterized by k and by both k and the treewidth of the input graph. Also, we will prove that the P_3 Hull Number Problem is W[1]-hard when parameterized by k .

The W[1]-hardness results in this section use two problems in their parameterized reductions: the MULTICOLORED CLIQUE problem and the GRID TILING problem. The MULTICOLORED CLIQUE problem is a W[1]-hard problem when parameterized by its input k [25] and consists in, given an integer k and a k -colored graph G , finding a multicolored k -clique, that is, a k -clique such that all vertices are colored with distinct colors. We will actually work with a restriction of the MULTICOLORED CLIQUE problem where the input graph G has minimum degree at least 2; for each pair of colors x and y , there is at least one pair of adjacent vertices, one colored with x and the other with y ; and the set vertices colored with each color is an independent set. MULTICOLORED CLIQUE problem is still W[1]-hard even with these restrictions because (a) any pendant vertex cannot be part of a multicolored k -clique, since MULTICOLORED CLIQUE problem is W[1]-hard even for $k \geq 3$, (b) if there are two colors in G such that there is no edge between any two vertices colored each with one of these colors, then G cannot have a multicolored k -clique and (c) any edge between two vertices with the same color does not take part in any multicolored k -clique.

The Grid Tiling problem is also a W[1]-hard problem when parameterized by its input k [33] and consists in, given integers k and n and a collection \mathcal{S} of k^2 non-empty sets $S_{i,j} \subseteq [n]^2$, for $1 \leq i, j \leq k$, finding, for each $1 \leq i, j \leq k$, an ordered pair $s_{i,j} \in S_{i,j}$ such that, if $s_{i,j} = (a, b)$ and $s_{i+1,j} = (a', b')$, then $a = a'$ and, if $s_{i,j} = (a, b)$ and $s_{i,j+1} = (a', b')$, then $b = b'$.

Before we present the next theorem, let us define some concepts. Let v be a vertex in a graph G , a false twin v' of v is a vertex of G that has the exact same neighborhood as v , that is, $N_G(v') = N_G(v)$. Also, let us define a strictly binary many-tree. A strictly binary many-tree is any graph that can be constructed following these instructions:

1. Add a strictly binary tree T ;
2. While there are nodes in T that do not have a false twin, pick one such node v of T and add a false twin of v .

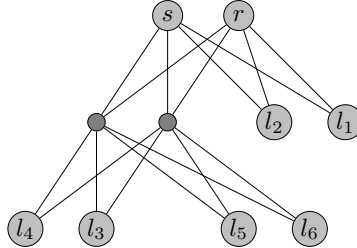


Figure 1: Strictly binary many-tree with many-root $\{r, s\}$ and three many-leaves $\{l_1, l_2\}$, $\{l_3, l_4\}$ and $\{l_5, l_6\}$.

Figure 1 illustrates a strictly binary many-tree with three many-leaves. We define the many-root of a strictly binary many-tree as the set composed by the root of the original strictly binary tree and its false twin. Similarly, a many-leaf is a set composed by a leaf of the original strictly binary tree and its false twin. Note that the maximum degree of a strictly binary many-tree is six.

Theorem 1. *Determining whether $\text{hnp}_3(G) \leq k$ is W[1]-hard, when k is the parameter, even if G is triangle-free with maximum degree six.*

Proof. We are going to prove that the P_3 Hull Number problem is W[1]-hard when parameterized by k , even in K_3 -free graphs, by showing a parameterized reduction from the Multicolored Clique Problem parameterized by the input k where the input graph G has minimum degree at least 2; for each pair of distinct colors x and y , there is at least one pair of adjacent vertices, one colored with x and the other with y , and the set vertices colored with each color is an independent set.

Before we describe the reduction, let us define a *strictly binary tree*. A *strictly binary tree* is a rooted binary tree where all internal vertices have either zero or two children.

Let (G, k) be an instance of the Multicolored Clique problem. Let $n = |V(G)|$ and $m = |E(G)|$. Let $V_x = \{v_1^x, v_2^x, \dots, v_{t_x}^x\}$ be the set of vertices of G with color x , where $t_x = |V_x|$, and let $d_i^x = d_G(v_i^x)$.

For each color $1 \leq x \leq k$, construct a *color gadget* in the following manner:

1. Make four paths $a_1^x, a_2^x, \dots, a_{t_x}^x$; $b_1^x, b_2^x, \dots, b_{t_x}^x$; $c_1^x, c_2^x, \dots, c_{t_x}^x$ and $d_1^x, d_2^x, \dots, d_{t_x}^x$ and let A_x, B_x, C_x and D_x be the set of vertices of each path, respectively.

2. For each vertex v_p^x in V_x , add the vertices y_p^x and w_p^x . After that, add an edge between the vertices a_p^x and y_p^x , b_p^x and y_p^x , c_p^x and w_p^x and d_p^x and w_p^x .
3. For each vertex v_p^x in V_x , add a strictly binary tree rooted at the node y_p^x and with d_p^x leaves $L_p^x = \{l_p^{x,1}, l_p^{x,2}, \dots, l_p^{x,d_p^x}\}$. Add an edge between w_p^x and both children of y_p^x . Also, let T_p^x be the set of all nodes of this tree including w_p^x .
4. Now, for each vertex v_p^x in V_x , add any strictly binary many-tree with many-root $\{r_p^x, s_p^x\}$ and with d_p^x many-leaves. Link the vertices r_p^x and s_p^x to both y_p^x and w_p^x . Let Q_p^x be this many-tree.
5. Finally, for each many-leaf $\{l, l'\}$ of Q_p^x , link both l and l' to one vertex in L_p^x so that each vertex in L_p^x is adjacent to only two vertices of Q_p^x .

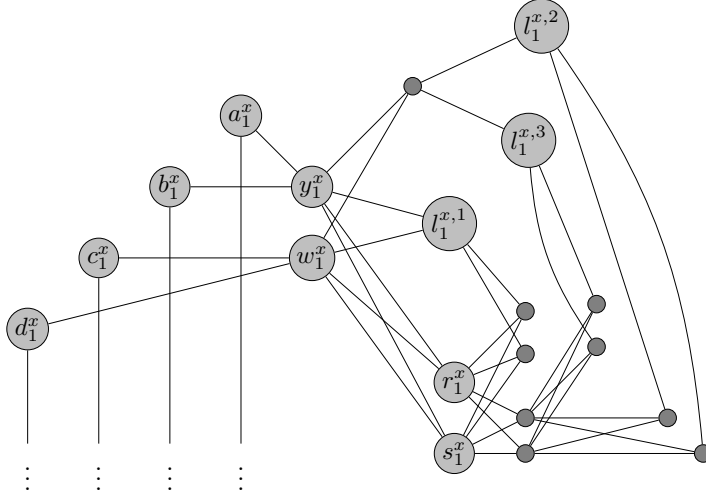


Figure 2: Partial Color gadget of a color x for $d_1^x = 3$.

Figure 2 partially illustrates a color gadget of a color x for $d_1^x = 3$. Let $L^x = \bigcup_{j=1}^{t_x} L_j^x$ and let $L = \bigcup_{x=1}^k L^x$. Finally, to end our construction, for each pair of different colors $1 \leq x, x' \leq k$, construct an *edge gadget* in the following manner:

1. For each edge between some pair of vertices v_p^x and $v_{p'}^{x'}$ in G , add the vertices $e_{p,p'}^{x,x'}$ and $f_{p,p'}^{x,x'}$ and link both to the same vertex in L_p^x and to the same vertex in $L_{p'}^{x'}$. This has to be done so every different pair of vertices $e_{p,p'}^{x,x'}$ and $f_{p,p'}^{x,x'}$ in all edge gadgets is linked to a unique vertex in L_p^x and a unique vertex in $L_{p'}^{x'}$ and, thus, by the end of the construction, every vertex in L must be linked to only two vertices in some edge gadget.
2. Now, add any strictly binary many-tree such that each many-leaf is one set $\{e_{p,p'}^{x,x'}, f_{p,p'}^{x,x'}\}$.

Let $E_{x,x'}$ be the strictly binary many-tree of the edge gadget of the colors x and x' . The Figure 3 illustrates an edge gadget such that there is an edge between the vertices v_1^x and $v_3^{x'}$, v_1^x and $v_4^{x'}$ and v_2^x and $v_3^{x'}$.

Note, by our description, that the graph resulting from this construction is a K_3 -free graph with maximum degree six that has $2n + 16m - 2\binom{k}{2}$ vertices. Also, the reduction runs in $O(m + n + k^2)$ time.

Let G' be the resulting graph when we apply our reduction to the graph G . Let us prove that G contains a multicolored clique if and only if $\text{hn}(G') \leq 4k$. First, suppose that $C = \{v_{c_1}^1, v_{c_2}^2, \dots, v_{c_k}^k\}$ is a multicolored k -clique in G , where each $v_{c_x}^x$ has color x . Define $S = \bigcup_{x=1}^k \{a_{c_x}^x, b_{c_x}^x, c_{c_x}^x, d_{c_x}^x\}$. We claim that S is a hull set of G' .

For each vertex $v_{c_x}^x$ in C , we have that $y_{c_x}^x \in I[a_{c_x}^x, b_{c_x}^x]$. Thus, we have that, for each vertex $v_{c_x}^x$ in C , $y_{c_x}^x \in I[a_{c_x}^x, b_{c_x}^x]$ and $w_{c_x}^x \in I[c_{c_x}^x, d_{c_x}^x]$, which implies that $T_{c_x}^x \cup V(Q_{c_x}^x) \subseteq H[S]$. Since C is a multicolored k -clique in G , then, for each $1 \leq x, x' \leq k$, the pair of vertices $e_{c_x, c_{x'}}^{x, x'}$ and $f_{c_x, c_{x'}}^{x, x'}$ are in $H[S]$ and, consequently, $V(E_{x, x'}) \subseteq H[S]$. Since, for each $1 \leq x, x' \leq k$, $V(E_{x, x'}) \subseteq H[S]$, then $L \subseteq H[S]$.

Let v_p^x be an arbitrary vertex in $V(G) \setminus C$ with color x . Since $L \subseteq H[S]$, then $T_p^x \subseteq H[S]$, which implies that $V(Q_p^x) \subseteq H[S]$.

Finally, since, for each $1 \leq x \leq k$ and $1 \leq p \leq t_x$, $\{y_p^x, w_p^x\} \subseteq H[S]$ and there is one vertex in $S \cap A_x$, $S \cap B_x$, $S \cap C_x$ and $S \cap D_x$, we have that $A_x \cup B_x \cup C_x \cup D_x \subseteq H[S]$. Therefore, we have that $H[S] = V(G')$, which implies that $\text{hn}(G') \leq |S| = 4k$.

Now, suppose that G does not have any multicolored k -clique. Since every hull set must have, for each $1 \leq x \leq k$, at least one vertex in each set A_x , B_x , C_x and D_x , then any hull set has size at least $4k$.

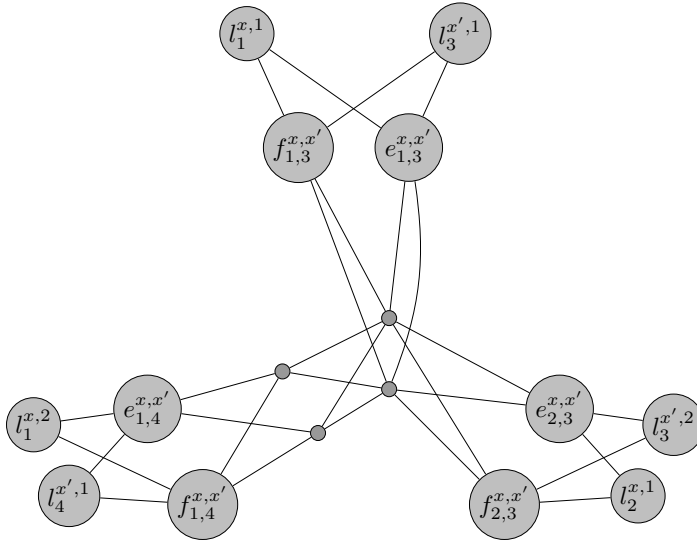


Figure 3: Edge gadget of the colors x and x' such that there is an edge between the vertices v_1^x and $v_3^{x'}$, v_1^x and $v_4^{x'}$ and v_2^x and $v_3^{x'}$.

Therefore, to prove that $hn(G') \geq 4k + 1$, is enough to prove that $H[S] \neq V(G)$ for any set S that contains exactly one vertex in each set A_x, B_x, C_x and D_x , for each $1 \leq x \leq k$.

Thus, henceforth, let S be an arbitrary set that contains exactly one vertex in each set A_x, B_x, C_x and D_x , for each $1 \leq x \leq k$. Let us split this proof in two cases.

The first case is when, for all $1 \leq x \leq k$, $S \cap (A_x \cup B_x \cup C_x \cup D_x) = \{a_{i_x}^x, b_{i_x}^x, c_{i_x}^x, d_{i_x}^x\}$. Thus, for each x , $T_{i_x}^x \cup V(Q_{i_x}^x) \subseteq H[S]$.

However, since there is no multicolored k -clique in G , there are colors $1 \leq x, x' \leq k$ such that the vertices $v_{i_x}^x$ and $v_{i_{x'}}^{x'}$ are not neighbors in G . Hence, we have that $H[L_{i_x}^x \cup L_{i_{x'}}^{x'}] = \emptyset$.

This implies that $E_{x,x'} \not\subseteq H[S]$ because in order to have $E_{x,x'} \subseteq H[S]$, we must have two vertices $l_p^{x,i}$ and $l_q^{x',j}$ in $H[S]$ such that $\{v_p^x, v_q^{x'}\} \in E(G)$, which is impossible because, for any $v_p^x \neq v_{i_x}^x$ and $v_q^{x'} \neq v_{i_{x'}}^{x'}$, we have that $\{y_p^x, w_p^x\} \subseteq H[S] \Leftrightarrow L_p^x \subseteq H[S]$ and $\{y_q^{x'}, w_q^{x'}\} \in H[S] \Leftrightarrow L_q^{x'} \subseteq H[S]$. Thus, S is not a hull set.

The second case happens when the first case do not. Thus, let x be an integer such that $S \cap (A_x \cup B_x \cup C_x \cup D_x) = \{a_{i_1}^x, b_{i_2}^x, c_{i_3}^x, d_{i_4}^x\}$. Since there are integers $1 \leq p, q \leq 4$ such that $i_p^x \neq i_q^x$, then we have that no vertex in L^x is in $H[S]$, which implies that $E_{x,x'} \cap H[S] = \emptyset$, for all $1 \leq x' \leq k$. Thus, using the same arguments we used in the first case, we have that S is not a hull set.

Therefore, since S is not a hull set we have that $hn(G') \geq 4k + 1$. □

In [4], Araújo, Sampaio and Szwarcfiter introduced the P_3^* convexity. The P_3^* convexity is a graph convexity such that its interval function $I[x, y]$ returns all the vertices in an induced path of order three between x and y . In Theorem 3.1 of this work, they stated that, for every m and graph G , $hn_g(G) = hn_{P_3^*}(G')$, where $G' = G + K_m$, K_m is a clique with m vertices and $G_1 + G_2$ is the graph with vertex set $V(G_1) \cup V(G_2)$ and all the original edges of G_1 and G_2 plus all edges between a vertex of G_1 and a vertex of G_2 .

Also, in triangle-free graphs, since every P_3 is induced, the P_3^* convexity is the same as the P_3 convexity, that is, a set S is convex in the P_3^* convexity if and only if it is convex in the P_3 convexity. Therefore, we have the following lemma.

Lemma 1 (Consequence of Theorem 3.1 of [4]). *Let $G = (V, E)$ be a triangle-free graph such that $|V| \geq 3$, let $G' = (V \cup \{v\}, E \cup \{(v, w) \mid w \in V\})$. Then $hn_g(G) = hn_{P_3}(G')$.*

Theorem 2. *Determining whether $hn_g(G) \leq k$ is $W[1]$ -hard, when k is the parameter, even if G has diameter two.*

Proof. Let (G, k) be an instance of the Multicolored Clique problem and let $G' = (V', E')$ be the resulting graph from the reduction detailed in the proof of the Theorem 1. Since G' is triangle-free, applying Lemma 1, we have that there is a multicolored k -clique in G if and only if $hn_g(H) \leq 4k$, where $H = (V' \cup \{v\}, E' \cup \{(v, w) \mid w \in V'\})$. This is because, we proved that there is a multicolored k -clique in G if and only if $hn_{P_3}(G') \leq 4k$. Also, because

v is a universal vertex, H has diameter two. Therefore, determining whether $\text{hg}(G) \leq k$ is W[1]-hard, when k is the parameter, even if G has diameter two. \square

Theorem 3. *Determining whether $\text{hg}(G) \leq k$ is W[1]-hard, when $\text{tw}(G) + k$ is the parameter.*

Proof. We are going to prove that the Geodesic Hull Number problem is W[1]-hard when parameterized by both the treewidth of the input graph and k by showing a parameterized reduction from the Grid Tiling Problem parameterized by its input k .

Let (k, n, \mathcal{S}) be an instance of the Grid Tiling problem. First, let $Q = \bigcup_{S \in \mathcal{S}} \bigcup_{(x,y) \in S} \{x, y\}$ be the set of all the different positive integers that appear in some pair of some set in \mathcal{S} and $n' = |Q|$. Let $\pi : Q \rightarrow [n']$ be an arbitrary injective function. Note that $\sum_{S \in \mathcal{S}} |S| \geq |Q| = n'$.

Let $A_i = 6n' + i$ and $B_i = 6n' - i$ for each $1 \leq i \leq n'$. Note that $3B_{n'} > 2A_{n'}$, $A_i > B_j$, for any $1 \leq i, j \leq n'$ and, for every $1 \leq i \neq j \leq n'$, $A_i + B_i = A_j + B_j = 12n'$ and $A_i - B_i \neq A_j - B_j$. Let I be the interval function of the geodesic convexity and $f(a, b) = k(a - 1) + b$. Also, we say that two distinct pairs (i, j) and (i', j') are neighbors, where i, j, i' and j' are positive integers, if $i = i'$ and $|j' - j| = 1$ or if $j = j'$ and $|i' - i| = 1$. Finally, we say that a pair (i, j) is a corner pair if $i \in \{1, k\}$ and $j \in \{1, k\}$ and we say that it is a boundary pair if either $i \in \{1, k\}$ and $1 < j < k$ or $j \in \{1, k\}$ and $1 < i < k$.

To save some space in the figures, a weighted edge will represent a path like in Figure 4 and a weightless edge will represent an edge with weight $A_{n'}$.



Figure 4: A weighted edge on the right and the real structure on the left.

Before we describe the reduction, let us define a *complete binary tree*. A *complete binary tree* is a rooted binary tree where all internal vertices have two children and all leaves are at the same distance from the root.

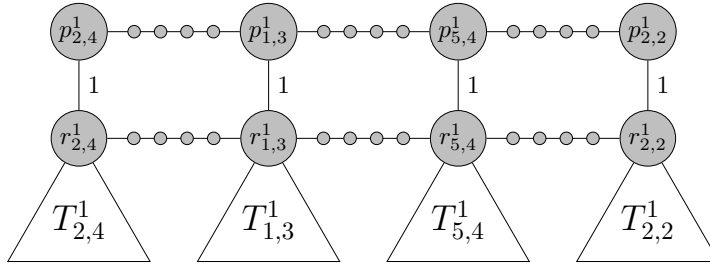


Figure 5: Paths $P_{1,1}$ and $R_{1,1}$ for $S_{1,1} = \{(2, 4), (1, 3), (5, 4), (2, 2)\}$.

Now, let us describe the reduction. First, add the vertices u, z and z' to the graph and link z' to z . For each set $S_{i,j} \in \mathcal{S}$, construct the *set gadget* $G_{i,j}$, where $f = f(i, j)$, $s = |S_{i,j}|$ and $S_{i,j} = \{(x_1, y_1), (x_2, y_2), \dots, (x_s, y_s)\}$, in the following manner:

1. Add two paths with length s , $P_f = p_{x_1, y_1}^f, p_{x_2, y_2}^f, \dots, p_{x_s, y_s}^f$ and $R_f = r_{x_1, y_1}^f, r_{x_2, y_2}^f, \dots, r_{x_s, y_s}^f$. Then, subdivide each edge $5A_{n'} - 1$ times in both paths. Link each vertex $p_{x,y}^f \in P_f$ to the vertex $r_{x,y}^f \in R_f$. In Figure 5, there is an example of the paths P_1 and R_1 , for $S_{1,1} = \{(2, 4), (1, 3), (5, 4), (2, 2)\}$;
2. For each $r_{x,y}^f \in R_f$, add a complete binary tree with four leaves $L_{x,y}^f = \{l_{x,y}^{f,e}, l_{x,y}^{f,w}, l_{x,y}^{f,n}, l_{x,y}^{f,s}\}$ and three internal vertices $z_{x,y}^{f,0}, z_{x,y}^{f,1}$ and $r_{x,y}^f$, where $l_{x,y}^{f,e}, l_{x,y}^{f,w}$ are the children of $z_{x,y}^{f,0}$, $l_{x,y}^{f,n}, l_{x,y}^{f,s}$ are the children of $z_{x,y}^{f,1}$ and $r_{x,y}^f$ is the root. For each $c \in \{e, w, n, s\}$, let $L_f^c = \{l_{x_1, y_1}^{f,c}, l_{x_2, y_2}^{f,c}, \dots, l_{x_s, y_s}^{f,c}\}$, $L_f = L_f^e \cup L_f^w \cup L_f^n \cup L_f^s$ and $L = \bigcup_{f=1}^{k^2} L_f$. Subdivide the edges between $\{r_{x,y}^f\}$ and its children $2A_{n'} - 1$ times and subdivide the edge between each leaf in $L_{x,y}^f$ and its parent $A_{n'} - 1$ times. Let $T_{x,y}^f$ be the tree rooted in $r_{x,y}^f$ with leaves $L_{x,y}^f$, which, with the subdivisions of the edges, is not a complete binary tree anymore;
3. For each set $L_{x,y}^f$, link each leaf in $L_{x,y}^f$ to u and subdivide that edge $A_{n'} - 1$ times. Let $L_r = (\bigcup_{j=1}^k L_f^n \cup L_f^s) \cup L_{f(i,k)}^e \cup (\bigcup_{i=1}^k L_f^w \cup L_f^e)$. Link each vertex in $L_{x,y}^f \cap L_r$ to z and, then, subdivide each of these edges $A_{n'} - 1$ times. Note that the vertices resulting from the subdivision of the edge between a node in $L_{x,y}^f$ and u and between a node in $L_{x,y}^f$ and z are in $V(G_{i,j})$ but not in $V(T_{x,y}^f)$ while u and z are in neither sets.

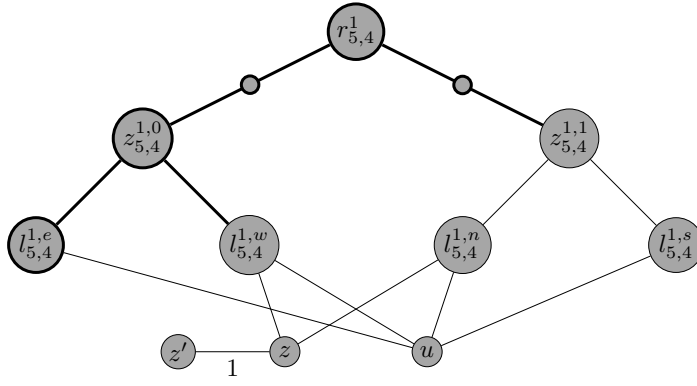


Figure 6: Vertices u , z and z' and tree $T_{5,4}^1$. The subtree $T_{5,4}^{1,e}$ is highlighted in bold. Note that $l_{5,4}^{1,w}$ and $z_{5,4}^{1,1}$ are not in $V(T_{5,4}^{1,e})$.

Let $T_f = V(T_{x_1, y_1}^f) \cup V(T_{x_2, y_2}^f) \cup \dots \cup V(T_{x_s, y_s}^f)$ and $T = \bigcup_{i=1, j=1}^k T_f(i, j)$. Let $T_{x, y}^{f, c}$, for any $c \in \{n, s, e, w\}$, be the subtree of $T_{x, y}^f$ induced by the vertices $(I[r_{x, y}^f, l_{x, y}^{f, c}] \cup I[r_{x, y}^f, l_{x, y}^{f, c'}] \cup I[r_{x, y}^f, z_{x, y}^{f, t}]) \setminus \{l_{x, y}^{f, c'}, z_{x, y}^{f, t}\}$, where, if $c = e$ or $c = w$, then $t = 1$, or, otherwise, $t = 0$ and c' represents the opposite direction of c , that is, if $c = e$, $c' = w$ and vice-versa and, if $c = n$, $c' = s$ and vice-versa. In Figure 6, there is an example that shows the tree $T_{5,4}^1$ and the edges from u and z to the vertices in $L_{5,4}^1$ and also shows the subtree $T_{5,4}^{1,e}$ highlighted in bold.

For each pair of neighbor pairs (i, j) and (i', j') , where $i \leq i'$ and $j \leq j'$, construct the *edge gadget* $E_{f, f'}$, where $c = e$ and $c' = w$, if $i = i'$, and $c = s$ and $c' = n$, if $j = j'$, let $f = f(i, j)$ and $f' = f(i', j')$, in the following manner:

1. Add the vertices $a_f^{f'}$ and $b_f^{f'}$, link them both to u and subdivide both edges $A_{n'} - 1$ times;
2. Link each vertex in L_f^c and in $L_{f'}^{c'}$ to both $a_f^{f'}$ and $b_f^{f'}$;
3. For each $l_{x, y}^{f, c} \in L_f^c$, subdivide the edge that links it to $a_f^{f'}$ either $A_{\pi(y)} - 1$ times, if $c = e$, or $A_{\pi(x)} - 1$ times, if $c = s$, and the edge that links it to $b_f^{f'}$ either $B_{\pi(y)} - 1$ times, if $c = e$, or $B_{\pi(x)} - 1$ times, if $c = s$;
4. For each $l_{x', y'}^{f', c'} \in L_{f'}^{c'}$, subdivide the edge that links it to $a_f^{f'}$ either $B_{\pi(y')} - 1$ times, if $c' = w$, or $B_{\pi(x')} - 1$ times, if $c' = n$, and the edge that links it to $b_f^{f'}$ either $A_{\pi(y')} - 1$ times, if $c' = w$, or $A_{\pi(x')} - 1$ times, if $c' = n$.

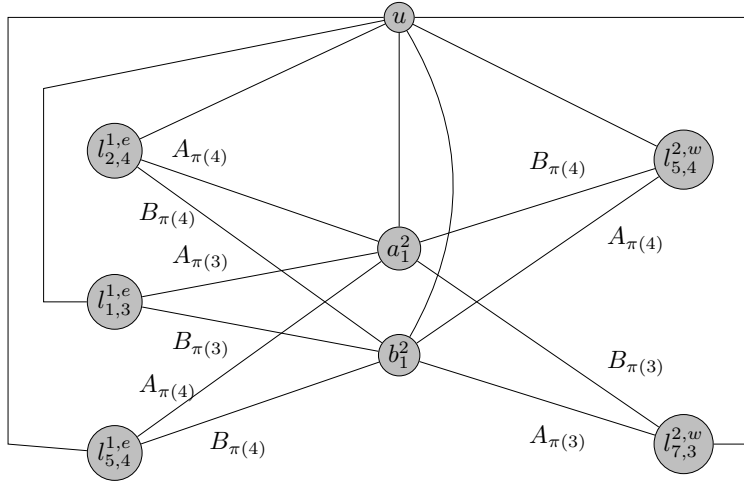


Figure 7: Edge gadget $E_{1,2}$, where $S_{1,1} = \{(2, 4), (1, 3), (5, 4)\}$ and $S_{1,2} = \{(5, 4), (7, 3)\}$.

In Figure 7, there is an example of the edge gadget $E_{1,2}$, where $S_{1,1} = \{(2, 4), (1, 3), (5, 4)\}$ and $S_{1,2} = \{(5, 4), (7, 3)\}$. In Figure 8, there is a more complete example of the sets gadgets $G_{1,1}$ and $G_{1,2}$ and the edge gadget $E_{1,2}$ for $S_{1,1} = \{(1, 3), (5, 4)\}$ and $S_{1,2} = \{(5, 4), (7, 3)\}$.

Note that the vertices resulting from the subdivision of the edge from a vertex $l \in L_f \cap L_{f'}$ and the vertices $a_f^{f'}$ and $b_f^{f'}$ are in $V(E_{f, f'})$, but l is not. Also, the vertices resulting from the subdivision of the edge from the vertex u and the vertices $a_f^{f'}$ and $b_f^{f'}$ are in $V(E_{f, f'})$, but u is not.

Let E_f be the set of vertices resulting from the subdivision of the edge from a vertex in $l \in L_f$ and the vertices $a_f^{f'}$ and $b_f^{f'}$, for all f' such that there are vertices $a_f^{f'}$ and $b_f^{f'}$. Note that l , $a_f^{f'}$ and $b_f^{f'}$ themselves are not in E_f .

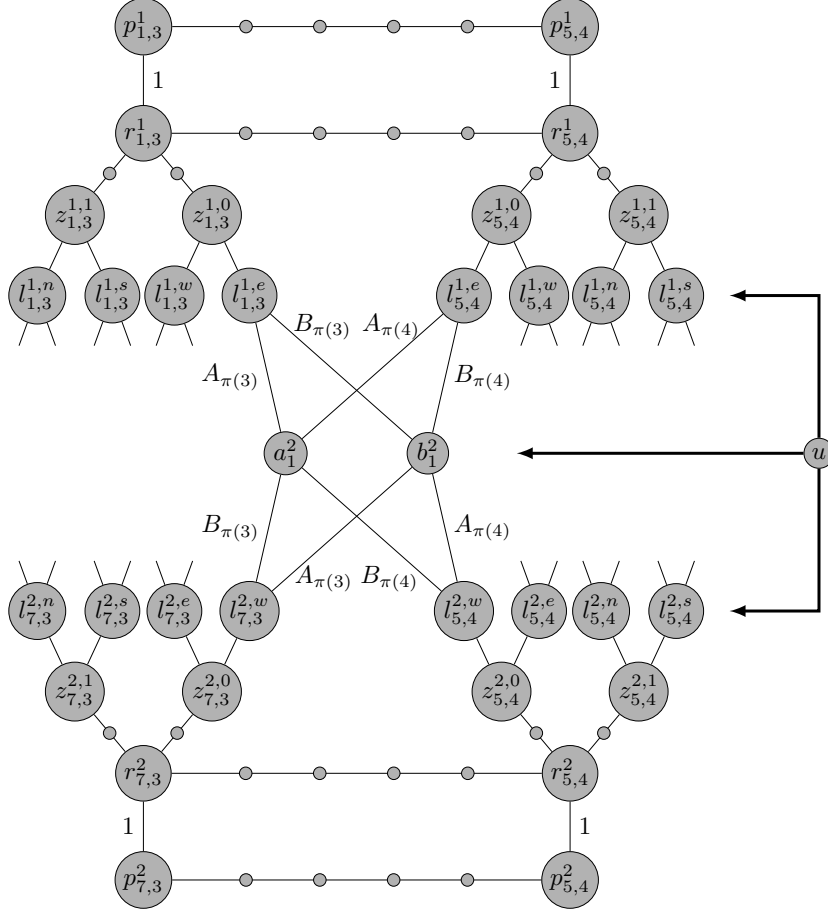


Figure 8: Set gadgets $G_{1,1}$ and $G_{1,2}$ and edge gadget $E_{1,2}$ without the vertices z and z' . For the sake of clarity, the links between u and the vertices in $L \cup \{a_1^2, b_1^2\}$ were omitted.

Let G' be the graph resulting from the application of the reduction to an instance (k, n, \mathcal{S}) of the Grid Tiling problem. The reduction runs in $\Theta(n'S) = O(S^2)$ time, where $S = \sum_{S' \in \mathcal{S}} |S'|$, which is polynomial time in the size of the input.

Now, let us prove that $tw(G') \leq 8k + 3$. First, let us define some sets. Let:

$$\begin{aligned} C_{1,1} &= \{a_{f(1,1)}^{f(1,2)}, b_{f(1,1)}^{f(1,2)}, a_{f(1,1)}^{f(2,1)}, b_{f(1,1)}^{f(2,1)}\} \\ C_{1,k} &= \{a_{f(1,k-1)}^{f(1,k)}, b_{f(1,k-1)}^{f(1,k)}, a_{f(1,k)}^{f(2,k)}, b_{f(1,k)}^{f(2,k)}\} \\ C_{k,1} &= \{a_{f(k-1,1)}^{f(k,1)}, b_{f(k-1,1)}^{f(k,1)}, a_{f(k,1)}^{f(k,2)}, b_{f(k,1)}^{f(k,2)}\} \\ C_{k,k} &= \{a_{f(k-1,k)}^{f(k,k)}, b_{f(k-1,k)}^{f(k,k)}, a_{f(k,k-1)}^{f(k,k)}, b_{f(k,k-1)}^{f(k,k)}\} \end{aligned}$$

Also, for every $1 < i < k$, let:

$$\begin{aligned} C_{1,i} &= \{a_{f(1,i-1)}^{f(1,i)}, b_{f(1,i-1)}^{f(1,i)}, a_{f(1,i)}^{f(1,i+1)}, b_{f(1,i)}^{f(1,i+1)}, a_{f(1,i)}^{f(2,i)}, b_{f(1,i)}^{f(2,i)}\} \\ C_{k,i} &= \{a_{f(k,i-1)}^{f(k,i)}, b_{f(k,i-1)}^{f(k,i)}, a_{f(k,i)}^{f(k,i+1)}, b_{f(k,i)}^{f(k,i+1)}, a_{f(k-1,i)}^{f(k,i)}, b_{f(k-1,i)}^{f(k,i)}\} \\ C_{i,1} &= \{a_{f(i-1,1)}^{f(i,1)}, b_{f(i-1,1)}^{f(i,1)}, a_{f(i,1)}^{f(i+1,1)}, b_{f(i,1)}^{f(i+1,1)}, a_{f(i,1)}^{f(i,2)}, b_{f(i,1)}^{f(i,2)}\} \\ C_{i,k} &= \{a_{f(i-1,k)}^{f(i,k)}, b_{f(i-1,k)}^{f(i,k)}, a_{f(i,k)}^{f(i+1,k)}, b_{f(i,k)}^{f(i+1,k)}, a_{f(i,k-1)}^{f(i,k)}, b_{f(i,k-1)}^{f(i,k)}\} \end{aligned}$$

And, for every $1 < i, j < k$, let:

$$C_{i,j} = \{a_{f(i,j)}^{f(i+1,j)}, b_{f(i,j)}^{f(i+1,j)}, a_{f(i-1,j)}^{f(i,j)}, b_{f(i-1,j)}^{f(i,j)}, a_{f(i,j-1)}^{f(i,j)}, b_{f(i,j-1)}^{f(i,j)}, a_{f(i,j)}^{f(i,j+1)}, b_{f(i,j)}^{f(i,j+1)}\}.$$

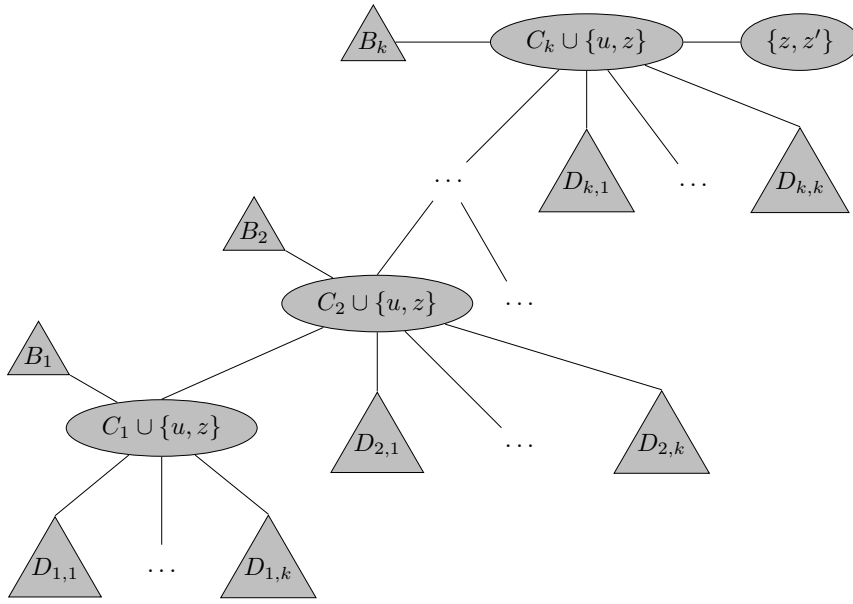


Figure 9: Tree decomposition of G' .

Finally, for $1 \leq i \leq k$, let $C_i = \bigcup_{j=1}^k C_{i,j}$. Note that $|C_i| < 8k$. Also, note that each set $C_i \cup \{u, z\}$ is a separator of G' , more specifically, $C_i \cup \{u, z\}$ separates each set gadget $G_{i,j}$, for $1 \leq j \leq k$, from the rest of the graph. Consider the tree decomposition D of G' in Figure 9, where, for all $1 \leq i, j \leq k$, $D_{i,j}$ is a tree decomposition of the graph induced by the vertices $V(G_{i,j}) \cup E_{f(i,j)}$ whose all bags are supersets of $C_i \cup \{u, z\}$ and, for all $1 \leq i \leq k$, B_i is a tree decomposition of the subgraph induced by the vertices $C_i \cup \{u\}$ union all the paths of length A_n having u and some vertex in C_i as extremities. Thus, since the graph induced by the vertices $V(G_{i,j}) \cup E_{f(i,j)}$ has width two, then the decomposition $D_{i,j}$ has width, at most, $|C_i \cup \{u, z\}| + 2 < 8k + 4$. Therefore, since each decomposition B_i has width one, we have that $\text{width}(D) \leq 8k + 3$ and, hence, $\text{tw}(G') \leq 8k + 3$.

Now, let us prove that (k, n, \mathcal{S}) is an yes instance of the Grid Tiling problem if and only if $(G', k^2 + 1)$ is an yes instance of the Geodesic Hull Number problem.

First, suppose that (k, n, \mathcal{S}) is an yes instance of the Grid Tiling problem. Then, let $H = \{s_{i,j} = (x_j, y_i) : 1 \leq i, j \leq k\}$ be a solution for this instance of the problem and let $S = \{p_{x_j, y_i}^{f(i,j)} : 1 \leq i, j \leq k\} \cup \{z'\}$. Note that $|S| = k^2 + 1$. First, let us analyze the shortest paths between two vertices in S .

For each pair of neighbor pairs (i, j) and (i', j') , where $i \leq i'$ and $j \leq j'$, let $x = x_j, y = y_i, x' = x_{j'}, y' = y_{i'}$, let $c = e$ and $c' = w$, if $i = i', c = s$ and $c' = n$, if $j = j', f = f(i, j)$ and $f' = f(i', j')$. Since H is a solution and, for any $1 \leq i \leq n', 2A_{n'} > A_i + B_i$, we have two shortest paths between $p_{x,y}^f$ and $p_{x',y'}^{f'}$, which are $p_{x,y}^f$ to $r_{x,y}^f$, then the shortest to $l_{x,y}^{f,c}$, then the shortest path to either $a_f^{f'}$ or $b_f^{f'}$, then the shortest path to $l_{x',y'}^{f',c'}$, then the shortest path to $r_{x',y'}^{f'}$ and, finally, to $p_{x',y'}^{f'}$.

For each pair of non-neighbor pairs (i, j) and (i', j') , let $x = x_j, y = y_i, x' = x_{j'}, y' = y_{i'}, f = f(i, j)$ and $f' = f(i', j')$. We have 16 shortest paths between $p_{x,y}^f$ and $p_{x',y'}^{f'}$, which are $p_{x,y}^f$ to $r_{x,y}^f$, then the shortest path to $l_{x,y}^{f,n}$ or $l_{x,y}^{f,s}$ or $l_{x,y}^{f,e}$ or $l_{x,y}^{f,w}$, then the shortest path to u , then the shortest path to $l_{x',y'}^{f',n}$ or $l_{x',y'}^{f',s}$ or $l_{x',y'}^{f',e}$ or $l_{x',y'}^{f',w}$, then the shortest path to $r_{x',y'}^{f'}$ and, finally, to $p_{x',y'}^{f'}$.

For each boundary pair (i, j) , let $x = x_j, y = y_i$, let $c = n$ if $i = 1$, let $c = s$ if $i = k$, let $c = w$ if $j = 1$, let $c = e$ if $j = k$, and, finally, let $f = f(i, j)$. There is one shortest path between z' and $p_{x,y}^f$ which is $p_{x,y}^f$ to $r_{x,y}^f$, then the smallest path to $l_{x,y}^{f,c}$, then to z and, finally, to z' .

For each corner pair (i, j) , let $x = x_j, y = y_i$, let $c = n$ and $c' = w$ if $i = 1$ and $j = 1$, let $c = n$ and $c' = e$ if $i = 1$ and $j = k$, let $c = s$ and $c' = w$ if $i = k$ and $j = 1$, let $c = s$ and $c' = e$ if $i = k$ and $j = k$, and, finally, let $f = f(i, j)$. There are two shortest path between z' and $p_{x,y}^f$ which are $p_{x,y}^f$ to $r_{x,y}^f$, then the shortest path to either $\{l_{x,y}^{f,c}$ or $l_{x,y}^{f,c'}\}$, then the shortest path to z and, finally, to z' .

Finally, for each pair (i, j) that is neither a corner nor a boundary pair, let $x = x_j, y = y_i$ and $f = f(i, j)$. There are $|L_{x,y}^f| \cdot |L_r| = 16k$ shortest paths between z' and $p_{x,y}^f$, which are z' to z , then to any vertex in L_r , then the shortest path to u , then the shortest path to any vertex in $L_{x,y}^f$, then the shortest path to $r_{x,y}^f$ and, finally, to $p_{x,y}^f$.

Hence, we have that all of those paths are in $I[S]$. Since $A_i + B_i = 12n'$, for any $1 \leq i \leq n'$ and every $a_f^{f'}$

and $b_f^{f'}$ are in $I[S]$, then $L \subseteq I^2[S]$ and all vertices in every path of length $A_{n'}$ between u and every vertex $a_f^{f'}$ and $b_f^{f'}$ are in $I^3[S]$. Thus, all vertices in every path between a vertex in L and u of length $A_{n'}$ are in $I^3[S]$. Also, since $L \subseteq I^2[S]$, we have that $T \subseteq I^4[S]$. Since there is one vertex of $P_{f(i,j)}$ in S , for each $1 \leq i, j \leq k$, then $P_{f(i,j)} \subseteq I^{|S_{i,j}|+4}[S]$. Therefore, we have that $V(G') = I_h[S]$ and, hence, $\text{hn}_g(G') \leq k^2 + 1$.

Now, suppose that (k, n, \mathcal{S}) is a no instance of the Grid Tiling problem. Note that each path $P_{f(i,j)}$ is a co-convex set, which implies that, for each path $P_{f(i,j)}$, no vertex in $P_{f(i,j)}$ is in a minimum path between two vertices outside of $P_{f(i,j)}$. Also, note that z' has degree 1. Thus, we have that every hull set of G' must contain z' and, at least, one vertex of each path $P_{f(i,j)}$. Since there are k^2 such paths, then every hull set has size at least $k^2 + 1$. Hence, to prove that $\text{hn}_g(G') \geq k^2 + 2$, is enough to prove that the convex hull of any set that only contain z' and exactly one vertex of each path $P_{f(i,j)}$ is not $V(G')$.

Let S be an arbitrary set that only contains z' and exactly one vertex of each path $P_{f(i,j)}$. Since (k, n, \mathcal{S}) is a no instance, then there will be, at least, one pair of neighbor coordinates (i, j) and (i', j') , with $i \leq i'$ and $j \leq j'$ such that either $a_f^{f'} \notin I[S]$ and $b_f^{f'} \in I[S]$ or $a_f^{f'} \in I[S]$ and $b_f^{f'} \notin I[S]$, where $f = f(i, j)$ and $f' = f(i', j')$. Let $c = e$ and $c' = w$, if $i = i'$, or $c = s$ and $c' = n$, if $j = j'$. Also, let $p_{x,y}^f, p_{x',y'}^{f'} \in S$ and let v be the only vertex in the set $\{a_f^{f'}, b_f^{f'}\} \cap I[S]$.

We claim that the set

$$\begin{aligned} R = & ((P_f \cup R_f \cup P_{f'} \cup R_{f'}) \setminus \{p_{x,y}^f, p_{x',y'}^{f'}, r_{x,y}^f, r_{x',y'}^{f'}\}) \cup \\ & \left(\bigcup_{\substack{(q,o) \neq (x,y) \\ (q,o) \in (S_{i,j})}} V(T_{q,o}^{f,c}) \bigcup_{\substack{(q,o) \neq (x',y') \\ (q,o) \in (S_{i',j'})}} V(T_{q,o}^{f',c'}) \right) \cup \\ & (V(E_{f,f'}) \setminus I[\{u, v, l_{x,y}^{f,c}, l_{x',y'}^{f',c'}\}]) \end{aligned}$$

is a co-convex set, which implies, since $S \cap R = \emptyset$, that $I_h[S] \neq V(G')$ and, consequently, that $\text{hn}_g(G') \geq k^2 + 2$. To prove that R is a co-convex set, it is enough to prove that no shortest path between two vertices in $N(R) \setminus R$ contains a vertex in R , where $N(R)$ is the union of the neighborhoods of all the vertices in R . Note that

$$\begin{aligned} N(R) \setminus R = & \{p_{x,y}^f, p_{x',y'}^{f'}, r_{x,y}^f, r_{x',y'}^{f'}, l_{x,y}^{f,c}, l_{x',y'}^{f',c'}, v, u\} \cup \\ & \bigcup_{\substack{(q,o) \neq (x,y) \\ (q,o) \in (S_{i,j})}} \{l_{q,o}^{f,c}, z_{q,o}^{f,t}\} \bigcup_{\substack{(q,o) \neq (x',y') \\ (q,o) \in (S_{i',j'})}} \{l_{q,o}^{f',c'}, z_{q,o}^{f',t}\} \end{aligned}$$

where $t = 0$, if $c = e$ or $c = w$, or $t = 1$, otherwise. If $d_1, d_2 \in \{l_{x,y}^{f,c}, l_{x',y'}^{f',c'}, v, u\}$, by the definition of R , $I[d_1, d_2] \cap R = \emptyset$. Hence, we have that any shortest path between two vertices $d_1, d_2 \in \{p_{x,y}^f, p_{x',y'}^{f'}, r_{x,y}^f, r_{x',y'}^{f'}, l_{x,y}^{f,c}, l_{x',y'}^{f',c'}, v\}$ does not pass by any vertex in R since it only passes by vertices in $V(T_{x,y}^f) \cup V(T_{x',y'}^{f'}) \cup I[l_{x,y}^{f,c}, l_{x',y'}^{f',c'}]$. Additionally, it is easy to check that, for any other case of $d_1, d_2 \in N(R) \setminus R$, the shortest paths between d_1 and d_2 go through the vertex u and do not pass by any vertex in R , even if d_1 and d_2 are in the same tree $T_{q,o}^f$.

Therefore, we can conclude that R is a co-convex set and, as argued before, we have that $\text{hn}_g(G') \geq k^2 + 2$. \square

2.1. Exponential Time Hypothesis Consequences

In 2001, Impagliazzo et al. [27, 28] introduced the Exponential Time Hypothesis (ETH). The ETH states that there is a constant $\delta_3 > 0$ such that **3SAT** cannot be solved in time $O^*(2^{cN})$ for any constant $c < \delta_3$, where N is the number of variables of the 3SAT instance.

One consequence of the ETH is that neither the Multicolored Clique problem [32] nor the Grid Tiling problem [33] can be solved in time $O(f(k)|V(G)|^{o(k)})$, for any computable function f . The following three corollaries are a byproduct of the theorems 1, 2 and 3 respectively, where $n = |V(G)|$ e $w = tw(G)$.

Corollary 1. *The P_3 Hull Number problem cannot be solved in $O(f(k)n^{o(k)})$ time, even in K_3 -free graphs, for any computable function f , unless the ETH fails.*

Corollary 2. *The Geodesic Hull Number problem cannot be solved in $O(f(k)n^{o(k)})$ time, for any computable function f , unless the ETH fails.*

Corollary 3. *The Geodesic Hull Number problem cannot be solved in $O(f(w,k)n^{o(w)+o(\sqrt{k})})$ time, for any computable function f , unless the ETH fails.*

3. XP Algorithm for the Geodesic Hull Number Problem

In this section we are going to present an algorithm that, given a graph G and a tree decomposition of G with width tw , uses dynamic programming to compute the hull number of G in $O(f(tw)n^{O(tw)})$, where $n = |V(G)|$. Since we can compute a tree decomposition of a graph in **FPT** time, this implies that the Geodesic Hull Number problem is in **XP** when parameterized by the treewidth of the input graph. Henceforth, let G be the connected graph instance of the Geodesic Hull Number problem that we will work with.

First, let us first present some definitions. Let $n = |V(G)|$ and tw be the treewidth of G , where $V(G)$ denote the set of vertices of G . Let $(\mathcal{B} = \{B_j : j \in T\}, \mathcal{T} = (T, F))$ be a nice tree decomposition of G , where \mathcal{T} is rooted in the node $r \in T$. Also, denote by G_j , for any $j \in T$, the subgraph of G induced by B_j union the bags of all of the descendants of j in \mathcal{T} , including j itself. Note that $G_r = G$.

For a path P , denote by $Ext(P)$ the set consisting the extremities of P , $Int(P)$ the set consisting of the internal vertices of P and $Int_j(P) = Int(P) \cap V(G_j)$. Note that $|Int(P)|$ can be zero. Let, for any function f , $Dom(f)$ and $Img(f)$ be the domain and image of f respectively.

For any $j \in T$, let H_j be the set of all functions $h : B_j \rightarrow \{0, 1, \dots, tw + 1\}$. We say that two functions $h \in H_j$ and $h' \in H_{j'}$ have the same relative order, where $j, j' \in T$ and $B_j \subseteq B_{j'}$, if and only if $\{u \in B_j \mid h(u) = 0\} = \{u \in B_j \mid h'(u) = 0\}$ and, for all $v, v' \in B_j$, $h(v) < h(v')$ if and only if $h'(v) < h'(v')$. Also, we say that a function $h \in H_j$ is *nice* if and only if there is an integer k such that either $Img(h) = \{0, 1, 2, \dots, k - 1, k\}$ or $Img(h) = \{1, 2, \dots, k - 1, k\}$.

For any $j \in T$ and $h \in H_j$, let $F_{j,h}$ be the set of all functions $f : Img(h) \setminus \{0\} \rightarrow \binom{V(G)}{2}$. Also, let $Q_j^{h,f}(x)$ be the set of all minimum paths P between the vertices in $f(x)$ such that $Int(P) \cap B_j = \{v \in B_j \mid h(v) = x\}$.

Let $U_j(S) = (V(G) \setminus V(G_j)) \cup S$. Let $R_j(S)$, for any $S \subseteq V(G_j)$, be the set of all paths P such that P is a minimum path between any two vertices in S , $Int(P) \cap B_j = \emptyset$ and $Int_j(P) \setminus S \neq \emptyset$.

For any $j \in T$, $h \in H_j$ and $f \in F_{j,h}$, let $K_j^h(S) = \{v \in B_j \mid \forall w \in B_j \setminus S, h(v) \leq h(w)\}$. Also, for $B_j \setminus S \neq \emptyset$, $A_j^h(S) = \min\{h(v) \mid v \in B_j \setminus S\}$. Furthermore, let \mathcal{P} be the set of all paths of G .

Definition 1. For any $j \in T$, $h \in H_j$ and $f \in F_{j,h}$, let $C_j^{h,f}(S)$, for all $S \subseteq V(G_j)$, be the set of paths P that satisfies one of the three following conditions:

1. $P \in R_j(S)$;
2. $Ext(P) \subseteq U_j(S)$ and $P \in Q_j^{h,f}(A_j^h(S))$;
3. $Int(P) = \emptyset$.

Definition 2. For any $j \in T$, $h \in H_j$ and $f \in F_{j,h}$, let $C_j^{h,f}$ be the set of all functions $c : 2^{V(G_j)} \rightarrow \mathcal{P}$ such that, for all $S \subseteq V(G_j)$, $c(S) \in C_j^{h,f}(S)$.

Note that for any $j \in T$, $h \in H_j$ and $f \in F_{j,h}$, $C_j^{h,f}$ is never empty because, for any $S \in \mathcal{Z}_j^h$, any path P with only one vertex is such that $Int(P) = \emptyset$, which implies that $P \in C_j^{h,f}(S)$. In fact, we have that $|C_j^{h,f}| = \prod_{S \subseteq V(G_j)} |C_j^{h,f}(S)|$.

Therefore, for any $j \in T$, $h \in H_j$, $f \in F_{j,h}$, if we have sets $S_1, S_2, \dots, S_x \subseteq V(G)$ and paths $P_1, P_2, \dots, P_x \in \mathcal{P}$, such that, for every $1 \leq i \leq x$, $P_i \in C_j^{h,f}(S_i)$, then there is a function $c \in C_j^{h,f}$ such that, for $1 \leq i \leq x$, $c(S_i) = P_i$.

Thus, in order to prove that there is a function $c \in C_j^{h,f}$ such that, for a given sequence of sets $S_1, S_2, \dots \subseteq V(G)$ and paths $P_1, P_2, \dots \in \mathcal{P}$, $c(S_i) = P_i$, it is enough to show, for each i , that $P_i \in C_j^{h,f}(S_i)$.

Definition 3. For any $j \in T$, $h \in H_j$, $c : 2^{V(G_j)} \rightarrow \mathcal{P}$ and set $S \subseteq V(G_j)$, let $I_j^c(S) = S \cup Int_j(c(S))$.

Let $I_j^c(S, 0) = S$, $I_j^c(S, i) = I_j^c(I_j^c(S, i - 1))$, for any $i > 0$, and let $Y_j^c(S) = I_j^c(S, i)$, where i is the minimum positive integer such that $I_j^c(S, i + 1) = I_j^c(S, i)$. By a simple argument of induction on i , we have that, for any $i \geq 0$, $I_j^c(S, i) = I_j^c(I_j^c(S, i_1), i_2)$, where $i = i_1 + i_2$.

let $\mathcal{Z}_k^{j,h}$ be the family of all subsets Z of $V(G_j)$ such that $Z \cap B_j = \{v \in B_j \mid h(v) \leq k\}$ and let $\mathcal{Z}^{j,h} = \bigcup_{k=0}^{tw+1} \mathcal{Z}_k^{j,h}$.

Definition 4. Let $j \in T$, $h \in H_j$ and $f \in F_{j,h}$. A set S is a (h, f) -hull set of the subgraph G_j if and only if $S \in \mathcal{Z}_0^{j,h}$ and there is a function $c \in C_j^{h,f}$ such that $Y_j^c(S) = V(G_j)$.

Let $S \in \mathcal{Z}_0^{j,h}$ and $c \in C_j^{h,f}$. Then, for each $i > 0$, either $I_j^c(S, i) \cap K_j^h(I_j^c(S, i - 1)) = \emptyset$ or $K_j^h(I_j^c(S, i - 1)) \subseteq I_j^c(S, i)$, which implies that $I_j^c(S, i) \in \mathcal{Z}^{j,h}$, for any $i \geq 0$.

Also, we have that a vertex $w \in B_j$ is in $I_j^c(S, i)$ if and only if $I_j^c(S, i) \in \mathcal{Z}_{h(w)}^{j,h}$. In fact, for any two vertices $v, w \in B_j$ such that $v, w \in Y_j^c(S)$, we have that $i_v < i_w$ if and only if $h(v) < h(w)$, where i_v and i_w are the least value such that $v \in I_j^c(S, i_v)$ and $w \in I_j^c(S, i_w)$.

Henceforth, denote by $M_j(h, f)$, for each $j \in T$, $h \in H_j$ and $f \in F_{j,h}$, the cardinality of the minimum (h, f) -hull set of G_j or ∞ , if G_j has no (h, f) -hull set.

The next lemma relates $\text{hn}_g(G)$ to the values of the table M_r .

Lemma 2. *S is a hull set of G if and only if there are functions $h \in H_r$ and $f \in F_{r,h}$ such that S is a (h, f) -hull set of G_r .*

Proof. First, suppose that S is a (h, f) -hull set of G_r . Let $c \in C_r^{h,f}$ be a function such that $Y_r^c(S) = V(G_r) = V(G)$. First, let us prove that $\text{Int}(c(I_r^c(S, y))) \setminus I_r^c(S, y) \subseteq I[I_r^c(S, y)] \setminus I_r^c(S, y)$, for all $y \geq 0$.

Let $I_y = I_r^c(S, y)$ and $M = c(I_y)$, for an arbitrary $y \geq 0$. If $M \in R_r(I_y)$, since $\text{Ext}(M) \subseteq I_y$ and M is a minimum path, we have that $\text{Int}(M) \setminus I_y \subseteq I[I_y] \setminus I_y$. If $\text{Ext}(M) \subseteq U_r(I_y)$ and $M \in Q_r^{h,f}(A_r^h(I_y))$, then, since $\text{Ext}(M) \subseteq U_r(I_y) = I_y$, we have that $\text{Int}(M) \setminus I_y \subseteq I[I_y] \setminus I_y$. Finally, if $\text{Int}(M) = \emptyset$, then $\text{Int}(M) \setminus I_y \subseteq I[I_y] \setminus I_y$.

So, we have that $\text{Int}(c(I_r^c(S, y))) \setminus I_r^c(S, y) \subseteq I[I_r^c(S, y)] \setminus I_r^c(S, y)$, for all $y \geq 0$ such that $I_r^c(S, y) \neq V(G)$. Hence, for each $y \geq 0$, we have that $I_r^c(S, y+1) = (\text{Int}(c(I_r^c(S, y))) \setminus I_r^c(S, y)) \cup I_r^c(S, y) \subseteq (I[I_r^c(S, y)] \setminus I_r^c(S, y)) \cup I_r^c(S, y) = I[I_r^c(S, y)]$. Now, let us prove by induction on y , that $I_r^c(S, y) \subseteq I^y[S]$, for any $y \geq 0$. For $y = 0$, we have that $I_r^c(S, 0) = S = I^0[S]$. We have that $I_r^c(S, y+1) \subseteq I[I_r^c(S, y)]$. By our inductive hypothesis, we have that $I_r^c(S, y) \subseteq I^y[S]$. Thus, we have that $I_r^c(S, y+1) \subseteq I[I_r^c(S, y)] \subseteq I[I^y[S]] = I^{y+1}[S]$.

Thus, we have that $I_r^c(S, y) \subseteq I^y[S]$, for any $y \geq 0$, which implies that $V(G) = Y_r^c(S) \subseteq I_h[S]$. Hence, S is a hull set of G .

Now, suppose that S is a hull set of G . Let us define function $h \in H_r$, $f \in F_{r,h}$ such that S is a (h, f) -hull set of G_r . For each vertex $v \in B_r$, let $i : B_j \rightarrow \mathbb{N}$ be the function such that $i(v)$ is the least integer such that $v \in I^{i(v)}[S]$. For each $y \in \text{Img}(i) \setminus \{0\}$, let $B_r(y) = \{w \in B_r \mid i(w) = y\}$. For each $y \in \text{Img}(i) \setminus \{0\}$, let $O_y = M_1^y, M_2^y, \dots, M_{l(y)}^y$ be any ordering of minimum paths such that $(\bigcup_{1 \leq k \leq l(y)} \text{Int}(M_k^y)) \setminus I^{y-1}[S] = I^y[S] \setminus I^{y-1}[S]$ and, for each $1 \leq i \leq l(y)$, $\text{Ext}(M_i^y) \subseteq I^{y-1}[S] \cup \bigcup_{1 \leq k < i} \text{Int}(M_k^y)$, $\text{Int}(M_i^y) \cap B_r \subseteq B_r(y) \setminus \bigcup_{1 \leq k < i} \text{Int}(M_k^y)$ and $\text{Int}(M_i^y) \setminus (I^{y-1}[S] \cup \bigcup_{1 \leq k < i} \text{Int}(M_k^y)) \neq \emptyset$. For instance, for the graph of the Figure 10 and set $S = \{v_1, v_5\}$, we can have that $O_1 = M_1^1, M_2^1, M_3^1$ and $O_2 = M_1^2, M_2^2$, where $M_1^1 = v_1, v_2, v_3, v_4, v_5$; $M_2^1 = v_1, v_6, v_7, v_9, v_5$; $M_3^1 = v_1, v_6, v_8, v_9$; $M_1^2 = v_3, v_{10}, v_8$ and $M_2^2 = v_3, v_{11}, v_8$.

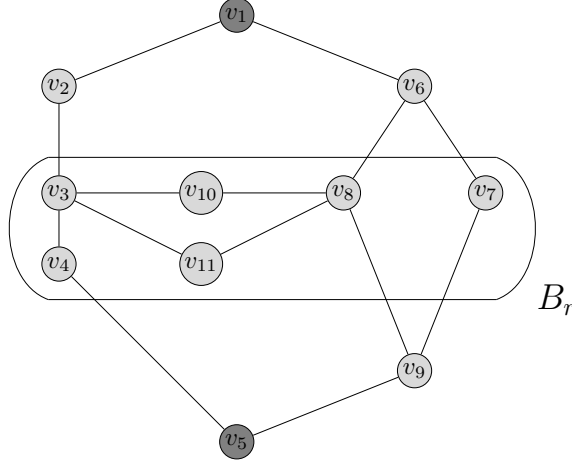


Figure 10: Example of paths in the orderings O_1 and O_2 for $S = \{v_1, v_5\}$ are $M_1^1 = v_1, v_2, v_3, v_4, v_5$; $M_2^1 = v_1, v_6, v_7, v_9, v_5$; $M_3^1 = v_1, v_6, v_8, v_9$; $M_1^2 = v_3, v_{10}, v_8$ and $M_2^2 = v_3, v_{11}, v_8$.

Thus, Let h be a nice function in H_j such that, for each vertex $v \in B_j$, $i(v) = 0$ if and only if $h(v) = 0$, and, for each two vertices $v, w \in B_j$, $i(v) < i(w)$ if and only if $h(v) < h(w)$. Furthermore, for each $y \in \text{Img}(i) \setminus \{0\}$ and any two vertices $v \in \text{Int}(M_j^y) \cap B_r$ and $w \in \text{Int}(M_k^y) \cap B_r$, we have that $h(v) < h(w)$ if and only if $j < k$. Furthermore, for each $y \in \text{Img}(i) \setminus \{0\}$ and $v \in \text{Int}(M_j^y)$, let $f(h(v)) = \text{Ext}(M_j^y)$. Since any two vertices $v, w \in \text{Int}(M_j^y)$ are such that $h(v) = h(w)$, then f is well defined. For instance, for the graph in Figure 10, set $S = \{v_1, v_5\}$ and paths $M_1^1 = v_1, v_2, v_3, v_4, v_5$; $M_2^1 = v_1, v_6, v_7, v_9, v_5$; $M_3^1 = v_1, v_6, v_8, v_9$; $M_1^2 = v_3, v_{10}, v_8$ and $M_2^2 = v_3, v_{11}, v_8$, we have that $h(v_3) = h(v_4) = 1$, $h(v_7) = 2$, $h(v_8) = 3$, $h(v_{10}) = 4$ and $h(v_{11}) = 5$ and $f(1) = f(2) = \{v_1, v_5\}$, $f(3) = \{v_1, v_9\}$, $f(4) = f(5) = \{v_3, v_8\}$.

First, let us define a function $c \in C_r^{h,f}$ such that, for any $y \geq 0$ such that $I^y[S] \neq V(G)$, we have that $I_r^c(I^y[S], l(y+1)) = I^{y+1}[S]$.

For every $0 \leq i < l(y+1)$, let $c(I_r^c(I^y[S], i)) = M_{i+1}^{y+1}$. For each $0 \leq i < l(y+1)$, if $\text{Int}(M_{i+1}^{y+1}) \cap B_r = \emptyset$, since $\text{Ext}(M_{i+1}^{y+1}) \subseteq I^y[S] \cup \bigcup_{1 \leq k < i} \text{Int}(M_k^{y+1}) \subseteq I_r^c(I^y[S], i)$ and $\text{Int}(M_{i+1}^{y+1}) \setminus (I^y[S] \cup \bigcup_{1 \leq k < i} \text{Int}(M_k^{y+1})) = \text{Int}(M_{i+1}^{y+1}) \setminus I_r^c(I^y[S], i) \neq \emptyset$, we have that $M_{i+1}^{y+1} \in R_r(I_r^c(I^y[S], i))$, which implies that $M_{i+1}^{y+1} \in C_r^{h,f}(I_r^c(I^y[S], i))$.

If $\text{Int}(M_{i+1}^{y+1}) \cap B_r \neq \emptyset$, we have that $\text{Int}(M_{i+1}^{y+1}) \cap B_r = K_r^h(I_r^c(I^y[S], i))$. This is because, every vertex $v \in \text{Int}(M_{i+1}^{y+1}) \cap B_r$ is such that $v \notin I_r^c(I^y[S], i)$, for any $w \in B_r \setminus I_r^c(I^y[S], i)$, we have, by definition of h , that $h(v) \leq h(w)$. This implies that $M_{i+1}^{y+1} \in Q_r^{h,f}(A_r^h(I_r^c(I^y[S], i)))$. Furthermore, we have that $\text{Ext}(M_{i+1}^{y+1}) \subseteq U_r(I_r^c(I^y[S], i)) = I_r^c(I^y[S], i)$. Therefore, we have that $M_{i+1}^{y+1} \in C_r^{h,f}(I_r^c(I^y[S], i))$.

Now that we prove that, for any $y \geq 0$ such that $I^y[S] \neq V(G)$, we have that $I_r^c(I^y[S], l(y+1)) = I^{y+1}[S]$, since $I^0[S] = I_r^c(S, 0) = S$, by a simple argument of induction on y , we have that there is a value z such that $I_r^c(S, z) = I^y[S]$, for any $y \geq 0$, which implies that $Y_r^c(S) = I_h[S] = V(G)$.

Furthermore, since $S \cap B_r = \{v \in B_r \mid h(v) = 0\}$, we have that $S \in \mathcal{Z}_0^{r,h}$ and, hence, S is a (h, f) -hull set of G_r . \square

One consequence of Lemma 2 is that $\text{hn}_g(G) = \min_{h,f} M_r(h, f)$, where h and f iterates over all functions in H_r and $F_{r,h}$ respectively.

We say that (h, f) is a invalid index of the table M_j if and only if there is an integer $x \in \text{Img}(h) \setminus \{0\}$ such that $Q_j^{h,f}(x) = \emptyset$ or there is a vertex $v \in B_j$ and an integer $x \in \text{Img}(h) \setminus \{0\}$ such that $v \in f(x)$ and $h(v) \geq x$.

If $Q_j^{h,f}(x) = \emptyset$, for some $x \in \text{Img}(h) \setminus \{0\}$, then, for any $S \in \mathcal{Z}_0^{j,h}$, $Y_j^c(S) \cap \{v \in B_j \mid h(v) = x\} = \emptyset$, which implies that there can be no (h, f) -hull set of G_j . Also, by Lemma 3, if there is a vertex $v \in B_j$ and an integer $x \in \text{Img}(h) \setminus \{0\}$ such that $v \in f(x)$ and $h(v) \geq x$, then there is no (h, f) -hull set of G_j . Hence, if (h, f) is an invalid index, then $M_j(h, f) = \infty$.

Lemma 3. *Let $j \in T$, $h \in H_j$ and $f \in F_{j,h}$. If there is a vertex $v \in B_j$ and an integer $x \in \text{Img}(h) \setminus \{0\}$ such that $v \in f(x)$ and $h(v) \geq x$ then there is no (h, f) -hull set of G_j .*

Proof. Suppose that S be a (h, f) -hull set of G_j and there is a vertex $v \in B_j$ and an integer $x \in \text{Img}(h) \setminus \{0\}$ such that $v \in f(x)$ and $h(v) \geq x$. Let w be a vertex in B_j such that $h(w) = x$. Let $c \in C_j^{h,f}$ be a function such that $Y_j^c(S) = V(G_j)$ and let i_z be the least integer such that $z \in I_j^c(S, i_z)$. Since $h(w) = x > 0$, we have that $i_w > 0$. Then, since $w \in I_j^c(S, i_w) \setminus I_j^c(S, i_w - 1)$, we have that $v \in f(x) = U_j(I_j^c(S, i_w - 1))$, which implies that $v \in I_j^c(S, i_w - 1)$, since $v \notin V(G) \setminus V(G_j)$. So, we have that $i_v < i_w$. But, then, we have that $i_v < i_w$ and $h(v) \geq x = h(w)$, which is a contradiction. Therefore, if there is a vertex $v \in B_j$ and an integer $x \in \text{Img}(h) \setminus \{0\}$ such that $v \in f(x)$ and $h(v) \geq x$, then there is no (h, f) -hull set of G_j . \square

Given any index (h, f) of a table M_j , we can check weather there is a vertex $v \in B_j$ and an integer $x \in \text{Img}(h) \setminus \{0\}$ such that $v \in f(x)$ and $h(v) \geq x$ in $O(tw)$, since $|f(x)| = 2$, for any x .

Now, let (h, f) be some index of a table M_j such that there is no vertex $v \in B_j$ and integer $x \in \text{Img}(h) \setminus \{0\}$ such that $v \in f(x)$ and $h(v) \geq x$.

Let $f(x) = \{w_0, w_{z+1}\}$, $W = \{w \in B_j \mid h(w) = x\} \cup f(x)$ and let $w_0, w_1, w_2, \dots, w_z, w_{z+1}$ be an ordering of the vertices in W such that, for each $1 \leq i \leq z$, the vertex v_i is the nearest from w_{i-1} among the vertices in $W \setminus \{w_0, w_1, w_2, \dots, w_{i-1}\}$. Since every subpath of a minimum path is also a minimum path on its own, we have that there is a minimum path M such that $W \setminus f(x) \subseteq \text{Int}(M)$ if and only if, for each $1 \leq i \leq z$, w_i is in a minimum path between w_{i-1} and w_{z+1} .

So, the following algorithm verifies weather $Q_j^{h,f}(x) = \emptyset$:

1. For each $1 \leq i \leq z$, find the vertex of W nearest from w_{i-1} , name it w_i and check weather w_i is in a minimum path between w_{i-1} and w_{z+1} . If yes, remove w_i from W . If no, then $Q_j^{h,f}(x) = \emptyset$.
2. If, for each $1 \leq i \leq z$, w_i is in a minimum path between w_{i-1} and w_{z+1} , then $Q_j^{h,f}(x) \neq \emptyset$.

Furthermore, given the table of distances between any two vertices of G , we can check if a vertex v is in a minimum path between any two vertices w and z in $O(1)$ time because a vertex v is in a minimum path between w and z if and only if $d(w, z) = d(w, v) + d(v, z)$, where $d(v_1, v_2)$ is the distance between the vertices v_1 and v_2 . Hence, we can check whether there is some value $x \in \text{Img}(h) \setminus \{0\}$ such that $Q_j^{h,f}(x) = \emptyset$ in $O(tw^2)$ time.

Therefore, we can verify whether an index (h, f) of a table is invalid in $O(tw^2 + tw) = O(tw^2)$ time.

Our algorithm will compute the table M_j of all nodes j in the tree \mathcal{T} in a bottom-up fashion. Each index of the table M_j will be computed by looking up the values of some indexes in the table of the child, or possible children, of j , except if j is a leaf node. When we compute the table M_r , as discussed before, we can compute $\text{hn}_g(G)$ by finding the minimum $M_r(h, f)$ for all indexes (h, f) of M_r .

Before we proceed to show how to compute M_j depending on the type of node j , let us prove the following two useful lemmas.

Lemma 4. *Let G be a graph and $S = \{s_1, s_2, \dots, s_x\}$ be subset of $V(G)$. Let P be a minimum path in G between s_1 and s_x such that $S \subseteq V(P)$ and every subpath R of P between the vertices s_i and s_{i+1} , for every $1 \leq i < x$, is such that $\text{Int}(R) \cap S = \emptyset$, that is, the vertices in the set S appear in the order $s_1, s_2, \dots, s_{x-1}, s_x$ in the path P . Then, any minimum path P' between s_1 and s_x , where $S \subseteq V(P')$, is such that the vertices in the set S also appear in the order $s_1, s_2, \dots, s_{x-1}, s_x$ in P' .*

Proof. Let P' be a minimum path between s_1 and s_x and suppose by contradiction that there is a subpath R' of P' such that $Ext(R') = \{s_i, s_{i+1}\}$ and some vertex s_j is in $Int(R') \cap S$. Thus, since P' is a minimum path, then the subpath of P' between s_i and s_j and between s_j and s_{i+1} are also minimum paths and are strictly shorter than the subpath of P' between s_i and s_{i+1} , which implies that they are strictly shorter than the subpath of P between s_i and s_{i+1} , since both subpaths are minimum between s_i and s_{i+1} .

If $j > i$, let M be the path between s_1 and s_x that is composed by the subpath of P between s_1 and s_i and between s_j and s_x and by the subpath of P' between s_i and s_j . Since the subpath of P' between s_i and s_j is strictly shorter than the subpath of P between s_i and s_{i+1} , then M is a path between s_1 and s_x shorter than P , which is a contradiction because P is a minimum path.

If $j < i$, let M be the path between s_1 and s_x that is composed by the subpath of P between s_1 and s_j and between s_{i+1} and s_x and by the subpath of P' between s_j and s_{i+1} . Since the subpath of P' between s_j and s_{i+1} is strictly shorter than the subpath of P between s_i and s_{i+1} , then M is a path between s_1 and s_x shorter than P , which is a contradiction because P is a minimum path.

Therefore, we have that any minimum path P' between s_1 and s_x , where $S \subseteq V(P')$, is such that the vertices in S appear in P' in the same order as they appear in P , that is, $s_1, s_2, \dots, s_{x-1}, s_x$. \square

Lemma 5. *Let $j \in T$, $h \in H_j$ and $f \in F_{j,h}$. If there is a function $c \in C_j^{h,f}$ such that $Y_j^c(S) = V(G_j)$, then, for all superset $S' \in \mathcal{Z}_j^h$ of S such that $K_j^h(S) = K_j^h(S')$, there is a function $c' \in C_j^{h,f}$ such that $Y_j^{c'}(S') = V(G_j)$.*

Proof. Let $S' \in \mathcal{Z}_j^h$ be a superset of S and, for any $y \geq 0$, $I_y = I_y^c(S, y)$. Let us define a function $c' \in C_j^{h,f}$ such that $Y_j^{c'}(S') = V(G_j)$. For any $y \geq 0$, let $I'_y = I_y^{c'}(S', y)$, $x(y')$ be the greatest value such that $I_{x(y')} \subseteq I'_y$ and $c'(I'_y) = c(I_{x(y)})$. This implies that $Y_j^{c'}(S') = V(G_j)$, since, for all $y \geq 0$ such that $I'_y \neq V(G_j)$, we have that $Int_j(c'(I'_y)) = Int_j(c(I_{x(y)})) \neq \emptyset$. Also, since $K_j^h(S) = K_j^h(S')$, by induction on y , for all $y \geq 0$, we have that $K_j^h(I_{x(y)}) = K_j^h(I'_y)$.

Now, let us prove that $c' \in C_j^{h,f}$, by proving that, for every $y \geq 0$, $c'(I'_y) \in C_j^{h,f}(I'_y)$. Since $S' \in \mathcal{Z}_j^h$ and $c \in C_j^{h,f}$, we have, by a simple argument of induction on y , that $I'_y \in \mathcal{Z}_j^h$, for any $y \geq 0$.

Let $y \geq 0$ be an arbitrary integer such that $I'_y \neq V(G_j)$ and let $M = c(I_{x(y)})$. First, suppose that $M \in R_j(I_{x(y)})$. Since $I_{x(y)} \subseteq I'_y \not\subseteq I_{x(y)+1}$, we have that $Int_j(M) \setminus I'_y \neq \emptyset$. Also, we have that $Ext(M) \subseteq I_{x(y)} \subseteq I'_y$ and $Int(M) \cap B_j = \emptyset$. Thus, we have that $M \in R_j(I'_y)$, which implies that $M \in C_j^{h,f}(I'_y)$.

Now, suppose that $M \in Q_j^{h,f}(A_j^h(I_{x(y)}))$ and $Ext(M) \subseteq U_j(I_{x(y)})$. Since $K_j^h(I_{x(y)}) = K_j^h(I'_y)$, we have that $A_j^h(I_{x(y)}) = A_j^h(I'_y)$, which implies that $M \in Q_j^{h,f}(A_j^h(I'_y))$. Also, since $I_{x(y)} \subseteq I'_y$, we have that $Ext(M) \subseteq U_j(I'_y)$. Thus, we have that $M \in C_j^{h,f}(I'_y)$.

Then, we conclude that $c' \in C_j^{h,f}$. \square

As a consequence of the Lemma 4, given a node $j \in T$, we have that any two minimum paths P and P' that passes by the same vertices in B_j , passes by them in the same order. This fact will be frequently used in the following lemmas when we want to construct a minimum path M from two minimum paths P and P' , where all three passes by the same vertices in B_j .

In the next four lemmas, in several occasions, we will define a function c on only some sets $S_1, S_2, \dots \subseteq V(G)$ and prove that c are in $C_j^{h,f}$, for some $j \in T$, $h \in H_j$ and $f \in F_{j,h}$, by proving that $c(S_i) \in C_j^{h,f}(S_i)$, for every i . One could argue that c is not totally defined, since we did not define c on all the subsets of $V(G_j)$. However, we define c in this manner because we do not care about the value of the function c on the other sets and, then, for all other sets W , $c(W)$ can be any path in $C_j^{h,f}(W)$.

The next four lemmas will show how to compute M_j , depending on the type of node j .

Lemma 6 (Leaf Node). *Let $j \in T$ be a leaf node. For any $h \in H_j$ and $f \in F_{j,h}$ such that (h, f) is a valid index of M_j , we have that $M_j(h, f) = |\{v \in B_j \mid h(v) = 0\}|$.*

Proof. By the definition of (h, f) -hull set, since $V(G_j) = B_j$, the only set that could be a (h, f) -hull set of $V(G_j)$ is $S = \{v \in B_j \mid h(v) = 0\}$ since S is the only set in $\mathcal{Z}_0^{j,h}$. Let us prove that S is indeed a (h, f) -hull set of $V(G_j)$.

First, note that, since j is a leaf node, for each $k \in Img(h)$, there is only one set in the family $\mathcal{Z}_k^{j,h}$, which is $Z_k = \{v \in B_j \mid h(v) \leq k\}$.

Let k be an arbitrary integer such that $k \in Img(h)$ and $0 \leq k < \max_{v \in B_j} h(v)$. So let P be any path in $Q_f(A_j^h(Z_k))$ and $c(Z_k) = P$. Each vertex $v \in Ext(P)$ either is in $V(G) \setminus B_j$ or, since (h, f) is a valid index, is such that $h(v) < A_j^h(Z_k)$, which implies that it is in Z_k . Thus, we have that $Ext(P) \subseteq U_j(Z_k)$. Hence, we can conclude that $P \in C_j^{h,f}(Z_k)$, which implies that $c \in C_j^{h,f}$. Since $Int_j(P) = K_j^h(Z_k)$, we have that $I_j^c(Z_k) = Z_{k'}$, where $k' = A_j^h(Z_k)$.

So, since $Z_0 = S$, for any $i > 0$, by a simple argument of induction on i , we have that $I_j^c(S, i) = Z_k$, where k is the i th smaller integer in $\text{Img}(h) \setminus \{0\}$. Hence, we have that $Y_j^c(S) = B_j$ and, thus, we conclude that S is a (h, f) -hull set of G_j . \square

Lemma 7 (Forget Node). *Let $j \in T$ be a forget node and $B_j = B_{j'} \setminus \{v\}$. For any $h \in H_j$ and $f \in F_{j,h}$ such that (h, f) is a valid index of M_j , we have that*

$$M_j(h, f) = \min_{h', f'} M_{j'}(h', f'),$$

where h' iterates over all functions in $H_{j'}$ such that h' is nice and has the same relative order as h , and f' iterates over all functions in $F_{j',h'}$ such that, for all $u \in \text{Dom}(h)$, we have that $f'(h'(u)) = f(h(u))$.

Proof. Let L be the set of functions $l \in H_{j'}$ such that $l \in L$ if and only if l is nice and has the same relative order as h and let X_l , where $l \in L$, be the set of functions $g \in F_{j',l}$ such that, for all $u \in \text{Dom}(h)$, we have that $g(l(u)) = f(h(u))$. Since $V(G_j) = V(G_{j'})$, for any $S \subseteq V(G_j)$ and any path P , we have that $\text{Int}_j(P) = \text{Int}_{j'}(P)$ and $U_j(S) = U_{j'}(S)$, which implies that $\text{Int}_j(P) \setminus S = \emptyset$ if and only if $\text{Int}_{j'}(P) \setminus S = \emptyset$ and $P \in U_j(S)$ if and only if $P \in U_{j'}(S)$.

Let S be a subset of $V(G_j)$. To prove what we want, it is enough to prove that a set S is a (h, f) -hull set of G_j if and only if there are functions $h' \in L$ and $f' \in X_{h'}$ such that S is a (h', f') -hull set of $G_{j'}$. This is because this equivalence implies that, if we let Q be a minimum (h, f) -hull set of G_j , X be a minimum (l, g) -hull set of $G_{j'}$, where $l \in L$ and $g \in X_l$ are functions that satisfy the minimum $\min_{h', f'} M_{j'}(h', f')$, then, by the equivalence, we have that there are $h' \in L$ and $f' \in X_{h'}$, such that Q is a (h', f') -hull set of $G_{j'}$ and, thus, $\min_{h', f'} M_{j'}(h', f') = |X| \leq |Q| = M_j(h, f)$. Also, by the equivalence, we have that X is a (h, f) -hull set of G_j , and, hence, $M_j(h, f) = |Q| \leq |X| = \min_{h', f'} M_{j'}(h', f')$.

So, first, suppose that S is a (h, f) -hull set of G_j . Let c be a function in $C_j^{h,f}$ such that $Y_j^c(S) = V(G_j)$. Also, let $I_y = I_j^c(S, y)$, for any y . Note that, for any y such that $I_y \neq V(G_j)$, $\text{Int}_j(c(I_y)) \neq \emptyset$. To prove that S is a (l, g) -hull set of $G_{j'}$, for some functions $l \in L$ and $g \in X_l$, it is enough to prove that $S \in \mathcal{Z}_0^{j',l}$ and that there is a function $c' \in C_{j'}^{l,g}$ such that, for all $y \geq 0$ such that $I_y \neq V(G_j)$, $c'(I_y) = c(I_y)$. This is because, if so, for any $y \geq 0$, we would have that $I_y = I_{j'}^{c'}(S, y)$ and, thus, $Y_{j'}^{c'}(S) = Y_j^c(S) = V(G_j) = V(G_{j'})$.

Let i be the least integer such that $v \in I_i$. If $i > 0$, let $D = c(I_{i-1})$. In this case, note that $v \in \text{Int}_j(D) \setminus I_{i-1}$. For every $w \in B_{j'}$, let i_w be the least integer such that $w \in I_{i_w}$ (note that $i_v = i$). Let l be the function such that, for every $w \in B_{j'}$ such that $i_w = 0$, $l(w) = 0$ and, for every $w \in B_{j'}$ such that $i_w > 0$, $l(w) = |\{x \in \mathbb{N} \mid \exists q \in B_{j'}, x = i_q \text{ and } 0 < x \leq i_w\}|$. So, for instance, if we have $B_{j'} = \{v, x, y, w, z\}$ and $i_v = 0, i_x = 2, i_y = 3, i_w = 3$ and $i_z = 5$, then we have $l(v) = 0, l(x) = 1, l(y) = 2, l(w) = 2$ and $l(z) = 3$.

By definition of l , we have that, for any $w \in B_j$, $w \in S$ if and only if $l(w) = 0$, which implies that $h(w) = 0$ if and only if $l(w) = 0$. Since, for any $w, w' \in B_j$, $i_w \leq i_{w'}$ if and only if $h(w) < h(w')$, then, for any $w, w' \in B_j$, $l(w) < l(w')$ if and only if $h(w) < h(w')$, which implies that h and l have the same relative order. Additionally, we have that:

- If there is a vertex $w \in B_j$ such that $i = i_w$ and there is a vertex in $B_{j'} \cap S$, then $\text{Img}(l) = \{0, 1, \dots, |\text{Img}(h)| - 1\}$;
- If there is a vertex $w \in B_j$ such that $i = i_w$ and there is no vertex in $B_{j'} \cap S$, then $\text{Img}(l) = \{1, 2, \dots, |\text{Img}(h)|\}$;
- If there no vertex $w \in B_j$ such that $i = i_w$ and there is a vertex in $B_{j'} \cap S$, then $\text{Img}(l) = \{0, 1, \dots, |\text{Img}(h)|\}$;
- If there is no vertex $w \in B_j$ such that $i = i_w$ and there is no vertex in $B_{j'} \cap S$, then $\text{Img}(l) = \{1, 2, \dots, |\text{Img}(h)| + 1\}$;

Thus, l is nice and, then, we conclude that $l \in L$. Additionally, let $g \in F_{j',l}$ such that, for all $w \in B_j$ such that $l(w) > 0$, let $g(l(w)) = f(h(w))$ and, if $l(v) > 0$, let $g(l(v)) = \text{Ext}(D)$. Note that if there is a vertex $w \in B_j$ such that $l(w) = l(v)$, then $g(l(v)) = f(h(w)) = \text{Ext}(D)$. Hence, we have that $g \in X_l$.

Now that we defined $l \in L$ and $g \in X_l$, let us prove that S is a (l, g) -hull set of $G_{j'}$. We have that $S \cap B_{j'} = (S \cap B_j) \cup (S \cap \{v\}) = \{z \in B_j \mid h(z) = 0\} \cup (S \cap \{v\}) = \{z \in B_{j'} \mid l(z) = 0\}$, which implies that $S \in \mathcal{Z}_0^{j',l}$.

Claim 1. *If, for all $y \geq 0$ such that $I_y \neq V(G_j)$, there is a function $c'_y \in C_{j'}^{l,g}$ and an integer $z_y > 0$ such that $I_{y+1} \subseteq I_{j'}^{c'_y}(I_y, z_y)$ and $K_{j'}^l(I_{y+1}) = K_{j'}^l(I_{j'}^{c'_y}(I_y, z_y))$, then there is a function $c' \in C_{j'}^{l,g}$ such that $Y_{j'}^{c'}(S) = V(G_{j'})$.*

Claim's proof. For any $W \subseteq V(G_j)$, let us say that $L(W)$ is true if and only if there is a function $c' \in C_{j'}^{l,g}$ such that $Y_{j'}^{c'}(W) = V(G_{j'})$. If, for all $y \geq 0$ such that $I_y \neq V(G_j)$, there is a function $c'_y \in C_{j'}^{l,g}$ and an integer $z_y > 0$ such that $I_{y+1} \subseteq I_{j'}^{c'_y}(I_y, z_y)$, then, for all $y \geq 0$ such that $I_y \neq V(G_j)$, we have that $L(I_{j'}^{c'_y}(I_y, z_y)) \implies L(I_y)$ because, if there is a function $c' \in C_{j'}^{l,g}$ such that $Y_{j'}^{c'}(I_{j'}^{c'_y}(I_y, z_y)) = V(G_{j'})$ then the function d , where, for all $0 \leq y' < z_y$, $d(I_{j'}^{c'_y}(I_y, y')) = c'_y(I_{j'}^{c'_y}(I_y, y'))$ and, for all the other sets W , $d(W) = c'(W)$, is such that $d \in C_{j'}^{l,g}$, since both c'_y and c' are in $C_{j'}^{l,g}$, and $Y_{j'}^d(I_y) = V(G_{j'})$.

Also, by Lemma 5, we have that $L(I_{y+1}) \implies L(I_{j'}^{c'_y}(I_y, z_y))$. Thus, we have that, for all $y \geq 0$ such that $I_y \neq V(G_j)$, $L(I_{y+1}) \implies L(I_y)$. Since, for all $y \geq 0$ such that $I_y \neq V(G_j)$, $I_y \subset I_{y+1}$, then there is a value y' such that $I_{y'} = V(G_j)$.

Therefore, since we have $L(V(G_j))$, then we have that there is a function $c' \in C_{j'}^{l,g}$ such that $Y_{j'}^{c'}(S) = V(G_{j'})$. \square

So, let us show a function $c' \in C_{j'}^{l,g}$ such that, for every y such that $I_y \neq V(G_j)$, $I_{y+1} \subseteq I_{j'}^{c'}(I_y, z)$ and $K_{j'}^l(I_{y+1}) = K_{j'}^l(I_{j'}^{c'}(I_y, z))$, for some $z > 0$. Let $W = I_y$, for some arbitrary $y \geq 0$ such that $I_y \neq V(G_j)$, and $M = c(W)$. Since $W = I_y$, we have that $W \in \mathcal{Z}_j^h$.

Let us analyze two cases. Recall that $Int_j(P) = Int_{j'}(P)$ and $U_j(S) = U_{j'}(S)$ for any path P and set of vertices S .

The first case happens when $M \in R_j(W)$ and let us divide this case in three sub-cases. The first one happens when either $y > i - 1$ and $v \notin Int(M)$ or $y < i - 1$. In this sub-case, let $c'(W) = M$. We have that either $y < i - 1$, which implies that $v \notin Int(M)$, or $y > i - 1$, which also implies that $v \notin Int(M)$. Thus, we have that $Int(M) \cap B_{j'} = \emptyset$. Also, we have that $Int_{j'}(M) \setminus W \neq \emptyset$. Hence, $M \in R_{j'}(W)$, which implies that $M \in C_{j'}^{l,g}(W)$. Also, we have that $I_{j'}^{c'}(W, 1) = W \cup Int_{j'}(M) = I_{y+1}$ and, thus, $K_{j'}^l(I_{y+1}) = K_{j'}^l(I_{j'}^{c'}(W, 1))$.

The second sub-case happens when, $y > i - 1$ and $v \in Int(M)$. Let $Ext(M) = \{q, q'\}$, P be the subpath of M such that $Ext(P) = \{q, v\}$ and P' be the subpath of M such that $Ext(P') = \{q', v\}$. Since $y > i - 1$, we have that $v \in W$ and, since $v \in Int(M)$ and $Int(M) \cap B_j = \emptyset$, then $Int(M) \cap B_{j'} = \{v\}$, which implies that $Int_{j'}(P) \cap B_{j'} = Int_{j'}(P') \cap B_{j'} = \emptyset$. If $Int_{j'}(P) \setminus W \neq \emptyset$, let $c'(W) = P$, or, otherwise, let $c'(W) = P'$.

So, if $Int_{j'}(P) \setminus W = \emptyset$ or $Int_{j'}(P') \setminus W = \emptyset$, we have that $Int_{j'}(c'(W)) \setminus W = Int_j(M) \setminus W \neq \emptyset$, which implies that $I_{j'}^{c'}(W, 1) = W \cup Int_{j'}(c'(W)) = W \cup Int_j(M) = I_{y+1}$ and, thus, $K_{j'}^l(I_{y+1}) = K_{j'}^l(I_{j'}^{c'}(W, 1))$. Also, we have that $c'(W) \in R_{j'}(W)$ as $Ext(c'(W)) \subseteq W$, $Int_{j'}(c'(W)) \cap B_{j'} = \emptyset$ and $Int_{j'}(c'(W)) \setminus W \neq \emptyset$, which implies that $c'(W) \in C_{j'}^{l,g}(W)$.

If $Int_{j'}(P) \setminus W \neq \emptyset$ and $Int_{j'}(P') \setminus W \neq \emptyset$, additionally to have $c'(W) = P$, let $W_2 = W \cup Int_{j'}(P)$ and $c'(W_2) = P'$. We have that $Int_{j'}(P) \cap B_{j'} = \emptyset$ and $Int_{j'}(P') \cap B_j = \emptyset$. Also, we have that $Int_{j'}(P) \cap Int_{j'}(P') = \emptyset$, since P and P' are subpaths of a minimum path, which implies that $Int_{j'}(P') \setminus W_2 \neq \emptyset$. Furthermore, we have that $Ext(P) \subseteq W$ and $Ext(P') \subseteq W \subseteq W_2$. Thus, we have that $P \in R_{j'}(W)$ and $P' \in R_{j'}(W_2)$, which implies that $P \in C_{j'}^{l,g}(W)$ and $P' \in C_{j'}^{l,g}(W_2)$. Furthermore, since $v \in W$, we have that $(Int_{j'}(P) \cup Int_{j'}(P')) \setminus W = Int_j(M) \setminus W$, which implies that $I_{j'}^{c'}(W, 2) = W \cup Int_{j'}(P) \cup Int_{j'}(P') = W \cup Int_j(M) = I_{y+1}$ and, thus, $K_{j'}^l(I_{y+1}) = K_{j'}^l(I_{j'}^{c'}(W, 2))$.

The last sub-case happens when $y = i - 1$. In this sub-case, let $c'(W) = M$. So, we have that $M = D$, $A_{j'}^l(W) = l(v)$, $v \in Int(M) \setminus W$ and, since $Int(M) \cap B_j = \emptyset$, there is no vertex $w \in B_j$ such that $l(w) = l(v)$. Thus, we have that $K_{j'}^l(W) = \{v\}$ and, since $Int(M) \cap B_{j'} = \{v\}$, then $M \in Q_{j'}^{l,g}(A_{j'}^l(W))$. Also, we have that $Ext(M) \subseteq W \subseteq U_{j'}(W)$. Then, we can conclude that $M \in C_{j'}^{l,g}(W)$. Also, we have that $I_{j'}^{c'}(W, 1) = W \cup Int_{j'}(M) = I_{y+1}$ and, hence, $K_{j'}^l(I_{y+1}) = K_{j'}^l(I_{j'}^{c'}(W, 1))$.

The second case happens when $M \in Q_j^{h,f}(A_j^h(W))$ and $Ext(M) \in U_j(W)$. We will also divide this case further in three sub-cases. The first one happens when either $y > i - 1$ and $v \notin Int(M)$ or $y < i - 1$. In this sub-case, let $c'(W) = M$. Since $y < i - 1$, we have that $v \notin Int(M)$. Also, we have that $v \notin K_j^h(W)$, which implies that $K_j^h(W) = K_j^l(W)$. Hence, we have that $M \in Q_{j'}^{l,g}(A_{j'}^l(W))$. Also, we have that $Ext(M) \in U_j(W) = U_{j'}(W)$. Thus, we have $M \in C_{j'}^{l,g}(W)$. Furthermore, $I_{j'}^{c'}(W, 1) = W \cup Int_{j'}(M) = I_{y+1}$ and, thus, $K_{j'}^l(I_{y+1}) = K_{j'}^l(I_{j'}^{c'}(W, 1))$.

The second sub-case happens when $y = i - 1$, which implies that $v \in Int(M) \setminus W$ and $v \in K_j^h(W) = K_{j'}^l(W)$. In this sub-case, let $c'(W) = M$. Thus, since $v \in Int(M) \setminus W$ and $Int(M) \cap B_j = K_j^h(W)$, then $Int(M) \cap B_{j'} = K_{j'}^l(W)$, which implies that $M \in Q_{j'}^{l,g}(A_{j'}^l(W))$. Also, we have that $Ext(M) \subseteq U_j(W) = U_{j'}(W)$. Then, we can conclude that $M \in C_{j'}^{l,g}(W)$. Furthermore, we have that $I_{j'}^{c'}(W, 1) = W \cup Int_{j'}(M) = I_{y+1}$ and, then, $K_{j'}^l(I_{y+1}) = K_{j'}^l(I_{j'}^{c'}(W, 1))$.

The last sub-case happens when $y > i - 1$ and $v \in \text{Int}(M)$. Let P be the shortest subpath of M such that $\text{Ext}(P) \subseteq \text{Ext}(M) \cup K_j^h(W)$ and $v \in \text{Int}(P)$. Since $y > i - 1$, then $v \in W$, $K_j^h(W) = K_{j'}^l(W)$. Let P' be the subpath of any path in $Q_{j'}^{l,g}(A_{j'}^l(W))$ such that $\text{Ext}(P') = \text{Ext}(P)$, which implies that $\text{Int}(P') \cap B_{j'} = \emptyset$. Let M' be the path W when we replace P by P' , let $c'(W) = M'$ and let $W_2 = W \cup \text{Int}_{j'}(M')$. Thus, since $\text{Int}(P') \cap B_{j'} = \emptyset$, we have that $v \notin \text{Int}(M')$, which implies that $\text{Int}(M') \cap B_{j'} = K_j^h(W) = K_{j'}^l(W)$. So, we have that $M' \in Q_{j'}^{l,g}(A_{j'}^l(W))$. Also, $\text{Ext}(M') = \text{Ext}(M) \subseteq U_j(W) = U_{j'}(W)$. Therefore, we have that $M' \in C_{j'}^{l,g}(W)$. If $\text{Int}_{j'}(P) \setminus W_2 = \emptyset$, we have that $I_{j'}^c(W, 1) = W_2 = W \cup \text{Int}_{j'}(M') \supseteq I_{y+1}$ and, since $\text{Int}(P') \cap B_{j'} = \emptyset$, we have that $K_{j'}^l(I_{y+1}) = K_{j'}^l(I_{j'}^c(W, 1))$.

If $\text{Int}_{j'}(P) \setminus W_2 \neq \emptyset$, let $\text{Ext}(P) = \{q_1, q_2\}$ and P_1 and P_2 be the subpaths of P such that $\text{Ext}(P_1) = \{q_1, v\}$ and $\text{Ext}(P_2) = \{q_2, v\}$. Since $y > i - 1$, we have that $v \in W$ and, since $v \in \text{Int}(P)$ and $\text{Int}(P) \cap B_j = \emptyset$, then $\text{Int}(P) \cap B_{j'} = \{v\}$, which implies that $\text{Int}(P_1) \cap B_{j'} = \text{Int}(P_2) \cap B_{j'} = \emptyset$. Also, at least one of the two subpaths P_1 and P_2 have an internal vertex in $V(G_{j'}) \setminus W_2$, since $\text{Int}_{j'}(P) \setminus W_2 \neq \emptyset$, $\text{Int}_{j'}(P_1) \cup \text{Int}_{j'}(P_2) = \text{Int}_{j'}(P) \setminus \{v\}$ and $v \in W \subseteq W_2$.

If $\text{Int}_{j'}(P_1) \setminus W_2 \neq \emptyset$, let $c'(W_2) = P_1$, or, otherwise, let $c'(W_2) = P_2$. So, if either $\text{Int}_{j'}(P_1) \setminus W_2 = \emptyset$ or $\text{Int}_{j'}(P_2) \setminus W_2 = \emptyset$, we have that $\text{Int}_{j'}(c'(W_2)) \setminus W_2 = \text{Int}_{j'}(P) \setminus W_2 \neq \emptyset$, which implies that $I_{j'}^c(W, 2) = W \cup \text{Int}_{j'}(M') \cup \text{Int}_{j'}(c'(W_2)) = W \cup \text{Int}_{j'}(M') \cup \text{Int}_{j'}(P) = W \cup \text{Int}_{j'}(M) \cup \text{Int}_{j'}(P) \supseteq I_{y+1}$ and, since $\text{Int}(c'(W_2)) \cap B_{j'} = \emptyset$, we have that $K_{j'}^l(I_{y+1}) = K_{j'}^l(I_{j'}^c(W, 2))$. Also, we have that $c'(W_2) \in R_{j'}(W_2)$ as $\text{Ext}(c'(W_2)) \subseteq W_2$, $\text{Int}_{j'}(c'(W_2)) \cap B_j = \emptyset$ and $\text{Int}_{j'}(c'(W_2)) \setminus W_2 \neq \emptyset$, which implies that $c'(W_2) \in C_{j'}^{l,g}(W_2)$.

If $\text{Int}_{j'}(P_1) \setminus W_2 \neq \emptyset$ and $\text{Int}_{j'}(P_2) \setminus W_2 \neq \emptyset$, additionally to have $c'(W_2) = P_1$, let $W_3 = W_2 \cup \text{Int}_{j'}(P_1)$ and $c'(W_3) = P_2$. Since $\text{Int}(P) \cap B_{j'} = \{v\}$, we have that $\text{Int}_{j'}(P_1) \cap B_{j'} = \text{Int}_{j'}(P_2) \cap B_{j'} = \emptyset$. Also, we have that $\text{Int}_{j'}(P_1) \cap \text{Int}_{j'}(P_2) = \emptyset$ because P_1 and P_2 are subpaths of a minimum path, which implies that $\text{Int}_{j'}(P_2) \setminus W_3 \neq \emptyset$. Furthermore, we have that $\text{Ext}(P_1) \subseteq W_2$ and $\text{Ext}(P_2) \subseteq W_2 \subseteq W_3$.

Thus, we have that $P_1 \in R_{j'}(W_2)$ and $P_2 \in R_{j'}(W_3)$, which implies that $P_1 \in C_{j'}^{l,g}(W_2)$ and $P_2 \in C_{j'}^{l,g}(W_3)$. Furthermore, since $v \in W$, we have that $(\text{Int}_{j'}(P_1) \cup \text{Int}_{j'}(P_2)) \setminus W = \text{Int}_{j'}(P) \setminus W$, which implies that $I_{j'}^c(W, 3) = W \cup \text{Int}_{j'}(M') \cup \text{Int}_{j'}(P_1) \cup \text{Int}_{j'}(P_2) = W \cup \text{Int}_{j'}(M') \cup \text{Int}_{j'}(P) = W \cup \text{Int}_{j'}(M) \cup \text{Int}_{j'}(P) \supseteq I_{y+1}$ and, since $\text{Int}(P_1) \cap B_{j'} = \text{Int}(P_2) \cap B_{j'} = \emptyset$, we have that $K_{j'}^l(I_{y+1}) = K_{j'}^l(I_{j'}^c(W, 3))$.

Therefore, we have that there is a function c' in $C_{j'}^{l,f}$, where, for all $y \geq 0$ such that $I_y \neq V(G_j)$, there is a value $z > 0$ such that $I_{y+1} \subseteq I_{j'}^c(I_y, z)$ and $K_{j'}^l(I_{y+1}) = K_{j'}^l(I_{j'}^c(I_y, z))$. Thus, we conclude that there are functions $l \in L$, $g \in X_l$ such that S is a (l, g) -hull set of $G_{j'}$.

Now, suppose that there are functions $l \in L$, $g \in X_l$ such that S is a (l, g) -hull set of $G_{j'}$. Let c' be a function in $C_{j'}^{l,g}$ such that $Y_{j'}^{c'}(S) = V(G_{j'})$. Also, let $I'_x = I_{j'}^c(S, x)$, for any x . Note that, for any y such that $I'_y \neq V(G_j)$, $\text{Int}_{j'}(c'(I'_y)) \neq \emptyset$. Since $l(v) = 0$ if and only if $v \in S$, then $S \cap B_j = S \cap (B_{j'} \setminus \{v\}) = \{z \in B_{j'} \mid l(z) = 0\} \setminus (S \cap \{v\}) = \{z \in B_j \mid l(z) = 0\} = \{z \in B_j \mid h(z) = 0\}$, which implies that $S \in \mathcal{Z}_0^{j,h}$.

Therefore, in order to prove that S is a (h, f) -hull set of G_j , we only have to prove that there is a function $c \in C_j^{h,f}$ such that, for all $y \geq 0$ such that $I'_y \neq V(G_{j'})$, $\text{Int}_j(c(I'_y)) = \text{Int}_{j'}(c'(I'_y))$. This is because, if so, we would have that $I'_y = I_j^c(S, y)$, for all $y \geq 0$, which implies that $Y_j^c(S) = Y_{j'}^{c'}(S) = V(G_{j'}) = V(G_j)$.

Let $W' = I'_y$, for some $y \geq 0$ such that $W' \neq V(G_{j'})$, and let $M' = c'(W')$. First, suppose that $M' \in R_{j'}(W')$. Let $c(W') = M'$. Since $B_j \subseteq B_{j'}$, we have that $\text{Int}(M') \cap B_j = \emptyset$. Also, we have that $\text{Ext}(M') \subseteq W'$ and $\text{Int}_j(M') \setminus W' \neq \emptyset$. Then, we have that $M' \in R_j(W')$, which implies that $M' \in C_j^{h,f}(W')$.

Now, suppose that $M' \in Q_{j'}^{l,g}(A_{j'}^l(W'))$ and $\text{Ext}(M') \subseteq U_{j'}(W')$. Let us analyze two cases. The first one happens when $K_{j'}^l(W') \neq \{v\}$. Let $c(W') = M'$. In this case, we have that there is a vertex $v' \in B_j$ in $K_{j'}^l(W')$, which implies that $A_{j'}^l(W') = l(v')$ and $A_j^h(W') = h(v')$. Since $Q_{j'}^{l,g}(l(v')) \subseteq Q_j^{h,f}(h(v'))$, we have that $M' \in Q_j^{h,f}(A_j^h(W'))$. Also, we have that $\text{Ext}(M') \subseteq U_{j'}(W') = U_j(W')$. Hence, we conclude that $M' \in C_j^{h,f}(W')$.

The second case happens when $K_{j'}^l(W') = \{v\}$. In this case, since B_j separates v from $V(G) \setminus V(G_j)$ and $\text{Int}(M') \cap B_{j'} = \{v\}$, we have that $\text{Ext}(M') \subseteq W'$. Also, we have that $\text{Int}(M') \cap B_j = \emptyset$ and $\text{Int}_j(M') \setminus W' \neq \emptyset$, since $v \in \text{Int}_j(M') \setminus W'$. Therefore, we conclude that $M' \in R_j(W')$, which implies that $M' \in C_j^{h,f}(W')$.

Thus, there is a function $c \in C_j^{h,f}$ such that, for all $y \geq 0$ such that $I'_y \neq V(G_{j'})$, $\text{Int}_j(c(I'_y)) = \text{Int}_{j'}(c'(I'_y))$ and, hence, as argued before, S is a (h, f) -hull set of G_j . \square

Lemma 8 (Introduce Node). *Let $j \in T$ be a introduce node and $B_j = B_{j'} \cup \{v\}$. For any $h \in H_j$ and $f \in F_{j,h}$ such that (h, f) is a valid index of M_j , we have that*

$$M_j(h, f) = \begin{cases} M_{j'}(l, g) + 1 & \text{if } h(v) = 0 \\ M_{j'}(l, g) & \text{otherwise,} \end{cases}$$

where l is h restricted to $B_{j'}$ and $g \in F_{j',l}$ is the function such that, for all $x \in \text{Img}(l) \setminus \{0\}$, $g(x) = f(x)$.

Proof. Let S be a subset of $V(G_j)$ such that $h(v) = 0$ if and only if $v \in S$. Also, let $S' = S \setminus \{v\}$. Suppose we have the following equivalence: S is a (h, f) -hull set of G_j if and only if S' is a (l, g) -hull set of $G_{j'}$. Let S be a minimum (h, f) -hull set of G_j and X be a minimum (l, g) -hull set of $G_{j'}$.

If $h(v) = 0$, then $v \in S$ and $S \setminus \{v\}$ is a (l, g) -hull set of $G_{j'}$, which implies that $M_j(h, f) = |S| = |S \setminus \{v\}| + 1 \geq |X| + 1 = M_{j'}(l, g) + 1$. Also, we have that $X \cup \{v\}$ is a (h, f) -hull set of G_j , which implies that $M_{j'}(l, g) + 1 = |X| + 1 = |X \cup \{v\}| \geq |S| = M_j(h, f)$. Thus, if $h(v) = 0$, we have that $M_j(h, f) = M_{j'}(l, g) + 1$.

If $h(v) > 0$, then $v \notin S$ and $S \setminus \{v\}$ is a (l, g) -hull set of $G_{j'}$, which implies that $M_j(h, f) = |S| = |S \setminus \{v\}| \geq |X| = M_{j'}(l, g)$. Also, we have that X is a (h, f) -hull set of G_j , which implies that $M_{j'}(l, g) = |X| \geq |S| = M_j(h, f)$. Thus, if $h(v) > 0$, we have that $M_j(h, f) = M_{j'}(l, g)$.

Therefore, in order to prove what we want, it is enough to prove that S is a (h, f) -hull set of G_j if and only if S' is a (l, g) -hull set of $G_{j'}$, where $h(v) = 0$ if and only if $v \in S$.

So, let $S \subseteq V(G_j)$ such that $h(v) = 0$ if and only if $v \in S$. First, let us prove that, if S is a (h, f) -hull set of G_j , then S' is a (l, g) -hull set of $G_{j'}$. Let S be a (h, f) -hull set of G_j and let c be a function in $C_j^{h,f}$ such that $Y_j^c(S) = V(G_j)$ and let $I_y = I_j^c(S, y)$, for all y . Note that, for all y such that $I_y \neq V(G_j)$, $I_y \subset I_{y+1}$.

In order to prove that S' is a (l, g) -hull set of $G_{j'}$, it is enough to prove that $S' \in \mathcal{Z}_0^{j',l}$ and there is a function $c' \in C_{j'}^{l,g}$ such that, for all $y \geq 0$ such that $I_y \neq V(G_j)$ and $I_{y+1} \neq I_y \cup \{v\}$, $c(I_y) = c'(I_y \setminus \{v\})$. This is because, if so, since $S' = Y_0 \setminus \{v\}$, we have that, for all $y \geq 0$ such that $I_y \neq V(G_j)$, there is a $y' \geq y$ such that $I_{y'+1} \setminus \{v\} = I_{y'}^{c'}(S', y')$, which implies that $V(G_{j'}) = V(G_j) \setminus \{v\} = Y_j^c(S) \setminus \{v\} = Y_{j'}^{c'}(S')$.

We have that $S' \cap B_{j'} = (S \setminus \{v\}) \cap (B_j \setminus \{v\}) = (S \cap B_j) \setminus \{v\} = \{w \in B_j \mid h(w) = 0\} \setminus \{v\} = \{w \in B_{j'} \mid l(w) = 0\}$, which implies that $S' \in \mathcal{Z}_0^{j',l}$.

So, let $W = I_y$, $W' = I_y \setminus \{v\}$ and $M = c(W)$. Suppose that $M \in R_j(W)$. Since $B_{j'}$ separates v from $V(G_{j'}) \setminus B_{j'}$, $\text{Int}(M) \cap B_j = \emptyset$ and $\text{Int}_j(W) \setminus W \neq \emptyset$, then $v \notin \text{Ext}(M)$, which implies that $\text{Ext}(M) \subseteq W'$. Also, $\text{Int}(M) \cap B_{j'} = \emptyset$ and, since $v \notin \text{Int}(M)$, $\text{Int}_{j'}(M) \setminus W' \neq \emptyset$. Thus, we have that $M \in R_{j'}(W')$, which implies that $M \in C_{j'}^{l,g}(W')$.

Finally, suppose that $M \in Q_j^{h,f}(A_j^h(W))$ and $\text{Ext}(M) \subseteq U_j(W)$. We have that $K_j^h(W) \neq \{v\}$ because otherwise, since $B_{j'}$ separates $V(G_{j'}) \setminus B_{j'}$ from v and $\text{Int}(M) \cap B_j = \{v\}$, we would have that $\text{Int}_j(M) = \{v\}$, which would imply that $I_{y+1} = I_y \cup \{v\}$ and that is a contradiction.

Thus, we have that there is a vertex $w \in B_{j'}$ that is in $K_j^h(W)$, which implies that $A_j^h(W) = A_{j'}^l(W') = l(w)$. Since, for all $x \in \text{Img}(l) \setminus \{0\}$, $Q_j^{h,f}(x) \subseteq Q_{j'}^{l,g}(x)$, we have that $M \in Q_{j'}^{l,g}(A_{j'}^l(W'))$. Also, $\text{Ext}(M) \subseteq U_j(W) \subseteq U_{j'}(W')$. Thus, we have that $M \in C_{j'}^{l,g}(W')$.

With this, we proved the function c' , where, for each $y \geq 0$ such that $I_y \neq V(G_j)$ and $I_{y+1} \neq I_y \cup \{v\}$, $c'(I_y \setminus \{v\}) = c(I_y)$, is in $C_{j'}^{l,g}$, which, as discussed before, implies that S' is a (l, g) -hull set of $G_{j'}$.

Now, suppose that S' is a (l, g) -hull set of $G_{j'}$. Let $c' \in C_{j'}^{l,g}$ be a function such that $Y_{j'}^{c'}(S') = V(G_{j'})$. Also, let $I'_y = I_{j'}^{c'}(S', y)$, for all y . Note that, for all y such that $I'_y \neq V(G_{j'})$, $I'_y \subset I'_{y+1}$, which implies that $\text{Int}_{j'}(c'(I'_y)) \setminus I'_y \neq \emptyset$.

So, let us prove that S is a (h, f) -hull set of $V(G_j)$. Since we have that $S \cap B_j = ((S \setminus \{v\}) \cap B_{j'}) \cup (S \cap \{v\}) = \{w \in B_{j'} \mid l(w) = 0\} \cup (S \cap \{v\})$ and $h(v) = 0$ if and only if $v \in S$, then $S \cap B_j = \{w \in B_{j'} \mid l(w) = 0\} \cup (S \cap \{v\}) = \{w \in B_j \mid h(w) = 0\}$, which implies that $S \in \mathcal{Z}_0^{h,f}$.

Now, we only have to show that there is a function $c \in C_j^{h,f}$ such that $Y_j^c(S) = V(G_j)$. Let e_y be the least integer such that $I'_{e_y+1} = V(G_{j'})$.

Let x be the least integer such that $x \in \text{Img}(l)$ and either $x \geq h(v)$ or, if there is no such integer, let $x = tw + 2$. Also, let i be the value such that, if $x = 0$, $i = -1$; if $0 < x < tw + 2$, $c'(I'_i) \in Q_{j'}^{l,g}(A_{j'}^l(I'_i))$ and $x = A_{j'}^l(I'_i)$; and if $x = tw + 2$, let $i = y_f$.

Let $W_y = I'_y \cup \{v\}$, if $y > i$, and $W_y = I'_y$, otherwise. Note that $I'_0 = S'$ and $W_0 = S$.

Claim 2. *If, for all $y \geq 0$ such that $I'_y \neq V(G_{j'})$, there is a function $c_y \in C_{j'}^{h,f}$ and an integer $z_y > 0$ such that $W_{y+1} \subseteq I_j^{c_y}(W_y, z_y)$ and $K_j^h(W_{y+1}) = K_j^h(I_j^{c_y}(W_y, z_y))$, then there is a function $c \in C_j^{h,f}$ such that $Y_j^c(S) = V(G_j)$.*

Claim's proof. For any $W \subseteq V(G_j)$, let us say that $L(W)$ is true if and only if there is a function $c \in C_j^{h,f}$ such that $Y_j^c(W) = V(G_j)$. If, for all $y \geq 0$ such that $W'_y \neq V(G_{j'})$, there is a function $c_y \in C_{j'}^{h,f}$ and an integer $z_y > 0$ such that $W_{y+1} \subseteq I_j^{c_y}(W_y, z_y)$, then, for all $y \geq 0$ such that $W'_y \neq V(G_{j'})$, we have that $L(I_j^{c_y}(W_y, z_y)) \implies L(W_y)$ because, if there is a function $c \in C_j^{h,f}$ such that $Y_j^c(I_j^{c_y}(W_y, z_y)) = V(G_j)$ then the function c' , where, for all $0 \leq y' < z_y$, $c'(I_j^{c_y}(W_y, y')) = c_y(I_j^{c_y}(W_y, y'))$ and, for all the other sets W , $c'(W) = c(W)$, is such that $c' \in C_j^{h,f}$, since both c_y and c are in $C_{j'}^{h,f}$, and $Y_j^{c'}(W_y) = V(G_j)$.

Also, by Lemma 5, we have that $L(W_{y+1}) \implies L(I_j^{c_v}(W_y, z_y))$. Thus, we have that, for all $y \geq 0$ such that $W'_y \neq V(G_{j'})$, $L(W_{y+1}) \implies L(W_y)$. Since, for all $y \geq 0$ such that $I'_y \neq V(G_{j'})$, $W_y \subset W_{y+1}$ and there is a value y such that $I'_y \neq V(G_{j'})$ and $v \in W_y$, which is when $y > i$, then there is a value y such that $W_y = V(G_j)$. Hence, by transitivity, we have that $L(V(G_j)) \implies L(S)$ and, since we have $L(V(G_j))$, we also have $L(S)$. \square

So, let $W' = I'_y$, for some arbitrary $y \geq 0$ such that $I'_y \neq V(G_{j'})$, $M' = c'(I'_y)$ and $W = W_y$. Let us prove that, for all $y \geq 0$ such that $W' \neq V(G_{j'})$, there is a function $c \in C_j^{h,f}$ and an integer $z > 0$ such that $W_{y+1} \subseteq I_j^c(W, z)$ and $K_j^h(W_{y+1}) = K_j^h(I_j^{c_v}(W_y, z_y))$.

First, suppose that $M' \in R_{j'}(W')$. Then, we have that either $y \neq i$ or $y = i = ey$. Since $B_{j'}$ separates $W' \setminus B_{j'}$ from v , $Int(M') \cap B_{j'} = \emptyset$ and $Int_{j'}(M') \setminus W' \neq \emptyset$, we have that $v \notin Int(M')$. So, let $c(W) = M'$. We have that $Ext(M') \subseteq W' \subseteq W$. Also, since $v \notin Int(M')$, we have that $Int_j(M') \cap B_j = \emptyset$. Finally, since $Int_{j'}(M') \setminus W' \neq \emptyset$, $W \subseteq W' \cup \{v\}$ and $v \notin Int(M')$, then $Int_j(M') \setminus W \neq \emptyset$. Thus, we have that $M' \in R_j(W)$, which implies that $M' \in C_j^{h,f}(W)$.

If $y > i$, then $W = W' \cup \{v\}$, which implies that $I_j^c(W, 1) = (W' \cup \{v\}) \cup Int_j(M') = I'_{y+1} \cup \{v\} = W_{y+1}$. If $y < i$, then $W = W'$, which implies that $I_j^c(W, 1) = W' \cup Int_j(M')I'_{y+1} = W_{y+1}$ as $v \notin Int_j(M')$. Also, since $I_j^c(W, 1) = W_{y+1}$, for $y \neq i$, then $K_j^h(I_j^c(W, 1)) = K_j^h(W_{y+1})$.

If $y = i = ey$, then $W = W'$ and, then, $W_1 = I_j^c(W, 1) = I'_{y+1} = V(G_{j'})$. Thus, by definition of i , we have that $h(v) > h(w)$ for every $w \in B_{j'}$, which implies that $A_j^h(W_1) = h(v)$. Let P be any path in $Q_j^{h,f}(A_j^h(W_1))$ and let $c(W_1) = P$. Since $P \in Q_j^{h,f}(A_j^h(I'_{y+1}))$, we have that $v \notin Ext(P)$, which implies that $Ext(P) \subseteq V(G) \setminus \{v\} = U_j(W_1)$. This implies that $P \in C_j^{h,f}(W_1)$. Also, we have that $I_j^c(W, 2) = I_j^c(W_1, 1) = I'_{y+1} \cup \{v\} = W_{y+1}$ and, thus, $K_j^h(I_j^c(W, 2)) = K_j^h(W_{y+1})$.

Now, suppose that $M' \in Q_j^{l,g}(A_{j'}^l(W'))$ and $Ext(M') \subseteq U_{j'}(W')$. Let us further divide this case in four sub-cases.

The first sub-case happens when $y = i$ and $h(v) < A_{j'}^l(W')$. Thus, we have that $W = W'$ and there is no vertex in $w \in B_{j'}$ such that $h(w) = h(v)$ and, thus, $A_j^h(W') = h(v)$. Let P be a path in $Q_j^{h,f}(A_j^h(W'))$ and let $c(W) = P$. Since (h, f) is a valid index of M_j , we have that $Ext(P) \cap B_{j'} \subseteq \{w \in B_j \mid h(w) < h(v)\}$. Also, since $B_{j'}$ separates v from $V(G_{j'}) \setminus B_{j'}$ and $Int(P) \cap B_j = \{v\}$, then $Ext(P) \subseteq B_{j'} \cup (V(G) \setminus V(G_j)) = \{w \in B_j \mid h(w) < h(v)\} \cup (V(G) \setminus V(G_j))$. However, since $\{w \in B_j \mid h(w) < h(v)\} \subseteq W$, then $Ext(P) \subseteq W \cup (V(G) \setminus V(G_j)) = U_j(W')$. Thus, we have that $P \in C_j^{h,f}(W')$. Furthermore, we have that $Int_j(P) = \{v\}$, which implies that $W_2 = I_j^c(W, 1) = W' \cup Int_j(c(W')) = W' \cup \{v\}$. Also, note that $A_{j'}^l(W') = A_j^h(W_2)$.

If $v \notin Int(M')$, let $M = M'$ and $c(W_2) = M$. In this case, since $v \notin Int(M')$, we have that $M' \in Q_j^{h,f}(A_j^h(W_2))$. If $v \in Int(M')$, let P' be the smallest subpath of M' such that $Ext(P') \subseteq Ext(M') \cup K_{j'}^l(W')$ and $v \in Int(P')$, which implies that $Int_j(P') = \{v\}$ as $B_{j'}$ separates v from $V(G_{j'}) \setminus B_{j'}$. Let P be the subpath of any path in $Q_j^{h,f}(A_j^h(W_2))$ such that $Ext(P) = Ext(P')$ and let M be the path M' when we replace P' for P . Let $c(W_2) = M$.

So, in any case, we have that $M \in Q_j^{h,f}(A_j^h(W_2))$. Also, since (h, f) is a valid index for M_j , $v \notin Ext(M')$, we have that $Ext(M) = Ext(M') \subseteq U_j(W_2)$ since $U_j(W_2) = U_{j'}(W) \cup \{v\}$. Thus, we have that $M \in C_j^{h,f}(W_2)$. Also, since $Int_{j'}(P') = \emptyset$, we have that $Int_{j'}(M') \subseteq Int_j(M)$, which implies that $I_j^c(W, 2) = I_j^c(W_2) = (W' \cup \{v\}) \cup Int_j(M) \supseteq I'_{y+1} \cup \{v\} = W_{y+1}$. Also, we have that $K_j^h(I_j^c(W, 2)) = K_j^h(W_{y+1})$ since either $M = M'$, which implies that $I_j^c(W, 2) = W_{y+1}$, or $Int(P) \cap B_j = \emptyset$, which implies that $Int(M) \cap B_j = Int(M') \cap B_j$ and, thus, $B_j \setminus I_j^c(W, 2) = B_j \setminus W_{y+1}$.

The second sub-case happens when $y = i$ and $h(v) = A_{j'}^l(W')$. Thus, we have that $W = W'$ and that there is a vertex w in $B_{j'}$ such that $h(w) = h(v) = A_j^h(W) = A_{j'}^l(W')$, which implies that $K_j^h(W) = K_{j'}^l(W') \cup \{v\}$. First, if $v \in Int(M')$ let $M = M'$ and $c(W) = M$. Since $v \in Int(M')$, $M' \in Q_j^{h,f}(A_j^h(W))$. If $v \notin Int(M')$, let P be the smallest subpath of any path N in $Q_j^{h,f}(A_j^h(W))$ such that $Ext(P) \subseteq Ext(N) \cup K_{j'}^l(W')$ and $v \in Int(P)$, which implies that $Int_j(P) = \{v\}$. We have that $Ext(N) = Ext(M')$ and $K_{j'}^l(W) \subseteq Int(M')$. So, let P' be the subpath of M' such that $Ext(P') = Ext(P)$. Note that, $Int(P') \cap B_j = \emptyset$, since $Int(M') \cap B_j = K_{j'}^l(W')$, $Ext(P') = Ext(P)$ and P is the smallest subpath of N such that $Ext(P) \subseteq Ext(N) \cup K_{j'}^l(W)$. Let M be the path M' when we replace P' for P .

Since $v \in Int(M)$ and $M' \in Q_j^{l,g}(A_{j'}^l(W'))$, then, in any case, $M \in Q_j^{h,f}(A_j^h(W))$. Also, since (h, f) is a valid index for M_j , $v \notin Ext(M')$, which implies that $Ext(M) = Ext(M') \subseteq U_j(W)$ since $Ext(M') \subseteq U_{j'}(W') = U_j(W) \cup \{v\}$. Thus, we have that $M \in C_j^{h,f}(W)$. If $Int_j(P') \setminus W = \emptyset$, since $Int_j(P) = \{v\}$, we have that $W_2 = I_j^c(W, 1) = W \cup Int_j(M) = W' \cup (Int_j(M') \cup \{v\}) = I'_{y+1} \cup \{v\} = W_{y+1}$ and, thus, we have that $K_j^h(I_j^c(W, 1)) = K_j^h(W_{y+1})$.

If $Int_j(P') \setminus W \neq \emptyset$, let $c(W_2) = P'$. In this case, we have that $Ext(P') \subseteq V(G_j)$ because, otherwise, we would have one extremity q_1 of P' in $V(G) \setminus V(G_j)$, which would be also an extremity of M' , and the other

extremity $q_2 \in K_j^l(W')$, which would imply that $\text{Int}_j(P') \setminus W = \emptyset$, since $\text{Ext}(P) = \text{Ext}(P')$, $v \in \text{Int}(P)$ and $\text{Int}(P) \cap B_j = \emptyset$, and that is a contradiction.

We have that $\text{Ext}(M') \cap V(G_j) \subseteq W'$ as $\text{Ext}(M') \in U_{j'}(W')$ and $v \notin \text{Ext}(M')$. Thus, since $\text{Ext}(P') \subseteq V(G_j)$, $\text{Ext}(P') \subseteq \text{Ext}(M') \cup K_j^l(W')$, $K_j^l(W') \subseteq W_2$ and $\text{Ext}(M') \cap V(G_j) \subseteq W'$, we have that $\text{Ext}(P') \subseteq W_2$. Also, since $\text{Int}_j(P) = \{v\}$, we have that $\text{Int}_j(P) \cap \text{Int}_j(P') = \emptyset$, which implies that $\text{Int}_j(P') \setminus W_2 \neq \emptyset$, as $\text{Int}_j(P') \setminus W \neq \emptyset$. Furthermore, we have that $\text{Int}(P') \cap B_j = \emptyset$. Therefore, $P' \in R_j(W_2)$, which implies that $P' \in C_j^{h,f}(W_2)$. Thus, in this case, we have that $I_j^c(W, 2) = I_j^c(W_2) = W \cup \text{Int}_j(M) \cup \text{Int}_j(P') = W' \cup \text{Int}_j(M') \cup \text{Int}_j(P) = I'_{y+1} \cup \{v\} = W_{y+1}$ and, thus, we have that $K_j^h(I_j^c(W, 1)) = K_j^h(W_{y+1})$.

The third sub-case happens when $y \neq i$. If $y > i$, we have that $W = W' \cup \{v\}$ and $h(v) < A_j^h(W)$, which implies that $A_j^l(W') = A_j^h(W)$ and $K_j^l(W') = K_j^h(W)$. If $y < i$, we have that $W = W'$ and $h(v) > A_j^h(W)$, which also implies that $A_j^l(W') = A_j^h(W)$. Thus, we have that $h(v) \neq A_j^l(W') = A_j^h(W)$. Also, since (h, f) is a valid index of M_j , we have that, if $y < i$, $v \notin \text{Ext}(M')$, which implies that $\text{Ext}(M') \subseteq U_j(W)$ as $\text{Ext}(M') \subseteq U_{j'}(W')$ and either $v \in W$, if $y > i$, or $v \notin \text{Ext}(M')$, if $y < i$.

First, assume that $v \notin \text{Int}(M')$. Let $c(W) = M'$. Then, since $v \notin \text{Int}(M')$, M' is in $Q_j^{h,f}(A_j^h(W))$. Thus, we have that $M' \in C_j^{h,f}(W)$. Also, if $y > i$, we have that $I_j^c(W, 1) = W' \cup \{v\} \cup \text{Int}_j(M') = I'_{y+1} \cup \{v\} = W_{y+1}$. If $y < i$, $I_j^c(W, 1) = W' \cup \text{Int}_j(M') = W' \cup \text{Int}_{j'}(M') = I'_{y+1} = W_{y+1}$. Furthermore, we have that $K_j^h(I_j^c(W, 1)) = K_j^h(W_{y+1})$.

Now, assume that $v \in \text{Int}(M')$. Let P' be the smallest subpath of M' such that $\text{Ext}(P') \subseteq \text{Ext}(M') \cup K_j^l(W')$ and $v \in \text{Int}(P')$, which implies that $\text{Int}_{j'}(P') = \emptyset$. Let P be the subpath of any path in $Q_j^{h,f}(A_j^h(W))$ such that $\text{Ext}(P) = \text{Ext}(P')$ and let M be the path M' when we replace P' for P . Since $v \notin \text{Int}(P)$, then $v \notin \text{Int}(M)$. Let $c(W) = M$. We have that $\text{Ext}(M) = \text{Ext}(M') \subseteq U_j(W)$. Furthermore, since $M' \in Q_j^{h,f}(A_j^h(W'))$, $A_j^l(W') = A_j^h(W)$ and $v \notin \text{Int}(M)$, then $M \in Q_j^{h,f}(A_j^h(W))$, which implies that $M \in C_j^{h,f}(W)$. Note that, since $\text{Int}_{j'}(P') = \emptyset$, $\text{Int}_{j'}(M') \subseteq \text{Int}_j(M)$. Also, note that $\text{Int}(P) \cap B_j = \emptyset$ since $v \notin \text{Int}(P)$, $\text{Ext}(P) = \text{Ext}(P')$ and P' be the smallest subpath of M' such that $\text{Ext}(P') \subseteq \text{Ext}(M') \cup K_j^l(W')$.

So, if $y > i$, we have that $I_j^c(W, 1) = (W' \cup \{v\}) \cup \text{Int}_j(M) \supseteq W' \cup \{v\} \cup \text{Int}_{j'}(M') = I'_{y+1} \cup \{v\} = W_{y+1}$. If $y < i$, $I_j^c(W, 1) = W' \cup \text{Int}_j(M) \supseteq W' \cup \text{Int}_{j'}(M') \supseteq I'_{y+1} = W_{y+1}$. Furthermore, since $\text{Int}(P) \cap B_j = \emptyset$, we have that $\text{Int}(M') \cap B_j = \text{Int}(M) \cap B_j = \emptyset$, which implies that $K_j^h(I_j^c(W, 1)) = K_j^h(W_{y+1})$.

Therefore, by our claim, we have a function $c \in C_j^{h,f}$ such that $Y_j^c(S) = V(G_j)$, which, since we already showed that $S \in \mathcal{Z}_0^{j,h}$, implies that S is a (h, f) -hull set of G_j . \square

Lemma 9 (Join Node). *Let $j \in T$ be a join node and $B_j = B_{j_1} = B_{j_2}$. For any $h \in H_j$ and $f \in F_{j,h}$ such that (h, f) is a valid index of M_j , we have that $M_j(h, f) = M_{j_1}(h, f) + M_{j_2}(h, f) - |\{v \in B_j \mid h(v) = 0\}|$.*

Proof. Let S be a subset of $V(G_j)$. Suppose that we have the following equivalence: S is a (h, f) -hull set of G_j if and only if $S_1 = S \cap V(G_{j_1})$ is a (h, f) -hull set of G_{j_1} and $S_2 = S \cap V(G_{j_2})$ is a (h, f) -hull set of G_{j_2} .

Then, let S be a minimum (h, f) -hull set of G_j , which implies that S_1 and S_2 are (h, f) -hull set of G_{j_1} and G_{j_2} respectively. So, we have that $M_j(h, f) = |S| = |S \cap V(G_{j_1})| + |S \cap V(G_{j_2})| - |S \cap V(G_{j_1}) \cap V(G_{j_2})| = |S_1| + |S_2| - |\{v \in B_j \mid h(v) = 0\}| \geq M_{j_1}(h, f) + M_{j_2}(h, f) - |\{v \in B_j \mid h(v) = 0\}|$.

On the other hand, let S be a subset of $V(G_j)$ such that S_1 is a minimum (h, f) -hull set of G_{j_1} and S_2 is a minimum (h, f) -hull set of G_{j_2} , which implies that S is a (h, f) -hull set of G_j . Then, we have that $M_j(h, f) \leq |S| = |S_1| + |S_2| - |S \cap S_1 \cap S_2| = M_{j_1}(h, f) + M_{j_2}(h, f) - |\{v \in B_j \mid h(v) = 0\}|$.

Thus, to prove that $M_j(h, f) = M_{j_1}(h, f) + M_{j_2}(h, f) - |\{v \in B_j \mid h(v) = 0\}|$, it is enough to prove that S is a (h, f) -hull set of G_j if and only if S_1 is a (h, f) -hull set of G_{j_1} and S_2 is a (h, f) -hull set of G_{j_2} .

First, let us prove that if S is a (h, f) -hull set of G_j , then S_1 is a (h, f) -hull set of $V(G_{j_1})$. Since S is a (h, f) -hull set of G_j , let $c \in C_j^{h,f}$ be the function such that $Y_j^c(S) = V(G_j)$. Since $B_j = B_{j_1}$, $S_1 \cap B_{j_1} = S \cap B_j = \{w \in B_{j_1} \mid h(w) = 0\}$, which implies that $S_1 \in \mathcal{Z}_0^{j_1,h}$.

Let $I_y = I_j^c(S, y)$, for any y . Let us present a function $c' \in C_{j_1}^{h,f}$ such that, for any $y \geq 0$ where $I_y \neq V(G_j)$ and $I_y \cap V(G_{j_1}) \subset I_{y+1} \cap V(G_{j_1})$, we have that $c(I_y) = c'(I_y \cap V(G_{j_1}))$. This implies that $I_{y+1} \cap V(G_{j_1}) = (I_y \cup \text{Int}_j(c(I_y))) \cap V(G_{j_1}) = (I_y \cap V(G_{j_1})) \cup (\text{Int}_j(c(I_y)) \cap V(G_{j_1})) = (I_y \cap V(G_{j_1})) \cup \text{Int}_{j_1}(c'(I_y \cap V(G_{j_1}))) = I_{j_1}^{c'}(I_y \cap V(G_{j_1}))$, for any $y \geq 0$ where $I_y \neq V(G_j)$ and $I_y \cap V(G_{j_1}) \subset I_{y+1} \cap V(G_{j_1})$. Hence, since $I_0 \cap V(G_{j_1}) = S_1$ and there is a y such that $I_y = V(G_j)$, this proves that $Y_{j_1}^{c'}(S_1) = V(G_{j_1})$.

So, let $W = I_y$, where $y \geq 0$, $I_y \neq V(G_j)$ and $I_y \cap V(G_{j_1}) \subset I_{y+1} \cap V(G_{j_1})$. Also, let $M = c(W)$, $W' = W \cap V(G_{j_1})$ and $c'(W') = M$.

First, suppose that $M \in R_j(W)$. Since $\text{Ext}(M) \subseteq W$, $\text{Int}(M) \cap B_j = \emptyset$ and $\text{Int}_j(M) \setminus W \neq \emptyset$, we have that either $\text{Ext}(M) \subseteq W \cap V(G_{j_1})$ or $\text{Ext}(M) \subseteq W \cap V(G_{j_2})$. However, if $\text{Ext}(M) \subseteq W \cap V(G_{j_2})$, since $\text{Int}(M) \cap B_j = \emptyset$, we would have that $\text{Int}_j(M) \subseteq V(G_{j_2})$, which would imply that $I_y \cap V(G_{j_1}) = I_{y+1} \cap V(G_{j_1})$, which is a contradiction. Therefore, we have that $\text{Ext}(M) \subseteq W \cap V(G_{j_1}) = W'$.

Furthermore, $\text{Int}(M) \cap B_{j_1} = \emptyset$, as $B_j = B_{j_1}$, and $\text{Int}_{j_1}(M) \setminus W' \neq \emptyset$ because $I_y \cap V(G_{j_1}) \subset I_{y+1} \cap V(G_{j_1})$. Therefore, $M \in R_{j_1}(W')$, which implies that $M \in C_{j_1}^{h,f}(W')$.

Now, suppose that $M \in Q_j^{h,f}(A_j^h(W))$ and $\text{Ext}(M) \subseteq R_j(W)$. In this case, we have that $\text{Ext}(M) \subseteq U_j(W) \subseteq R_{j_1}(W')$. Since $B_j = B_{j_1}$, we have that, for all $x \in \text{Img}(h) \setminus \{0\}$, $Q_j^{h,f}(x) = Q_{j_1}^{h,f}(x)$ and, since $W' \cap B_j = W \cap B_{j_1}$, we have that $A_j^h(W) = A_{j_1}^h(W')$. Therefore, we have that $M \in Q_{j_1}^{h,f}(A_{j_1}^h(W'))$, which implies that $M \in C_{j_1}^{h,f}(W')$.

Thus, there is a function $c' \in C_{j_1}^{h,f}$ such that, for any $y \geq 0$ where $I_y \neq V(G_j)$ and $I_y \cap V(G_{j_1}) \subset I_{y+1} \cap V(G_{j_1})$, we have that $I_{y+1} \cap V(G_{j_1}) = I_{j_1}^{c'}(I_y \cap V(G_{j_1}))$. Hence, we can conclude that S_1 is a (h, f) -hull set of G_{j_1} . Analogously, we can also conclude that S_2 is a (h, f) -hull set of G_{j_2} .

Now, let us prove that if S_1 is a (h, f) -hull set of $V(G_{j_1})$ and S_2 is a (h, f) -hull set of $V(G_{j_2})$ then S is a (h, f) -hull set of $V(G_j)$. Thus, let $c_1 \in C_{j_1}^{h,f}$ and $c_2 \in C_{j_2}^{h,f}$ be functions such that $Y_{j_1}^{c_1}(S_1) = V(G_{j_1})$ and $Y_{j_2}^{c_2}(S_2) = V(G_{j_2})$. First, we have that $S \cap B_j = S_1 \cap B_j = \{w \in B_j \mid h(w) = 0\}$, which implies that $S \in \mathcal{Z}_0^{j,h}$. Let $I_y^1 = I_{j_1}^{c_1}(S_1, y)$ and $I_y^2 = I_{j_2}^{c_2}(S_2, y)$, for all $y \geq 0$.

Claim 3. *If there is a function $c \in C_j^{h,f}$ and an integer $z > 0$ such that, for every $y_1 \geq 0$ and $y_2 \geq 0$, where $I_{y_1}^1 \cup I_{y_2}^2 \neq V(G_j)$, we have that $I_{y_1+x_1(y_1)}^1 \cup I_{y_2+x_2(y_2)}^2 \subseteq I_j^c(I_{y_1}^1 \cup I_{y_2}^2, z_{y_1}^{y_1})$, where $K_j^h(I_{y_1+x_1(y_1)}^1 \cup I_{y_2+x_2(y_2)}^2) = K_j^h(I_j^c(I_{y_1}^1 \cup I_{y_2}^2))$ and:*

- If $c_1(I_{y_1}^1) \in R_{j_1}(I_{y_1}^1)$, $x_1(y_1) = 1$ and $x_2(y_2) = 0$,
- If $c_1(I_{y_1}^1) \notin R_{j_1}(I_{y_1}^1)$ and $c_2(I_{y_2}^2) \in R_{j_2}(I_{y_2}^2)$, $x_1(y_1) = 0$ and $x_2(y_2) = 1$,
- If $c_1(I_{y_1}^1) \in Q_{j_1}^{h,f}(A_{j_1}^h(I_{y_1}^1))$, $\text{Ext}(c_1(I_{y_1}^1)) \subseteq U_{j_1}(I_{y_1}^1)$, $c_2(I_{y_2}^2) \in Q_{j_2}^{h,f}(A_{j_2}^h(I_{y_2}^2))$, $\text{Ext}(c_2(I_{y_2}^2)) \subseteq U_{j_2}(I_{y_2}^2)$ and $K_{j_1}^h(I_{y_1}^1) = K_{j_2}^h(I_{y_2}^2)$, $x_1(y_1) = x_2(y_2) = 1$,

then, we have that $Y_j^c(S) = V(G_j)$.

Claim's proof. For any $W \subseteq V(G_j)$, let us say that $L(W)$ is true if and only if there is a function $c' \in C_j^{h,f}$ such that $Y_j^{c'}(W) = V(G_j)$. For every $y_1 \geq 0$ and $y_2 \geq 0$ such that $I_{y_1}^1 \cup I_{y_2}^2 \neq V(G_j)$ and $K_{j_1}^h(I_{y_1}^1) = K_{j_2}^h(I_{y_2}^2)$, we have that $L(I_j^c(I_{y_1}^1 \cup I_{y_2}^2, z_{y_1}^{y_1})) \implies L(I_{y_1}^1 \cup I_{y_2}^2)$ because, if there is a function $d \in C_j^{h,f}$ such that $Y_j^d(I_j^c(I_{y_1}^1 \cup I_{y_2}^2)) = V(G_j)$ then the function c' , where $c'(I_j^c(I_{y_1}^1 \cup I_{y_2}^2, i)) = c(I_j^c(I_{y_1}^1 \cup I_{y_2}^2, i))$, for each $0 \leq i < z_{y_1}^{y_1}$, and, for all the other sets W , $c'(W) = d(W)$, is in $C_j^{h,f}$, since both d and c are in $C_j^{h,f}$, and $Y_j^{c'}(I_{y_1}^1 \cup I_{y_2}^2) = V(G_j)$.

Also, by Lemma 5, we have that $L(I_{y_1+x_1(y_1)}^1 \cup I_{y_2+x_2(y_2)}^2) \implies L(I_j^c(I_{y_1}^1 \cup I_{y_2}^2, z))$. Thus, we have that, for every $y_1 \geq 0$ and $y_2 \geq 0$ such that $I_{y_1}^1 \cup I_{y_2}^2 \neq V(G_j)$ and $K_{j_1}^h(I_{y_1}^1) = K_{j_2}^h(I_{y_2}^2)$, $L(I_{y_1+x_1(y_1)}^1 \cup I_{y_2+x_2(y_2)}^2) \implies L(I_{y_1}^1 \cup I_{y_2}^2)$.

Let $V(G_{j_1}) = I_{e_{y_1}}^1$ and $V(G_{j_2}) = I_{e_{y_2}}^2 = V(G_j)$. Now, let us prove that $L(V(G_j)) \implies L(S)$. To do that, it is enough to prove that the sequence of pairs $(z_1(i), z_2(i))$, for $i \geq 0$, here $z_1(0) = 0$, $z_2(0) = 0$ and, for any $y > 0$, $z_1(y) = z_1(y-1) + x_1(z_1(y-1))$ and $z_2(y) = z_2(y-1) + x_2(z_2(y-1))$, is such that there is a value a such that $z_1(a) = e_{y_1}$ and $z_2(a) = e_{y_2}$ and, for each pair $(z_1(i), z_2(i))$ in the sequence, we have that $c_1(I_{z_1(i)}^1)$ and $c_2(I_{z_2(i)}^2)$ falls in one of the three cases in the statement of this claim.

This is because, if we prove that, then we would have that

$$L(I_{z_1(a)}^1 \cup I_{z_2(a)}^2) \implies L(I_{z_1(a-1)}^1 \cup I_{z_2(a-1)}^2) \implies \dots \implies L(I_{z_1(0)}^1 \cup I_{z_2(0)}^2)$$

which, since $V(G_j) = I_{z_1(a)}^1 \cup I_{z_2(a)}^2$ and $S = I_{z_1(0)}^1 \cup I_{z_2(0)}^2$, implies that $L(V(G_j)) \implies L(S)$.

So, since $K_{j_1}^h(I_0^1) = K_{j_2}^h(I_0^2)$, by a simple argument of induction on y , we have that $K_{j_1}^h(I_{z_1(y)}^1) = K_{j_2}^h(I_{z_2(y)}^2)$, for any $y \geq 0$. Thus, we cannot have that, for any $y \geq 0$ such that $I_{z_1(y)}^1 \cup I_{z_2(y)}^2 \neq V(G_j)$, $\text{Int}(c_1(I_{z_1(y)}^1)) = \emptyset$ and $c_2(I_{z_2(y)}^2) \in Q_{j_2}^{h,f}(A_{j_2}^h(I_{z_2(y)}^2))$ or vice-versa because that would imply that $K_{j_1}^h(I_{z_1(y)}^1) \neq K_{j_2}^h(I_{z_2(y)}^2)$. Also, we cannot have that $\text{Int}(c_1(I_{z_1(y)}^1)) = \text{Int}(c_2(I_{z_2(y)}^2)) = \emptyset$ because that would imply that $I_{z_1(y)}^1 \cup I_{z_2(y)}^2 = V(G_j)$. Thus, $c_1(I_{z_1(i)}^1)$ and $c_2(I_{z_2(i)}^2)$ falls in one of the three cases in the statement of this claim.

Since, for any y , we have that $z_1(y) + z_2(y) > z_1(y-1) + z_2(y-1)$, then there is a value a such that $V(G_j) = I_{z_1(a)}^1 \cup I_{z_2(a)}^2$.

Therefore, as discussed before, we have that we have that $L(V(G_j)) \implies L(S)$ and, since we have $L(V(G_j))$, then we have $L(S)$. \square

So, let us show a function $c \in C_j^{h,f}$ as stated of the Claim 3. let $W = I_{y_1}^1 \cup I_{y_2}^2$, for an arbitrary y_1 and y_2 such that $I_{y_1}^1 \cup I_{y_2}^2 \neq V(G_j)$. Let $c_1(I_{y_1}^1) = M_1$ and $c_2(I_{y_2}^2) = M_2$. Let us divide this proof in three cases.

The first case happens when $M_1 \in R_{j_1}(I_{y_1}^1)$. In this case, let $c(W) = M_1$. We have that $Int(M_1) \cap B_j = \emptyset$, since $B_j = B_{j_1}$. Also, since B_j separates $V(G_{j_1}) \setminus B_j$ from $V(G_{j_2}) \setminus B_j$ and $Int(M_1) \cap B_j = \emptyset$, we have that $Int_j(M_1) \cap I_{y_2}^2 = \emptyset$. Therefore, we have that $Int_j(M_1) \setminus W \neq \emptyset$, since $Int_j(M_1) \setminus I_{y_1}^1 \neq \emptyset$. Also, we have that $Ext(M_1) \subseteq I_{y_1}^1 \subseteq W$. Thus, we have that $M_1 \in R_j(W)$, which implies that $M_1 \in C_j^{h,f}(W)$. Furthermore, we have that $I_j^c(W) = W \cup Int_j(M_1) = I_{y_1+1}^1 \cup I_{y_2}^2$ and, thus, $K_j^h(I_{y_1+1}^1 \cup I_{y_2}^2) = K_j^h(I_j^c(I_{y_1}^1 \cup I_{y_2}^2))$.

The second case happens when $M_1 \notin R_{j_1}(I_{y_1}^1)$ and $M_2 \in R_{j_2}(I_{y_2}^2)$. In this case, let $c(W) = M_2$. Similarly to the first case, we have that $M_2 \in R_j(W)$, which implies that $M_2 \in C_j^{h,f}(W)$. Furthermore, we have that $I_j^c(W) = W \cup Int_j(M_2) = I_{y_1}^1 \cup I_{y_2+1}^2$ and, thus, $K_j^h(I_{y_1}^1 \cup I_{y_2+1}^2) = K_j^h(I_j^c(I_{y_1}^1 \cup I_{y_2}^2))$.

The last case happens when $c_1(I_{y_1}^1) \in Q_{j_1}^{h,f}(A_{j_1}^h(I_{y_1}^1))$, $Ext(c_1(I_{y_1}^1)) \subseteq U_{j_1}(I_{y_1}^1)$, $c_2(I_{y_2}^2) \in Q_{j_2}^{h,f}(A_{j_2}^h(I_{y_2}^2))$, $Ext(c_2(I_{y_2}^2)) \subseteq U_{j_2}(I_{y_2}^2)$ and $K_{j_1}^h(I_{y_1}^1) = K_{j_2}^h(I_{y_2}^2)$.

Since $K_{j_1}^h(I_{y_1}^1) = K_{j_2}^h(I_{y_2}^2)$ and $B_j = B_{j_1} = B_{j_2}$, we have that $K_{j_1}^h(I_{y_1}^1) = K_{j_2}^h(I_{y_2}^2) = K_j^h(W)$, which implies that $A_{j_1}^h(I_{y_1}^1) = A_{j_2}^h(I_{y_2}^2) = A_j^h(W)$. Let $z = A_j^h(W)$. Thus, we have that $Q_{j_1}^{h,f}(z) = Q_{j_2}^{h,f}(z) = Q_j^{h,f}(z)$. Let $c(W) = M_1$. We have that $M_1 \in Q_j^{h,f}(z)$.

Since $f(z) = Ext(M_1) \subseteq U_{j_1}(I_{y_1}^1)$ and $f(z) = Ext(M_2) \subseteq U_{j_2}(I_{y_2}^2)$, we have that $Ext(M_1) = Ext(M_2) \subseteq U_{j_1}(I_{y_1}^1) \cap U_{j_2}(I_{y_2}^2)$. Also, since $K_{j_1}^h(I_{y_1}^1) = K_{j_2}^h(I_{y_2}^2)$ and $I_{y_1}^1, I_{y_2}^2 \in \mathcal{Z}^{j,h}$, then $I_{y_1}^1 \cap B_j = I_{y_2}^2 \cap B_j$, which implies that $U_{j_1}(I_{y_1}^1) \cap U_{j_2}(I_{y_2}^2) = U_j(W)$. Then, we have that $Ext(M_1) \subseteq U_j(W)$. Hence, we can conclude that $M_1 \in C_j^{h,f}(W)$. Furthermore, we have that $W' = W \cup Int_j(M_1) \supseteq I_{y_1+1}^1 \cup I_{y_2}^2$. Note that $K_j^h(W) \subseteq W'$.

If $Int_j(M_2) \setminus W' = \emptyset$, we have that $W \cup Int_j(M_1) = W \cup Int_j(M_1) \cup Int_j(M_2) \supseteq I_{y_1+1}^1 \cup I_{y_2+1}^2$ and $K_j^h(I_j^c(I_{y_1}^1 \cup I_{y_2}^2)) = K_j^h(I_{y_1}^1 \cup I_{y_2}^2) \cup \{w \in B_j \mid h(w) = A_j^h(W)\} = K_j^h(I_{y_1+1}^1 \cup I_{y_2+1}^2)$.

So, suppose that $Int_j(M_2) \setminus W' \neq \emptyset$. For any $q, q' \in Int(M)$, let $M_2^{q,q'}$ be the subpath of M_2 such that $Ext(M_2^{q,q'}) = \{q, q'\}$. For any such subpath $M_2^{q,q'}$ such that $q, q' \in Ext(M_2) \cup K_j^h(W)$, we have that $Ext(M_2^{q,q'}) \subseteq U_j(W')$ as $K_j^h(W) \subseteq W'$. Also, for any two such subpaths $M_2^{q_1,q'_1}$ and $M_2^{q_2,q'_2}$, since they are subpaths of M_2 and M_2 is a minimum path, we have that $Int(M_2^{q_1,q'_1}) \cap Int(M_2^{q_2,q'_2}) = \emptyset$.

Let $O = \{q_1, q'_1, \{q_2, q'_2\}, \dots, \{q_z, q'_z\}$ be an arbitrary ordering of all sets with two vertices such that, for any $1 \leq i \leq z$, $q_i, q'_i \in Ext(M_2) \cup K_j^h(W)$, $Int(M_2^{q_i,q'_i}) \cap B_j = \emptyset$ and $Int_j(M_2^{q_i,q'_i}) \setminus W' \neq \emptyset$. Note that $\bigcup_{1 \leq i \leq z} Int_j(M_2^{q_i,q'_i}) \setminus W' = Int_j(M_2) \setminus W' \supseteq Int_{j_2}(M_2) \setminus W'$ and, since $Int_j(M_2) \setminus W' \neq \emptyset$, then z is at least one.

Let $W_1 = W'$ and, for all $1 \leq i \leq z$, $W_{i+1} = W_i \cup Int_j(M_2^{q_i,q'_i})$ and $c(W_i) = M_2^{q_i,q'_i}$. Since $Int(M_2^{q_i,q'_i}) \cap Int(M_2^{q_{i'},q'_{i'}}) = \emptyset$, for any $1 \leq i \neq i' \leq z$, we have that $Int_j(M_2^{q_i,q'_i}) \setminus W_i \neq \emptyset$. Thus, for any $1 \leq i \leq z$ since $Int(M_2^{q_i,q'_i}) \cap B_j = \emptyset$ and $\{q_i, q'_i\} \subseteq Ext(M_2) \cup K_j^h(W) \subseteq W' \subseteq W_i$, we have that $M_2^{q_i,q'_i} \in R_j(W_i)$, which implies that $M_2^{q_i,q'_i} \in C_j^{h,f}(W_i)$. Also, we have that $I_j^c(W, z+1) = W' \cup \bigcup_{1 \leq i \leq z} Int_j(M_2^{q_i,q'_i}) = W' \cup Int_j(M_2) \supseteq I_{y_1+1}^1 \cup I_{y_2+1}^2$ and, since $Int(M_2^{q_i,q'_i}) \cap B_j = \emptyset$, for each $1 \leq i \leq z$, then $K_j^h(I_j^c(I_{y_1}^1 \cup I_{y_2}^2, z)) = K_j^h(I_{y_1}^1 \cup I_{y_2}^2) \cup \{w \in B_j \mid h(w) = A_j^h(W)\} = K_j^h(I_{y_1+1}^1 \cup I_{y_2+1}^2)$. \square

Theorem 4. *The Geodesic Hull Number of a graph can be computed in $O((tw+1)^{tw+6}n^{2tw+5})$ time.*

Proof. First, our algorithm shall compute the distance between all pairs of vertices in G . Since G is unweighted and undirected, and $|E(G)| \leq tw \cdot n$, our algorithm can compute the table in $O(tw \cdot n^2)$ time. Then, given a tree decomposition of G with width tw , our algorithm finds a nice tree decomposition of G with width tw and $O(tw \cdot n)$ nodes in $O(tw^2 \cdot n)$ time.

Then it will compute the values in all tables M_j . For each $j \in T$, there are $O((tw+1)^{tw+2})$ functions in H_j and, for each $h \in H_j$, there are $O(n^{2tw+2})$ functions in $F_{j,h}$. Given an index (h, f) of a table M_j , we take $O((tw+1)^2)$ time to check if (h, f) is valid and, in the worst case, which is achieved when j is a forget node, we take $O((tw+1) \cdot n^2)$ time to actually compute $M_j(h, f)$. Thus, to compute a table M_j , we take $O((tw+1)^{tw+5}n^{2tw+4})$ time. Since there are $O(tw \cdot n)$ tables to compute, our algorithm takes $O((tw+1)^{tw+6}n^{2tw+5} + tw \cdot n^2 + tw^2 \cdot n)$ to compute the table M_r and find the minimum value there. \square

[1] Albenque N.; Knauer K.: Convexity in partial cubes: The hull number. Discrete Mathematics 339 (2), 866–876 (2016).

[2] Araujo, J.; Campos, V.; Giroire, F.; Sampaio, L.; Soares, R.: On the hull number of some graph classes. Theoretical Computer Science 475, 1–12 (2013).

- [3] Araujo, J.; Morel, G.; Sampaio, L.; Soares, R.; Weber, V.: Hull number: P5-free graphs and reduction rules. *Discrete Applied Mathematics* 210, 171–175 (2016).
- [4] Araújo, R. T.; Sampaio, R. M.; Szwarcfiter J. L. The convexity of induced paths of order three. *Electronic Notes in Discrete Mathematics* 44, 109–114 (2013).
- [5] Babel, L., Olariu, S.: On the structure of graphs with few P_4 's. *Discrete Applied Mathematics* 84, 1–13 (1998).
- [6] Benevides, F.; Przykucki, M.: On slowly percolating sets of minimal size in bootstrap percolation. *The Electronic Journal of Combinatorics* 20 (2), 46p (2013).
- [7] Benevides, F.; Przykucki, M.: Maximum percolation time in two-dimensional bootstrap percolation. *SIAM Journal on Discrete Mathematics* 29 (1), 224–251 (2015).
- [8] Ben-Zwi, O., Hermelin, D., Lokshtanov, D., Newman, I.: Treewidth governs the complexity of target set selection. *Discrete Optimization* 8 (1), 87–96 (2011). *Parameterized Complexity of Discrete Optimization*.
- [9] Bodlaender, H.L.: Some classes of graphs with bounded treewidth. *Bulletin of the EATCS* 36, 116–126 (1988).
- [10] Calder, J.: Some elementary properties of interval convexities. *Journal of the London Mathematical Society* 3, 422–428 (1971).
- [11] Campos, V., Sampaio, R.M., Silva, A., Szwarcfiter, J.L.: Graphs with few P_3 's under the convexity of paths of order three. *Discrete Applied Mathematics* 192, 28–39 (2015).
- [12] Centeno, C.C., Dourado, M.C., Penso, L.D., Rautenbach, D., Szwarcfiter, J.L.: Irreversible conversion of graphs. *Theoretical Computer Science* 412 (29), 3693–3700 (2011).
- [13] Chen, N.: On the Approximability of Influence in Social Networks. *SIAM Journal on Discrete Mathematics* 23 (3), 1400–1415 (2009).
- [14] Coelho, E.M.M., Dourado, M.C., Sampaio, R.M.: Inapproximability results for graph convexity parameters. *Theoretical Computer Science* 600, 49–58 (2015).
- [15] Dourado, M.C., Gimbel, J.G., Kratochvíl, J., Protti, F., Szwarcfiter, J.L.: On the computation of the hull number of a graph. *Discrete Mathematics* 309 (18), 5668–5674 (2009). *Combinatorics 2006, A Meeting in Celebration of Pavol Hell's 60th Birthday (May 1-5, 2006)*.
- [16] Dourado, M. C.; Protti F.; Rautenbach D.; Szwarcfiter J. L.: On the hull number of triangle-free graphs. *SIAM Journal on Discrete Mathematics* v. 23 (4), 2163–2172 (2010).
- [17] Dourado, M. C.; Penso, L. D.; Rautenbach, D.: On the geodetic hull number of P_k -free graphs. *Theoretical Computer Science* 640, 52–60 (2016).
- [18] Dourisboure, Y., Gavaille, C.: Tree-decompositions with bags of small diameter. *Discrete Mathematics* 307 (16), 2008–2029 (2007).
- [19] Dourisboure, Y., Gavaille, C.: Spanners for bounded tree-length graphs. *Theoretical Computer Science* 383 (1), 34–44, (2007).
- [20] Downey, R. G.; Fellows, M. R.: *Parameterized Complexity*. Springer (2012).
- [21] Duchet, P.: Convex sets in graphs, II. Minimal path convexity. *Journal of Combinatorial Theory B* 44, 307–316 (1988).
- [22] Erdős, P., Fried, E., Hajnal, A., Milner, E.C.: Some remarks on simple tournaments. *Algebra Universalis* 2, 238–245 (1972).
- [23] Everett, M.; Seidman, S.: The hull number of a graph. *Discrete Mathematics* 57, 217–223 (1985).
- [24] Farber, M., Jamison, R.E.: Convexity in graphs and hypergraphs. *SIAM Journal on Algebraic Discrete Methods* 7, 433–444 (1986).
- [25] Fellows, M. R., Hermelin, D., Rosamond, F., Vialette, S.: On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410 (1), 53–61 (2009).
- [26] Flum, J., Grohe, M.: *Parameterized Complexity Theory*. Springer (2010).

- [27] Impagliazzo, R., Paturi, R.: On the complexity of k-sat. *Journal of Computer and System Sciences* 62, 367–375 (2001).
- [28] Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity. *Journal of Computer and System Sciences* 63 (4), 512–530 (2001).
- [29] Lampis, M.: Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica* 64, 19–37 (2012).
- [30] Levi, F.W.: On Helly’s theorem and the axioms of convexity. *Journal Indian Mathematical Society* 15, 65–76 (1951).
- [31] Lokshtanov, D.: On the complexity of computing treelength. *Discrete Applied Mathematics* 158 (7), 820–827 (2010).
- [32] Lokshtanov, D., Marx, D., Saurabh, S.: Lower bounds based on the Exponential Time Hypothesis. *Bulletin of the EATCS* 105, 41–72 (2011).
- [33] Marx, D.: On the optimality of planar and geometric approximation schemes. In: 48th Symposium on Foundations of Computer Science (FOCS’07), Providence, Rhode Island, USA (2007). *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, 338–348 (2007).
- [34] Moon, J.: Embedding tournaments in simple tournaments. *Discrete Mathematics* 2 (4), 389–395 (1972).
- [35] Nichterlein, A.; Niedermeier, R.; Uhlmann, J.; Weller, M.: On tractable cases of Target Set Selection. *Social Network Analysis and Mining* 3 (2), 233–256 (2013).
- [36] Penso, L. D.; Protti, F.; Rautenbach, D.; Souza, U. S.: Complexity analysis of P_3 -convexity problems on bounded-degree and planar graphs. *Theoretical Computer Science* 607, 83–95 (2015).
- [37] Robertson, N., Seymour, P.: Graph minors III: Planar tree-width. *Journal of Combinatorial Theory B* 36 (1), 49–64 (1984).
- [38] Robertson, N., Seymour, P.: Graph minors II: Algorithmic aspects of tree-width. *Journal of Algorithms* 7 (3), 309–322 (1986).
- [39] Rockafellar, R. T. *Convex analysis*. Princeton University Press (1970).
- [40] Van de Vel, M.L.J.: *Theory of Convex Structures*. North-Holland, Amsterdam (1993).