UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS

DEPARTAMENTO DE COMPUTAÇÃO

MESTRADO E DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO (MDCC)

MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

CICERO SAMUEL SANTOS MORAIS

RESTRICTING INFECTIONS ON GRAPHS: IMMUNIZATION PROBLEMS

FORTALEZA

2025

CICERO SAMUEL SANTOS MORAIS

RESTRICTING INFECTIONS ON GRAPHS: IMMUNIZATION PROBLEMS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Mestrado e Doutorado em Ciência da Computação (MDCC) do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Algoritmos e Otimização.

Orientadora: Prof.ª Dr.ª Ana Karolinna Maia

Coorientador: Prof. Dr. Carlos Vinícius G. C. Lima

FORTALEZA

2025

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

M825r Morais, Cicero Samuel Santos.
Restricting Infections on Graphs : Immunization Problems / Cicero Samuel Santos Morais. – 2025.
74 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2025.
Orientação: Profa. Dra. Ana Karolinna Maia.
Coorientação: Prof. Dr. Carlos Vinícius Gomes Costa Lima.

1. Algoritmos. 2. Imunização. 3. Complexidade Parametrizada. 4. Processos Irreversíveis. 5. Processos Dinâmicos em Grafos. I. Título.

CDD 005

CICERO SAMUEL SANTOS MORAIS

RESTRICTING INFECTIONS ON GRAPHS: IMMUNIZATION PROBLEMS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Mestrado e Doutorado em Ciência da Computação (MDCC) do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Algoritmos e Otimização.

Aprovada em: 22 de abril de 2025

BANCA EXAMINADORA

Prof.ª Dr.ª Ana Karolinna Maia (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Dr. Carlos Vinícius G. C. Lima (Coorientador)
Universidade Federal do Cariri (UFCA)

Prof. Dr. Rudini Menezes Sampaio
Universidade Federal do Ceará (UFC)

Prof. Dr. Nicolas Almeida Martins
Universidade da Integração Internacional da
Lusofonia Afro-Brasileira (UNILAB)

AGRADECIMENTOS

Hoje, o dia em que eu escrevo estes agradecimentos, foi um dia muito cheio. Dias como este têm sido comuns, especialmente neste último ano. Porém, apesar de eu saber que não dá para ir muito longe sozinho, o mestrado tem me mostrado isso na prática. Esta dissertação só está pronta agora porque eu tive a ajuda de muitas pessoas incríveis.

Agradeço a meus pais por sempre me apoiar. Obrigado por sempre priorizarem a minha educação, não medindo esforços para que eu consiga atingir o que eu busco. Obrigado por todo o suporte dado em todas as áreas da minha vida e por terem me dado a estrutura necessária para que eu chegasse até aqui.

Obrigado à minha noiva Andreia que acompanhou todo o processo até chegar aqui e que me apoiou desde quando o ingresso no mestrado era apenas uma ideia. Agradeço por sempre me incentivar a buscar o que eu quero e por tornar fáceis os dias difíceis através de uma simples conversa. Obrigado também por sempre me mostrar os momentos em que eu precisava parar um pouco e descansar e por aguentar a saudade e a distância durante este período.

Agradeço muito aos amigos que o Pargo me deu. À Natália e ao Jota, por terem encarado praticamente as mesmas disciplinas que eu, passando por todas as coisas boas e por todas as dificuldades que elas nos proporcionaram juntos comigo. Obrigado também por terem tornado este processo tão mais fácil, me ajudando sempre. Ao Brito que, apesar de não ter feito nenhuma disciplina conosco, virou membro indispensável do nosso grupo. Vocês três não fazem ideia de como tornaram este ano mais leve e mais prazeroso. Além de tudo isso, obrigado por todas as conversas que tivemos na sala de convivência, por todos os eventos e momentos vividos juntos, e pelas discussões de problemas uns dos outros.

À Karol, por ter me acolhido como seu aluno e por ter sido a melhor orientadora que alguém poderia pedir. Obrigado por todas as reuniões, as discussões, os eventos, as conversas e as risadas. Obrigado por ter escutado e entendido minhas reclamações de desconforto e solidão no início do mestrado e por ter sido sempre compreensiva. Obrigado mais ainda por ter orientado não só este trabalho, mas me orientado também neste período de mudanças e adaptação. Ao Vinícius, por ter aceitado continuar me orientando à distância e algumas vezes presencialmente também, mesmo durante a greve ou férias. A vocês dois, obrigado por todos os ensinamentos e por terem aceitado correr nesse mestrado junto comigo.

Ao Pargo e a todos os seus professores, em especial a Karol e o Vinícius que me orientaram e aos professores com os quais tive aulas neste ano: Cláudia, Júlio, Manoel, Rafael,

Rudini e Victor. Obrigado a todos vocês por terem criado um laboratório tão acolhedor. Obrigado também por não aderirem ao esteriótipo conhecido de professores de exatas carrascos, durões e fechados. Obrigado por todas as aulas, eventos, seminários, conversas, caronas e brincadeiras.

“Não sou nada. [...]

À parte isso, tenho em mim todos os sonhos do mundo.”

(Fernando Pessoa)

RESUMO

Considere um grafo G em que $c_v(\tau) \in \{0,1\}$ denota o estado do vértice $v \in V(G)$ em um dado tempo $\tau \in \mathbb{N}$. Se $c_v(\tau) = 0$, dizemos que o vértice v está inativo ou não-infectado no tempo τ; caso contrário, dizemos que v está ativo ou infectado no tempo τ. Uma coleção de estados $C_\tau = (c_v(\tau))_{v \in V(G)}$ é dita uma configuração de G no tempo τ. Uma sequência de configurações $\mathscr{P} = (C_\tau)_{\tau \in \mathbb{N}}$ de G é chamada de um processo dinâmico discreto em G. Dados um grafo G, uma função $t : V(G) \to \mathbb{N}$ que atribui a cada vértice v de G um valor chamado de limiar de v, e um conjunto $S \subseteq V(G)$ de vértices inicialmente infectados, dizemos que $\mathscr{P} = I_t(G,S)$ é um processo t-irreversível em G se $\mathscr{P}$ é um processo dinâmico discreto em G tal que, para todo $v \in V(G)$, temos que $c_v(0) = 1$ se e somente se $v \in S$ e $c_v(\tau+1) = 1$ se e somente se $|\{u \in N_G(v) \mid c_u(\tau) = 1\}| \geq t(v)$. Ou seja, em um processo t-irreversível $\mathscr{P} = I_t(G,S)$, iniciamos com todos os vértices de S infectados no tempo $\tau = 0$ e, a cada passo de tempo sucessivo, um vértice não-infectado v torna-se infectado se tiver pelo menos seu limiar $t(v)$ de vizinhos infectados no passo de tempo anterior. Além disso, uma vez que um vértice é infectado, ele permanece infectado. Dizemos que o processo termina quando mais nenhum vértice pode ser infectado. Processos irreversíveis são utilizados para modelar diversos fenômenos, entre eles: difusão de (des-)informações e doenças contagiosas. Processos t-irreversíveis são bastante estudados na literatura principalmente com o objetivo de encontrar um conjunto de vértices inicialmente infectados que satisfaça algum critério. Este critério geralmente é maximizar o número de vértices infectados; ou minimizar o tempo necessário para infectar todos os vértices; entre outros similares. Porém, em contextos de contágio e difusão como os citados acima, torna-se natural pensar em como conter a infecção, ou seja, restringir o número de vértices infectados no fim do processo a um conjunto pequeno. Fazemos isto através do que chamamos de imunização de vértices. Um vértice imunizado não pode ser infectado e não contribui para infectar outros vértices. (CORDASCO *et al.*, 2023) introduziu o problema INFLUENCE IMMUNIZATION BOUNDING (IIB) em que, dado um processo t-irreversível $\mathscr{P} = I_t(G,S)$ e naturais k e ℓ, o objetivo é encontrar um conjunto $Y \subseteq V(G)$ de vértices a serem imunizados tal que $|Y| \leq \ell$ e o número de vértices infectados ao fim do processo é no máximo k. Dizemos que Y é um conjunto k-restritor. No mesmo artigo, os autores mostraram que o problema é W[1]-difícil e W[2]-difícil parametrizado por alguns parâmetros, entre eles k, ℓ, a largura em árvore do grafo e a diversidade de vizinhança do grafo. Eles também mostraram alguns algoritmos FPT parametrizados por combinações destes parâmetros. Neste trabalho, nós estudamos IIB em diversas classes de grafos. Para grafos bipartidos e split, nós mostramos que o problema continua W[2]-difícil parametrizado por ℓ. Mostramos também que IIB é NP-completo mesmo em grafos planares bipartidos subcúbicos. Para árvores, conjecturamos que o problema também permanece NP-completo. Para grafos completos em que os limiares de todos os vértices são iguais, mostramos como encontrar um conjunto k-restritor mínimo. Mostramos também alguns limitantes superiores para o tamanho de um conjunto k-restritor de caminhos, árvores, grafos planares e exoplanares e outras classes de grafos quando k é suficientemente grande e o subgrafo induzido por S é conexo.

Palavras-chave: Algoritmo; Imunização; Complexidade Parametrizada; Processos Irreversíveis; Processos Dinâmicos em Grafos; Teoria dos Grafos.

ABSTRACT

Consider a graph G in which $c_v(\tau) \in \{0,1\}$ denotes the state of the vertex $v \in V(G)$ at a given time $\tau \in \mathbb{N}$. If $c_v(\tau) = 0$, we say that vertex v is inactive or uninfected at time τ; otherwise, we say that v is active or infected at time τ. A collection of states $C_\tau = (c_v(\tau))_{v \in V(G)}$ is said to be a configuration of G at time τ. A sequence of configurations $\mathscr{P} = (C_\tau)_{\tau \in \mathbb{N}}$ of G is called a discrete dynamic process on G. Given a graph G, a function $t : V(G) \to \mathbb{N}$ that assigns to each vertex v of G a value called the threshold of v, and a set $S \subseteq V(G)$ of initially infected vertices, we say that $\mathscr{P} = I_t(G,S)$ is a t-irreversible process in G if $\mathscr{P}$ is a discrete dynamical process in G such that, for all $v \in V(G)$, we have that $c_v(0) = 1$ if and only if $v \in S$ and $c_v(\tau+1) = 1$ if and only if $|\{u \in N_G(v) \mid c_u(\tau) = 1\}| \geq t(v)$. In other words, in a t-irreversible process $\mathscr{P} = I_t(G,S)$, we start with all the vertices of S infected at time $\tau = 0$ and, at each successive time step, an uninfected vertex v becomes infected if it has at least its threshold $t(v)$ of neighbors infected at the previous time step. Furthermore, once a vertex is infected, it remains infected. We say that the process ends when no more vertices can be infected. Irreversible processes are used to model various phenomena, including the spread of (dis-)information and contagious diseases. Irreversible t-processes are widely studied in the literature, mainly with the aim of finding a set of initially infected vertices that satisfies some criterion. This criterion is usually to maximize the number of infected vertices; or to minimize the time needed to infect all vertices; among other similar ones. In contexts of contagion and diffusion such as those mentioned above, however, it becomes natural to think about how to contain the infection, that is, to restrict the number of infected vertices at the end of the process to a small set. We do this through what we call vertex immunization. An immunized vertex cannot be infected and does not contribute to infecting other vertices. (CORDASCO *et al.*, 2023) introduced the problem INFLUENCE IMMUNIZATION BOUNDING (IIB) in which, given a t-irreversible process $\mathscr{P} = I_t(G,S)$ and naturals k and ℓ, the goal is to find a set $Y \subseteq V(G)$ of vertices to be immunized such that $|Y| \leq \ell$ and the number of infected vertices at the end of the process is at most k. We say that Y is a k-restricting set. In the same paper, the authors showed that the problem is W[1]-hard and W[2]-hard parameterized by some parameters, including k, ℓ, the treewidth of the graph and the neighborhood diversity of the graph. They also showed some FPT algorithms parameterized by combinations of these parameters. In this work, we study IIB on several classes of graphs. For bipartite and split graphs, we show that the problem remains W[2]-hard parameterized by ℓ. We also show that IIB is NP-complete even for subcubic bipartite planar graphs. For trees, we conjecture that the problem also remains NP-complete. For complete graphs where the thresholds of all vertices are equal, we show how to find a minimum k-restricting set. We also show some upper bounds for the size of a k-restricting set for paths, trees, planar and outerplanar graphs and other classes of graphs when k is large enough and the subgraph induced by S is connected.

Keywords: Algorithms; Dynamical Processes on Graphs; Graph Theory; Immunization; Irreversible Processes; Parameterized Complexity.

LIST OF FIGURES

CONTENTS

1 INTRODUCTION

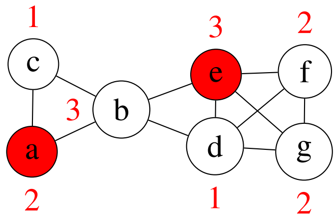
What our acquaintances and other people around us do or think influences us. We can think of several situations in which that happens. Banerjee (1992) has listed some of them: investors choose their assets based on what their fellow investors are investing (KEYNES, 1937); voters tend to be influenced by opinion polls predictions (CUKIERMAN, 1991); people tend to adopt new technologies or innovations if their colleagues are adopting them (RYAN; GROSS, 1950; COLEMAN *et al.*, 1957; GRANOVETTER, 1978); academics tend to research topics that are "hot" in their communities; we often choose which restaurant to go to or what school to attend based on how popular they are or on what our friends and family think of them; our opinions tend to be similar to those of the people we trust.

These phenomena can be seen as a cascading effect. A non-influenced individual is influenced by his contacts and starts to influence other contacts, which will influence others, and so on. Other contagion-like events can also be seen in this framework. For example, we can contract a contagious disease by coming into contact with an infected individual. Or we can get a computer virus by receiving an infected packet or file from some computer in our network.

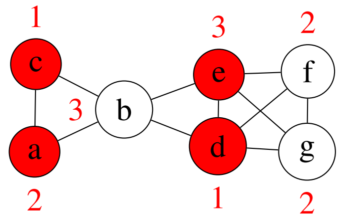
That is the point: individuals or things connected in a network can influence each other. And they can be connected by several ways. Two people can be connected because they are friends. Or they can be connected because they live in the same neighborhood. Two computers can be connected because they share the same router. And the structure of the network – the graph – formed by these connections determines how the contagious agent will spread, whether it is a piece of information, an innovation, a virus, or something else. As evidence of this, we can cite a recent and familiar real-world event: the Covid-19 pandemic. In (COLIZZA *et al.*, 2006), the authors show that the airline transportation network structure provided reliable information to predict how the virus would spread.

To model networks such as those we have mentioned, we are going to use a simple and undirected graph $G = (V,E)$. The vertices in $V(G)$ will represent the subjects. The connections between two distinct subjects will be represented by the edges in $E(G)$. Each vertex will belong to one of two states: either active (infected) or inactive (uninfected). To each vertex $v \in V(G)$, we assign a natural value $t(v)$, called vertex threshold. This value represents how susceptible a vertex is to be infected by its contacts. More specifically, $t(v)$ is the number of infected neighbors necessary for v to get infected. The infection then happens in discrete time steps synchronously, that is, all the infected vertices update their states at each time step.

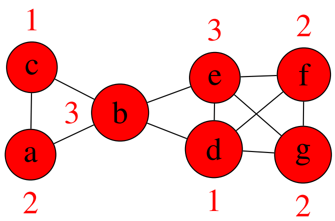
Starting with a given subset of initially infected vertices, $S \subseteq V(G)$, new vertices are getting infected. Once a vertex gets infected, it stays infected. The infection stops when there are no more susceptible vertices with enough infected neighbors to get infected. This is an irreversible threshold process (DREYER; ROBERTS, 2009), which we refer to as the *threshold model*. Figure 1 illustrates such a process.



(a) The set of initially infected vertices is $S = \{a, e\}$.



(b) At time step $\tau = 1$, vertices c and d get infected.

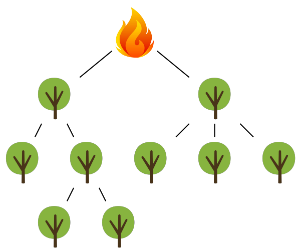


(c) At time step $\tau = 2$, the remaining vertices get infected.

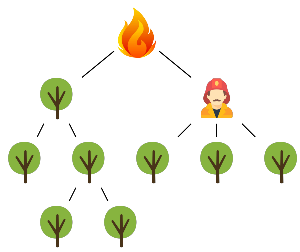
Figure 1 – Example of an infection diffusion in a graph with thresholds.
Source: Made by the author.

The threshold model and other contagion models have gained a lot of attention from several areas, such as Economics (BANERJEE, 1992; MORRIS, 2000); Marketing (GRANOVETTER, 1978; WANG; STREET, 2018); Epidemiology (BARABÁSI; PÓSFAI, 2016; NEWMAN, 2010); Physics (MELLO *et al.*, 2021); and, of course, Computer Science. Kempe *et al.* (2003) were the first to study problems related to the Threshold Model from the computational point of view. They introduced the problems INFLUENCE MAXIMIZATION and TARGET SET SELECTION. Given a graph $G = (V,E)$ and a threshold function $t : V(G) \to \mathbb{N}$, both problems consist of finding a small subset of initially infected vertices maximizing the number of infected vertices at the end of the process. In TARGET SET SELECTION the goal is to infect all the vertices. The motivations for both problems come from viral marketing.

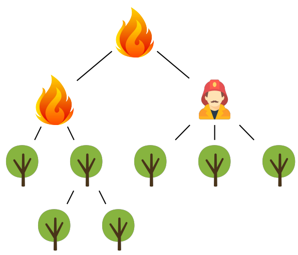
But rather than trying to influence or infect everyone, it is natural to think about how to contain the infection. A very studied problem in this direction is the FIREFIGHTER (FINBOW; MACGILLIVRAY, 2009; HARTNELL, 1995). In the classical version of this problem, we are given a graph $G = (V,E)$ and a vertex $s \in V(G)$. The vertex s is the source of the fire, and this fire will spread. One neighbor in fire is necessary for a vertex to catch fire too. In other words, the threshold of every vertex is equal to one. The problem works like a solitaire game. At each round, the fire spreads and we can choose a non-burning vertex to place a firefighter. Once we place a firefighter on a vertex, it is protected against the fire from the current round on. That is, the vertex with a firefighter on it cannot catch fire. The problem consists then in saving the maximum number of vertices from the fire. Figure 2 shows an example of FIREFIGHTER. Other variations of the problem are also studied, such as one in which the number of available firefighters at each round is strictly greater than one (BAZGAN *et al.*, 2013), or the one in which the goal is to protect a given vertex set $S \subseteq V(G)$ (KING; MACGILLIVRAY, 2010).



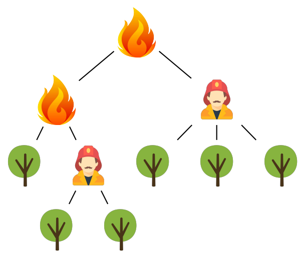
(a) In this example, the fire starts at the root of the tree.



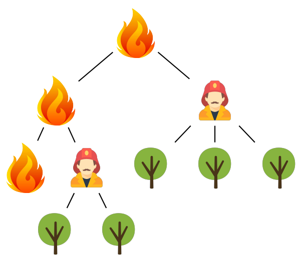
(b) In this case, when we place a firefighter in a vertex, all its descendants are protected.



(c) After we place a firefighter, the fire continues to spread to adjacent vertices.



(d) In round 2, we place a firefighter in another vertex.



(e) The game then ends when no more vertices can get burned.

Figure 2 – Example of FIREFIGHTER problem.
Source: Made by the author.

A very similar problem is the (k,ℓ)-INFLUENCE IMMUNIZATION BOUNDING, introduced by Cordasco *et al.* (2023). In this problem, we are also given a graph $G = (V,E)$ with threshold function $t : V(G) \to \mathbb{N}$ and a set of initially infected vertices $S \subseteq V(G)$. Just like in the FIREFIGHTER problem, the goal here is to choose a set of vertices to protect (the word we use is to immunize) so that the infection gets restricted to a small number of vertices. To immunize a vertex means that it cannot get infected, nor it cannot infect others. Notice that this is analogous to removing the vertex from the graph. There are some key differences from the FIREFIGHTER though. In this problem, the thresholds of the vertices do not need to be equal to 1. Another difference is that we do not have the game-like dynamics here. There are no rounds in which we choose vertices to immunize, we only choose a subset of vertices to immunize before the process begins and that is it.

Figure 3 shows an example of an entry for (k,ℓ)-IIB and a solution. The thresholds of the graph are shown in red and the set of initially infected vertices in black. The vertices highlighted in blue form a solution when $k = 3$ and $\ell = 2$. By immunizing $Y = \{c,h\}$ in the graph of the figure, the infection gets restricted to only the 3 already infected vertices.

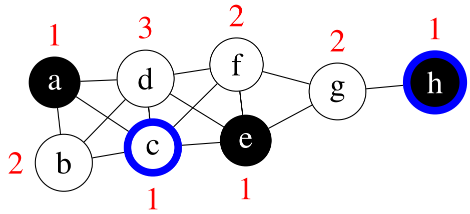


Figure 3 – Example of a entry and a solution for (k,ℓ)-INFLUENCE IMMUNIZATION BOUNDING with $k = 3$ and $\ell = 2$.

Problems like FIREFIGHTER and (k,ℓ)-INFLUENCE IMMUNIZATION BOUNDING have clear applications in stopping the spreading of (mis-)information, viruses, and other undesirable objects in a network.

The problems TARGET SET SELECTION, INFLUENCE MAXIMIZATION, FIREFIGHTER, and (k,ℓ)-INFLUENCE IMMUNIZATION BOUNDING mentioned above are all NP-hard. In this work, however, we are going to study the (k,ℓ)-INFLUENCE IMMUNIZATION BOUNDING problem. Cordasco *et al.* (2023) introduced the problem and studied it from the parameterized complexity point of view. We study the (k,ℓ)-INFLUENCE IMMUNIZATION BOUNDING in different graph classes.

For bipartite graphs and split graphs, we show that the (k,ℓ)-IIB remains W[2]-hard parameterized by the maximum number of vertices to be immunized, ℓ. We show that the problem is NP-complete even for subcubic bipartite planar graphs. All these results are for instances in which the threshold of every uninfected vertex is exactly equal to its degree and

$k = |S|$. They are presented in Chapter 4. For trees, we conjecture that the problem also remains NP-complete. In Chapter 5, we show how to solve (k,ℓ)-IIB exactly in some graph classes. We solve (k,ℓ)-IIB in complete graphs when all vertices have the same threshold value. In the case that $k = |S|$, which is called *total restriction*, we show how to solve the problem in trees with any thresholds and in general graphs when each vertex has threshold equal to 1. We also present upper bounds on the number of immunized vertices needed to restrict the infection for paths, trees, planar graphs, outerplanar graphs and other graph classes when k is large enough and the subgraph induced by S is connected. Chapters 2 and 3 are dedicated to the concepts of graph theory and complexity theory needed throughout this dissertation and to a review of the main results about irreversible processes and other related problems in graphs, respectively. Finally, we make our final remarks in Chapter 6, summarizing our results and pointing open questions and future work.

2 PRELIMINARIES

In this chapter, we present the basic concepts needed throughout this dissertation. In Section 2.1, we present the definitions concerning Graph Theory. Then, in Section 2.2, we talk about Computational Complexity and Parameterized Complexity.

2.1 Graph Theory

The majority of definitions in this section follow those from (WEST, 2000). Thus, we refer the reader to this reference to take more details about the concepts we introduce here. Other sources that may be consulted are (DIESTEL, 2000; BONDY; MURTY, 2008).

A *graph* $G = (V, E, \varphi)$ is a triple defined by a non-empty set of elements V called *vertices*, a set E of elements called *edges*, and an *incidence function* φ that maps each edge from E to an *unordered* and *not necessarily distinct* pair of vertices. Figure 4 illustrates a graph. Its vertex set is $V = \{w, x, y, z\}$ and its edge set is $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$. Its function φ maps e_1 to the pair of vertices $\{x, w\}$; e_2 to the same pair of vertices; e_3 to the pair $\{w, z\}$; and so on. If e is an edge of a graph $G = (V, E, \varphi)$ and φ maps e to a pair $u, v \in V$, we say that u and v are the *endpoints* of e; u and v are *neighbors* or *adjacent*; and e is *incident* to u and v. For example, in the graph of Figure 4, w and x are endpoints of the edges e_1 and e_2.

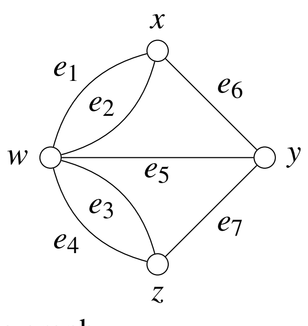


Figure 4 – Visual representation of a graph.
Source: Adapted from West (2000).

A *loop* is an edge e in which both endpoints are equal. Two edges e_1 and e_2 are *multiple edges* if their endpoints are equal. A graph is *simple* if it has no loops nor multiple edges. Throughout this work, we are going to deal only with simple graphs. Thus, from now on, unless stated otherwise, every time we refer to a graph, it is simple. Notice that, when we are dealing with simple graphs, we can omit the incidence function φ, writing each edge as its endpoints. If φ maps an edge e of a graph to vertices $\{u, v\}$, for example, we simply write $e = uv$. Thus, when we are referring to a simple graph, we write it as pair $G = (V,E)$, where V is its vertex set and E is a set of non-ordered pairs of vertices. In Figure 5, we exemplify a simple graph with vertex set $V = \{a,b,c,d,e,f,g\}$ and edges $E = \{ab,ag,bc,bg,cd,ce,de,eg,eg,fg\}$. We often write $V(G)$ and $E(G)$ to denote the vertex set and edge set of a graph G, respectively. The order of the graph is denoted by $n(G)$ or simply by n, and it is equal to the number of vertices $|V(G)|$. The number of edges $|E(G)|$ is denoted by $m(G)$, or simply by m.

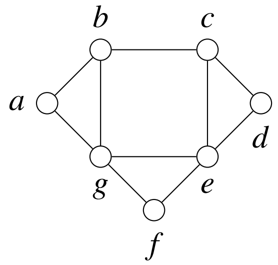


Figure 5 – Example of a simple graph $\mathscr{G}$.
Source: Made by the author.

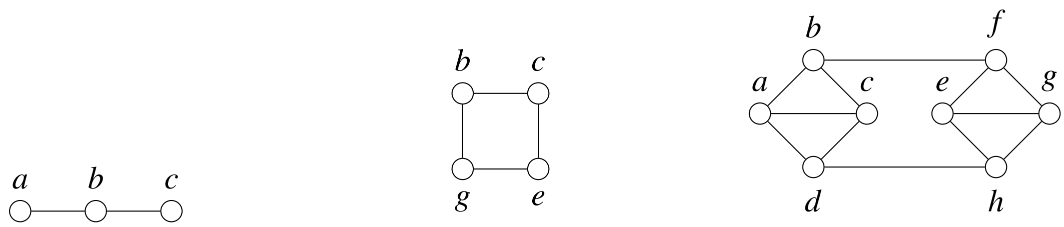
Given a graph G and a vertex $v \in V(G)$, the *degree* of v, denoted by $d_G(v)$, is the number of edges that are incident to v. Its *neighborhood*, denoted by $N_G(v)$ is the set of neighbors of v. The neighborhood of a set of vertices $S \subseteq V(G)$ of G is the union of the neighborhood of every vertex in S. The *closed neighborhood* of v in G, denoted by $N_G[v]$, is $N_G(v) \cup \{v\}$. Analogously, the closed neighborhood of a set of vertices $S \subseteq V(G)$ of G, denoted by $N_G[S]$, is $N_G(S) \cup S$. We denote by $\Delta(G)$ the *maximum degree* over all vertices of G. Similarly, we denote by $\delta(G)$ the *minimum degree* over all vertices of G. In Figure 5, the degree of a is $d_{\mathscr{G}}(a) = 2$ and its neighborhood is $N_{\mathscr{G}}(a) = \{b,g\}$, while $\Delta(\mathscr{G}) = 4$ and $\delta(\mathscr{G}) = 2$. A graph G is *k-regular* if $\Delta(G) = \delta(G) = k$, i.e., the degree of *all* vertices of G is equal to k. We say that a 3-regular graph is a *cubic graph* (Figure 6c).

A sequence of vertices $v_1, v_2, \ldots, v_k$ is a path of G if v_i is adjacent to v_{i+1}, for $i = 1, \ldots, k-1$, in G. We say then that v_1 and v_k are the *endpoints* of the path; $v_2, \ldots, v_{k-1}$ are the *internal vertices* of the path; and the path is a v_1, v_k-path. The *length* of a path is the number of edges in the path. The *distance* between two vertices u and v of a graph G, denoted by $d_G(u,v)$, is the minimum length of a u,v-path in G or $+\infty$ if such a path does not exist. The *diameter* of G, denoted by $\operatorname{diam}(G)$ is the maximum distance between any two vertices of G. A path graph is a graph such that its vertex set can be listed in an order $v_1, v_2, \ldots, v_k$ such that its edges are $v_i v_{i+1}$ for $i = 1, \ldots, k-1$. We denote a path graph of order k as P_k (Figure 6a).

A *subgraph* H of G is a graph such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. We denote it by $H \subseteq G$. A subgraph H of G is said to be a *spanning subgraph* of G if $V(H) = V(G)$. For a graph G and an edge $e \in E(G)$, we denote by $G - e$ the graph obtained by removing e from G. If $M \subseteq E(G)$, then $G - M$ is analogous. Similarly, for a graph G and a vertex $v \in V(G)$, we denote by $G - v$ the graph obtained by removing v and all the edges incident to v from G. If $S \subseteq V(G)$, then $G - S$ is analogous. A subgraph H of G is said to be *induced* if it can be obtained from G by sequentially removing vertices not in $S = V(H)$. In this case, we say that H is the subgraph of G induced by S and denote it by $H = G[S]$. The graph from Figure 6b, for example, is the subgraph of the graph $\mathscr{G}$ from Figure 5 induced by $\{b, c, e, g\}$.

A graph G is *connected* if, for every pair of vertices $s, t \in V(G)$, there is an s,t-path in G. If this condition is not met, then we say that G is *disconnected*. A maximal connected subgraph of G is called a *component* of G. A 2-regular connected graph is called a *cycle*, and we denote a cycle of order k by C_k (Figure 6b). If a cycle has an odd number of vertices, we say that it is an *odd cycle*, otherwise it is an *even cycle*.

Let x, y be two distinct vertices of a graph G such that x and y belongs to the same component of G. A set $S \subseteq V(G)$ is said to be a *x,y-separator* or simply a *separator* of G if x and y are in different components of $G - S$. Given $S, T \subseteq V(G)$, we denote by $[S, T]$ the set of edges of G with one endpoint in S and the other in T. If $T = \overline{S} = V(G) \setminus S$, we call $[S, \overline{S}]$ an *edge-cut* of G.



(a) A path graph of order 3. (b) A cycle graph of order 4. (c) A cubic graph.

Figure 6 – A P_3, a C_4, and a cubic graph.

Source: Made by the author.

A graph G is said to be *acyclic* or a *forest* if it has no cycle as a subgraph. A *connected* acyclic graph is called a *tree*. Path graphs, for example, are trees. Let T be a tree graph and $v \in V(T)$, we say that v is a *leaf vertex* if $d_T(v) = 1$ and that v is *internal vertex* otherwise. Trees are a widely studied class of graphs and they have several useful properties.

Theorem 2.1. *(West (2000)) Every tree with at least 2 vertices has at least 2 leaves.*

Theorem 2.2. *(West (2000)) Let T be a tree. Then, the following are equivalent:*

I. T is connected and has no cycles;

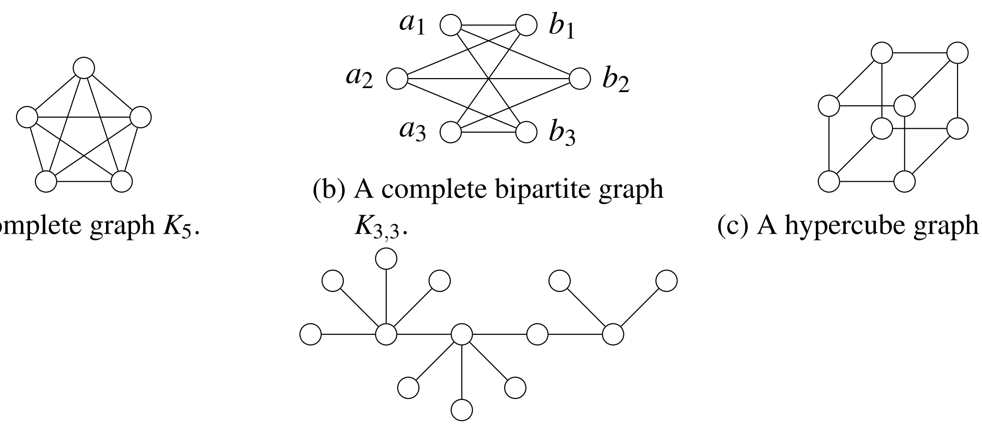
II. T is connected and has $n-1$ edges;

III. T has $n-1$ edges and no cycles;

IV. For every $u, v \in V(G)$, T has exactly one u, v-path.

A tree T is said to be a *caterpillar tree* if, by removing all the leaves of T, the remaining graph is a path graph. Figure 7d shows a caterpillar tree.

Given a graph G and a set $S \subseteq V(G)$, we say that S is an *independent set* of G if, for every $u, v \in S$, we have that $uv \notin E(G)$. In other words, there are no edges between any two vertices of S in G. The size of a maximum independent set of G is called the *independence number* of G, denoted by $\alpha(G)$. Similarly, we say that S is a *clique* of G if there is an edge between any two vertices of S in G. The size of a maximum clique of G is called the *clique number* of G, denoted by $\omega(G)$.



(a) A complete graph K_5.

(b) A complete bipartite graph $K_{3,3}$.

(c) A hypercube graph Q_3.

(d) A caterpillar tree.

Figure 7 – Some examples of other graph classes.
Source: Made by the author.

We say that G is a *complete graph* if $V(G)$ is a clique. We denote complete graphs on n vertices by K_n. Figure 7a shows a K_5 graph. A graph G is said to be a *bipartite graph* if it is possible to partite $V(G)$ into two independent sets A and B and, in this case, we say that (A, B) is the bipartition of G. The following theorem characterizes the bipartite graphs.

Theorem 2.3. *(König (1937)) A graph is bipartite if and only if it has no odd cycles.*

A bipartite graph G with bipartition (A, B) is said to be *complete* if, for every $u \in A$ and $v \in B$, we have that $uv \in E(G)$. We denote it by $K_{n,m}$ where $n = |A|$ and $m = |B|$ without loss of generality. Figure 7b shows a $K_{3,3}$ graph.

The *hypercube graph*, denoted by Q_n, is the simple graph whose vertices are all the binary n-tuples and the edges are the pairs of n-tuples that differ in exactly one position. Figure 7c shows a Q_3 graph.

Theorem 2.4. *(West (2000)) Q_n is n-regular and bipartite.*

Given a graph G, a set $D \subseteq V(G)$ is a *dominating set* of G if every vertex in $V(G) \setminus D$ has a neighbor in D. The size of a minimum dominating set of G is called the *domination number* of G, denoted by $\gamma(G)$. A set $S \subseteq V(G)$ is a *vertex cover* if, for every edge $e \in E(G)$, e has an endpoint in S. The *vertex cover number* of G, denoted by $\beta(G)$ is the size of a minimum vertex cover of G. A *matching* of a graph G is a subset of its edges $M \subseteq E(G)$ such that no edge in M share an endpoint. The *matching number* of G, denoted by $\alpha'(G)$, is the size of a maximum matching of G.

Theorem 2.5. *(West (2000)) If G is a bipartite graph, then $\alpha'(G) = \beta(G)$.*

Theorem 2.6. *(Edmond's Blossom Algorithm (WEST, 2000; EDMONDS, 1965)) Given any graph G, we can compute a maximum matching of G in polynomial time.*

A vertex v of a graph G is said to be a *simplicial vertex* if $N_G(v)$ is a clique of G. A *perfect elimination ordering* of G is an ordering $v_n, \dots, v_1$ of $V(G)$ such that v_i is a simplicial vertex of $G[\{v_1, \dots, v_i\}]$. A *chord* of G is an edge connecting two non-consecutive vertices of a cycle in G. A cycle of G is said to be *chordless* if it has no chords in G. A *hole* of a graph G is a chordless cycle of G with at least 4 vertices. A graph G is *chordal* if it has no holes, i.e., every cycle with at least 4 vertices have a chord in G.

A graph G is called a *split graph* if its vertex set can be partitioned into a clique and an independent set. Observe that split graphs and complete graphs are subclasses of chordal graphs. Chordal graphs are characterized by the existence of a perfect elimination ordering.

Theorem 2.7 (Dirac (1961))**.** *A graph G is chordal if and only if G has a perfect elimination ordering.*

The following properties of chordal graphs are also very useful.

Theorem 2.8 (Dirac (1961))**.** *If G is chordal, then any induced subgraph of G is also chordal.*

Theorem 2.9 (Dirac (1961))**.** *If G is a chordal graph and is not a complete graph, then G has at least two non-adjacent simplicial vertices.*

A graph G is said to be a *planar graph* if it can be drawn on the plane without crossing edges. The graphs from Figure 6, for example, are all planar graphs. The graphs K_5 and $K_{3,3}$ from Figures 7a and 7b, however, are not planar graphs. In fact, they are the minimal non-planar graphs and all planar graphs are characterized by the exclusion of them as minors, which we will define next.

(a) Graph G before the edge contraction. The contracted edge is uv, highlighted in red.

(b) Graph G/uv. The new vertex created by the contraction is highlighted in red.

Figure 8 – Example of edge contraction in a graph.

Source: Made by the author.

Given a graph G and an edge $e = uv \in E(G)$, the operation of *edge contraction*, denoted by G/e, is the operation of removing e from G and merging its endpoints u and v into a new vertex w whose neighborhood is $(N_G(u) \cup N_G(v)) \setminus \{u, v\}$. A graph H is a *minor* of a graph G if we can obtain a copy of H by deleting and/or contracting edges of G. Now, we can state the following characterization of planar graphs.

Theorem 2.10. *(Wagner (1937)) A graph G is planar if and only if G has no K_5 nor $K_{3,3}$ as minors.*

An *apex graph* is a graph G in which there is a vertex $v \in V(G)$, called *an apex*, such that $G - v$ is planar. An *outerplanar* is a graph that excludes K_4 and $K_{2,3}$ as minors. Outerplanar graphs are also planar by Theorem 2.10.

A graph class $\mathscr{C}$ is said to be minor-closed if, for any graph G in $\mathscr{C}$, the graphs obtained by taking minors of G are also in the $\mathscr{C}$.

A graph parameter that has proved very useful for making graph problems easier to solve is the *tree width*, or *treewidth*. Intuitively, this is a parameter that seeks to measure how "tree-like" a graph is. Using dynamic programming algorithms on graphs that have limited treewidth, computer scientists have been able to solve various problems in polynomial or even linear time. To understand what treewidth is, we first need to understand the concept of *tree decomposition*.

A tree decomposition of a graph $G = (V,E)$ is a pair $(\{X_i \mid i \in I\}, T = (I,F))$ where each node $i \in I$ of the tree T has associated to it a subset of vertices $X_i \subseteq V(G)$, called the *bag* of i, such that:

I. Each vertex of G belongs to at least one bag: $\bigcup_{i \in I} X_i = V(G)$;

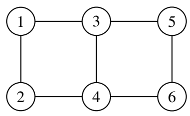
II. For every $vw \in E(G)$, there exists $i \in I$ such that $v, w \in X_i$;

III. For every $v \in V(G)$, the set $\{i \in I \mid v \in X_i\}$ induces a subtree of T.

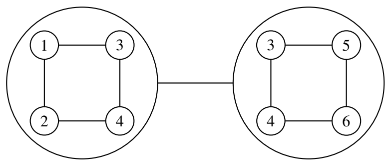
The same graph can have several tree decompositions. Take, for example, the graph $\mathscr{H}$ in Figure 9a. Figures 9b and 9c show two different tree decompositions for $\mathscr{H}$.

Figure 9 – Example of tree decomposition.

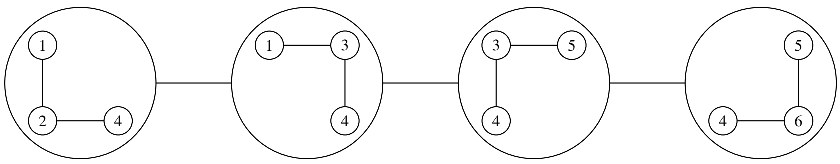
(a) A graph $\mathscr{H}$



(b) A tree decomposition of $\mathscr{H}$



(c) Another tree decomposition of $\mathscr{H}$



Source: Made by the author.

Given a tree decomposition of a graph G, we can define the *width* of *a* tree decomposition. The width of a tree decomposition is the size of its largest bag minus one. That is, let $\mathscr{T} = (\{X_i \mid i \in I\}, T = (I,F))$ be a tree decomposition. Then the width of $\mathscr{T}$ is $\max_{i \in I} |X_i| - 1$.

Observe that the tree decomposition in Figure 9b has width 3, while the one in Figure 9c has width 2. Based on the width, we define *treewidth*. The *treewidth* of a graph G is the smallest width of a tree decomposition among all the tree decompositions of G. We denote the treewidth of a graph G by $tw(G)$.

A path decomposition of a graph G is a tree decomposition of G in which the underlying tree is a path. Both tree decompositions shown in Figure 9 are path decompositions. The smallest width of a path decomposition of a graph G is the *pathwidth* of G, denoted by $pw(G)$.

2.2 Classical and Parameterized Computational Complexity

A string x is a finite sequence of symbols that are chosen from a finite alphabet Σ. For a natural number n, we denote the set of all strings of size n over the alphabet Σ as Σ^n. In addition, we denote the size of x as $|x|$.

Let Σ be an alphabet. The *Kleene closure* of Σ, denoted by Σ^* is the set of all strings of size i, for $i \geq 0$, formed by the elements of Σ. We denote the string of size 0, i.e. the *empty string*, as ε. Note that for any alphabet Σ, we always have $\varepsilon \in \Sigma^*$.

A *Turing Machine* is a quadruple $M = (K, \Sigma, \delta, s)$, where K is a finite set of states; $s \in K$ is the initial state; Σ is a finite set of symbols (we say that Σ is the alphabet of M); and δ is a transition function that maps $K \times \Sigma$ into $(K \cup \{h, \text{"}yes\text{"}, \text{"}no\text{"}\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\}$. We assume that K and Σ are disjoint sets; Σ always contains the symbols $\sqcup$ (empty) and $\triangleright$ (first); h is the stop state, "*yes*" is the accept state, "*no*" is the reject state; and the pointer directions $\leftarrow$ ("go left"), $\rightarrow$ ("go right") and $-$ ("stay in the same position") do not belong to $K \cup \Sigma$.

The function δ is the *"program"* of the Turing Machine. It specifies, for a current state $q \in K$ and a symbol $\sigma \in \Sigma$, the triple $\delta(q, \sigma) = (p, \rho, D)$, where p is the next state, ρ is the symbol to be written over σ and $D \in \{\leftarrow, \rightarrow, -\}$ is the direction in which the pointer should move.

The program starts in state s and the Turing Machine receives as *input* a string whose first character is $\triangleright$ followed by a finite string $x \in (\Sigma - \{\sqcup\})^*$. The symbol $\triangleright$ always directs the pointer to the right and is never deleted. The pointer starts at $\triangleright$ and the Turing Machine follows its program according to the rules described by δ, until it reaches one of the three stop states h, "*yes*" or "*no*".

Let $L \subseteq (\Sigma - \{\sqcup\})^*$ be a *language*, that is, a set of strings composed of the symbols of $\Sigma - \{\sqcup\}$. Let M be a Turing Machine such that, for every string $x \in (\Sigma - \{\sqcup\})^*$, if $x \in L$, then $M(x) = \text{"}yes\text{"}$, and if $x \notin L$, then $M(x) = \text{"}no\text{"}$. In this case, we say that *M decides L*.

Looking at the definitions given so far, we can think of a Turing Machine as a machine that solves problems related to strings. But observe that we can *encode* other finite mathematical objects using the format of strings and describe them using languages.

An *encoding* of a set S is a mapping e of elements of S into the set of binary strings $\{0,1\}^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \dots\}$. A common example is encoding decimal numbers in binary numbers. For example, $e(25) = 11001$. To represent more complex objects, such as graphs, we can combine the encodings of their constituent parts, generating an encoding for the entire object. A graph can be encoded in different ways. Despite this, we usually agree that the size of an encoding of a graph G is $O(n+m)$, where $n = |V(G)|$ and $m = |E(G)|$.

We can see then that a Turing Machine can solve other problems defined on more general mathematical objects and, thus, we can visualize Turing Machines as *algorithms*.

A *decision problem* is a binary relation between a set I of instances and a set of solutions {"*yes*", "*no*"}. We can then think of an algorithm that solves a *decision problem* as a Turing Machine that accepts a string x_i that encodes an instance $i \in I$ if it represents an instance "*yes*" of the decision problem and rejects it otherwise. In this case, we say that the algorithm *decides* the problem. From now on, the terms "algorithm" and "Turing Machine" will be used interchangeably.

Let $f, g : \mathbb{N}^+ \to \mathbb{R}^+$. Suppose that $g(n)$ is the number of steps that an algorithm $\mathscr{A}$ takes to produce the solution to a problem as a function of the size n of its input. If there are constants $c_1, c_2, n_0 \in \mathbb{R}^+$ such that for all $n \in \mathbb{N}^+$, $n \geq n_0$, we have that $g(n) \leq c_1 \cdot f(n) + c_2$, so we say that $g(n) \in O(f(n))$, i.e., $f(n)$ is an asymptotic upper bound of $g(n)$. We also denote $g(n) \in O(f(n))$ by $g(n) = O(f(n))$.

Let M be a Turing Machine that decides a language $L \subseteq \{0,1\}^*$. Let f be a function from the non-negative integers to the non-negative integers. We say that the machine M *operates in time* $f(n)$, if, for any input string $x \in L$ with $n = |x|$, we have that the number of operations needed until M halts is $O(f(|x|))$.

Let $L \subseteq \{0,1\}^*$ be a language. If there exists a Turing Machine M that decides L such that M operates in time n^k, for some constant $k > 0$, then we say that L is a language *decidable in polynomial time* and that there exists a *polynomial Turing Machine* (or *polynomial algorithm*) that decides L. We then denote by P the class of languages (or, equivalently, decision problems) that are *decidable in polynomial time*.

A language $L \subseteq \{0,1\}^*$ is in NP if there is a polynomial function $p : \mathbb{N} \to \mathbb{N}$ and a polynomial Turing Machine M such that, for all $x \in \{0,1\}^*$, we have that x belongs to L if and only if there exists $u \in \{0,1\}^{p(|x|)}$ such that $M(x,u) =$ "*yes*". In this case, we say that M is a *verifier* of L and u is a *certificate* of x. Intuitively, the verifier (or *verification algorithm*) checks the validity of a proposed solution u for an instance x of a decision problem in polynomial time.

See that $\mathsf{P} \subseteq \mathsf{NP}$, since, for a language $L \in \mathsf{P}$, we can simply use the machine M_L that decides L to be the verifier and make $u = \varepsilon$. However, it is not yet known whether $\mathsf{P} = \mathsf{NP}$. In fact, this is one of the most important open mathematical problems today.

This difference between the complexities of different decision problems has led computer scientists to wonder whether it is possible to capture the notion of *"hardness"* between problems. This led to the concept of *reduction* and NP-*hardness*.

A language $L \in \{0,1\}^*$ is *reducible in polynomial time* to a language $L' \in \{0,1\}^*$, denoted by $L \leq_p L'$, if there exists a polynomial-time computable function $f : \{0,1\}^* \rightarrow \{0,1\}^*$ such that, for all $x \in \{0,1\}^*$, $x \in L$ if and only if $f(x) \in L'$.

We then say that L' is NP-*hard* if $L \leq_p L'$, *for all* $L \in \mathsf{NP}$. When L' is NP-hard and $L' \in \mathsf{NP}$, then we say that L' is NP-*complete*.

However, in order to use polynomial reductions, a first NP-complete problem had to exist. The first problem to be shown to be NP-complete consists of deciding whether a set of logical clauses can be satisfied. It is called SAT and was shown to be NP-complete through a famous theorem, called the Cook-Levin Theorem (GAREY; JOHNSON, 1979). From this theorem, we began to show that other problems are NP-complete, by reducing from SAT. We now know several NP-complete problems that are also used to prove the NP-completeness of other problems.

NP-hardness captures the intuitive notion that a problem is "at least as hard" as another problem. Moreover, by the very definition of NP-hardness, if there is a polynomial algorithm for some NP-hard problem, then there is a polynomial algorithm for all problems in NP, implying that $\mathsf{P} = \mathsf{NP}$. However, no one has managed to find a polynomial algorithm that solves any NP-complete or NP-hard problem. So NP-hardness or NP-completeness is seen as evidence that these problems are intractable.

A problem is said to be *easy* or *tractable* if and only if there is a polynomial algorithm that solves *all* of its instances. A problem is said to be *intractable* if and only if *no polynomial algorithm is known* that solves *all* of its instances. Note that we may know of polynomial algorithms that solve a subset of the instances of an intractable problem, but not all of them. We still can't rule out the possibility that an algorithm exists that solves all instances of an (supposedly) intractable problem. If such an algorithm is discovered, we say the problem it solves is no longer intractable.

It is often possible to turn NP-hard or NP-complete problems into "easier" ones by considering more restricted cases and fixing the size of certain parameters of the input. Consider, for example, the classical problem VERTEX COVER.

VERTEX COVER

Input: A graph G and $k \in \mathbb{N}$.

Question: Does G have a vertex cover of size at most k?

As this is an NP-complete problem (GAREY; JOHNSON, 1979), there is no known algorithm that solves all instances of VERTEX COVER in polynomial time. However, now consider the following *parameterized problem*.

k-VERTEX COVER

Input: A graph G and $k \in \mathbb{N}$.

Parameter: k.

Question: Does G have a vertex cover of size at most k?

For this new problem, now with a fixed parameter k, there is an algorithm that solves it in time $O(2^k|V(G)|)$ (DOWNEY; FELLOWS, 2012), which is generally feasible in most cases. That said, let's now define parameterized problems and the complexity classes related to them.

A parameterized problem is a language $L \subseteq \Sigma^* \times \mathbb{N}$, that is, a pair (P,q), where $P \in \Sigma^*$ is a decision problem and $q \in \mathbb{N}$ is what we call a *parameter*.

We say that a parameterized problem L is in FPT (or *is* FPT or *treatable by fixed parameter*) if there exists a computable function f and an algorithm that decides L in time $f(k) \cdot |x|^{O(1)}$ for all $(x,k) \in L$. The P-VERTEX COVER problem is an FPT problem, since it has an algorithm that decides it in time $O(2^k \cdot n)$, where $n = |V(G)|$.

Similarly, we say that a parameterized problem L is in XP (or *is* XP) if there are two computable functions f and g and an algorithm that decides L in time $f(k) \cdot n^{g(k)}$, for all $(x,k) \in L$. The problem k-CLIQUE, shown below, when parameterized by the solution size k, has an algorithm that decides it in time $O(k^2 \cdot n^k)$ and is therefore in XP.

k-CLIQUE

Input: A graph G and $k \in \mathbb{N}$.

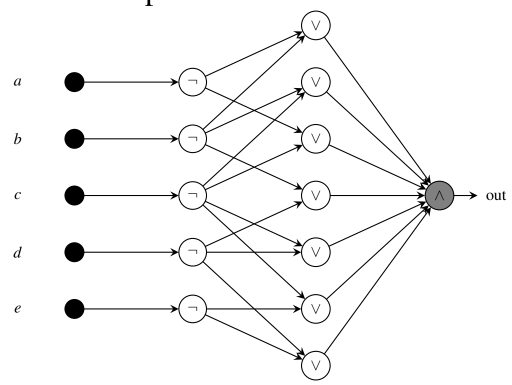
Parameter: k.

Question: Does G have a clique of size at least k?

Just as classical complexity has the concepts of polynomial reduction, NP-hardness and NP-completeness to capture the intractability of problems, parameterized complexity has the concepts of *parameterized reduction*, $W[t]$*-difficulty* and $W[t]$*-completeness* for the same purpose. To understand these new concepts, we first need some definitions.

A *Boolean circuit* C of n variables is a DAG (*acyclic directed graph*) (the reader can find more information about DAGs in (WEST, 2000)) such that: every vertex with entry degree 0 is an *entry vertex*; every vertex with entry degree 1 is a *negation vertex*; and every vertex with entry degree at least 2 is either an *and vertex* or an *or vertex*. In addition, *exactly one* of the vertices of output degree 0 is designated as the *output vertex* (in addition to being, for example, an *and* vertex). We say that the *depth* of C is the size of the longest path between an input vertex and the output vertex. We say that a vertex of C is a *small vertex* if it has entry degree at most 2. In the case where the entry degree of a vertex is greater than 2, we say that it is a *large vertex*. The *weft* of C is then defined as the maximum number of large vertices on a path between an input vertex and the output vertex. See Figure 10 for an example of a Boolean circuit.

Figure 10 – A boolean circuit with depth 3 and weft 1.



Source: Adapted from Cygan *et al.* (2012).

By assigning values TRUE (1) or FALSE (0) to the input vertices of C, the values of the other vertices are automatically obtained by the operations of each vertex. We say that C has been satisfied if, given an *assignment* of values to the input vertices, we have that the value of the output vertex is TRUE. The *weight* of an assignment τ is defined as the number of variables assigned the value TRUE in τ.

Let's denote by $\mathscr{C}_{t,d}$ the class of circuits with weft at most t and depth at most d. The circuit in Figure 10, for example, is in the class of circuits $\mathscr{C}_{1,3}$. We then define the parameterized problem P-WEIGHTED CIRCUIT SATISFIABILITY restricted to a class of circuits $\mathscr{C}_{t,d}$, also called $WCS[\mathscr{C}_{t,d}]$, as follows:

P-WEIGHTED CIRCUIT SATISFIABILITY ($WCS[\mathscr{C}_{t,d}]$)

Input: A circuit $C \in \mathscr{C}_{t,d}$ and $k \in \mathbb{N}$.

Parameter: k.

Question: Is there an assignment τ with weight at most k such that τ satisfies C?

The evidence so far suggests that $WCS[\mathscr{C}_{t,d}]$, for all $t \geq 1$ and $d \geq 1$, is not in FPT. Parameterized intractability theory is built on this hypothesis. Once WCS is considered intractable (or "hard"), we can transfer the intractability of WCS to another parameterized problem via a parameterized reduction.

Let $A, B \subseteq \Sigma^* \times \mathbb{N}$ be two parameterized problems. A *parameterized reduction* from A to B is an algorithm that, given an instance (x,k) of A, returns an instance (x',k') of B such that: (x,k) is a "yes" instance of A if and only if (x',k') is a "yes" instance of B; $k' \leq g(k)$ for some computable function g; and the algorithm operates in time $f(k) \cdot |x|^{O(1)}$ for some computable function f.

Let L be a parameterized problem. For $t \geq 1$, we say that L *belongs to the class* $\mathsf{W}[t]$ if there exists a parameterized reduction of L to $WCS[\mathscr{C}_{t,d}]$ for some $d \geq 1$. Conversely, we say that L *is* $\mathsf{W}[t]$*-hard* if $WCS[\mathscr{C}_{t,d}]$ can be reduced to L via a parameterized reduction, for $t \geq 1$ and $d \geq 1$. If, for some $t \geq 1$, L *belongs* to $\mathsf{W}[t]$ and *is* $\mathsf{W}[t]$-hard, then we say that L is $\mathsf{W}[t]$*-complete*.

A famous $\mathsf{W}[1]$-complete problem that is often used to show that other problems are $\mathsf{W}[1]$-hard or $\mathsf{W}[1]$-complete is the k-CLIQUE problem.

Finally, we have that $\mathsf{FPT} \subseteq \mathsf{W}[1] \subseteq \mathsf{W}[2] \subseteq \cdots \subseteq \mathsf{XP}$. Although we don't know if $\mathsf{FPT} = \mathsf{W}[1]$, most computer scientists believe that $\mathsf{FPT} \neq \mathsf{W}[1]$ and therefore showing that a problem is $\mathsf{W}[t]$-hard or $\mathsf{W}[t]$-complete is considered strong evidence that such a problem is not in FPT.

For more details on classic and parameterized computational complexity theory, we refer the reader to Garey and Johnson (1979), Cygan *et al.* (2012).

3 BIBLIOGRAPHIC REVIEW AND RELATED PROBLEMS

3.1 Irreversible Processes on Graphs

Suppose you are a worker in a factory whose set of workers is denoted by V. Imagine that different workers have different safety requirements and different levels of satisfiability with their jobs. A more radical worker starts a strike, but the strike will be successful only if many workers join. A crucial concept here to describe the variations amongst the individuals is that of a *threshold*. In this situation, the threshold of a worker will be defined as the number of fellow workers in the strike he needs to also join it. Define a graph $G = (V, E)$ in which the edges represent a friendship between two workers. We can represent the above situation in the graph G by assigning a state (1:"joined the strike"; 0:"did not join the strike") and a threshold value $t(v)$ to each vertex v. Then, step by step, workers check their neighborhood and update their state according to its threshold.

This simple model, as well as the example above, was introduced by Granovetter (1978) in his work "Threshold Models for Collective Behavior", and other models follow along similar ideas, for example (BANERJEE, 1992; BIKHCHANDANI *et al.*, 1992; DEGROOT, 1974; JR, 1956). Granovetter (1978) provides us with some extra examples and applications of this model, from which we highlight the following.

Diffusion of innovations and (mis-)information. Several empirical works have verified that innovations and information spread in a "threshold-like" manner. Some physicians are more prone (have small thresholds) to innovations and start using a new medicine. Then, as their fellow physicians start adopting the new medicine, other physicians adopt it too (COLEMAN *et al.*, 1957). A similar behavior was verified in farmers who began to use a new corn seed as their neighbors were using too (RYAN; GROSS, 1950). Information and misinformation spread from person to person, but some may have more credulity than others, and some may need to see many friends talking about it to believe.

Disease spreading. We can replace "credulity" by "vulnerability" and see how this model also applies to diseases. In many diseases, some kind of contact with an infected person is needed to get infected. The number of contacts needed (the threshold) for a person to get infected may vary according to their vulnerability.

We can formally describe the model above, which we will refer to as the *threshold model*, by using an *automata network* (GOLES; MARTÍNEZ, 1990; WOLFRAM, 1986). An automata network is a triple $\mathscr{A} = (G, Q, (f_v)_{v \in V(G)})$ defined by a graph G, a finite set of states Q, and transition functions $f_v : Q^n \to Q$ for each vertex $v \in V(G)$. Each vertex $v \in V(G)$ will be in a state $c_v(\tau) \in Q$ in a non-negative integer time step τ. From time step τ to $\tau + 1$, the transition functions $(f_v)_{v \in V(G)}$ are applied, updating the state of a vertex (*sequential iteration*) or a set of vertices (*synchronous iteration*). In the synchronous iteration, we apply the rule $c_i(\tau+1) = f_i(c_1(\tau), \ldots, c_n(\tau))$ for every $i \in V(G)$. In the sequential iteration, the rule is applied one vertex at a time. In this work, the iteration type we deal with is synchronous. For more details on automata networks, the reader can check (GOLES; MARTÍNEZ, 1990).

Automata networks have been used to model and study several themes and phenomena, such as spin glass physics (DERRIDA, 1988); statistical physics (WOLFRAM, 1986; WOLFRAM, 1985); ergodic theory and finite Markov fields (KINDERMANN; SNELL, 1980; WOLFRAM, 1986; DOBRUSHIN *et al.*, 1978); crystal growth chemistry (ULAM *et al.*, 1962); genetic interactions (KAUFFMAN, 2017; KAUFFMAN, 1969) and other biological applications (DOBRUSHIN *et al.*, 1978); urban growth and organization (LIU *et al.*, 2014); social self-organization (MEDINA *et al.*, 2017) and behavior (SEITZ *et al.*, 2014). Famous automata networks are the von Neumann Cellular Automata (NEUMANN *et al.*, 1966), which is known for having the same computability power as a Turing Machine, and John Conway's Game of Life (CONWAY *et al.*, 1970).

The threshold model we study here falls into a specific kind of automata networks, called *dynamic systems on graphs*. A dynamic system on a graph is an automata network in which we have only two states, for example $Q = \{0, 1\}$ or $Q = \{+, -\}$, and the transition function f_v for a vertex v depends on the states of the neighborhood of v.

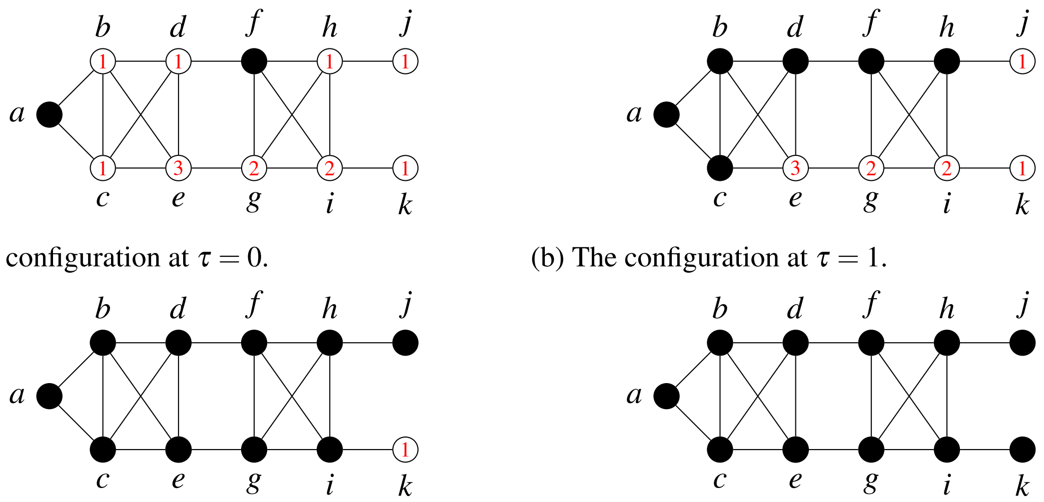
Let G be a graph. To each vertex $v \in V(G)$ is assigned one of two states. We will refer to the states as *active* or *infected* (1), and *inactive* or *uninfected* (0). Given a set of initially infected vertices, which we call a *seed set*, $S \subseteq V(G)$, the initial states of the vertices, $(c_i(0))_{i \in V(G)}$, is defined as follows.

$$c_i(0) = \begin{cases} 1 \text{ (``active'')} & \text{if } i \in S, \\ 0 \text{ (``inactive'')} & \text{otherwise.} \end{cases}$$

Each vertex is associated with a *threshold* value determined by the *threshold function* $t : V(G) \to \mathbb{N}$. As we have discussed before, the threshold value of a vertex models its susceptibility to change its state from inactive to active by taking into account its neighbors' states. An inactive vertex $v \in V(G)$ at time τ changes its state to active at time $\tau+1$ if it has at least $t(v)$ active neighbors at time τ. At each time step $\tau > 0$, the following update rule is applied on all vertices of G.

$$c_i(\tau+1) = \begin{cases} 1 \text{ (``active'')} & \text{if } c_i(\tau) = 1 \textit{ or } |\{u \in N_G(i) \mid c_u(\tau) = 1\}| \geq t(i), \\ 0 \text{ (``inactive'')} & \text{otherwise.} \end{cases} \tag{3.1}$$

A *configuration* of a graph G at a time $\tau \geq 0$ is a collection of states of the vertices of G at time τ, $C_\tau = (c_i(\tau))_{i \in V(G)}$. An infinite sequence of configurations $\mathscr{P} = (C_\tau)_{\tau \in \mathbb{N}}$ of G is called a *discrete dynamic process* on G. Observe that, in the process we defined above, once a vertex gets active it stays active. Such processes are called *irreversible processes*. This means that there is a minimum time step τ_F in which either all vertices are active or no more vertices can get active. We say that τ_F is the *final time* of a irreversible process. We call a threshold process based on the rule defined on Equation 3.1 of a *t-irreversible process* (JR, 2000; DREYER; ROBERTS, 2009). We illustrate such a process in Figure 11.



(a) The configuration at $\tau = 0$.

(b) The configuration at $\tau = 1$.

(c) The configuration at $\tau = 2$.

(d) The configuration at $\tau = 3$.

Figure 11 – Example of an irreversible t-process. The thresholds are shown in red and the active vertices are in black.

Source: Made by the author.

As imagined, one can also define *reversible* (threshold) processes. That kind of processes, however, are not the object of study of this work even though they are also very interesting. For more on reversible processes, the reader may refer to (LIMA, 2017).

Adapting the notation used in (LIMA, 2017), given a graph G with thresholds $t: V(G) \to \mathbb{N}$ and a seed set $S \subseteq V(G)$, we will denote a t-irreversible process in G by $I_t(G,S) =$

$(C_\tau)_{\tau\in\mathbb{N}}$.

Other irreversible processes exist in the literature, with a variety of names and updating rules, both threshold-based and non-threshold-based. In the *zero forcing* process (AIM Minimum Rank Special Graphs Work Group, 2008; BURGARTH *et al.*, 2009; KALINOWSKI *et al.*, 2019), we are given a graph G with a set of initially active vertices $B \subseteq V(G)$. An inactive vertex v of G at time τ gets active at time $\tau+1$ if it has an active neighbor u such that the *only* inactive neighbor of u is v at time τ. The idea is that if an active vertex has only one inactive neighbor, it *forces* him to become active. Such a process appears in the literature also under the name of *power domination* (DORBEC, 2020) and has applications in quantum computing and controllability of quantum networks (BURGARTH *et al.*, 2009).

The *k-forcing process* (AMOS *et al.*, 2015) is a generalization of the zero forcing process. Starting with a set of initially active vertices B from a graph G, if a vertex u has at most k inactive neighbors at time τ, then all of them get active at time $\tau+1$. When $k=1$, this is equivalent to the zero forcing process.

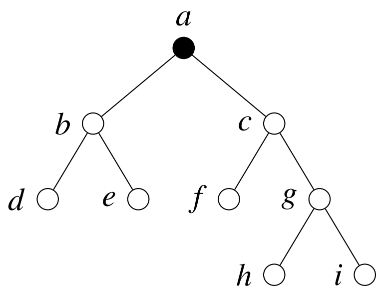
When the thresholds $t(v)$ of each vertex v of a graph G are all equal to 1, the t-irreversible process defined by the rule 3.1 appears in the literature under the name of *graph burning* (FINBOW; MACGILLIVRAY, 2009; BONATO *et al.*, 2014; BONATO *et al.*, 2016; BESSY *et al.*, 2017; BONATO, 2020). Figure 12 shows a graph burning process in a tree. Similarly, when $t(v) = r$ for all $v \in V(G)$, the process receives the name of *r-neighbour bootstrap percolation* (RIEDL, 2010; BENEVIDES *et al.*, 2015; MARCILON; SAMPAIO, 2018), or simply of *r-irreversible process* (DREYER; ROBERTS, 2009).

Convexities on graphs can also be framed as irreversible processes. Under the P_3 convexity (CENTENO *et al.*, 2011), an inactive vertex u gets active at time $\tau+1$ if it lies on a path of length 3 between two active vertices at time τ. Under the geodesic (monophonic) convexity (HARARY; NIEMINEN, 1981; JAMISON; NOWAKOWSKI, 1984), an inactive vertex u gets active at time $\tau+1$ if it lies on a minimum (induced) path between two active vertices at time τ.

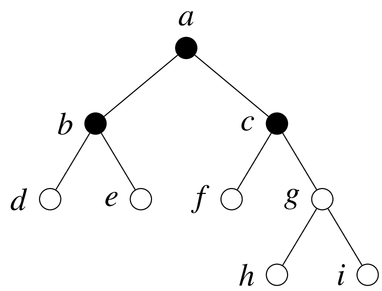
Given a t-irreversible process $\mathscr{P} = I_t(G,S) = (C_\tau)_{\tau\in\mathbb{N}}$, we define the set of *active vertices at time* τ as

$$A_\tau(\mathscr{P}) = \{i \in V(G) \mid c_i(\tau) = 1\}.$$

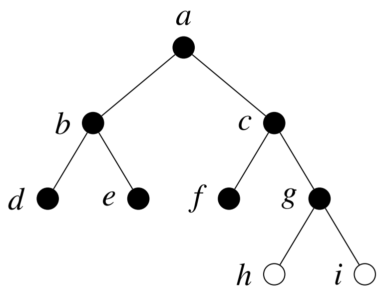
Observe that $A_0(\mathscr{P}) \subseteq A_1(\mathscr{P}) \subseteq A_2(\mathscr{P}) \subseteq \ldots$. The set of *final active vertices* of



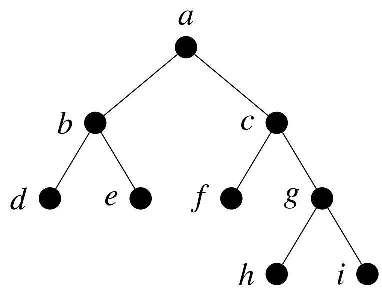
(a) The configuration at $\tau = 0$.



(b) The configuration at $\tau = 1$.



(c) The configuration at $\tau = 2$.



(d) The configuration at $\tau = 3$.

Figure 12 – Example of a graph burning process. Every vertex that is adjacent to an active vertex gets active in the next time step.

Source: Made by the author.

$\mathscr{P}$ is defined as

$$A^*(\mathscr{P}) = \bigcup_{\tau \geq 0} A_\tau(\mathscr{P}).$$

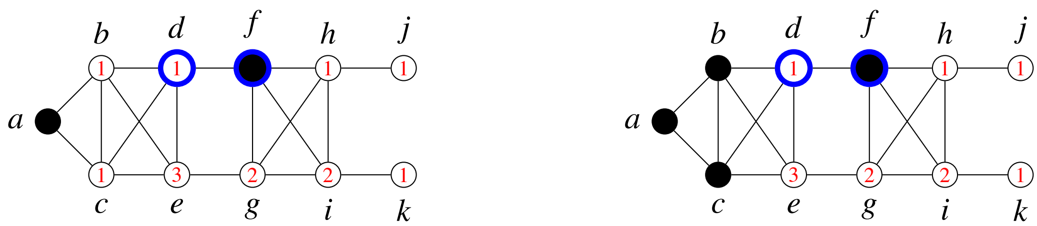
Several NP-complete or NP-hard problems arise from irreversible processes on graphs such as those we have discussed. Given a graph G and a threshold function $t : V(G) \to \mathbb{N}$, the TARGET SET SELECTION problem (KEMPE *et al.*, 2003; CHEN, 2009) asks to find a small set of initially active vertices S of G such that all vertices become active, i.e., $A^*(\mathscr{P} = I_t(G,S)) = V(G)$. This problem also appears in the literature under different nomenclatures, such as finding *dynamic monopolies* (ZAKER, 2012; EHARD; RAUTENBACH, 2019). All the other irreversible processes we mentioned in this section also have analogous problems associated with them.

However, problems such as these, in which the goal is to maximize the number of active vertices, are not the object of study of this dissertation. Rather, we study a problem that falls into a class of problems that we call *immunization problems*. They are the subject of the next section.

3.2 Immunization Problems in Graphs

Recall the two applications of the threshold model we have seen in the last chapter: diffusion of innovations and *(mis-)information*, and *disease spreading*. In contexts like these, it is natural to think about restricting the number of active or infected vertices rather than maximizing them. We do that by *immunizing* or *protecting* vertices.

Let G be a graph with thresholds $t : V(G) \to \mathbb{N}$. As far as we know, there are two variants for the immunization of a vertex. In the first variant, we immunize a vertex by raising its threshold $t(v)$ to $+\infty$ or $d_G(v)+1$ (EHARD; RAUTENBACH, 2019). In the second variant, the immunization of a vertex is analogous to its removal from the graph (CORDASCO *et al.*, 2023). In both variants, an *inactive* immunized vertex cannot get active in any t-irreversible process on G. However, observe that these two variants behave differently. In the first one, if an immunized vertex is *already active*, its immunization makes no difference – the vertex will still influence its neighbors. In the second one, this does not happen: an immunized vertex, already infected or not, cannot influence others nor get active. In this work, we focus on the second variant of immunization.



(a) The configuration at $\tau = 0$ with vertices f and d immunized.

(b) The configuration at $\tau = 1$ with vertices f and d immunized. No more vertices can be infected.

Figure 13 – Example of an irreversible t-process with immunized vertices. The immunized vertices are circled in blue.

Source: Made by the author.

Given a t-irreversible process $\mathscr{P} = I_t(G,S)$ and a set $Y \subseteq V(G)$ of immunized vertices, we define a *t-irreversible process with immunized vertices* $\mathscr{P}_Y = I_t(G-Y, S-Y)$. Observe that we did not add the restriction that $Y \cap S = \emptyset$. In fact, we allow initially active vertices to be immunized. However, this does not mean that an initially active vertex that is immunized becomes inactive. It just means that it is not "contagious" anymore, i.e., it will not influence the states of its neighbors. Figure 13 shows an example of a t-irreversible process with immunized vertices. In this example, the seed set is $S = \{a, f\}$ and the set of immunized vertices is $Y = \{d, f\}$. Observe that $f \in S$ and $f \in Y$. In time $\tau = 1$, if f were not immunized, the vertex

h would get active, but since h is immunized, it does not influence the states of its neighbors anymore. Observe that we can assume that $t(v) \leq d_G(v)$ for all $v \in V(G) \setminus S$, since a vertex v with $t(v) > d_G(v)$ would never get active.

For a natural $k \geq |S|$, we say that Y is a *k-restricting* set of $\mathscr{P}$ if $|A^*(\mathscr{P}_Y) \cup S| \leq k$. That is, by immunizing the vertices in Y, the number of final active vertices, *counting with the initially infected ones*, does not exceed k. In other words, the infection is *restricted* to at most k vertices by immunizing Y. In the example of Figure 13, $Y = \{d, f\}$ is a 4-restricting set of that process. If Y is a $|S|$-restricting set, we also call it a *total restriction set* of $\mathscr{P}$. Observe that, if Y is a total restriction set of $\mathscr{P}$, then $A^*(\mathscr{P}_Y) \cup S = S$, i.e., all initially inactive vertices remain inactive for the whole process. We define the *k-restricting number* of $\mathscr{P}$, denoted by $\mathfrak{R}(\mathscr{P}, k)$, as the cardinality of a smallest k-restricting set of $\mathscr{P}$. Observe that, for any $k \geq |S|$, such a set always exists. Similarly, the *total restriction number* of $\mathscr{P}$, denoted by $\mathfrak{R}_T(\mathscr{P})$, is the cardinality of a smallest total restriction set of $\mathscr{P}$. Figure 14 shows a total restriction set of size 1. The following observations are going to be useful throughout this work.

Observation 3.1. *For any $k \geq |S|$, we have that $\mathfrak{R}(\mathscr{P}, k) \leq \mathfrak{R}_T(\mathscr{P})$.*

Proof. If Y is a total restriction set for $\mathscr{P}$, then Y is also a k-restricting set for $\mathscr{P}$, for any $k \geq |S|$. □

Observation 3.2. *Let $\mathscr{P} = I_t(G,S)$ be a t-irreversible process and $\mathscr{P}^1 = I_{t'}(G,S)$ be the t-irreversible process defined by the same graph and seed set but with thresholds $t'(v) = 1$ for all $v \in V(G)$. Then, for any $k \geq |S|$, we have that $\mathfrak{R}(\mathscr{P}, k) \leq \mathfrak{R}(\mathscr{P}^1, k)$.*

Proof. If $t(v) = t'(v) = 1$ for all $v \in V(G)$, the equality $\mathfrak{R}(\mathscr{P}, k) = \mathfrak{R}(\mathscr{P}^1, k)$ follows. Otherwise, $t(v) > t'(v)$ for some vertex $v \in V(G)$. Let Y be the k-restricting set of $\mathscr{P}^1$. Then, the immunization of Y also restricts the infection to at least k vertices in $\mathscr{P}$. Thus $\mathfrak{R}(\mathscr{P}, k) \leq \mathfrak{R}(\mathscr{P}^1, k)$ follows. □

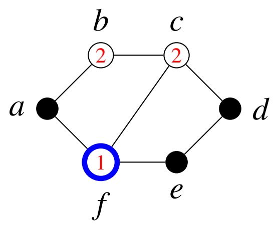


Figure 14 – Example of a total restriction set. The immunized vertices are circled in blue.
Source: Made by the author.

Cordasco *et al.* (2023) introduced the problem (k,ℓ)-INFLUENCE IMMUNIZATION BOUNDING ((k,ℓ)-IIB), which we state bellow. Even though we do not agree with the name of the problem, we are going to stick to it and keep calling it by (k,ℓ)-IIB throughout the rest of this work.

(k,ℓ)-INFLUENCE IMMUNIZATION BOUNDING ((k,ℓ)-IIB)
Input: A t-irreversible process $\mathscr{P} = I_t(G,S)$, and naturals $k \geq |S|$, and ℓ.
Question: Is there a k-restricting set Y of $\mathscr{P}$ such that $|Y| \leq \ell$?

Cordasco *et al.* (2023) studied the (k,ℓ)-IIB problem from a parameterized complexity point of view. They showed several W[1]-hardness and W[2]-hardness results for a set of parameters, and FPT algorithms for combinations of some of these parameters. Their results on (k,ℓ)-IIB are summarized on Table 1. As this is a fairly recent problem, their results are the only results on this problem to the extent of our knowledge. Thus, we focus in this chapter on reviewing problems with similar concepts and connections to (k,ℓ)-IIB.

Parameter	**Hardness**	**Restriction on the entry**
k	W[1]-hard	$t(v) = 1 \quad \forall v$
ℓ	W[1]-hard	$t(v) = 1 \quad \forall v$
$k+\ell$	FPT	
$\lvert S\rvert+\ell$	W[2]-hard	Bipartite graphs
$k+\lvert S\rvert$	FPT	
$\Delta(G)+\ell$	W[2]-hard	$t(v) \leq 2 \quad \forall v$
$tw(G)$	W[1]-hard	
$nd(G)$	W[1]-hard	
$k+nd(G)$	FPT	
$\ell+nd(G)$	FPT	
$\min(\Delta(G),k)+tw(G)+\ell$	FPT	

Table 1 – Summary of parameterized complexity results of Cordasco *et al.* (2023).
Source: Made by the author with data from Cordasco *et al.* (2023).

Observe that, when we restrict $\mathscr{P}$ to be a graph burning process, i.e., the thresholds of all vertices are equal to 1, then (k,ℓ)-IIB is equivalent to the problem of finding a small unbalanced vertex separator of the graph. This is the problem studied in (FOMIN *et al.*, 2013). Given a graph G, a source vertex $s \in V(G)$ and parameters $k,\ell \in \mathbb{N}$, they study the problem of finding a separator $\mathscr{S}$ of G that separates $V(G) \setminus \mathscr{S}$ in two sets A and B such that $|\mathscr{S}| \leq \ell$, $|A| \leq k$, and $s \in A$. This problem was called CUTTING k VERTICES WITH TERMINAL (CVT-k). Fomin *et al.* (2013) showed that CVT-k is W[1]-hard with respect to the parameters k and ℓ.

Observe the resemblance of CVT-k to (k,ℓ)-IIB and the correspondence of parameters k and ℓ in both problems. In fact, the reductions of Cordasco *et al.* (2023) showing that (k,ℓ)-IIB is W[1]-hard for these parameters are straightforward from CVT-k.

This problem falls under the umbrella of a wide class of problems called *Node-Deletion Problems*. In a problem of this class, we are given a graph G and a property π and are asked to find a small subset of vertices of G such that, after their deletion from G, the remaining graph satisfies π. The complexity of such problems is intrinsically related to the property π. A property π is said to be *hereditary on induced subgraphs* if, for any graph G satisfying π, the graph resulting from the deletion of any vertex from G also satisfies π. Similarly, a property π is said to be *non-trivial* for a class of graphs $\mathscr{C}$ if it is satisfied by the trivial graph, and there are infinitely many graphs in $\mathscr{C}$ that satisfy π and infinitely many graphs in $\mathscr{C}$ that does not satisfy π.

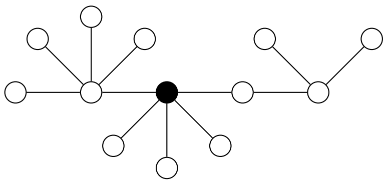
If π is hereditary on induced subgraphs, Lewis and Yannakakis (1980) showed that the associated Node-Deletion Problem is NP-complete on general (undirected and directed) graphs. Complementing this result, Yannakakis (1981) showed that, for any non-trivial property π, the associated Node-Deletion Problem is NP-complete even when restricted to *bipartite graphs*, except for the VERTEX COVER problem.

Node-Deletion Problems in which the property π involves connectivity measures of the graph are called *Critical Node Detection Problems* (LALOU *et al.*, 2018). One example is the NETWORK DISMANTLING problem, which asks for a small set of vertices of a graph G, such that its removal dismantles G into components of small size (BRAUNSTEIN *et al.*, 2016; REN *et al.*, 2019). Node-Deletion Problems such as these also have applications in epidemic and misinformation control, thus we also framed them here as immunization problems.

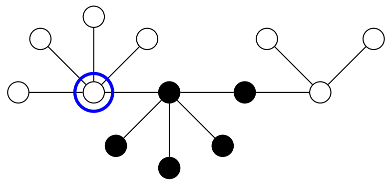
But not only vertices can be immunized. Mehta and Reichman (2022) studied a very similar problem to (k,ℓ)-IIB in which the goal is not to find a subset of vertices to immunize, but rather a subset of edges. The notion of immunization of an edge in their work is similar to ours of immunization of vertices: an immunized edge is analogous to a removed edge. They also do not work with general thresholds but with r-neighbour bootstrap percolation. Recall that, in the r-neighbour bootstrap percolation, the thresholds of all vertices in the graph are equal to r. Mehta and Reichman (2022) showed that, for $r \geq 2$, a given constant d, and a parameter $\kappa \leq n^{1-\varepsilon}$ with $\varepsilon \in (0,1)$, the problem can be solved with high probability on random graphs $\mathbb{G}(n,p)$, where $p = \frac{d}{n}$, in time $2^{o(\kappa)} \cdot \text{poly}(n)$.

Another similar problem to (k,ℓ)-IIB can be formulated in terms of an edge cut problem. The standard setup of the (min-)edge cut problem is the following: given a directed graph $D=(V,A)$, a source s, and a sink t, we wish to find a cut $[S,\overline{S}]$ of D with minimum size. When we look at a cut $[S,\overline{S}]$, we can think of S like the set of vertices "on the side" of the source s, and $\overline{S}$ the set of vertices "on the side" of the sink t. Hayrapetyan *et al.* (2005) studied a variation of the edge cut problem in which the goal is to minimize the number of vertices in the source side with an edge cut subject to a cardinality constraint. They showed an efficient $(\frac{1}{\lambda},\frac{1}{1-\lambda})$-bicriteria approximation algorithm for this problem, for any $0<\lambda<1$. This result comes from an analysis of the linear relaxation of an integer programming formulation of the problem. Let (x^*,y^*) be the optimal solution to the linear relaxation, and ℓ be a random uniformly chosen number from the interval $[1-\lambda,1]$, with $0<\lambda<1$. Then, define $S=\{v \mid x^*_v \geq \ell\}$. Hayrapetyan *et al.* (2005) showed that $[S,\overline{S}]$ is an edge cut of size at most $\frac{1}{\lambda}B$ such that the number of vertices on the source side is at most $\frac{1}{1-\lambda}$ times more vertices than the minimum for an edge cut with capacity B. They improved this result by showing a parametric maximum flow algorithm that avoids solving the linear programming model and outputs either a $(\frac{1}{\lambda},1)$-approximation or a $(1,\frac{1}{1-\lambda})$-approximation. They also showed that this problem is FPT parameterized by the treewidth of the graph.

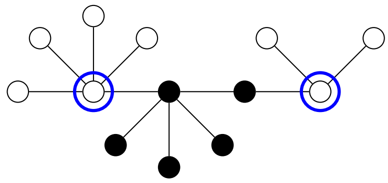
Another problem with a very similar concept to that of (k,ℓ)-IIB, and perhaps the most studied problem among the problems we mention in this chapter, is the FIREFIGHTER, introduced by Hartnell (1995). As this is a widely studied problem with similarities to (k,ℓ)-IIB, we are going to see next the main results about it. The setup of the FIREFIGHTER problem is a graph burning process. We have a graph G and a vertex $s \in V(G)$ where the fire starts, i.e., s is the only initially active vertex. We say that an active vertex is *burning*. Then, we are given b firefighters, $b \geq 1$, and the goal is to control the fire. The most common version of FIREFIGHTER is with $b=1$. The problem works like a solitaire game, where at each time step (or round of the game), the player needs to choose b vertices from G to place b firefighters. Once a firefighter is placed on a vertex $u \in V(G)$, then u will never burn in any of the next rounds and we say that u is *protected*. However, firefighters cannot protect already burned vertices. After the player has chosen the vertices to protect, the fire spreads in a graph burning manner: if a non-protected vertex has one burning neighbor, then it starts to burn too. The game stops when the fire can no longer spread. Figure 15 shows an example of the FIREFIGHTER problem. At the end of the game, we can partition $V(G)$ into two sets: the set of burned vertices B and the set of non-burned vertices, which we call *saved vertices*, S. Several variations of FIREFIGHTER appear in the literature with different goals for the player (FINBOW; MACGILLIVRAY, 2009). Common goals are to maximize the number of saved vertices (FINBOW *et al.*, 2007; HARTNELL, 1995), which is the classical FIREFIGHTER problem; and to decide if a given set $P \subseteq V(G)$ can be fully protected from the fire (KING; MACGILLIVRAY, 2010), which we call S-FIREFIGHTER.



(a) The configuration before the game begins.



(b) The configuration at the end of round 1.



(c) The configuration at the end of round 2.

Figure 15 – Example of a firefighting in a caterpillar graph.
Source: Made by the author.

FIREFIGHTER was studied in several different graph classes, such as trees, bipartite graphs, planar graphs, and regular graphs. The problem is NP-complete in all of these classes (FINBOW *et al.*, 2007; KING; MACGILLIVRAY, 2010; BAZGAN *et al.*, 2013). Even if we restrict the problem to the class of subcubic trees, it remains NP-complete (FINBOW *et al.*, 2007). Observe that, even though (k,ℓ)-IIB is very similar to FIREFIGHTER, the first is not a generalization of the latter. In (k,ℓ)-IIB, the thresholds are general and we do not have the notion of rounds: the immunization shall be made at the beginning of the process, not during it. Besides that, in (k,ℓ)-IIB several vertices may be initially active, whilst in FIREFIGHTER only one vertex starts the fire. Thus, those hardness results cannot be generalized to (k,ℓ)-IIB.

There are, however, some easy cases for FIREFIGHTER. Given G and $s \in V(G)$, we denote the maximum number of vertices that b firefighters can save when the fire starts in G at vertex s by $MVS(G,s;b)$. When $b = 1$, we may omit the last parameter, writing only $MVS(G,s)$.

Theorem 3.3. *(MACGILLIVRAY; WANG, 2003; FINBOW; MACGILLIVRAY, 2009) The following affirmations are true:*

I. For $n \geq 2$, $MVS(K_n,s) = 1$;

II. For $n \geq 2$, $MVS(P_n,s) = n-1$ *if* s *is a leaf and* $MVS(P_n,s) = n-2$ *otherwise;*

III. *For* $n \geq 3$, $MVS(C_n, s) = n - 2$;

IV. *For* $n \geq 2$, $MVS(Q_n, s) = n$;

V. *For* $2 \leq m \leq n$, $MVS(K_{m,n}, s) = 2$.

We define a *strategy* for a FIREFIGHTER instance as a sequence of valid choices of vertices for the player for each round $\tau \geq 1$. We say that a strategy is *optimal* with respect to a given goal if the outcome of the strategy is the best possible result concerning the goal. For example, if the goal is to maximize the number of saved vertices, we say that a strategy is optimal if, when the fire stops, the maximum number of vertices are saved by the strategy choices. MacGillivray and Wang (2003) proved that the following strategy saves the maximum amount of vertices in a caterpillar graph.

Theorem 3.4. *(MACGILLIVRAY; WANG, 2003) Let G be a caterpillar graph and the fire breaks out at the vertex* $s \in V(G)$. *If s is a leaf, then protect the only neighbor of s. Otherwise, protect the neighbor of s with the highest degree at round 1. If the fire has not stopped yet, protect the vertex with the highest degree that is adjacent to a burning vertex. This strategy saves the maximum possible number of vertices in G.*

Figure 15 is an example of this strategy. Another important result from (MACGILLIVRAY; WANG, 2003) is the following result concerning trees.

Theorem 3.5. *(MACGILLIVRAY; WANG, 2003) Let* T *be a tree and the fire breaks out at the vertex* $s \in V(T)$. *Then, in any optimal strategy, the vertex protected at each time is adjacent to a burning vertex.*

Recall that the FIREFIGHTER problem is NP-complete even for subcubic trees (FINBOW; MACGILLIVRAY, 2009). By following the above result, the following strategy for binary trees comes up.

Theorem 3.6. *(WAGNER, 2021) Let* T *be a rooted binary tree with root r and suppose the fire breaks out at r. The following strategy is optimal for* T *concerning the goals of maximizing the number of saved vertices and minimizing the time in which the fire stops. At each round, protect a vertex u that is adjacent to a burning vertex and for which the distance from u to its descendant of degree 2 is maximized.*

Another surprising result that improves on the result of Theorem 3.6 shows that if the fire breaks out on a subcubic tree at a vertex of degree 2, the problem is in P (FINBOW *et* *al.*, 2007; KING; MACGILLIVRAY, 2010). Thus, if $\mathsf{P} \neq \mathsf{NP}$, FIREFIGHTER is NP-complete on subcubic trees only if the fire breaks out at a vertex of degree exactly 3. Generalizing this result, Bazgan *et al.* (2013) showed that, for any $b \geq 1$, the problem is NP-complete on trees of maximum degree $b+2$, but polynomially solvable if the fire breaks out at a vertex with degree at most $b+1$. In (CAI *et al.*, 2008), a sub-exponential algorithm for FIREFIGHTER on trees is presented, which has running time $2^{O(\sqrt{n}\log n)}$.

From the point of view of parameterized complexity, there are also some hardness results and some fixed-parameter tractability results, which we summarize below. Let us denote by κ the parameter that measures the number of saved vertices and κ_ℓ the number of saved leaves in a tree. For a given graph G, denote by $cvd(G)$ the parameter *cluster vertex deletion* of G, which is defined as the minimum number of vertices to be removed from G such that the remaining graph is a disjoint union of complete graphs. Bazgan *et al.* (2013) showed that FIREFIGHTER is W[1]-hard parameterized by κ even when restricted to bipartite graphs. For the parameters κ and κ_ℓ, FIREFIGHTER is FPT in trees with running times $4^{\kappa}\kappa^{O(\log\kappa)} + O(n)$ and $2^{O(\kappa)} + O(n)$, respectively (CAI *et al.*, 2008). Bazgan *et al.* (2014) showed that, for planar graphs, the problem is FPT when parameterized by $\kappa + b$ with an algorithm of complexity $O(2^{\kappa^2+(b+1)+\kappa b}\mathrm{poly}(n))$, where b is the maximum degree. For the pathwidth of the graph $pw(G)$, Chlebíková and Chopin (2017) showed that the problem is para-NP-hard, but FPT when parameterized by the same parameter combined with the maximum degree $pw(G) + \Delta(G)$. They also showed that the problem is FPT parameterized by $cvd(G)$. Das *et al.* (2019) studied FIREFIGHTER parameterized by modulator parameters. They showed that the problem is W[1]-hard parameterized by modulators to split graphs and diameter 2 graphs. On the other hand, they showed that the problem is FPT parameterized by modulators to cographs and disjoint unions of stars.

3.3 Other Variations and Similar Problems

Even though this is a very hot and interesting topic, especially after the COVID-19 pandemic, there are not many studies from an algorithmic, combinatorial, and computational point of view, and even less when we talk about immunization in t-irreversible processes. Most works focus on stochastically modeling the spread of an epidemic in different settings and in different types of networks, and on empirically studying the allocation of vaccines using intuitive greedy or random algorithms, such as selecting vertices with high degrees or by using some centrality measure. Examples of works in this vein can be seen in the books (NEWMAN, 2010; BARABÁSI; PÓSFAI, 2016) and in (PASTOR-SATORRAS *et al.*, 2003; GIAKKOUPIS *et al.*, 2005; PRAKASH *et al.*, 2010; VENTRESCA; ALEMAN, 2015; YANG *et al.*, 2018). Some game-theoretical works model agents with different payoffs as vertices interacting in a graphical game. Examples of these works are (ASPNES *et al.*, 2006; MOSCIBRODA *et al.*, 2006; CHEN *et al.*, 2010; MEIER *et al.*, 2014).

Several works approach the problem of finding a set of vertices or edges to immunize by using methods from spectral graph theory. These works usually adopt the larger eigenvalue of the graph's adjacency matrix as a measure for the *vulnerability* of the graph (CHAKRABARTI *et al.*, 2008; CHEN *et al.*, 2016; TARIQ *et al.*, 2017; AHMAD *et al.*, 2020). They focus on identifying a set of influential vertices from the graph such that their immunization leads to the decrease of the larger eigenvalue to the minimum, thus decreasing the "vulnerability" of the graph to some infection. Similar spectral works focusing on edge sets rather than vertex sets can be found in (TONG *et al.*, 2012; KUHLMAN *et al.*, 2013).

Kimura *et al.* (2009) studied a different propagation model, the Independent Cascade, in which the infection spreads from an infected vertex to an uninfected neighbors with a certain probability. They studied the problem of finding a subset of the edges of the graph such that its removal would minimize the proportion of infected vertices and showed greedy heuristics and an approximation algorithm for this problem. Sheikhahmadi *et al.* (2023) also studied the problem of minimizing an infection in a network by immunizing some vertices, similar to (k,ℓ)-IIB. They approached it by identifying communities in the graph and vertices with a high centrality measure to immunize.

As we can see, there is a wide universe of different models and problems if one wants to study problems related to dynamic processes and immunizations in graphs. In this work, however, we focus on one of them: the (k,ℓ)-INFLUENCE IMMUNIZATION BOUNDING problem, which we introduced at the beginning of this section. From the next chapter on, we will talk about our results on (k,ℓ)-IIB.

4 HARDNESS RESULTS FOR IIB RESTRICTED TO SOME GRAPH CLASSES

Recall that, in this work, we study the (k,ℓ)-INFLUENCE IMMUNIZATION BOUNDING ((k,ℓ)-IIB) problem, in which the goal is to decide if a t-irreversible process $\mathscr{P} = I_t(G,S)$ has a k-restricting set of size at most ℓ. As we have seen in Table 1, IIB is W[1]-hard and W[2]-hard even for small threshold values and for bipartite graphs (CORDASCO *et al.*, 2023), implying also in NP-completeness of this problem in such instances. In this chapter, we complement these results by showing classical and parameterized hardness results for IIB in split graphs, a subclass of chordal graphs; bipartite graphs; and planar subcubic graphs with different thresholds.

4.1 Bipartite graphs and split graphs

One of the hardness results from (CORDASCO *et al.*, 2023) for (k,ℓ)-IIB was that the problem is W[2]-hard parameterized by $|S|+\ell$ even for bipartite graphs in which the threshold of every uninfected vertex is equal to 1 or equal to the vertex degree. They reduced from the HITTING SET problem, obtaining a bipartite graph in which all the vertices from one part are uninfected while the other has both infected and uninfected vertices mixed. Figure 16 illustrates a graph that Cordasco *et al.* (2023) creates during their reduction. What we are going to do now is to make a reduction that restricts the instance to a split graph in which all vertices from the clique are infected while the independent set has only uninfected vertices, showing that the problem is W[2]-hard parameterized by ℓ even in such instances. We are going to see that this reduction also leads us to another proof of W[2]-hardness parameterized by ℓ for bipartite graphs in which one of the parts has only infected vertices while the other has only uninfected vertices, thus a more restricted instance if compared with the one from Cordasco *et al.* (2023). Moreover, the thresholds of all vertices are going to be exactly equal to their degree.

The reduction is from SET COVER. In SET COVER, we are given a universe set $\mathscr{U} = \{a_1, a_2, \ldots, a_n\}$, a collection $\mathscr{S} = \{S_1, S_2, \ldots, S_m\}$ of subsets of $\mathscr{U}$, and a parameter $c \in \mathbb{N}$, and the goal is to find $C \subset \mathscr{S}$ such that $|C| \leq c$ and the union of all sets in C is equal to $\mathscr{U}$.

Theorem 4.1. (k,ℓ)-IIB *is* W[2]*-hard parameterized by the maximum number ℓ of immunized vertices even if we restrict the problem to total-restriction instances in which the graph is a split graph, the entire clique is initially infected and the threshold of every vertex is equal to its degree.*

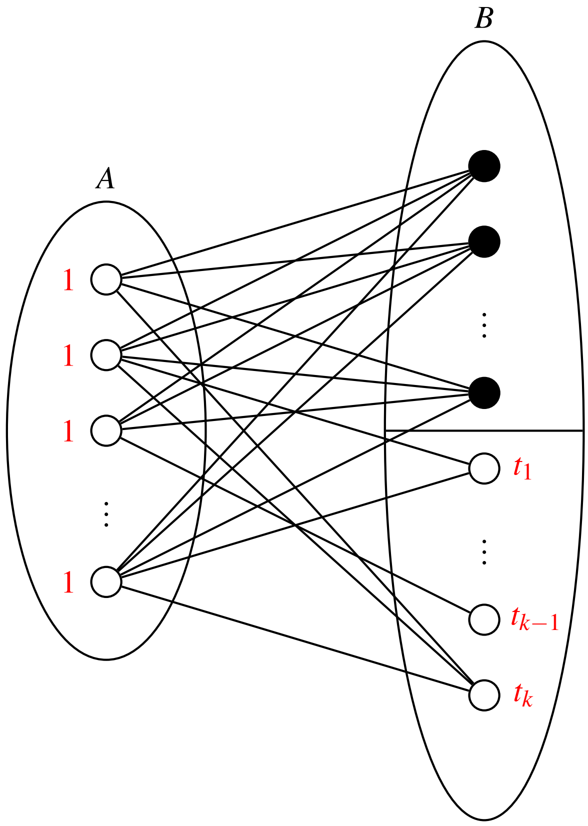


Figure 16 – Illustration of a bipartite graph obtained by the reduction in (CORDASCO *et al.*, 2023). The thresholds are shown in red and the initially infected vertices are shown in black. The part B has uninfected and infected vertices mixed.

Source: Made by the author.

Proof. Let $I = \langle \mathscr{U} = \{a_1, \ldots, a_n\}, \mathscr{S} = \{S_1, \ldots, S_m\}, c\rangle$ be an instance of SET COVER. Let's create a split graph G such that $V(G) = \mathscr{U} \cup \mathscr{S}$ and $E(G) = \{a_i S_j \mid a_i \in S_j\} \cup \{S_i S_j \mid S_i, S_j \in \mathscr{S}\}$. Define $k = |\mathscr{S}|$, $\ell = c$, and set the seed set $S = \mathscr{S}$. Finally, set $t(a_i) = |\{S_j \mid a_i \in S_j\}|$ or, equivalently, $t(a_i) = d_G(a_i)$, for all $i \in \{1, \ldots, n\}$. The entry for (k, ℓ)-IIB is going to be the t-irreversible process $\mathscr{P} = I_t(G, S)$, and the parameters k and ℓ. In other words, we want to find a total restricting set of $\mathscr{P}$ of size at most $\ell = c$. The infected vertices correspond to subsets of $\mathscr{U}$ and, by immunizing some S_j, we cover/protect all the elements of $\mathscr{U}$ that belong to S_j. The goal is then to cover/protect the whole $\mathscr{U}$, thus restricting the infection to $k = |\mathscr{S}|$ vertices. Figure 17 illustrates this reduction.

Now, suppose that $C \subset \mathscr{S}$ is a solution for SET COVER instance I. Notice that C is also a total restricting set of $\mathscr{P}$, since by immunizing the vertices from C, we have that every a_i will have at most $d_G(a_i) - 1 < t(a_i)$ infected and non-immunized neighbors. As $|C| \leq c = \ell$, we have that C is a solution for our instance of (k, ℓ)-IIB.

Conversely, let Y be a total restricting set of $\mathscr{P}$. First, let us show that if $Y \not\subseteq \mathscr{S}$, we can obtain a new total restricting set Y' of same size for $\mathscr{P}$. Suppose that $a_i \in Y$, for some $i \in \{1,\dots,n\}$. There is a set S_j such that $a_i \in S_j$. Define $Y' = (Y \setminus \{a_i\}) \cup \{S_j\}$. Observe that Y' is also a total restricting set of $\mathscr{P}$. We can keep on doing that until we get a total restricting set that is subset of $\mathscr{S}$. Assume then that $Y \subseteq \mathscr{S}$. Since Y is a total restricting set of G, every a_i is covered by at least one S_j. Thus, Y is a set cover of $\mathscr{U}$ and, as $|Y| \leq \ell = c$, we have that Y is a solution for I.

Observe that the size c of the set cover corresponds to the size ℓ of the restriction set. Thus, since SET COVER is W[2]-hard parameterized by c (DOWNEY; FELLOWS, 2012), we have that (k,ℓ)-IIB is W[2]-hard parameterized by ℓ.

□

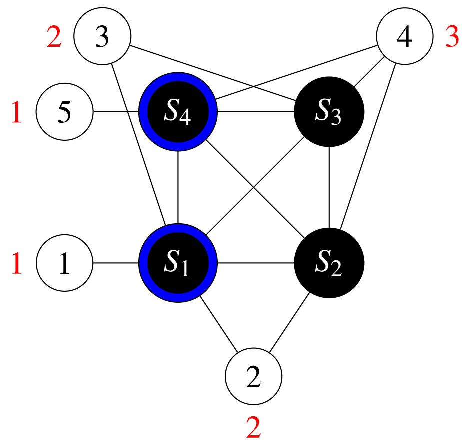


Figure 17 – Example of a split graph G obtained by our reduction to bipartite graphs and a total restricting set shown in blue.
Source: Made by the author.

Figure 17 shows an example of a graph obtained by the above reduction. In this example, suppose that $\mathscr{U} = \{1,2,3,4,5\}$, $\mathscr{S} = \{S_1 = \{1,2,3\}, S_2 = \{2,4\}, S_3 = \{3,4\}, S_4 = \{4,5\}\}$, and $c = 2$. A solution for this instance of SET COVER is $C = \{S_1, S_4\}$, which is also a total restricting set for the process.

The main observation we need to adapt this same reduction for bipartite graphs is to notice that edges between initially infected vertices do not interfere on the solution. Thus, let S be the set of initially infected vertices for an instance we create in Proposition 4.2. S is a clique, but we can turn S into an independent set of the graph by removing every edge between vertices in S and the solution remains the same. Thus, the reduction also works for bipartite graphs, which brings us to the following.

Corollary 4.2. (k,ℓ)-IIB *is* W[2]*-hard parameterized by* ℓ *even if we restrict the problem to total-restriction instances where the graph is a bipartite graph* $G=(A\cup B,E)$ *where the seed set is made by all vertices from one of the parts and the threshold of every vertex is equal to its degree.*

Figure 18 shows the bipartite graph obtained by the reduction using the same SET COVER instance from Figure 17.

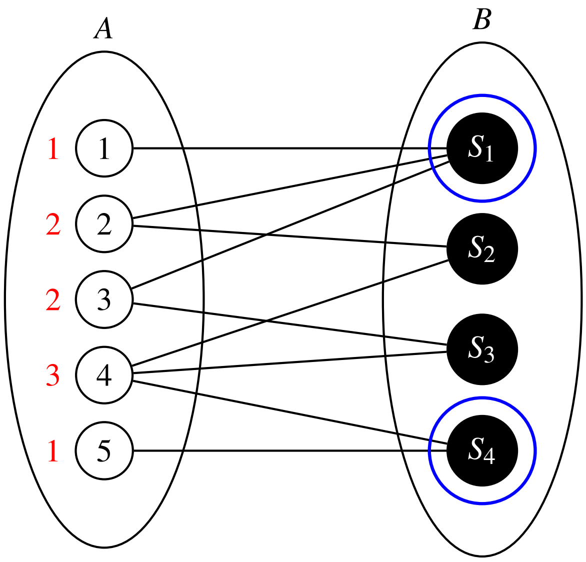


Figure 18 – Example from Figure 17 but now in a bipartite graph. The total restricting set is shown in blue.

Source: Made by the author.

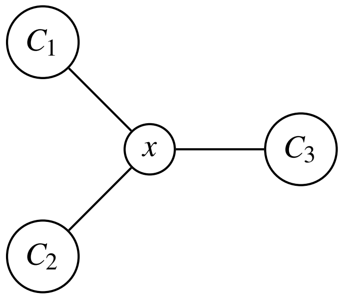
4.2 Subcubic bipartite planar graphs

We now turn our attention to another very important graph class: the planar graphs. We are going to show that (k,ℓ)-IIB is also NP-complete even in very restricted instances of this class.

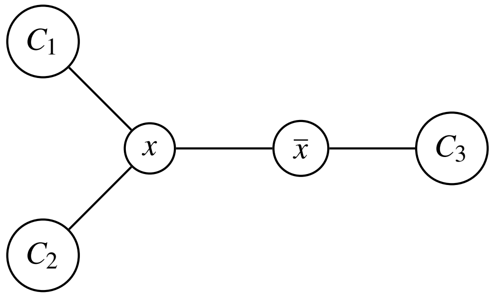
The reduction is from RESTRICTED PLANAR 3-SAT (R-PLANAR-3-SAT for short). In R-PLANAR-3-SAT, we are given a CNF formula in which every clause has at most 3 literals, and every variable occurs *exactly* 3 times: *twice* as a positive literal and *once* as a negative literal. Moreover, the clauses have size only 2 or 3 and the graph associated with the formula is a planar graph. R-PLANAR-3-SAT was proven to be NP-complete in (DAHLHAUS *et al.*, 1994).

Theorem 4.3. (k,ℓ)-IIB *is* NP*-complete even if we restrict the problem to total-restriction instances in which the graph is a subcubic planar bipartite graph.*

Proof. Let φ be an R-PLANAR-3-SAT formula, $V(\varphi)$ the set of variables of φ, and $\mathscr{C}(\varphi)$ the set of clauses of φ. In R-PLANAR-3-SAT, a graph G is associated to the formula φ, such that G is a bipartite graph with vertex set $V(\varphi) \cup \mathscr{C}(\varphi)$ and there is an edge between a variable x and a clause C if C has a literal x or $\overline{x}$. Observe that, in this graph, a variable vertex x is connected to exactly three clauses, one of them being a clause in which x is a negated literal. Let $C_{\overline{x}}$ be this clause. Then, there is an edge $xC_{\overline{x}}$ in the graph. We can subdivide this edge, obtaining a new vertex $\overline{x}$ without breaking the planarity of the graph (Figure 19). Now, in this graph, we have vertices representing the literals which are connected to their respective clauses.



(a) Illustration of x before the subdivision.



(b) Graph after the subdivision assuming that C_3 is the clause that contains the literal $\overline{x}$.

Figure 19 – Subdividing a edge to separate the literals x and $\overline{x}$ without breaking planarity.
Source: Made by the author.

We are going to create a graph $G_\varphi = (V, E)$ with thresholds t_φ and a set of initially infected vertices S_φ. For each variable $x \in V(\varphi)$, we are going to add to G_φ the gadget depicted in Figure 20. The gadget consists of two infected vertices x and $\overline{x}$, representing the positive and negative literals of variable x, respectively, together with an auxiliary vertex x' with threshold $t_\varphi(x') = 2$ that is connected with both x and $\overline{x}$.

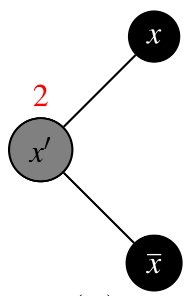


Figure 20 – The gadget for each variable of $V(\varphi)$. The infected vertices are shown in black and the auxiliary vertex is shown in gray. The threshold is shown in red.
Source: Made by the author.

Next, for each clause $C \in \mathscr{C}(\varphi)$, we are going to create an uninfected vertex C which is connected with the vertices relative to its literals. The threshold of a clause vertex is equal to the number of literals it has, i.e., if C is of the form $(x \vee y \vee z)$, then $t_\varphi(C) = 3$, otherwise if C is of the form $(x \vee y)$, then $t_\varphi(C) = 2$. Observe that this graph is exactly the graph we described in the first paragraph of this proof, with the addition of the auxiliary vertices x' for each variable x. But we can see x' as the vertex obtained from a subdivision of the edge $x\overline{x}$, thus the addition of the auxiliary vertices do not break the planarity of the graph.

With this, we have the t_φ-irreversible process $\mathscr{P}_\varphi = I_{t_\varphi}(G_\varphi, S_\varphi)$. Finally, we define $k = 2|V(\varphi)|$ and $\ell = |V(\varphi)|$, i.e., we want to find a total restricting set for $\mathscr{P}_\varphi$ with size at most $|V(\varphi)|$. The idea is that immunizing a vertex codifies the assignment of the value TRUE to the correspondent literal.

Let us exemplify such a construction. Take, for example, the following R-PLANAR-3-SAT formula:

$$\varphi = (u \vee v \vee \overline{w}) \wedge (\overline{u} \vee v \vee x) \wedge (u \vee \overline{x} \vee \overline{y}) \wedge (\overline{v} \vee x) \wedge (w \vee z \vee y) \wedge (y \vee \overline{z}) \wedge (w \vee z). \qquad (4.1)$$

We have that $V(\varphi) = \{u, v, w, x, y, z\}$ and $\mathscr{C}(\varphi) = \{C_1, C_2, \ldots, C_7\}$. Let C_i, $i \in \{1, \ldots, 7\}$, denote the i-th clause from left to right in φ. Figure 21 shows the graph obtained by this construction.

We have already argued that G_φ is planar. Let us show now that G_φ is also bipartite and subcubic.

Claim 4.4. *The graph G_φ is a subcubic bipartite planar graph.*

Proof. One can see that G_φ is subcubic by looking at the degrees for each type of vertex: a clause vertex has degree at most 3, a positive literal vertex has degree 3, a negative literal vertex has degree 2, and an auxiliary vertex has degree 2. Thus, G_φ is subcubic.

Let $A = \{x' \mid x \in V(\varphi)\}$ be the set of auxiliary vertices of G_φ. To see that G_φ is bipartite, define the partition (X, Y) of the set of vertices of G_φ such that $X = \mathscr{C}(\varphi) \cup A$ and $Y = \{x, \overline{x} \mid x \in V(\varphi)\}$. It's clear that the edges of G_φ are only from vertices in X to vertices in Y. □

Now, assume that φ has a satisfying assignment $\mathscr{S} : V(\varphi) \to \{\text{TRUE}, \text{FALSE}\}$.

Claim 4.5. $Y = (\{x \mid \mathscr{S}(x) = \text{TRUE}\} \cup \{\overline{x} \mid \mathscr{S}(x) = \text{FALSE}\})$ *is a total restricting set for $\mathscr{P}_\varphi$ and $|Y| \leq |V(\varphi)|$.*

Proof. For each variable $x \in V(\varphi)$, either $x \in Y$ or $\overline{x} \in Y$, but not both. By this, we have that $|Y| \leq |V(\varphi)|$. Moreover, this fact also implies that each vertex x', for $x \in V(\varphi)$, is safe, since its threshold is 2 and it is connected only with x and $\overline{x}$. Finally, since each clause has at least one

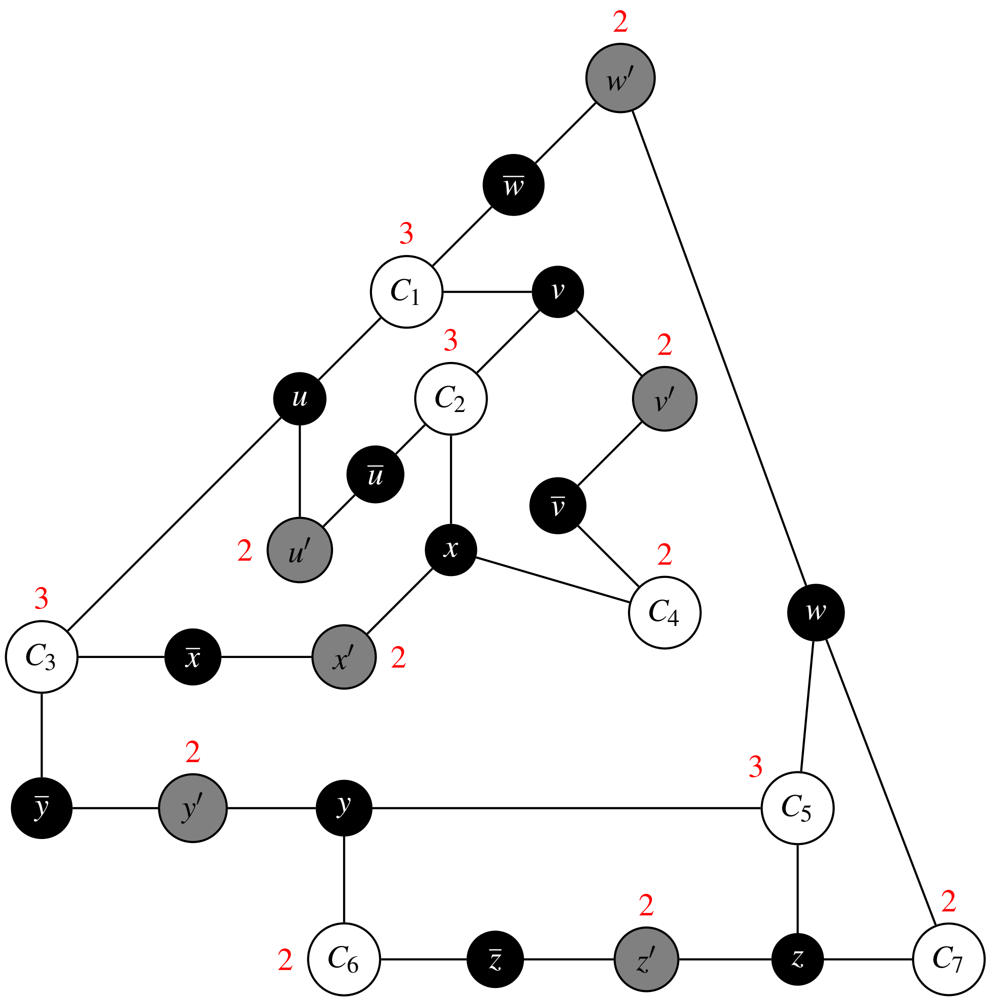


Figure 21 – The subcubic bipartite planar graph obtained in this reduction for the R-PLANAR-3-SAT formula 4.1.
Source: Made by the author.

TRUE literal, we have that every clause vertex is also protected by Y. Thus, Y is a total restricting set for $\mathscr{P}_\varphi$. □

Conversely, assume that $\mathscr{P}_\varphi$ has a total restricting set Y such that $|Y| \leq |V(\varphi)|$.

Claim 4.6. *Let $x \in V(\varphi)$. We have that exactly one vertex from the triple $(x, x', \overline{x})$ is in Y.*

Proof. Suppose that none of them are in Y. This implies that the vertex x' will be infected. Thus, Y is not a total restricting set of $\mathscr{P}_\varphi$.

Now, suppose that at least two of them are in Y. Since $|Y| \leq |V(\varphi)|$, there exists a variable, let us say y, such that both vertices correspondent to its literals, y and $\overline{y}$, are not in Y. But then y' gets infected. This means that Y is not a total restricting set of $\mathscr{P}_\varphi$. □

By the claim above, we can obtain an assignment $\mathscr{S} : V(\varphi) \to \{\text{TRUE}, \text{FALSE}\}$ for φ in the following way. For each $x \in V(\varphi)$, if $x \in Y$ we set $\mathscr{S}(x) = \text{TRUE}$, otherwise if $\overline{x} \in Y$, we set $\mathscr{S}(x) = \textsc{False}$. If $x' \in Y$, then any value assignment to variable x do not interefere in the formula satisfiability. Since each clause is protected by some immunized literal, we have that $\mathscr{S}$ satisfies the formula φ.

With this, we have proved that (k,ℓ)-IIB is NP-complete even for subcubic bipartite planar graphs. □

In this chapter, we have seen that even for very restricted graph classes it is still unlikely for (k,ℓ)-IIB to have efficient algorithms even when we parameterize by ℓ. In the next chapters, we are going to explore other classes, looking for positive results.

4.3 A brief commentary and a conjecture about the hardness of (k,ℓ)-IIB on trees

Trees are a very important and wide graph class and it is a subclass of planar graphs, chordal graphs, and bipartite graphs, all classes for which we have hardness results as seen in the last chapter. The FIREFIGHTER problem, which is very similar to (k,ℓ)-IIB, is also NP-complete for trees. Even though we do not have the game-like behavior in (k,ℓ)-IIB, which might make the problem "easier" if compared to FIREFIGHTER, we still have general thresholds and a seed set that can be big and disconnected instead of being a single vertex that might make the problem "harder".

Moreover, when we started to think and design an algorithm to solve (k,ℓ)-IIB on trees, there is a problem that appeared that is similar to the KNAPSACK problem. In the KNAPSACK problem, we are given a set of items $I = \{1,\dots,n\}$ with values $V = \{v_1,\dots,v_n\}$ and weights $P = \{p_1,\dots,p_n\}$ where v_i and p_i are the value and weight of item i, respectively, and a maximum weight W. The goal is to choose a subset of items such that the sum of their weights do not surpass the maximum weight W and the sum of their values is maximum. Suppose you have a rooted tree T with root r and we want to restrict the infection to at most k vertices. If r has p children, we may want to know how many vertices out from the maximum k infected vertices are going to be in subtrees $T_1,\dots,T_p$, where T_i is the subtree rooted at child i of r. In other words, we want to distribute the maximum number of infected vertices k or $k-1$ (in the case we let r get infected) among the subtrees. If we let less vertices get infected in a given subtree, we might need more vertices to immunize in that subtree to protect the rest. Now, we can make an analogy to the KNAPSACK in the following way. We can think on the number of vertices we immunize in a subtree as its weight and the number of vertices saved in that subtree as its value. Now, every possible way to immunize a given subtree can be viewed as an item with value and weight and we want to maximize the sum of the values, i.e., the total number of saved vertices, without surpassing the maximum number of immunized vertices k. This is then very similar to KNAPSACK.

We, however, do not have a proof of NP-completeness for (k,ℓ)-IIB restricted to trees (yet!). Since KNAPSACK and FIREFIGHTER are both NP-complete problems and, given the reasons outlined above, we make the following conjecture.

Conjecture 4.7. (k,ℓ)-IIB *is* NP*-complete even if the graph is a tree.*

5 BOUNDS AND EXACT VALUES FOR THE k-RESTRICTING NUMBER OF SOME GRAPH CLASSES

As seen in the last chapter and the work from Cordasco *et al.* (2023), (k,ℓ)-IIB is hard in bipartite, chordal, split, planar, and subcubic graphs. This Chapter shows that (k,ℓ)-IIB can be solved efficiently on some graph classes. Moreover, we also show some potentially useful observations about the problem and bounds for the k-restricting number of t-irreversible processes under some reasonable conditions.

We start first by proving the following result which is similar to Theorem 3.5 for FIREFIGHTER.

Theorem 5.1. *Let G be a graph with thresholds $t : V(G) \to \mathbb{N}$, $S \subseteq V(G)$ a seed set, $\mathscr{P} = I_t(G,S)$ a t-irreversible process, and Y a minimum total restricting set for $\mathscr{P}$. For each vertex $y \in Y$, we have that $y \in S$ or y is adjacent to some vertex from S.*

Proof. Suppose that there exists a minimum total restricting set Y for $\mathscr{P}$ that immunizes a vertex $x \in V(G)$ such that $x \notin S$ and x is not adjacent to any vertex from S. Then, $N_G[x] \not\subseteq S$. But then $Y' = Y \setminus \{x\}$ is a total restricting set of $\mathscr{P}$ with $|Y'| < |Y|$. □

Now, we show another observation about total restricting sets in graphs where the threshold of every vertex is equal to its degree.

Theorem 5.2. *Let $\mathscr{P} = I_t(G,S)$ be a t-irreversible process in a graph G with thresholds $t(v) = d_G(v)$ for each $v \in V(G)$, and seed set S. Then, there always exists a minimum total restricting set Y for $\mathscr{P}$ such that $Y \subseteq S$.*

Proof. Let v be an initially uninfected vertex from G. If the number of infected neighbors of v is strictly less than $d_G(v)$, then v is not in a minimum total restricting set, otherwise it would not be minimum. Suppose then that all neighbors of v are infected. Let Y' be a minimum total restricting set for $\mathscr{P}$ and suppose that $v \in Y'$. Since Y' is minimum, all neighbors of v are not in Y' because, if at least one neighbor of v is in Y', then v is already protected and does not need to be immunized. Let w be one such neighbor of v. Then, $Y = Y' \setminus \{v\} \cup \{w\}$ is a minimum total restriction set of $\mathscr{P}$. □

5.1 The *k*-restricting number for complete graphs with all thresholds equal

Complete graphs are a subclass of chordal graphs, a class of graphs for which (k,ℓ)-IIB is W[2]-hard as we have seen in Chapter 4. In this section, we give exact values of the k-restricting number of complete graphs. We start by looking at the case when all the thresholds of all uninfected vertices are equal.

Theorem 5.3. *Let $c = c(n)$ and $\mathscr{P} = I_t(G,S)$ be a t-irreversible process defined over a complete graph $G \cong K_n$ and thresholds $t(v) = t(u) = c$ for all $v,u \in V(G) \setminus S$. Then, for any $k \geq |S|$, the following holds:*

$$\mathfrak{R}(\mathscr{P},k) = \begin{cases} \min\{|S| - c + 1, n - k\}, & \text{if } |S| \geq c \\ 0, & \text{otherwise.} \end{cases}$$

Proof. If $|S| < c$, no vertex will ever be infected since there aren't enough infected vertices to surpass the threshold. So, in this case, $\mathfrak{R}(\mathscr{P},k) = 0$ (Figure 22a).

Suppose now that $|S| \geq c$ and, by absurd, that $\mathfrak{R}(\mathscr{P},k) < \min\{|S| - c + 1, n - k\}$. Let Y be a k-restricting set of $\mathscr{P}$ with size $\mathfrak{R}(\mathscr{P},k)$. We have three cases:

- $Y \subseteq S$. Since $\mathfrak{R}(\mathscr{P},k) < |S| - c + 1$, all vertices in $V(G) \setminus S$ are going to be infected because they all have at least c infected non-immunized neighbors.
- $Y \subseteq V(G) \setminus S$. Since $\mathfrak{R}(\mathscr{P},k) < n - k$, more than k vertices are going to be infected.
- $Y \nsubseteq V(G) \setminus S$ and $Y \nsubseteq S$. Neither the number of infected vertices decreases below c nor the infection is contained to k vertices.

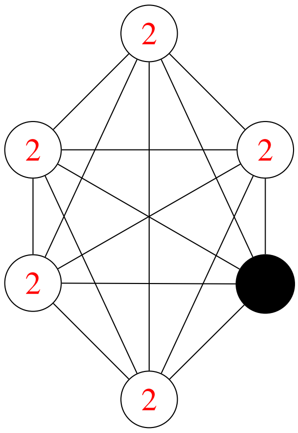
In all the three cases, we conclude that Y is not a k-restricting set of $\mathscr{P}$. □

5.2 Bounds for the *k*-restricting number for paths with thresholds 2

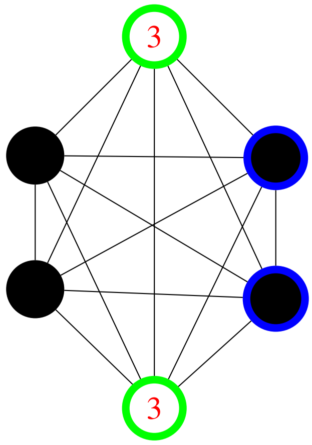
We establish the following simple bound on the k-restricting number of a process in which the graph is a path and the thresholds are all equal to 2.

Theorem 5.4. *Let $G \cong P_n$ be a path graph with n vertices and thresholds $t(v) = 2$ for each $v \in V(G)$, seed set $S \subseteq V(G)$, $\mathscr{P} = I_t(G,S)$. Then*

$$\mathfrak{R}_T(\mathscr{P}) \leq \left\lceil \frac{n}{4} \right\rceil$$



(a) When $|S| < c$, no vertex will ever be infected.



(b) Suppose $k = 4$. Then, we have two options: we either immunize $|S| - c + 1 = 2$ infected vertices (blue) or $n - k = 2$ uninfected vertices (green).

Figure 22 – Some examples of the immunization number on complete graphs when all thresholds are fixed. The thresholds are shown in red.

Proof. For $G \cong P_n$ and seed set $S \subseteq V(G)$, define an S-alternating path of G as a maximal path with 3 or more vertices of G in which the vertices alternate in pertinence to S (Figure 26). The main observation is that the only vertices that get infected are the vertices in S-alternating paths because they need 2 neighbors to get infected. Because of this, we can *split* G into S-alternating paths and consider them separately (Figure 26).

This proof follows by induction on the number of S-alternating paths of G.

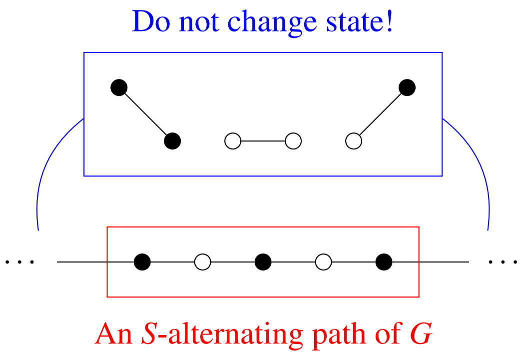
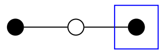


Figure 23 – The vertices in subgraphs of G that are not S-alternating paths will never change their states because all the thresholds are 2 and each such vertex has at most one infected neighbor. The infected vertices are shown in black.

Source: Made by the author.

Base case: G has exactly one S-alternating path. Let $P = v_1 v_2 \ldots v_p$ be the S-alternating path of G. If $p = 3$, we can stop the infection by immunizing any vertex from P (Figure 24). If $p = 4$, we also have that only one vertex is enough to stop the infection (Figure 25).



(a) The infection stops by immunizing any vertex. The immunized vertex is shown in blue.

(b) When the S-alternating path has length 3 and starts with an uninfected vertex, the infection does not spread.

Figure 24 – Possible options with S-alternating paths of length 3.

Source: Made by the author.

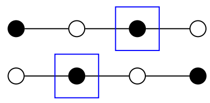


Figure 25 – When the S-alternating paths has length 4, we can inhibit the infection by immunizing only one vertex.

Source: Made by the author.

When $p \geq 5$, we can split the S-alternating path into several S-alternating paths of size 4 starting with an infected vertex, perhaps with the exception of the last one, which will have length $p \mod 4$. For each piece of the path, we can immunize only one vertex, inhibiting the infection in that piece. Doing this on all pieces, we have that $\mathfrak{R}(\mathscr{P},k) \leq \lceil \frac{n}{4} \rceil$.

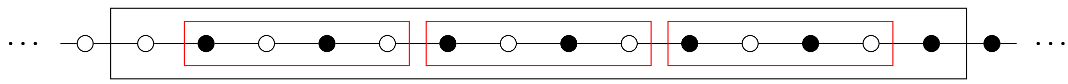


Figure 26 – Splitting a bigger S-alternating path into pieces of length 4 starting with infected vertices.

Source: Made by the author.

Now, let G be a path with thresholds 2 and $p > 1$ S-alternating paths. Then, G has two consecutive vertices u and v between two S-alternating paths such that both u and v have the same state. By removing the edge uv from G, we split G into two paths P_1 and P_2 with thresholds 2 and less S-alternating paths. Let $\mathscr{P}_1 = I_t(P_1, S \cap V(P_1))$ and $\mathscr{P}_2 = I_t(P_2, S \cap V(P_2))$.

By the inductive hypothesis, we have that for any k, $\mathfrak{R}(\mathscr{P}_1,k) \leq \left\lceil \frac{|V(P_1)|}{4} \right\rceil$ and $\mathfrak{R}(\mathscr{P}_2,k) \leq \left\lceil \frac{|V(P_2)|}{4} \right\rceil$. From this, we have:

$$\begin{aligned}
\mathfrak{R}(\mathscr{P},k) &\leq \mathfrak{R}(\mathscr{P}_1,k) + \mathfrak{R}(\mathscr{P}_2,k) \\
&\leq \left\lceil \frac{|V(P_1)|}{4} \right\rceil + \left\lceil \frac{|V(P_2)|}{4} \right\rceil \\
&\leq \left\lceil \frac{n}{4} \right\rceil.
\end{aligned}$$

□

5.3 Upper bounds on the k-restricting number based on the pathwidth

Several NP-complete problems have been solved when parameterized by the treewidth. This is one of the reasons why the treewidth is such a studied and important parameter of graphs. Several graph classes were shown to have bounded treewidth, leading to strong polynomial algorithms to solve a number of problems in these classes. (k,ℓ)-IIB, however, is W[1]-hard parameterized by the treewidth as shown by Cordasco *et al.* (2023), suggesting that we cannot solve (k,ℓ)-IIB in FPT time parameterized by this parameter. Another parameter that is closely related to the treewidth is the pathwidth.

Path decompositions are a special case of tree decompositions in which the subjacent tree of the decomposition is a path. The smallest width from a path decomposition of a graph is called the pathwidth of the graph, denoted by $pw(G)$. Of course, $tw(G) \leq pw(G)$, and then every graph with bounded treewidth also has bounded pathwidth. This list includes trees, caterpillars, planar and outerplanar graphs, series-parallel graphs, and Halin graphs. There are also some graph classes in which $pw(G) = tw(G)$, such as cographs.

We show next that, if some conditions about the seed set and the maximum number of allowed infected vertices hold, we can obtain a small (although possibly not optimal) restriction set which depends only on the pathwidth of the input graph.

5.3.1 *Definitions*

Let $\mathscr{T} = (X,T)$ be a tree decomposition of a graph G. Recall that, for a vertex v of G, the bags of $\mathscr{T}$ in which v is in induce a subtree T_v of T. When $\mathscr{T}$ is a path decomposition of a graph, then T_v is a subpath of T.

Let G be a graph, $\mathscr{P} = I_t(G,S)$ a t-irreversible threshold on G, and $\mathscr{T} = (X,T)$ be a path decomposition of G. For a vertex $v \in V(G)$, let us denote by $A^v_{\mathscr{T}}(\mathscr{P})$ the set of final active vertices of $\mathscr{P}$ that share a bag of $\mathscr{T}$ with v. That is:

$$A^v_{\mathscr{T}}(\mathscr{P}) = (\bigcup_{i \in V(T_v)} X_i) \cap A^*(\mathscr{P}).$$

For a graph G with thresholds $t : V(G) \to \mathbb{N}$ and seed set $S \subseteq V(G)$, let us define a transformation of G we call the *one-seed transformation*. We apply the one-seed transformation on (G,t,S), obtaining $(G',t',\{s\})$ and we call G' the *one-seed graph*, $t' : V(G') \to \mathbb{N}$ the *one-seed thresholds*, and s the *one-seed vertex* of (G,t,S).

The one-seed graph G' of (G,t,S) is obtained by identifying all the initially active vertices S in G. Let s be the vertex that is generated by such an identification, then s is the one-seed vertex. Now, we define the thresholds t' as follows.

$$t'(v) = \begin{cases} \max\{t(v) - |S \cap N_G(v)| + 1, 1\} & \text{if } v \in V(G) \setminus S \text{, and } N_G(v) \cap S \neq \emptyset, \\ 1, \text{ if } v = s \\ t(v), \text{ otherwise.} \end{cases}$$

Now, let $\mathscr{P}' = I_{t'}(G', \{s\})$ be the t'-irreversible process on the one-seed graph G' and one-seed thresholds t' of (G,t,S). Observe that $A_\tau(\mathscr{P}') \setminus \{s\} = A_\tau(\mathscr{P}) \setminus S$, for any $\tau \geq 0$. That is, at each time step, the vertices that get active in $\mathscr{P}'$ are that same vertices that get active in $\mathscr{P}$ at the same time step.

Now, let us state the condition that we need to prove our result. Let $\mathscr{P} = I_t(G,S)$ be a t-irreversible process and $\mathscr{P}' = I_{t'}(G', \{s\})$ be the t-irreversible process in which $(G', t', \{s\})$ is the one-seed transformation of (G,t,S). Let $\mathscr{T}' = (X,T)$ be the path decomposition of the one-seed graph G'. Now, for $k \geq |S|$, we state the Condition 5.5.

Condition 5.5. $k - |S| + 1 \geq |A^s_{\mathscr{T}'}(\mathscr{P}')|$.

Intuitively, Condition 5.5 means that the maximum number of allowed infected vertices (i.e., the parameter k) is at least the number of vertices that share a bag with s and that gets infected during the process.

5.3.2 Proof of bounds

Now, we are ready to state our result and prove it.

Theorem 5.6. *Let G be a graph with thresholds $t : V(G) \to \mathbb{N}$, $S \subseteq V(G)$, and $(G', t', \{s\})$ be the one-seed transformation of (G,t,S). Let $\mathscr{T}' = (X,T)$ be a path decomposition of G' with width $pw(G')$. Suppose that Condition 5.5 holds for $\mathscr{P} = I_t(G,S)$, $\mathscr{P}' = I_{t'}(G', \{s\})$, $\mathscr{T}'$ and $k \geq |S|$. Then*

$$\mathfrak{R}(\mathscr{P},k) \leq 2(pw(G') + 1).$$

Proof. Let X_i and X_f be the bags from $\mathscr{T}'$ which corresponds to the endpoints of the subpath T_s. Let X_{i-1} and X_{f+1} denote the predecessor and successor bags of X_i and X_f, respectively, if they exist. If X_{i-1} or X_{f+1} do not exist, let they be the empty set. Figure 27 illustrates this.

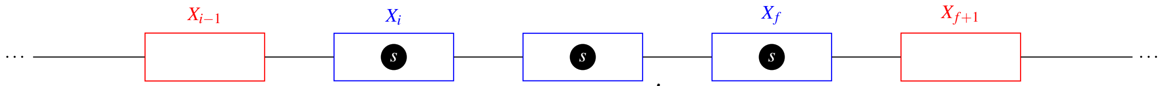


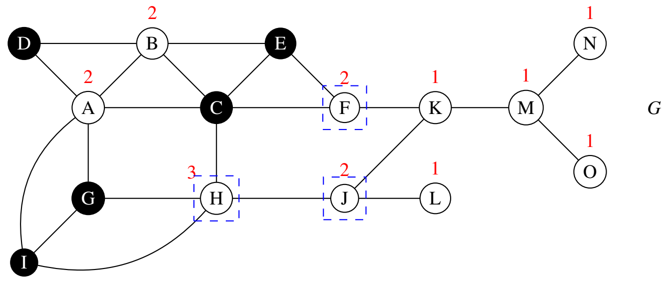
Figure 27 – Illustration of a path decomposition of G' with T_s in blue and X_{i-1}, X_{f+1} in red.
Source: Made by the author.

Define $Y = X_{i-1} \cup X_{f+1}$. Recall that every bag of a tree decomposition is a separator of the graph. Thus, by immunizing the vertices from Y, we are preventing every vertex outside of the T_s bags from being infected. Moreover, Y retains the infection to exactly $|A^s_{\mathscr{T}'}(\mathscr{P}')|$ vertices. Thus, Y is a $|A^s_{\mathscr{T}'}(\mathscr{P}')|$-restricting set of $\mathscr{P}'$.

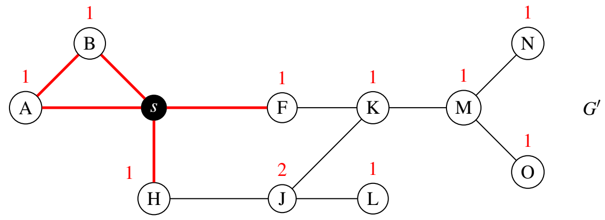
That means that, in $\mathscr{P}$, the same set Y will restrict the infection to $|A^s_{\mathscr{T}'}(\mathscr{P}')| + |S| - 1$ because, in $\mathscr{P}$, the seed set is the whole set S instead of a single vertex. Since Condition 5.5 holds, Y is a k-restricting set of $\mathscr{P}$. Moreover, the size of Y is $|Y| = |X_{i-1}| + |X_{f+1}| \leq 2(pw(G') + 1)$. This concludes our proof. □

If we receive a path decomposition of the graph as input, the proposition above gives us an algorithm to find such restriction set. In the example of Figure 28, we show a restriction set obtained from the path decomposition of the one-seed graph. Deciding if a graph has a path decomposition of width w is NP-hard for planar graphs, chordal graphs, and distance-hereditary bipartite graphs, among other classes (BODLAENDER, 1993). However, it is FPT parameterized by the width w and can be computed in polynomial time for forests, interval graphs, and some other graph classes with bounded treewidth. There are also some known upper bounds on $pw(G)$, which helps us derive better upper bounds for some graph classes by using Theorem 5.6 when Condition 5.5 holds.

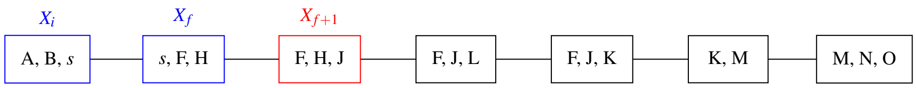
However, since the one-seed graph G' is obtained from G by identifying a set of vertices, the pathwidth of G' might differ a lot from the pathwidth of G. In Theorem 5.6 we are always dealing with G', but we want to say more about G by using G'. Few graph classes are closed under the identification of vertices and, as far as we know, none also have bounded treewidth or pathwidth. On the other hand, there are several graph classes closed under the contraction of edges that have bounded treewidth and pathwidth, such as trees and planar graphs. Then, if we have $G[S]$ connected in these graph classes, which is a reasonable assumption since the infection spreads “through the edges”, we know that G' is also from the same class because the identification of vertices from the seed set would be analogous to edge contractions. This is what brings us to the next results.



(a) Graph G with thresholds in red and the restriction set $Y = \{F, H, J\}$ obtained from the path decomposition of G' highlighted in blue. We can see that it is possible to further decrease the size of this immunizing set by removing J, obtaining a 7-restricting set for G of size 2.



(b) Graph G' obtained with the one-seed transformation of $(G, t, S = \{C, D, E, G, I\})$ and the subgraph induced by the vertices from the bags of T_s in red.



(c) The path decomposition of G' with X_i, X_f, and X_{f+1} highlighted.

Figure 28 – Illustration of the process to obtain a restriction set as in Theorem 5.6.

Source: Made by the author.

5.3.3 Application in classes that are closed under edge contractions

If a graph G belongs to a class which is closed under edge contractions and have bounded pathwidth, we can state Theorem 5.6 in terms of $pw(G)$ rather than $pw(G')$. Fortunately, there are several such graph classes which have bounded treewidth or pathwidth. In this subsection, we are going to show bounds for specific classes by using Theorem 5.6.

First, let us state the following result from Korach and Solel (1993).

Theorem 5.7. *(Korach and Solel (1993)) Let G be a graph with treewidth* $tw(G)$*. Then,* $pw(G) \in O(tw(G) \cdot \log n)$.

To get to the result above, Korach and Solel (1993) show that

$$pw(G) \leq 2 \cdot tw(G)(\log n + 2).$$

We are going to use this expression next.

Corollary 5.8. *Let T be a tree with n vertices, thresholds $t : V(G) \to \mathbb{N}$, seed set $S \subseteq V(T)$ such that $T[S]$ is connected, $\mathscr{P} = I_t(T,S)$, and $k \in \mathbb{N}$ such that Condition 5.5 holds. Then*

$$\mathfrak{R}(\mathscr{P},k) \leq 4 \cdot (\log(n - |S| + 1)) + 10.$$

Proof. Since S is connected, we have that the graph T' obtained by the identification of S is also a tree, thus $tw(T') = 1$ and, by the result from Korach and Solel (1993), we have that $pw(T') \leq 2 \cdot tw(T')(\log(|V(T')|) + 2) = 2 \cdot (\log(n - |S| + 1) + 2)$. By Theorem 5.6, we get that $\mathfrak{R}(\mathscr{P},k) \leq 2 \cdot (2 \cdot (\log(n - |S| + 1) + 2) + 1) = 4 \cdot (\log(n - |S| + 1)) + 10$. □

Trees are connected graphs that exclude K_3 as a minor. Now we show similar bounds for other two important graph classes: outerplanar, which excludes K_4 and $K_{2,3}$ as minors; and planar, which excludes K_5 and $K_{3,3}$ as minors. To show our bounds, we use the following result from Bodlaender (1998).

Theorem 5.9. *(Bodlaender (1998)) Let G be a graph on n vertices. If G is outerplanar, then $tw(G) \leq 2$. If G is planar, then $pw(G) \in O(\sqrt{n})$.*

Corollary 5.10. *Let G be an outerplanar graph with n vertices, thresholds $t : V(G) \to \mathbb{N}$, seed set $S \subseteq V(G)$ such that $G[S]$ is connected, $\mathscr{P} = I_t(G,S)$, and $k \in \mathbb{N}$ such that Condition 5.5 holds. Then*

$$\mathfrak{R}(\mathscr{P},k) \in O(\log(n - |S| + 1)).$$

Proof. Since $G[S]$ is connected, we have that the one-seed graph G' is also outerplanar. Thus, $tw(G') \leq 2$ by Theorem 5.9. Since Condition 5.5 holds, we have by Theorem 5.6 and Theorem 5.7 that $\mathfrak{R}(\mathscr{P},k) \leq 2(pw(G') + 1) \in O(tw(G') \cdot \log(|V(G')|))$. As $|V(G')| = n - |S| + 1$, we conclude that $\mathfrak{R}(\mathscr{P},k) \in O(\log(n - |S| + 1))$. □

Corollary 5.11. *Let G be a planar graph with n vertices, thresholds $t : V(G) \to \mathbb{N}$, seed set $S \subseteq V(G)$ such that $G[S]$ is connected, $\mathscr{P} = I_t(G,S)$, and $k \in \mathbb{N}$ such that condition 5.5 holds. Then*

$$\mathfrak{R}(\mathscr{P},k) \in O(\sqrt{n - |S| + 1}).$$

Proof. Since $G[S]$ is connected, we have that the one-seed graph G' is also planar. Thus, $pw(G') \in O(\sqrt{|V(G')|}) = O(\sqrt{n - |S| + 1})$ by Theorem 5.9. Since Condition 5.5 holds, then by Theorem 5.6 we conclude that $\mathfrak{R}(\mathscr{P},k) \in O(\sqrt{n - |S| + 1})$. □

Bodlaender (1998) has another result for planar graphs in terms of its radius. We can also use this result in a similar fashion to what we have been doing in this last few results.

Theorem 5.12. *(Bodlaender (1998)) The treewidth of a planar graph with radius d is at most* $3d+1$.

Corollary 5.13. *Let G be a planar graph with n vertices and radius d, thresholds $t : V(G) \to \mathbb{N}$, seed set $S \subseteq V(G)$ such that $G[S]$ is connected, $\mathscr{P} = I_t(G,S)$, and $k \in \mathbb{N}$ such that Condition 5.5 holds. Then*

$$\mathfrak{R}(\mathscr{P},k) \leq 4 \cdot \Big((3d+1) \cdot \log(n - |S| + 1) + 1\Big).$$

Proof. Since $G[S]$ is connected, we have that the one-seed graph G' is also planar. Moreover, the radius of G' is no more than d and, when we identify vertices, we might also decrease distances between vertices. Thus, by Theorems 5.12 and 5.7, and Theorem 5.6, we have that $\mathfrak{R}(\mathscr{P},k) \leq 2 \cdot (2 + 2 \cdot (3d+1) \cdot \log(n - |S| + 1)) = 4 \cdot \Big((3d+1) \cdot \log(n - |S| + 1) + 1\Big)$. □

Let us generalize results such as these. We are going to use the following results from Robertson and Seymour (1986) and Eppstein (2000).

Theorem 5.14. *(Robertson and Seymour (1986)) For every planar graph H there is an integer $t(H) > 0$ such that every graph in the class $\mathscr{C} = exc(H)$ has treewidth at most $t(H)$.*

Theorem 5.15. *(Eppstein (2000)) Let $\mathscr{C}$ be a graph class. We say that $\mathscr{C}$ has the diameter-treewidth property if every graph G from $\mathscr{C}$ has $tw(G) \in O(d(G))$, where $d(G)$ is the diameter of G. $\mathscr{C}$ has the diameter-treewidth property if and only if $\mathscr{C}$ does not contain every apex graph.*

Corollary 5.16. *Let H be a planar graph and $\mathscr{C} = exc(H)$ be the graph class that excludes H as a minor. For a graph G in $\mathscr{C}$ with n vertices, thresholds $t : V(G) \to \mathbb{N}$, seed set $S \subseteq V(G)$ such that $G[S]$ is connected, $\mathscr{P} = I_t(G,S)$, and $k \in \mathbb{N}$ such that Condition 5.5 holds, we have that*

$$\mathfrak{R}(\mathscr{P},k) \in O(\log(n - |S| + 1)).$$

Proof. Since $\mathscr{C}$ is minor-closed and $G[S]$ is connected, the one-seed graph G' is also in $\mathscr{C}$ and, by Theorem 5.14, $tw(G')$ is bounded. By using Theorem 5.7 and Theorem 5.6, we have that $\mathfrak{R}(\mathscr{P},k) \leq 2 \cdot (pw(G') + 1) \in O(tw(G') \cdot \log(n - |S| + 1)) = O(\log(n - |S| + 1))$. □

Corollary 5.17. *Let $\mathscr{C}$ be a minor-closed graph class with the diameter-treewidth property. For a graph G in $\mathscr{C}$ with n vertices and diameter d, thresholds $t : V(G) \to \mathbb{N}$, seed set $S \subseteq V(G)$ such that $G[S]$ is connected, $\mathscr{P} = I_t(G,S)$, and $k \in \mathbb{N}$ such that Condition 5.5 holds, we have that*

$$\mathfrak{R}(\mathscr{P},k) \in O(d \cdot \log(n - |S| + 1)).$$

Proof. Since $\mathscr{C}$ is minor-closed and $G[S]$ is connected, the one-seed graph G' is also in $\mathscr{C}$ and, by Theorem 5.15, $tw(G') \in O(d(G'))$. By using Theorem 5.7 and Theorem 5.6, we have that $\mathfrak{R}(\mathscr{P},k) \leq 2 \cdot (pw(G') + 1) \in O(tw(G') \cdot \log(n - |S| + 1)) = O(d \cdot \log(n - |S| + 1))$. □

We now finish this compilation of corollaries from Theorem 5.6 by showing a result that does not depend on the graph induced by S being connected nor on minor-closed graph classes. Rather, it says to us that the fewer the edges the easier it is to contain the infection. For this last corollary, we rely on the following result from Kneis *et al.* (2009).

Theorem 5.18. *(Kneis et al. (2009)) Let G be a graph with m edges. Then, $pw(G) \leq \frac{m}{5.217} + 3$.*

Corollary 5.19. *Let G be a graph with m edges, thresholds $t : V(G) \to \mathbb{N}$, seed set $S \subseteq V(G)$, $\mathscr{P} = I_t(G,S)$, and $k \in \mathbb{N}$ such that Condition 5.5 holds. then,*

$$\mathfrak{R}(\mathscr{P},k) \leq \frac{m}{2.607} + 8$$

.

6 CONCLUSIONS AND FUTURE WORK

In this work, we studied the problem (k,ℓ)-INFLUENCE IMMUNIZATION BOUNDING BOUNDING, introduced by Cordasco *et al.* (2023). In Chapter 4, we showed some hardness results. The problem remains W[2]-hard or NP-hard even if we restrict the graph to be in very restricted classes. Moreover, all our hardness results are for the specific case of (k,ℓ)-IIB when we set $k = |S|$, i.e., we want to restrict the infection only to the already infected vertices, which we call *total restriction*. We also conjectured that the general case of (k,ℓ)-IIB is NP-complete for trees. In Chapter 5, we showed that (k,ℓ)-IIB can be solved efficiently in complete graphs when all thresholds are equal.

These results allow us to provide a new complexity scenario for the problem, complementing Table 1.

Graph Class	(k,ℓ)**-IIB General Case**	(k,ℓ)**-IIB Total Restriction**
General Graphs	W-hard (several parameters) (CORDASCO *et al.*, 2023)	W[2]-hard (ℓ) [4.1] NP-complete [4.3]
Trees	**Open problem** [Conjecture 4.7]	**Open problem**
Complete	P for $t(v) = c \quad \forall v$ [5.3] **Open** for general thresholds	P for $t(v) = c \quad \forall v$ [5.3] **Open** for general thresholds
Split	W[2]-hard (ℓ) [4.1]	W[2]-hard (ℓ) [4.1]
Chordal	W[2]-hard (ℓ) [4.1]	W[2]-hard (ℓ) [4.1]
Bipartite	W[2]-hard (ℓ) [4.2]	W[2]-hard (ℓ) [4.2]
Planar	NP-complete [4.3]	NP-complete [4.3]
Subcubic	NP-complete [4.3]	NP-complete [4.3]

Table 2 – (k,ℓ)-IIB complexity table with our results for some graph classes.

We also proved a simple bound for the number of immunized vertices necessary to restrict the infection in paths with thresholds equal to 2 and bounds for minor-closed graph classes such as planar graphs, outerplanar graphs, and trees when k is sufficiently big and the seed set is connected.

As future work, we may want to keep studying other graph classes and parameters, searching for polynomial algorithms or FPT algorithms for (k,ℓ)-IIB. Interesting classes to be explored are subclasses of those in Table 2 and other similar classes. For example, paths, caterpillars, or regular graphs seem to be good classes to be explored. We think that the general case of (k,ℓ)-IIB is in P for paths. FIREFIGHTER, which is a very similar problem to (k,ℓ)-IIB, can be solved efficiently by a simple greedy algorithm in caterpillar, thus it might be a good class to also explore. Both paths and caterpillars are trees. Regular graphs such as grids are also a class with a very specific structure and well behaved structures might make it easier to restrict infections. We also want to fill the gaps in Table 2 by showing if the general case of (k,ℓ)-IIB is NP-complete or not for trees and complete graphs, also proving or disproving our Conjecture 4.7.

From the parameterized point of view, there are several parameters to be studied other than those studied by Cordasco *et al.* (2023) and showed in Table 1. Modulators are parameters that measure "how distant" a graph is from being in a certain class. For example, the vertex cover number of a graph measures how many vertices we need to remove from the graph for it to become an independent set. As future work, we want to study the problem parameterized by modulator parameters and/or other parameters, seeking for parameterized hardness results or FPT algorithms. We can also investigate how good the bounds established by Condition 5.5 are by investigating the maximum number of bags that a vertex might be in a path decomposition or in a tree decomposition of a graph.

The version of the problem for directed graphs, in which only the in-neighborhood of a vertex influences its state, is not studied yet. Another variation that is not studied is the version for reversible process, in which active vertices can get inactive and vice-versa. We want to introduce both variations of (k,ℓ)-IIB and study them as future work.

REFERENCES

AHMAD, M.; ALI, S.; TARIQ, J.; KHAN, I.; SHABBIR, M.; ZAMAN, A. Combinatorial trace method for network immunization. **Information Sciences**, Elsevier BV, v. 519, p. 215228, May 2020. ISSN 0020-0255. Available at: http://dx.doi.org/10.1016/j.ins.2020.01.037.

AIM Minimum Rank Special Graphs Work Group. Zero forcing sets and the minimum rank of graphs. **Linear Algebra and its Applications**, v. 428, n. 7, p. 1628–1648, 2008. ISSN 0024-3795. Available at: https://www.sciencedirect.com/science/article/pii/S0024379507004624.

AMOS, D.; CARO, Y.; DAVILA, R.; PEPPER, R. Upper bounds on the k-forcing number of a graph. **Discrete Applied Mathematics**, Elsevier BV, v. 181, p. 110, Jan. 2015. ISSN 0166-218X. Available at: http://dx.doi.org/10.1016/j.dam.2014.08.029.

ASPNES, J.; CHANG, K.; YAMPOLSKIY, A. Inoculation strategies for victims of viruses and the sum-of-squares partition problem. **Journal of Computer and System Sciences**, Elsevier BV, v. 72, n. 6, p. 10771093, Sep. 2006. ISSN 0022-0000. Available at: http://dx.doi.org/10.1016/j.jcss.2006.02.003.

BANERJEE, A. V. A simple model of herd behavior. **The quarterly journal of economics**, MIT Press, v. 107, n. 3, p. 797–817, 1992.

BARABÁSI, A.-L.; PÓSFAI, M. **Network science**. Cambridge: Cambridge University Press, 2016. ISBN 9781107076266 1107076269. Available at: http://barabasi.com/networksciencebook/.

BAZGAN, C.; CHOPIN, M.; CYGAN, M.; FELLOWS, M. R.; FOMIN, F. V.; van Leeuwen, E. J. Parameterized complexity of firefighting. **Journal of Computer and System Sciences**, v. 80, n. 7, p. 1285–1297, 2014. ISSN 0022-0000. Available at: https://www.sciencedirect.com/science/article/pii/S0022000014000245.

BAZGAN, C.; CHOPIN, M.; RIES, B. The firefighter problem with more than one firefighter on trees. **Discrete Applied Mathematics**, Elsevier BV, v. 161, n. 78, p. 899908, May 2013. ISSN 0166-218X. Available at: http://dx.doi.org/10.1016/j.dam.2012.11.011.

BENEVIDES, F.; CAMPOS, V.; DOURADO, M. C.; SAMPAIO, R. M.; SILVA, A. The maximum time of 2-neighbour bootstrap percolation: Algorithmic aspects. **European Journal of Combinatorics**, Elsevier BV, v. 48, p. 8899, Aug. 2015. ISSN 0195-6698. Available at: http://dx.doi.org/10.1016/j.ejc.2015.02.012.

BESSY, S.; BONATO, A.; JANSSEN, J.; RAUTENBACH, D.; ROSHANBIN, E. Burning a graph is hard. **Discrete Applied Mathematics**, Elsevier, v. 232, p. 73–87, 2017.

BIKHCHANDANI, S.; HIRSHLEIFER, D.; WELCH, I. A theory of fads, fashion, custom, and cultural change as informational cascades. **Journal of Political Economy**, University of Chicago Press, v. 100, n. 5, p. 9921026, Oct. 1992. ISSN 1537-534X. Available at: http://dx.doi.org/10.1086/261849.

BODLAENDER, H. L. A tourist guide through treewidth. **Acta cybernetica**, University of Szeged, v. 11, n. 1-2, p. 1–21, 1993.

BODLAENDER, H. L. A partial k-arboretum of graphs with bounded treewidth. **Theor. Comput. Sci.**, Elsevier BV, v. 209, n. 1-2, p. 1–45, Dec. 1998.

BONATO, A. A survey of graph burning. **arXiv preprint arXiv:2009.10642**, 2020.

BONATO, A.; JANSSEN, J.; ROSHANBIN, E. Burning a graph as a model of social contagion. In: SPRINGER. **Algorithms and Models for the Web Graph: 11th International Workshop, WAW 2014, Beijing, China, December 17-18, 2014, Proceedings 11**. [S.l.], 2014. p. 13–22.

BONATO, A.; JANSSEN, J.; ROSHANBIN, E. How to burn a graph. **Internet Mathematics**, Taylor & Francis, v. 12, n. 1-2, p. 85–100, 2016.

BONDY, J. A.; MURTY, U. S. R. **Graph Theory**. New York: Springer, 2008. ISBN 9781846289699 1846289696.

BRAUNSTEIN, A.; DALLASTA, L.; SEMERJIAN, G.; ZDEBOROVá, L. Network dismantling. **Proceedings of the National Academy of Sciences**, Proceedings of the National Academy of Sciences, v. 113, n. 44, p. 1236812373, Oct. 2016. ISSN 1091-6490. Available at: http://dx.doi.org/10.1073/pnas.1605083113.

BURGARTH, D.; BOSE, S.; BRUDER, C.; GIOVANNETTI, V. Local controllability of quantum networks. **Physical Review A**, American Physical Society (APS), v. 79, n. 6, Jun. 2009. ISSN 1094-1622. Available at: http://dx.doi.org/10.1103/PhysRevA.79.060305.

CAI, L.; VERBIN, E.; YANG, L. Firefighting on trees:(1- 1/e)–approximation, fixed parameter tractability and a subexponential algorithm. In: SPRINGER. **International symposium on algorithms and computation**. [S.l.], 2008. p. 258–269.

CENTENO, C. C.; DOURADO, M. C.; PENSO, L. D.; RAUTENBACH, D.; SZWARCFITER, J. L. Irreversible conversion of graphs. **Theoretical Computer Science**, Elsevier BV, v. 412, n. 29, p. 36933700, Jul. 2011. ISSN 0304-3975. Available at: http://dx.doi.org/10.1016/j.tcs.2011.03.029.

CHAKRABARTI, D.; WANG, Y.; WANG, C.; LESKOVEC, J.; FALOUTSOS, C. Epidemic thresholds in real networks. **ACM Transactions on Information and System Security**, Association for Computing Machinery (ACM), v. 10, n. 4, p. 126, Jan. 2008. ISSN 1557-7406. Available at: http://dx.doi.org/10.1145/1284680.1284681.

CHEN, C.; TONG, H.; PRAKASH, B. A.; TSOURAKAKIS, C. E.; ELIASSI-RAD, T.; FALOUTSOS, C.; CHAU, D. H. Node immunization on large graphs: Theory and algorithms. **IEEE Transactions on Knowledge and Data Engineering**, Institute of Electrical and Electronics Engineers (IEEE), v. 28, n. 1, p. 113126, Jan. 2016. ISSN 1041-4347. Available at: http://dx.doi.org/10.1109/TKDE.2015.2465378.

CHEN, N. On the approximability of influence in social networks. **SIAM Journal on Discrete Mathematics**, Society for Industrial & Applied Mathematics (SIAM), v. 23, n. 3, p. 14001415, Jan. 2009. ISSN 1095-7146. Available at: http://dx.doi.org/10.1137/08073617X.

CHEN, P.-A.; DAVID, M.; KEMPE, D. Better vaccination strategies for better people. In: **Proceedings of the 11th ACM conference on Electronic commerce**. ACM, 2010. (EC 10), p. 179188. Available at: http://dx.doi.org/10.1145/1807342.1807370.

CHLEBíKOVá, J.; CHOPIN, M. The firefighter problem: Further steps in understanding its complexity. **Theoretical Computer Science**, Elsevier BV, v. 676, p. 4251, May 2017. ISSN 0304-3975. Available at: http://dx.doi.org/10.1016/j.tcs.2017.03.004.

COLEMAN, J.; KATZ, E.; MENZEL, H. The diffusion of an innovation among physicians. **Sociometry**, JSTOR, v. 20, n. 4, p. 253, Dec. 1957.

COLIZZA, V.; BARRAT, A.; BARTHÉLEMY, M.; VESPIGNANI, A. The role of the airline transportation network in the prediction and predictability of global epidemics. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 103, n. 7, p. 2015–2020, 2006.

CONWAY, J. *et al.* The game of life. **Scientific American**, v. 223, n. 4, p. 4, 1970.

CORDASCO, G.; GARGANO, L.; RESCIGNO, A. A. Immunization in the threshold model: A parameterized complexity study. **Algorithmica**, Springer Science and Business Media LLC, v. 85, n. 11, p. 33763405, Apr. 2023. ISSN 1432-0541. Available at: http://dx.doi.org/10.1007/s00453-023-01118-y.

CUKIERMAN, A. Asymmetric information and the electoral momentum of public opinion polls. **Public Choice**, Springer, v. 70, n. 2, p. 181–213, 1991.

CYGAN, M.; FOMIN, F. V.; LEEUWEN, E. J. van. Parameterized complexity of firefighting revisited. In: MARX, D.; ROSSMANITH, P. (Ed.). **Parameterized and Exact Computation**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 13–26. ISBN 978-3-642-28050-4.

DAHLHAUS, E.; JOHNSON, D. S.; PAPADIMITRIOU, C. H.; SEYMOUR, P. D.; YANNAKAKIS, M. The complexity of multiterminal cuts. **SIAM Journal on Computing**, Society for Industrial & Applied Mathematics (SIAM), v. 23, n. 4, p. 864894, Aug. 1994. ISSN 1095-7111. Available at: http://dx.doi.org/10.1137/s0097539792225297.

DAS, B.; ENDURI, M. K.; KIYOMI, M.; MISRA, N.; OTACHI, Y.; REDDY, I. V.; YOSHIMURA, S. On structural parameterizations of firefighting. **Theoretical Computer Science**, Elsevier BV, v. 782, p. 7990, Aug. 2019. ISSN 0304-3975. Available at: http://dx.doi.org/10.1016/j.tcs.2019.02.032.

DEGROOT, M. H. Reaching a consensus. **Journal of the American Statistical Association**, Informa UK Limited, v. 69, n. 345, p. 118121, Mar. 1974. ISSN 1537-274X. Available at: http://dx.doi.org/10.1080/01621459.1974.10480137.

DERRIDA, B. Dynamics of automata, spin glasses and neural network models. In: **Nonlinear evolution and chaotic phenomena**. [S.l.]: Springer, 1988. p. 213–244.

DIESTEL, R. **Graph Theory**. Electronic. New York: Springer, 2000. ISBN 0387228977 9780387228976 9780585498935 0585498938.

DIRAC, G. A. On rigid circuit graphs. **Abh. Math. Semin. Univ. Hambg.**, Springer Science and Business Media LLC, v. 25, n. 1-2, p. 71–76, Apr. 1961.

DOBRUSHIN, R. L.; KRYUKOV, V.; TOOM, A. L. **Locally interacting systems and their application in biology**. [S.l.]: Springer, 1978. v. 653.

DORBEC, P. Power domination in graphs. In: **Topics in Domination in Graphs**. [s.n.], 2020. Available at: https://hal.science/hal-03073783.

DOWNEY, R. G.; FELLOWS, M. R. **Parameterized complexity**. [S.l.]: Springer Science & Business Media, 2012.

DREYER, 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. ISSN 0166-218X. Available at: https://www.sciencedirect.com/science/article/pii/S0166218X08004393.

EDMONDS, J. Paths, trees, and flowers. **Canad. J. Math.**, Canadian Mathematical Society, v. 17, n. 0, p. 449–467, 1965.

EHARD, S.; RAUTENBACH, D. Vaccinate your trees! **Theoretical Computer Science**, Elsevier BV, v. 772, p. 4657, Jun. 2019. ISSN 0304-3975. Available at: http://dx.doi.org/10.1016/j.tcs.2018.11.018.

EPPSTEIN, D. Diameter and treewidth in minor-closed graph families. **Algorithmica**, Springer, v. 27, p. 275–291, 2000.

FINBOW, S.; KING, A.; MACGILLIVRAY, G.; RIZZI, R. The firefighter problem for graphs of maximum degree three. **Discrete Mathematics**, Elsevier BV, v. 307, n. 16, p. 20942105, Jul. 2007. ISSN 0012-365X. Available at: http://dx.doi.org/10.1016/j.disc.2005.12.053.

FINBOW, S.; MACGILLIVRAY, G. The firefighter problem: a survey of results, directions and questions. **Australas. J Comb.**, v. 43, p. 57–78, 2009.

FOMIN, F. V.; GOLOVACH, P. A.; KORHONEN, J. H. On the parameterized complexity of cutting a few vertices from a graph. In: CHATTERJEE, K.; SGALL, J. (Ed.). **Mathematical Foundations of Computer Science 2013**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 421–432. ISBN 978-3-642-40313-2.

GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)**. First edition. [S.l.]: W. H. Freeman, 1979. ISBN 0716710455.

GIAKKOUPIS, G.; GIONIS, A.; TERZI, E.; TSAPARAS, P. Models and algorithms for network immunization. **Department of Computer Science, University of Helsinki, Tech. Rep. C-2005-75**, 2005.

GOLES, E.; MARTÍNEZ, S. **Neural and automata networks: dynamical behavior and applications**. USA: Kluwer Academic Publishers, 1990. ISBN 0792306325.

GRANOVETTER, M. Threshold models of collective behavior. **American journal of sociology**, University of Chicago Press, v. 83, n. 6, p. 1420–1443, 1978.

HARARY, F.; NIEMINEN, J. Convexity in graphs. **Journal of Differential Geometry**, Lehigh University, v. 16, n. 2, p. 185–190, 1981.

HARTNELL, B. Firefighter! an application of domination. In: **the 24th Manitoba Conference on Combinatorial Mathematics and Computing, University of Minitoba, Winnipeg, Cadada, 1995**. [S.l.: s.n.], 1995.

HAYRAPETYAN, A.; KEMPE, D.; PÁL, M.; SVITKINA, Z. Unbalanced graph cuts. In: BRODAL, G. S.; LEONARDI, S. (Ed.). **Algorithms – ESA 2005**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 191–202.

JAMISON, R. E.; NOWAKOWSKI, R. A helly theorem for convexity in graphs. **Discrete Mathematics**, Elsevier BV, v. 51, n. 1, p. 3539, 1984. ISSN 0012-365X. Available at: http://dx.doi.org/10.1016/0012-365X(84)90021-9.

JR, J. R. F. A formal theory of social power. **Psychological review**, American Psychological Association, v. 63, n. 3, p. 181, 1956.

JR, P. A. D. **Applications and variations of domination in graphs**. [S.l.]: Rutgers The State University of New Jersey, School of Graduate Studies, 2000.

KALINOWSKI, T.; KAMEV, N.; SUDAKOV, B. The zero forcing number of graphs. **SIAM Journal on Discrete Mathematics**, Society for Industrial & Applied Mathematics (SIAM), v. 33, n. 1, p. 95115, Jan. 2019. ISSN 1095-7146. Available at: http://dx.doi.org/10.1137/17M1133051.

KAUFFMAN, S. Behaviour of randomly constructed genetic nets: binary element nets. In: **Organization Stability and Process**. [S.l.]: Routledge, 2017. p. 18–37.

KAUFFMAN, S. A. Metabolic stability and epigenesis in randomly constructed genetic nets. **Journal of theoretical biology**, Elsevier, v. 22, n. 3, p. 437–467, 1969.

KEMPE, D.; KLEINBERG, J.; TARDOS, E. Maximizing the spread of influence through a social network. In: **Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: Association for Computing Machinery, 2003. (KDD ’03), p. 137146. ISBN 1581137370. Available at: https://doi.org/10.1145/956750.956769.

KEYNES, J. M. The general theory of employment. **The quarterly journal of economics**, MIT Press, v. 51, n. 2, p. 209–223, 1937.

KIMURA, M.; SAITO, K.; MOTODA, H. Blocking links to minimize contamination spread in a social network. **ACM Transactions on Knowledge Discovery from Data**, Association for Computing Machinery (ACM), v. 3, n. 2, p. 123, Apr. 2009. ISSN 1556-472X. Available at: http://dx.doi.org/10.1145/1514888.1514892.

KINDERMANN, R.; SNELL, L. **Markov random fields and their applications**. [S.l.]: American Mathematical Society, 1980. v. 1. ISBN 978-0-8218-5001-5.

KING, A.; MACGILLIVRAY, G. The firefighter problem for cubic graphs. **Discrete Mathematics**, Elsevier BV, v. 310, n. 3, p. 614621, Feb. 2010. ISSN 0012-365X. Available at: http://dx.doi.org/10.1016/j.disc.2009.05.007.

KNEIS, J.; MÖLLE, D.; RICHTER, S.; ROSSMANITH, P. A bound on the pathwidth of sparse graphs with applications to exact algorithms. **SIAM J. Discrete Math.**, Society for Industrial & Applied Mathematics (SIAM), v. 23, n. 1, p. 407–427, Jan. 2009.

KORACH, E.; SOLEL, N. Tree-width, path-width, and cutwidth. **Discrete Applied Mathematics**, Elsevier BV, v. 43, n. 1, p. 97101, May 1993. ISSN 0166-218X. Available at: http://dx.doi.org/10.1016/0166-218X(93)90171-J.

KUHLMAN, C. J.; TULI, G.; SWARUP, S.; MARATHE, M. V.; RAVI, S. Blocking simple and complex contagion by edge removal. In: **2013 IEEE 13th International Conference on Data Mining**. IEEE, 2013. Available at: http://dx.doi.org/10.1109/ICDM.2013.47.

KöNIG, D. Theorie der endlichen und unendlichen graphen. **Monatshefte für Mathematik und Physik**, Springer Science and Business Media LLC, v. 46, n. 1, p. A17A18, Dec. 1937. ISSN 1436-5081. Available at: http://dx.doi.org/10.1007/BF01792729.

LALOU, M.; TAHRAOUI, M. A.; KHEDDOUCI, H. The critical node detection problem in networks: A survey. **Computer Science Review**, Elsevier BV, v. 28, p. 92117, May 2018. ISSN 1574-0137. Available at: http://dx.doi.org/10.1016/j.cosrev.2018.02.002.

LEWIS, J. M.; YANNAKAKIS, M. The node-deletion problem for hereditary properties is np-complete. **Journal of Computer and System Sciences**, Elsevier BV, v. 20, n. 2, p. 219230, Apr. 1980. ISSN 0022-0000. Available at: http://dx.doi.org/10.1016/0022-0000(80)90060-4.

LIMA, C. V. G. C. Complexity and algorithms related to two classes of graph problems. Universidade Federal do Rio de Janeiro, 2017.

LIU, X.; MA, L.; LI, X.; AI, B.; LI, S.; HE, Z. Simulating urban growth by integrating landscape expansion index (lei) and cellular automata. **International Journal of Geographical Information Science**, Taylor & Francis, v. 28, n. 1, p. 148–163, 2014.

MACGILLIVRAY, G.; WANG, P. On the firefighter problem. **Journal of Combinatorial Mathematics and Combinatorial Computing**, v. 47, p. 83–96, 2003.

MARCILON, T.; SAMPAIO, R. The maximum time of 2-neighbor bootstrap percolation: Complexity results. **Theoretical Computer Science**, Elsevier BV, v. 708, p. 117, Jan. 2018. ISSN 0304-3975. Available at: http://dx.doi.org/10.1016/j.tcs.2017.10.014.

MEDINA, P.; GOLES, E.; ZARAMA, R.; RICA, S. Self-organized societies: On the sakoda model of social interactions. **Complexity**, Wiley Online Library, v. 2017, n. 1, p. 3548591, 2017.

MEHTA, H.; REICHMAN, D. Local treewidth of random and noisy graphs with applications to stopping contagion in networks. In: CHAKRABARTI, A.; SWAMY, C. (Ed.). **Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2022)**. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. (Leibniz International Proceedings in Informatics (LIPIcs), v. 245), p. 1–17. ISBN 978-3-95977-249-5. ISSN 1868-8969.

MEIER, D.; PIGNOLET, Y. A.; SCHMID, S.; WATTENHOFER, R. On the windfall and price of friendship: Inoculation strategies on social networks. **Computer Networks**, Elsevier BV, v. 62, p. 221236, Apr. 2014. ISSN 1389-1286. Available at: http://dx.doi.org/10.1016/j.bjp.2013.12.004.

MELLO, I. F.; SQUILLANTE, L.; GOMES, G. O.; SERIDONIO, A. C.; SOUZA, M. de. Epidemics, the ising-model and percolation theory: a comprehensive review focused on covid-19. **Physica A: Statistical Mechanics and its Applications**, Elsevier, v. 573, p. 125963, 2021.

MORRIS, S. Contagion. **The Review of Economic Studies**, Wiley-Blackwell, v. 67, n. 1, p. 57–78, 2000.

MOSCIBRODA, T.; SCHMID, S.; WATTENHOFER, R. When selfish meets evil: byzantine players in a virus inoculation game. In: **Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing**. ACM, 2006. (PODC06), p. 3544. Available at: http://dx.doi.org/10.1145/1146381.1146391.

NEUMANN, J. V.; BURKS, A. W. *et al.* Theory of self-reproducing automata. University of Illinois press Urbana, 1966.

NEWMAN, M. E. J. **Networks: an introduction**. Oxford; New York: Oxford University Press, 2010. ISBN 9780199206650 0199206651.

PASTOR-SATORRAS, R.; VESPIGNANI, A. *et al.* Epidemics and immunization in scale-free networks. **Handbook of Graphs and Networks, Wiley-VCH, Berlin**, Wiley Online Library, 2003.

PRAKASH, B. A.; TONG, H.; VALLER, N.; FALOUTSOS, M.; FALOUTSOS, C. Virus propagation on time-varying networks: Theory and immunization algorithms. In: BALCÁZAR, J. L.; BONCHI, F.; GIONIS, A.; SEBAG, M. (Ed.). **Machine Learning and Knowledge Discovery in Databases**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 99–114. ISBN 978-3-642-15939-8.

REN, X.-L.; GLEINIG, N.; HELBING, D.; ANTULOV-FANTULIN, N. Generalized network dismantling. **Proceedings of the National Academy of Sciences**, Proceedings of the National Academy of Sciences, v. 116, n. 14, p. 65546559, Mar. 2019. ISSN 1091-6490. Available at: http://dx.doi.org/10.1073/pnas.1806108116.

RIEDL, E. Largest minimal percolating sets in hypercubes under 2-bootstrap percolation. **The Electronic Journal of Combinatorics [electronic only]**, Prof. André Kündgen, Deptartment of Mathematics, California State University San Marcos, San Marcos, v. 17, n. 1, p. Research Paper R80, 13 p.–Research Paper R80, 13 p., 2010. Available at: http://eudml.org/doc/226637.

ROBERTSON, N.; SEYMOUR, P. Graph minors. v. excluding a planar graph. **Journal of Combinatorial Theory, Series B**, Elsevier BV, v. 41, n. 1, p. 92114, Aug. 1986. ISSN 0095-8956. Available at: http://dx.doi.org/10.1016/0095-8956(86)90030-4.

RYAN, B.; GROSS, N. **Acceptance and diffusion of hybrid corn seed in two Iowa communities**. [S.l.: s.n.], 1950.

SEITZ, M.; KÖSTER, G.; PFAFFINGER, A. Pedestrian group behavior in a cellular automaton. In: SPRINGER. **Pedestrian and Evacuation Dynamics 2012**. [S.l.], 2014. p. 807–814.

SHEIKHAHMADI, A.; BAHRAMI, M.; SAREMI, H. Minimizing outbreak through targeted blocking for disease control: a community-based approach using super-spreader node identification. **Sci. Rep.**, Springer Science and Business Media LLC, v. 13, n. 1, p. 14217, Aug. 2023.

TARIQ, J.; AHMAD, M.; KHAN, I.; SHABBIR, M. **Scalable Approximation Algorithm for Network Immunization**. arXiv, 2017. Available at: https://arxiv.org/abs/1711.00784.

TONG, H.; PRAKASH, B. A.; ELIASSI-RAD, T.; FALOUTSOS, M.; FALOUTSOS, C. Gelling, and melting, large graphs by edge manipulation. In: **Proceedings of the 21st ACM international conference on Information and knowledge management**. ACM, 2012. (CIKM12). Available at: http://dx.doi.org/10.1145/2396761.2396795.

ULAM, S. *et al.* On some mathematical problems connected with patterns of growth of figures. In: AM. MATH. SOC. VOL. 14, PROVIDENCE. **Proceedings of symposia in applied mathematics**. [S.l.], 1962. v. 14, n. 14, p. 215–224.

VENTRESCA, M.; ALEMAN, D. Efficiently identifying critical nodes in large complex networks. **Computational Social Networks**, Springer Science and Business Media LLC, v. 2, n. 1, Mar. 2015. ISSN 2197-4314. Available at: http://dx.doi.org/10.1186/s40649-015-0010-y.

WAGNER, C. **A new survey on the firefighter problem**. Thesis (Doctor in Philosophy) — University of Victoria, 2021.

WAGNER, K. Über eine eigenschaft der ebenen komplexe. **Math. Ann.**, Springer Science and Business Media LLC, v. 114, n. 1, p. 570–590, Dec. 1937.

WANG, W.; STREET, W. N. Modeling and maximizing influence diffusion in social networks for viral marketing. **Applied network science**, Springer, v. 3, p. 1–26, 2018.

WEST, D. B. **Introduction to Graph Theory**. 2. ed. [S.l.]: Prentice Hall, 2000. ISBN 0130144002.

WOLFRAM, S. Twenty problems in the theory of cellular automata. **Physica Scripta**, IOP Publishing, v. 1985, n. T9, p. 170, 1985.

WOLFRAM, S. **Theory and Applications of Cellular Automata: Including Selected Papers, 1983-1986**. World Scientific, 1986. (Advanced series on complex systems). ISBN 9789971501235. Available at: https://books.google.com.br/books?id=llBjQgAACAAJ.

YANG, D.; LIAO, X.; SHEN, H.; CHENG, X.; CHEN, G. Dynamic node immunization for restraint of harmful information diffusion in social networks. **Physica A: Statistical Mechanics and its Applications**, Elsevier BV, v. 503, p. 640649, Aug. 2018. ISSN 0378-4371. Available at: http://dx.doi.org/10.1016/j.physa.2018.02.128.

YANNAKAKIS, M. Node-deletion problems on bipartite graphs. **SIAM Journal on Computing**, Society for Industrial & Applied Mathematics (SIAM), v. 10, n. 2, p. 310327, May 1981. ISSN 1095-7111. Available at: http://dx.doi.org/10.1137/0210022.

ZAKER, M. On dynamic monopolies of graphs with general thresholds. **Discrete Mathematics**, Elsevier BV, v. 312, n. 6, p. 11361143, Mar. 2012. ISSN 0012-365X. Available at: http://dx.doi.org/10.1016/j.disc.2011.11.038.