UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

DAVI DE ANDRADE IÁCONO

(SUB)FALL COLORING OF GRAPHS

FORTALEZA
2024

DAVI DE ANDRADE IÁCONO

(SUB)FALL COLORING OF GRAPHS

Dissertation submitted to the Post-Graduation Program in Computer Science of the Federal University of Ceará, as a partial requirement for obtaining the title of Master in Computer Science. Concentration Area: Computer Science.

Advisor: Prof. Dr. Júlio César Silva Araújo.
Co-Advisor: Prof[a]. Dra. Ana Shirley Ferreira da Silva.

FORTALEZA

2024

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)

I12(Iácono, Davi de Andrade.
(Sub)Fall Coloring of Graphs / Davi de Andrade Iácono. – 2024.
74 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2024.
Orientação: Prof. Dr. Júlio César Silva Araújo.
Coorientação: Profa. Dra. Ana Shirley Ferreira da Silva.

1. Algoritmos. 2. Coloração de grafos. 3. Subfall coloração. 4. Complexidade computacional. 5. Complexidade parametrizada. I. Título.

CDD 005

DAVI DE ANDRADE IÁCONO

(SUB)FALL COLORING OF GRAPHS

Dissertation submitted to the Post-Graduation Program in Computer Science of the Federal University of Ceará, as a partial requirement for obtaining the title of Master in Computer Science. Concentration Area: Computer Science.

Approved on: 06/09/2024

EXAMINATION BOARD

Prof. Dr. Júlio César Silva Araújo (Advisor)
Universidade Federal do Ceará (UFC)

Prof[a]. Dra. Ana Shirley Ferreira da Silva (Co-Advisor)
Universidade Federal do Ceará (UFC)
Università degli Studi di Firenze (UniFI)

Prof[a]. Dra. Ana Karolinna Maia de Oliveira
Universidade Federal do Ceará (UFC)

Prof[a]. Dra. Diana Sasaki Nobrega
Universidade do Estado do Rio de Janeiro (UERJ)

ACKNOWLEDGEMENTS

Aos meus pais, Milena e Rafael, pelo amor infindo e por todos os preciosos ensinamentos, momentos de alegria, suporte incansável e, é claro, pela paciência infinita que têm comigo em todas as minhas aventuras. Além de grandes amigos, vocês são também o meu orgulho.

Ao meu orientador, Júlio Araújo, pelo apoio desde o início da minha jornada acadêmica e, mais recentemente, pela presença, motivação, paciência, competência e dedicação na orientação durante o mestrado. À minha coorientadora, Ana Shirley, pela paciência, competência e cuidado com que me orienta desde a graduação, sempre se fazendo disponível. A ambos, Julio e Ana, pela preocupação que transcende o âmbito acadêmico. Vocês são grandes inspirações como pessoas e pesquisadores. Às professoras Ana Karolinna e Diana Sasaki por terem aceitado o convite de integrar a banca examinadora, pelo tempo dedicado ao texto e pelas valiosas sugestões. Agradeço também aos demais professores do grupo ParGO (o que inclui o já pargeano Andrea) por todo o auxílio e conhecimento transmitido.

Aos meus amigos e companheiros do ParGO, Leonardo, João Lucas, João Luca, Kennedy, Brito, Jonas, Felipe e Caio, pela companhia e por todas as conversas, acadêmicas ou não, fazendo dessa jornada um momento divertido. Aos meus amigos de graduação Thiago, Jackson, Jhonatan e Guilherme pelos diversos momentos e, especialmente, ao meu amigo Matheus pela presença e pelas incontáveis conversas durante todos esses anos.

Ao meu amigo Marcos, pela amizade desde o colégio e por se fazer presente mesmo tão distante, e aos meus amigos Camila e Vinícios, pela forte amizade, preocupação, conversas, apoio e companheirismo durante esse período, em especial durante o tempo em que moramos juntos. Aos amigos da comunidade cubista, principalmente Gustavo, pelos nossos mais de nove anos de amizade, além de George, Gleiciano e Thales pelo incessante apoio e momentos de alegria. Vocês são extremamente importantes na minha vida.

À minha querida companheira Gabriela, pela paciência, apoio, amor incondicional, carinho, cuidado, respeito e por tornar minha vida mais feliz e mais leve, bem como sua família, Salvador, Ana, Graziela e Geymisson pelo acolhimento e também suporte incessante.

À CAPES, pelo apoio financeiro.

"Não venha com a problemática, que eu tenho a solucionática."

(Dadá Maravilha)

RESUMO

Dado um grafo G, uma *k-coloração (própria)* de G é uma função $f : V(G) \to \{1,\dots,k\}$ tal que $f(u) \neq f(v)$, para toda aresta $uv \in E(G)$. Dada uma k-coloração f de um grafo G, um vértice $u \in V(G)$ é dito *b-vértice* com respeito a f se, para toda cor $i \in \{1,\dots,k\} - \{f(u)\}$ existe pelo menos um vértice $v \in V(G)$ tal que $f(v) = i$ e $uv \in E(G)$. Uma k-coloração f de um grafo G é chamada de *fall k-coloração* se todo vértice $u \in V(G)$ é b-vértice com respeito a f. Se um grafo G admite uma fall k-coloração para algum k, o *número fall acromático*, denotado por $\psi_f(G)$, é o maior inteiro positivo k tal que G admite uma fall k-coloração. Dado um grafo G e um inteiro positivo k, uma *subfall k-coloração* de G é uma fall k-coloração de algum subgrafo induzido $H \subseteq G$; e o *número subfall acromático*, denotado por $\psi_{fs}(G)$, é o maior inteiro positivo k tal que G admite uma subfall k-coloração. Nesta dissertação apresentamos uma breve revisão dos resultados sobre fall k-coloração encontrados na literatura que são os resultados mais relacionados à subfall coloração. Além disso, provamos que o problema de decidir se um grafo G admite uma subfall k-coloração é NP-completo para todo inteiro $k \geq 4$, respondendo a uma pergunta levantada em (Dunbar *et al.*, 2000). Apresentamos também um algoritmo FPT de programação dinâmica para decidir se um grafo G admite subfall k-coloração quando parametrizado pela sua largura em árvore $\mathrm{tw}(G)$, com $k \geq 3$. Ademais, dado um grafo G, estabelecemos a continuidade do parâmetro $\psi_{fs}(G)$ e a sua relação com alguns parâmetros, sendo eles o *número b-cromático* $b(G)$ e o *número de Grundy* $\Gamma(G)$. Finalmente, definimos o *índice subfall acromático* de um grafo G como sendo o parâmetro correspondente para coloração de arestas e estabelecemos uma versão do Teorema de Vizing para o mesmo em grafos planares e periplanares.

Palavras-chave: algoritmos; subfall coloração; coloração de grafos; complexidade computacional; complexidade parametrizada.

ABSTRACT

Given a graph G, a *(proper) k-coloring* of G is a function $f : V(G) \to \{1,\dots,k\}$ such that $f(u) \neq f(v)$, for every edge $uv \in E(G)$. Given a k-coloring f of a graph G, a vertex $u \in V(G)$ is a *b-vertex* with respect to f if for every color $i \in \{1,\dots,k\} - \{f(u)\}$ there exists at least one vertex $v \in V(G)$ such that $f(v) = i$ and $uv \in E(G)$. A k-coloring f of a graph G is a *fall k-coloring* if every vertex $u \in V(G)$ is a b-vertex with respect to f; If a graph G admits a fall k-coloring for some k, the *fall achromatic number*, denoted by $\psi_f(G)$, is the maximum positive integer k such that G admits a fall k-coloring. Given a graph G and a positive integer k, a *subfall k-coloring* of G is a fall k-coloring of some induced subgraph $H \subseteq G$; and the *subfall achromatic number*, denoted by $\psi_{fs}(G)$, is the maximum positive integer k such that G admits a subfall k-coloring. In this preliminary work, we present a brief review of the results about fall k-coloring found in the literature which are the closest related to the subfall coloring. Furthermore, we prove that deciding whether a graph G admits a subfall k-coloring is an NP-complete problem for every integer $k \geq 4$, answering a question raised in (Dunbar *et al.*, 2000). We also give a dynamic programming algorithm to decide whether a graph G admits a subfall k-coloring when parameterized by its treewidth $\mathrm{tw}(G)$ in FPT time, when $k \geq 3$. In addition, given a graph G, we establish the continuity of the parameter $\psi_{fs}(G)$ and its relations with some parameters, which are the *b-chromatic number* $b(G)$ and the *Grundy number* $\Gamma(G)$. Finally, we define the *subfall achromatic index* of a graph G as the corresponding parameter for edge coloring and prove a Vizing-like theorem for it on planar and outerplanar graphs.

Keywords: algorithms; subfall coloring; graph coloring; computational complexity; parameterized complexity.

LIST OF FIGURES

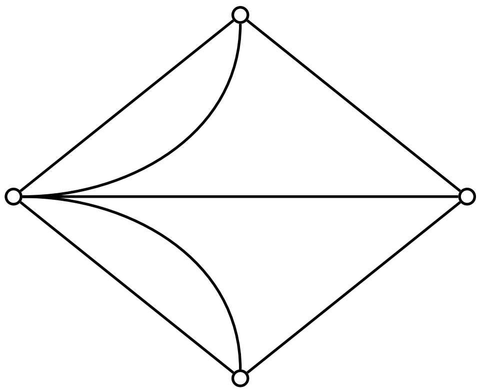
LIST OF TABLES

CONTENTS

1 INTRODUCTION

The history of Graph Theory is considered to have begun with the Seven Bridges of Königsberg problem, which asks whether it is possible to traverse all the bridges over the Pregel River, without crossing the same bridge more than once, ending the traverse in the same place the walk started. This problem was answered in the negative by the Swiss mathematician Leonhard Euler in (Euler, 1741), when he figured out that the actual path taken by the passer-by did not matter, but only the sequence of bridges crossed. This realization led him to ignore the shape of each piece of land, as well as its size, and just represented each of them by a node, making a connection between two nodes if the two corresponding pieces of land are the endpoints of some bridge. Euler then realized that in order to exist a walk passing through all the connections between the nodes and ending at the same node where it began, each node of the obtained object must have an even number of connections. He also proved that this is a sufficient condition, thus establishing what is believed to be the first theorem of Graph Theory. The object obtained by his formulation of the problem, shown in Figure 1, is called a *graph*, with its nodes being called *vertices* and the connections between two nodes being called *edges*.

Figure 1 – Graph representation of the Kőnigsberg's Bridges.



Source: prepared by the author

If we consider that, instead of bridges connecting two pieces of land, the edges of a graph represent land borders separating pieces of land, we then can use the structure of graphs to represent maps, where the vertices can represent countries, states, provinces, cities or towns, and then two vertices are connected by an edge if the two corresponding geographical areas share a border. Furthermore, such a representation of a map generates a graph that can be embedded in the plane and such that its edges intersect only at their endpoints, which we call *planar*. Indeed, planar graphs have been used to represent maps for many years, and one of the most famous theorems in Graph Theory derives from this type of representation: the Four Color Theorem. The Four Color Theorem, first proposed as a conjecture by Francis Guthrie in 1852, states that one needs at most 4 colors to color the vertices of a planar graph G in such a way that no two vertices sharing an edge have the same color. This is called *proper coloring*. Equivalently, the theorem states that $\chi(G) \leq 4$ for any planar graph G, where $\chi(G)$, called the *chromatic number* of G, is the minimum integer k such that $V(G)$ has a proper coloring using k colors. After 124 years and many false proofs and false counterexamples, the conjecture was finally proved by Wolfgang Haken and Kenneth Appel in (Appel; Haken, 1977). However, their proof initially met with resistance from the scientific community, as they used computers to check over 1800 configurations one by one, taking more than 1000 hours. Later, in (Appel; Haken, 1989), the same authors published a complete and detailed proof, with an appendix of over 400 pages. In order to find a more efficient algorithm and to reduce the controversy over the use of computers in Haken and Kenneth's proof, many researchers tried to improve their methods, leading to the quadratic time algorithm found by Neil Robertson, Paul Seymour, Daniel Sanders and Robin Thomas in (Robertson *et al.*, 1996).

As a result of the many applications that have arisen from the study of the Four Color Theorem, many researchers began to study other properties of colorings of planar graphs and other graph classes, as well as variations of the original problem of graph coloring, giving start to the area today called Graph Theory. As example of variations of graph coloring, we have: Fractional colorings (Larsen *et al.*, 1995), Complete colorings (Harary *et al.*, 1967), Exact colorings (Brown, 1972), List colorings (Erdős *et al.*, 1979), Weighted colorings (Guan; Xuding, 1997), Acyclic colorings (Grünbaum, 1973), Grundy colorings (Grundy, 1939) and b-colorings (Irving; Manlove, 1999).

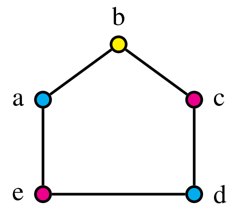
Formally, a *k-coloring* of a graph G is a labeling $f : V(G) \to X$, where X has k elements (often, $X = \{1,\ldots,k\}$). The labels are called *colors* and the vertices of one color forms a *color class*. A k-coloring is *proper* if there are no adjacent vertices in a same colors class. A graph is *k-colorable* if it has a proper k-coloring. The *chromatic number* of a graph G is the minimum integer k such that G has a proper k-coloring and it is denoted by $\chi(G)$. A k-coloring of a graph G is a *complete k-coloring* if, for every pair of colors $i, j \in \{1,\ldots,k\}$, $i \neq j$, there exists an edge in G whose endpoints are colored with i and j. This property induces a graph coloring heuristic, called *a-heuristic*, which is an algorithm that receives as input a graph G and a coloring f and searches for pairs of colors i and j such that there are no edges between vertices colored i

and j. If it finds a pair of colors i and j in such a way, it modifies the colors of the vertices colored j to i, and repeats this process until when there are no such pair of colors. Note that, given a graph G, every $\chi(G)$-coloring is complete. To the contrary, suppose that f is a $\chi(G)$-coloring of a graph G that is not complete. If this is the case, there exists a pair of colors i and j such that there are no edges between vertices colored i and j, and thus recoloring all vertices colored j to i would create a $(\chi(G)-1)$-coloring, which is not possible. In this sense, it is natural to study how good (or bad) this heuristic can be in order to construct a $\chi(G)$ coloring. In this sense, while $\chi(G)$ is the minimum number such that a graph admits a complete k-coloring, the *achromatic number* is the maximum number such that a graph G admits a complete k-coloring; it is denoted by $\psi(G)$ and was first defined in (Harary; Hedetniemi, 1970).

In order to present another studied heuristic for coloring a graph, known as *b-heuristic*, we give further definitions. Given a proper k-coloring f, we say that a vertex $v \in V(G)$ is a *b-vertex* with respect to f if it is adjacent to at least one vertex in each color class but its own. Again, this property induces a graph coloring heuristic, called *b-heuristic*, which is an algorithm that has as input a graph G and a coloring f and searches for color that do not contain a b-vertex. If it finds such a color i, then for each vertex v such that $f(v) = i$ we know that there exists a color j that no vertex in $N(v)$ is colored j, and thus it modifies the color of v to j while maintaining the coloring proper. Thus, the algorithm repeats this process until it obtains a coloring that every color has a b-vertex. Note that in every proper $\chi(G)$-coloring f of a graph G, there must exist at least one b-vertex u_i in each set $f^{-1}(c)$, for every $c \in \{1,\dots,k\}$. Indeed, if there is a color $c \in \{1,\dots,k\}$ such that $f^{-1}(c)$ contains no b-vertex, then for each vertex $u \in f^{-1}(c)$ there exists a color $i \in \{1,\dots,k\} \setminus c$ such that no vertex adjacent to u is colored with i and thus we can recolor u with color i; repeating such a process for each vertex $u \in f^{-1}(c)$ would give us a proper $(\chi(G)-1)$-coloring, which cannot happen. From this observation, a strategy to try to produce a proper $\chi(G)$-coloring of a graph G emerges, known as the *b-heuristic*. Given a k-coloring f, this heuristic involves changing the colors of the vertices in such a way that each color class contains a b-vertex. It was first introduced in (Irving; Manlove, 1999) and, in this sense, we say that a proper k-coloring f is a *b-coloring* if for each $c \in \{1,\dots,k\}$ the set $f^{-1}(c)$ has at least one b-vertex. The *b-chromatic number* of G is then defined as the maximum integer k such that G admits a b-coloring using k colors; it is denoted by $b(G)$. In other words, a proper k-coloring is a b-coloring if it can be obtained through the b-heuristic, and the parameter $b(G)$ indicates how bad the b-heuristic can perform.

It is known that deciding whether a graph G admits a b-coloring using at least k colors is a NP-complete problem. Hence, it is one of the most difficult problems to solve in terms of computational complexity and no polynomial-time algorithm to decide whether $b(G) \geq k$ for a given graph G and integer k is expected to exist, unless $\mathsf{P} = \mathsf{NP}$. More recently, it is proved in (Panolan *et al.*, 2017) that deciding $b(G) \geq k$ is $\mathsf{W}[1]$-hard when parameterized by k. In fact even an XP algorithm (algorithm that runs in time $\mathcal{O}(n^{f(k)})$) to decide whether $b(G) \geq k$ holds is not yet known. Before we introduce the coloring variation studied in this work, we first need to present a variation that precedes it and it is closely related to b-coloring. Introduced by Dunbar *et al.* in (Dunbar *et al.*, 2000), *fall colorings* can be seen as a variation of b-colorings in the sense that, while in b-colorings we ask for each color class to have at least one b-vertex, in fall colorings each vertex has to be a b-vertex. Because it is very constrained, this property leads to the existence of graphs that are not fall colorable, whereas this is not the case for b-coloring, since a $\chi(G)$-coloring is necessarily a b-coloring. Figure 2 gives a b-coloring of a graph G that does not admit a fall coloring. Indeed, it holds that $\chi(G) = 3$. Furthermore, because each vertex has degree equal to 2, no 4-coloring can admit a b-vertex, since there would always be a missing color for each vertex to be a b-vertex. Note now that G does not admit a fall 3-coloring: for a and c being b-vertices, it is necessary that d is colored the same as a and that e is colored the same as c. Hence, vertices d and e cannot be b-vertices.

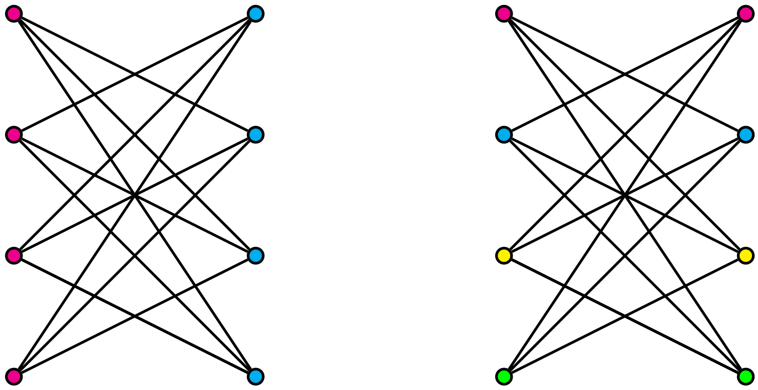
Figure 2 – A b-coloring of the graph cycle on 5 vertices, which is not fall colorable.



Source: prepared by the author

As in the case of b-colorings, it is natural to ask how bad this graph coloring heuristic can be. We then define the *fall achromatic number* of a graph G to be the greatest integer k such that G admits a fall k-coloring; this is denoted by $\psi_f(G)$. However, another peculiarity of fall coloring is that, even if a graph G admits fall colorings, G may not admit a fall k-coloring for some $k \in \{\chi(G)+1, \ldots, \psi_f(G)-1\}$. Figure 3 shows a graph G that is fall 2-colorable and fall 4-colorable, but not fall 3-colorable. Indeed, G does not admit a fall 3-coloring: since the graph is bipartite with partitions A and B of 4 vertices each, we know that in a 3-coloring of G there must exist two vertices in A with the same color, by the Pigeonhole principle. Let $u,v \in A$ be the two vertices such that $f(u) = f(v)$ and note that every vertex of B is adjacent to u or v, which gives us that no vertex $w \in B$ can satisfy $f(w) = f(u)$. Therefore, no vertex $a \in A$ such that $f(a) \neq f(u)$ can be a b-vertex.

Figure 3 – A graph G that is fall 2-colorable and fall 4-colorable, but not fall 3-colorable.

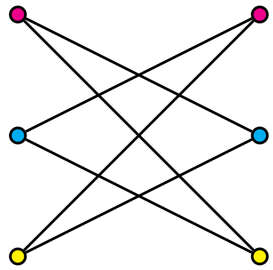


Source: prepared by the author

Dunbar *et al.* defined in (Dunbar *et al.*, 2000) the *subfall coloring*, which is the coloring variant studied in this work. Formally, given a graph G, a *subfall k-coloring* of G is a fall k-coloring of some induced subgraph $H \subseteq G$. Again, it is natural to define $\psi_{fs}(G)$, called *subfall achromatic number* of G, as the largest integer k such that a graph G has a subfall k-coloring.

Note that every graph with at least one vertex always has at least a subfall 1-coloring, since we can take the subgraph induced by only one vertex. Also, if a graph G has at least one edge, then it is subfall 2-colorable as well. As a result, each graph has at least one subfall coloring, which may not occur for fall colorings, as previously presented. Furthermore, in Section 4.3, we prove that if a graph G admits a subfall k-coloring, then it admits a subfall i-coloring for each $i \in \{1,\dots,k\}$, unlike fall colorings, as previously said. In Figure 4, we exhibit a subfall 3-coloring of the graph shown in Figure 3, in which we remove one vertex of each partition set in order to obtain a fall 3-colorable subgraph of G. In (Dunbar *et al.*, 2000), the authors left it as an open problem to decide whether $\psi_{fs}(G) \leq b(G)$ holds for any graph G. In this sense, in the third section of Chapter 4, we answer this question in the negative by showing, for every integer k, a graph G such that $\psi_{fs}(G) - b(G) = k$. We also extend the same study to other related parameters.

Figure 4 – A subfall 3-coloring of the graph shown in Figure 3.



Source: prepared by the author

In Section 4.1 we answer another open question from the seminal article, by settling the complexity of deciding whether a graph G satisfies $\psi_{fs}(G) \geq k$ for some integer k, or equivalently, deciding whether a graph G has a subfall k-coloring. In Section 4.2, we extend such a study. Since the problem is NP-complete, we do not expect it to have a polynomial-time algorithm and thus it is natural to try a different approach to solve it. We then give an algorithm running in FPT time when parameterized by the treewidth of a graph G that decides whether G admits a subfall k-coloring. In Section 4.4, we extend the study of subfall k-colorings of graphs by defining its edge version, with maximization parameter defined as *subfall achromatic index*.

In summary, this work is divided as follows:

- We give an overview of the results for fall and subfall colorings found in the literature, presenting proofs for a few of them in Chapter 3;
- We present our results for subfall k-coloring in Chapter 4: we prove that it is NP-complete when $k \geq 4$ and give an algorithm running in FPT time when parameterized by the treewidth of a graph G; we establish the continuity of the parameter ψ_{fs} and its relation with other related graph coloring parameters; we also define and give bounds for the subfall achromatic index for graphs; finally, we show a Vizing-like Theorem for planar and outerplanar graphs;
- We summarize the research developed in this work and point out present possible future research in Chapter 5.

All the results found in the first and third sections of Chapter 4 were presented at *VI Encontro de Teoria da Computação* (ETC) in 2021 (Andrade; Silva, 2021) and the results found in the second section of Chapter 4 were presented at *VII Encontro de Teoria da Computação* (ETC) in 2022 (Andrade; Silva, 2022).

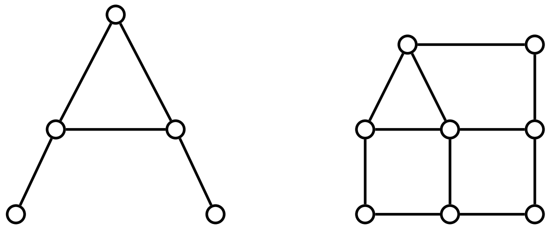
2 PRELIMINARIES

In this chapter, we focus on providing basic definitions and concepts for the full understanding of the content contained in this dissertation. We divide the concepts into two sections: one dedicated to Graph Theory and other dedicated to Computational Complexity. Each of the terms introduced in Section 2.1 can be found in more detail in (West, 2001) and the ones in Section 2.2 are presented in (Cormen *et al.*, 2009) and in (Cygan *et al.*, 2015).

2.1 Graph Theory

A *graph* G is a triple consisting of a *vertex set* $V(G)$, an *edge set* $E(G)$ and a function that associates with each edge two vertices called its *endpoints*; see Figure 5 for an example. When two vertices u,v are the endpoints of an edge, they are *adjacent* or, equivalently, *neighbors*. The *neighborhood* (*adjacency*) of a vertex v, denoted by $N_G(v)$, is the set of vertices adjacent to v. A *simple graph* is a graph such that there is no edge whose endpoints are equal and there are no edges sharing the same endpoints. For convenience, whenever we say "graph" in this text we refer to simple graphs, unless explicitly stated. Therefore, we can specify a simple graph simply by its vertex and edge sets, denoting each edge e with endpoints u and v by $e = uv$. A vertex u and an edge e are *incident* if u is an endpoint of e, and the *degree* of a vertex u in a graph is the number of incident edges. The *minimum degree* of a graph G is $\delta(G)$, the *maximum degree* is $\Delta(G)$ and a graph G is *k-regular* if $\delta(G) = \Delta(G) = k$. The *order* of a graph G is the number of vertices in G. A *subgraph* of a graph G is a graph H such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. We denote H being a subgraph of G by $H \subseteq G$. We also denote by $G-u$ or $G-X$ the subgraph obtained by removing the vertex $u \in V(G)$ and all the edges that u is endpoint of or the set of vertices X and all the edges that a vertex of X is endpoint of. Furthermore, an *induced subgraph* of a graph G is a graph $H \subseteq G$ obtained by deleting the set of vertices $V(G) \setminus V(H)$. The subgraph of G *induced by* a set of vertices X is the graph $G - \overline{X}$, where $\overline{X} = V(G) \setminus X$ and it is denoted by $G[X]$.

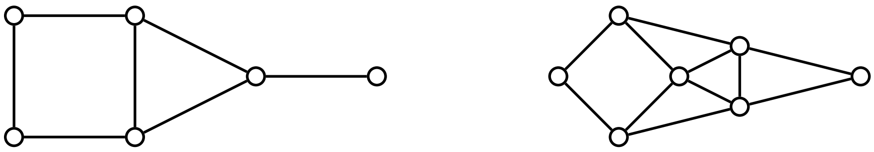
Figure 5 – Examples of graphs.



Source: prepared by the author

An *isomorphism* from a graph G to a graph H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ if and only if $f(u)f(v) \in E(H)$. Two graphs G and H are *isomorphic* if there is an isomorphism from G to H. The *line graph* of a graph G, denoted by $L(G)$, is the graph whose vertices are the edges of G, with $ef \in E(L(G))$ when e and f have a common endpoint in G. A graph G is a *line graph* if $G = L(H)$, for some graph H; see Figure 6 for an example.

Figure 6 – A graph G and its line graph $L(G)$.



Source: prepared by the author

A *k-coloring* of a graph G is a labeling $f : V(G) \to X$, where X has k elements; often, $X = [k]$. The labels are called *colors* and the vertices of one color form a *color class*. A k-coloring is *proper* if there are no adjacent vertices in a same colors class. A graph is *k-colorable* if it has a proper k-coloring. The *chromatic number* of a graph G is the minimum integer k such that G has a proper k-coloring; it is denoted by $\chi(G)$. Given a proper k-coloring of a graph G, a vertex $v \in V(G)$ is a *b-vertex* if it is adjacent to at least one vertex in each color class but its own. A k-coloring f is a *b-coloring* if each color class has at least one b-vertex, and the *b-chromatic number* $b(G)$ is the greatest integer such that a graph G admits a b-coloring; they were introduced in (Irving; Manlove, 1999). A k-coloring f of a graph G is a *fall k-coloring* if every vertex $v \in V(G)$ is a b-vertex with respect to f. The *fall achromatic number* $\psi_f(G)$ is the greatest integer k such that G admits a fall coloring and the *fall chromatic number* $\chi_f(G)$ is the least integer k such that G admits a fall coloring; they were introduced in (Dunbar *et al.*, 2000). The *fall spectrum* $\text{Fall}(G)$ of a graph G is the set of integers k such that G has a fall k-coloring. A graph G is *fall perfect* if $\text{Fall}(G) \subseteq \{\chi(G)\}$; this concept was introduced in (Silva, 2019). A graph G such that $\text{Fall}(G) \neq \emptyset$ is *fall continuous* if $\text{Fall}(G) = \{\chi_f(G), \ldots, \psi_f(G)\}$. In order to give an example of property of fall colorings, we present a result found in (Dunbar *et al.*, 2000) and that is used a large number of times through Chapters 3 and 4. The authors proved:

Proposition 2.1 (Dunbar *et al.*, 2000)**.** Let G be a graph with minimum degree $\delta(G)$. It holds that:

$$\psi_f(G) \leq \delta(G) + 1.$$

Proof. Let $v \in V(G)$ be a vertex of minimum degree. In any k-coloring of G with $k > \delta(G) + 1$, v can not be a b-vertex, since v has $\delta(G)$ neighbors and, thus, at most $\delta(G)$ colors in its neighborhood. □

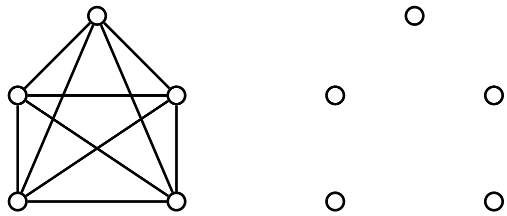
A *subfall k-coloring* of a graph G is a fall k-coloring of an induced subgraph H of G, and the *subfall achromatic number* ψ_{fs} is the greatest integer k such that G has a subfall k-coloring. We also denote the *fall achromatic index* and *subfall achromatic index* of a graph G by $\psi'_f(G) = \psi_f(L(G))$ and $\psi'_{fs}(G) = \psi_{fs}(L(G))$, respectively, where $L(G)$ is the line graph of G.

Let f be a proper coloring of G. We say that v is a *Grundy vertex* of color i with respect to f if $f(v) = i$ and v is adjacent to at least one vertex of color j for each color $j < i$. Additionally, f is a *Grundy k-coloring* if it is a k-coloring and each vertex $v \in V(G)$ is a Grundy vertex. The maximum value k such that G has a Grundy k-coloring is called *Grundy number* and is denoted by $\Gamma(G)$. Furthermore, a k-coloring f of a graph G is a *partial Grundy coloring* if every color class contains at least one Grundy vertex; this concept was first introduced in (Erdős *et al.*, 2003).

A *clique* in a graph G is a set of pairwise adjacent vertices, and an *independent set* is a set of pairwise non-adjacent vertices. Given a graph G, the maximum number of vertices found in a clique in a graph G is denoted by $\omega(G)$. A *complete graph* on n vertices is the graph K_n such that $V(K_n)$ is a clique. The *complement* of a graph G is the graph $\overline{G}$ with vertex set $V(G)$ and $uv \in E(\overline{G})$ if and only if $uv \notin E(G)$; see Figure 7 for an example. A *walk* is a list $v_0, e_1, v_1, \ldots, e_k, v_k$ of vertices and edges such that, for $1 \leq i \leq k$, the edge e_i has endpoints v_{i-1} and v_i. A *u,v-walk* has first vertex u and last vertex v, and these are its endpoints. A *closed walk* is a walk whose endpoints are the same. A *path* on n vertices P_n is a graph whose vertices can be ordered in a way that two vertices are adjacent if and only if they are consecutive in the ordering. A *u,v-path* is a path whose vertices of degree one are u and v. A *cycle* on n vertices

C_n is a graph with the same number of vertices and edges whose vertices can be placed around a circle such that two vertices are adjacent if and only if they are consecutive along the circle. A *cycle in* a graph G is a subgraph C of G that is a cycle. The *length* of a path or a cycle is its number of edges. A graph is *connected* if there exists a u,v-path for each pair of vertices u,v in G; otherwise, G is *disconnected*. A *component* of a graph G is a maximal connected subgraph of G. An *isolated vertex* is a vertex of degree 0 and a *leaf* is a vertex of degree 1.

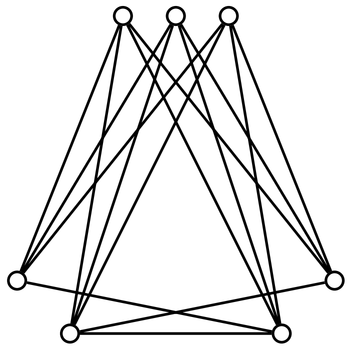
Figure 7 – A complete graph on 5 vertices and its complement.



Source: prepared by the author

A graph G is *k-partite* if $V(G)$ can be partitioned into at most k independent sets, called *partite sets*. Furthermore, a graph G is *bipartite* if it is 2-partite. A graph G is *complete k-partite* if every component of $\overline{G}$ is a complete graph; when $k \geq 2$, we denote the complete k-partite graphs with partite sets of sizes $n_1, n_2, \ldots, n_k$ by $K_{n_1,n_2,\ldots,n_k}$; see Figure 8 for an example.

Figure 8 – A complete 3-partite graph with partite sets of sizes $3,2,2$.



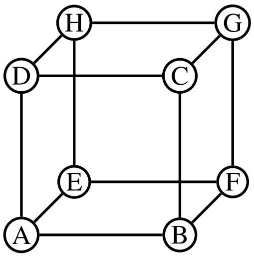
Source: prepared by the author

A graph G is *acyclic* if G has no cycles. A *forest* is an acyclic graph. A *tree* is a connected acyclic graph. A *matching* of a graph G is a set of edges with no shared endpoints. The vertices incident to the edges of a matching are *saturated*. A *perfect matching* of a graph G is a matching that saturates every vertex $v \in V(G)$.

The *k-dimensional hypercube* Q_k is the graph whose vertices are the k-tuples with entries in $\{0,1\}$ and whose edges are the pairs of k-tuples that differ in exactly one coordinate;

see Figure 9 for an example. A *cograph* is a graph with no P_4 as induced subgraph. A *chord* of a cycle C is an edge not in C whose endpoints lie in C. A graph G is *chordal* if every cycle C in G of length at least 4 has a chord. A family of graphs $\mathscr{G}$ is *hereditary* if every subgraph of a graph in $\mathscr{G}$ is also in $\mathscr{G}$. A *perfect graph* is a graph G such that $\chi(G) = \omega(G)$. Given a graph G, its *coloring number* is defined as $\text{col}(G) := \max_{H \subseteq G} \delta(H)$.

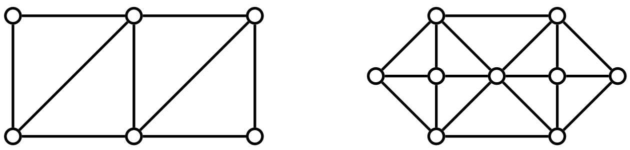
Figure 9 – A 3-dimensional hypercube with $A = (0,0,0)$, $B = (0,1,0)$, $C = (1,0,1)$, $D = (0,0,1)$, $E = (0,1,0)$, $F = (1,1,0)$, $G = (1,1,1)$ and $H = (0,1,1)$.



Source: prepared by the author

A *polygonal u,v-curve* is an image of a continuous map from $[0,1]$ to $\mathbb{R}^2$ composed of finitely many line segments that starts at u and finishes at v. A *drawing* of a graph G is a function defined on $V(G) \cup E(G)$ that assigns to each vertex v a distinct point $f(v)$ in the plane and assigns to each edge with endpoints u,v a polygonal $f(u), f(v)$-curve. A point in $f(e) \cap f(e')$ that is not a common endpoint is a *crossing*. A graph G is *planar* if it has a drawing without crossings. A *plane graph* is a drawing of a planar graph without crossings. An *open set* in the plane is a set $U \subset \mathbb{R}^2$ such that for every $p \in U$ there exists $\varepsilon > 0$ such that every point with a distance from p smaller than ε is in U. A *region* is an open set U that contains a polygonal u,v-curve for every pair $u,v \in U$. The *faces* of a plane graph are the maximal regions of the plane that contains no point used in the drawing. A plane graph has one unbounded face, called *outer face*. A graph is *outerplanar* if it has a drawing without any crossings such that every vertex lies in the outer face; see Figure 10 for an example.

Figure 10 – A chordal outerplanar graph and its line graph.



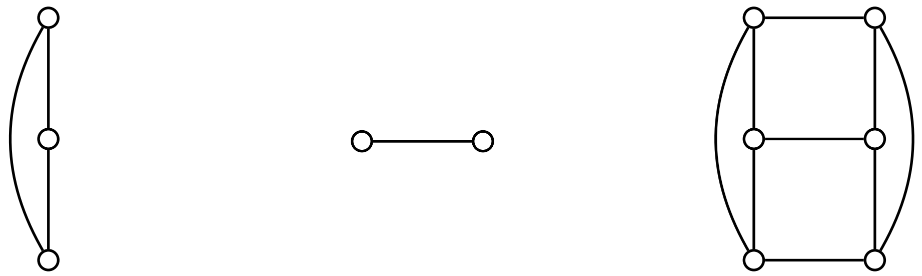
Source: prepared by the author

The *Cartesian product* of two graphs G and H, denoted by $G\square H$, is the graph with vertex set $V(G)\times V(H)$ specified by putting (u,v) adjacent to (u',v') if and only if:

- $u=u'$ and $vv'\in E(H)$ or;
- $v=v'$ and $uu'\in E(G)$.

See Figure 11 for an example of the cartesian product of two graphs.

Figure 11 – Cartesian product of the graphs C_3 and P_2.

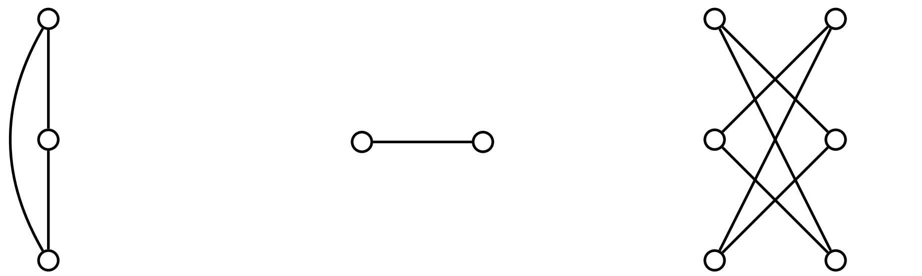


(a) K_3 (b) K_2 (c) $K_3\square K_2$

Source: prepared by the author

The *categorical product* (also known as: *direct product*, *tensor product*, *Kronecker product* and *conjunction*) of two graphs G and H is the graph $G\times H$ with vertex set $V(G)\times V(H)$ specified by putting (u,v) adjacent to (u',v') if and only if $uu'\in E(G)$ and $vv'\in E(H)$. See Figure 12 for an example of the categorical product of two graphs.

Figure 12 – Categorical product of the graphs C_3 and P_2.

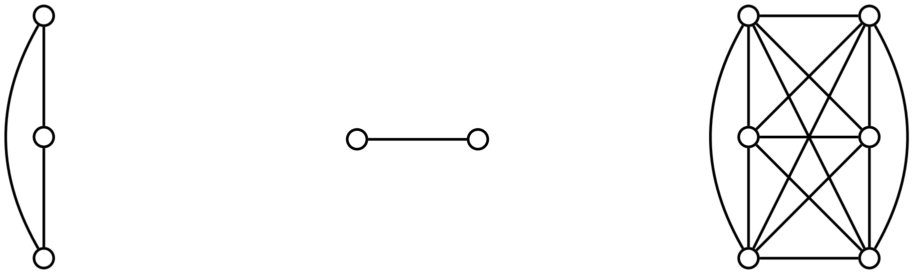


(a) K_3 (b) K_2 (c) $K_3\times K_2$

Source: prepared by the author

The *lexicographic product* of two graphs G and H is the graph $G[H]$ with vertex set $V(G)\times V(H)$ specified by putting (u,v) adjacent to (u',v') if and only if either $uu'\in E(G)$ or $u=u'$ and $vv'\in E(H)$. See Figure 13 for an example of the lexicographic product of two graphs.

Figure 13 – Lexicographic product of the graphs C_3 and P_2.



(a) C_3 (b) P_2 (c) $C_3[P_2]$

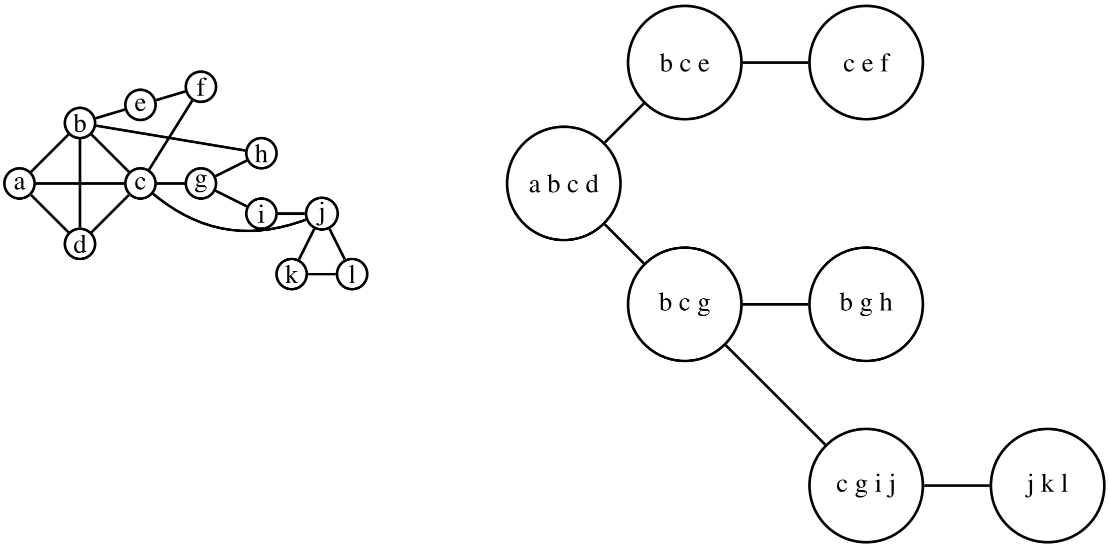
Source: prepared by the author

Given a graph G, a *tree decomposition* of G is a pair $\mathscr{T} = (\mathscr{X}, T)$, where $\mathscr{X} = \{X_t \subseteq V(G) \mid t \in V(T)\}$, T is a tree, and the following holds:

- $\bigcup \mathscr{X} = V(G)$;
- If $uv \in E(G)$, then there exists $X_t \in \mathscr{X}$ containing both u and v;
- For every $v \in V(G)$, $\{t \in V(T) \mid v \in X_t\}$ induces a connected subgraph of T.

For an example, see Figure 14. The *width* of $\mathscr{T}$ equals the maximum size of some X_t minus 1, and the *treewidth* of G equals the minimum possible width of a tree decomposition of G; the latter is denoted by $\mathrm{tw}(G)$. Observe that forests have treewidth 1, as proved in (Bodlaender, 1988).

Figure 14 – A graph G and a tree decomposition of G of width 3.



Source: prepared by the author

In addition, we say that a tree decomposition is *nice* if T is rooted at a node r with $X_r = \emptyset$, and each node $t \in V(T)$ is of one of the following types:

- *Leaf node*: t is a leaf in T and $|X_t| = 1$;
- *Forget node*: t has exactly 1 child t' and $X_t = X_{t'} \setminus \{u\}$, for some $u \in V(G)$;
- *Introduce node*: t has exactly 1 child t' and $X_t = X_{t'} \cup \{u\}$, for some $u \in V(G)$;
- *Join node*: t has two children t_1, t_2 and $X_t = X_{t_1} = X_{t_2}$.

Additionally, given a graph G, we say that $M \subseteq V(G)$ is a *module* if $M \neq \emptyset$ and, for every $w \in V(G) - M$ either w is adjacent to every vertex of M or either w is adjacent to no vertex of M. A module $M \subseteq V(G)$ is *trivial* if either $M = V(G)$ or $|M| \in \{0,1\}$. Furthermore, a module M is called *parallel* if $G[M]$ is disconnected, *series* if $\overline{G[M]}$ is disconnected, or *neighborhood* if both $G[M]$ and $\overline{G[M]}$ are connected. We say that a module M is *strong* if, for every module $N \subseteq V(G)$, either $M \cap N = \emptyset$ or $N \subseteq M$ or $M \subseteq N$. We also say that a module M is *maximal* if it there is no module $N \subseteq V(G)$ such that $M \subsetneq N$. A module M is a *maximal submodule* of $M \subsetneq N$ if there is no module $O \subsetneq N$ such that $M \subsetneq O$.

We now define the *modular decomposition tree* $T(G)$ recursively as follows, where $T(G)$ is rooted: the leaves of $T(G)$ represent the singletons $\{u\}$ with $u \in V(G)$; its internal nodes represent the strong modules of G, where the children of every internal node represent its maximal strong submodules in a way that the set of vertices represented by an internal node $v \in T(G)$ is exactly the union of the descendant leaves. Thus, the root represents the set of all vertices $V(G)$. Finally, each internal node is labelled parallel, series or neighborhood according to its type of module it represents. We comment that such a decomposition can be found in linear time (McConnell; Spinrad, 1994; Cournier; Habib, 1994). Given a modular decomposition MD of a graph G, the *quotient graph* of G with respect to MD, denoted by MD/G is the graph obtained by mapping each module M of MD to a single vertex such that two vertices of MD/G are adjacent if there is an edge between the corresponding modules in G.

Given two graphs G and H such that $V(G) \cap V(H) = \emptyset$, we say that the *join* $G + H$ is the graph whose set of vertices is $V(G+H) = V(G) \cup V(H)$ and whose set of edges is $E(G+H) = E(G) \cup E(H) \cup \{uv \mid u \in V(G) \wedge v \in V(H)\}$. The *union* $G \cup H$ of two graphs G and H such that $V(G) \cap V(H) = \emptyset$ is the graph whose set of vertices is $V(G \cup H) = V(G) \cup V(H)$ and whose set of edges is $E(G \cup H) = E(G) \cup E(H)$. A *rooted branch decomposition* of a graph G is a pair $(T, \mathscr{L})$ consisting of a rooted tree T of degree at most 3 and a bijection $\mathscr{L} : V(G) \to L(T)$, where $L(T)$ is the set of leaves of T. Furthermore, for $t \in V(T)$, we denote by T_t the subtree of T rooted at t and we define $V_t := \{v \in V(G) \mid \mathscr{L}(\sqsubseteq) \in \mathscr{L}(\mathscr{T}_{\sqcup})\}$, $\overline{V_t} := V(G) \setminus V_t$ and G_t as the subgraph of G induced by the vertices of V_t. Given a graph G, a rooted branch decomposition

$(T,\mathscr{L})$ of G and $t \in V(T)$, let $\sim_t$ be the equivalence relation on V_t such that, for every $u, v \in V_t$, $u \sim_t v$ if and only if $N_G(u) \cap \overline{V_t} = N_G(v) \cap \overline{V_t}$. The *module-width of a rooted branch decomposition* $(T,\mathscr{L})$ is equal to $\max_{t \in V(T)} |V_t / \sim_t|$. The *module-width of a graph G* is the minimum module-width over all rooted branch decompositions of G and it is denoted by $\mathrm{mw}(G)$; it was introduced in (Rao, 2006).

2.2 Computational complexity

An *algorithm* is a computational procedure that takes a set of values as *input* and produces a set of values as *output*. An algorithm is *correct* if it halts with the correct output for every given input. Given an algorithm A and an input x, we denote by $A(x)$ the output produced by A over x.

An *abstract problem Q* is a binary relation on a set I of problem *instances* and a set S of problem *solutions*. The *decision problems* are those having a yes/no solution, and the *optimization problems* are those which require some function to be maximize/minimized. Problems whose instance set is the set of binary strings are *concrete problems*. An algorithm *solves* a concrete problem in time $\mathscr{O}(T(n))$ if, when it is provided a problem instance i of length n, the algorithm can produce the solution in $\mathscr{O}(T(n))$ steps. A *polynomial-time algorithm* is an algorithm that, given an input x, outputs $A(x)$ in $\mathscr{O}(|x|^k)$, where $|x|$ is the number of elements of the input x. A concrete problem is *polynomial-time solvable* if there exists an algorithm to solve it in $\mathscr{O}(n^k)$ steps, for some constant k. The *complexity class* P is the set of concrete decision problems that are polynomial-time solvable.

An *alphabet* Σ is a finite set of symbols and Σ^* is the language of all strings over Σ; by simplification, we use $\Sigma = \{0,1\}$. A *language L* over Σ is any set of strings made up of symbols from Σ. An algorithm A *accepts* a string $x \in \{0,1\}^*$ if, given input x, the algorithm outputs 1. The language *accepted* by an algorithm A is the set of strings $L = \{x \in \{0,1\}^* \mid A(x) = 1\}$. An algorithm *rejects* a string x if $A(x) = 0$. The language L is *decided* by an algorithm A if every binary string in L is accepted by A and every binary string not in L is rejected by A. Furthermore, a decision problem is a language and the elements of the language are the "yes" instances. As an example, consider the problem of deciding whether a given graph G has a clique of size k. This problem has the corresponding language:

$$\text{CLIQUE} = \{\langle G, k\rangle : G \text{ is a graph containing a clique of size } k\}.$$

A deciding algorithm for this language receives as input a graph G and, for every subset $V' \subset V(G)$ with size $|V'| = k$, checks whether V' is a clique. If this is the case, then the algorithm outputs 1 and halts. If the algorithm finds no subset of $V(G)$ that is a clique of size k, then it outputs 0 and halts.

A *verification algorithm* to a problem/language is a two-argument algorithm A, where one argument is an ordinary input string x and the other is a binary string y called a *certificate*. A two-argument algorithm *verifies* an input string x if there exists a certificate y such that $A(x,y) = 1$. The *languages verified* by a verification algorithm A is $L = \{x \in \{0,1\}^* \mid$ there exists $y \in \{0,1\}^*$ such that $A(x,y) = 1\}$. The *complexity class* NP is the class of languages that can be verified by a polynomial-time algorithm.

A function $f : \{0,1\}^* \to \{0,1\}^*$ is *polynomial-time computable* if there exists a polynomial-time algorithm A that, given any input $x \in \{0,1\}^*$ produces as output $f(x)$.A language L_1 is *polynomial-time reducible* to a language L_2, written $L_1 \leq_p L_2$, if there exists a polynomial-time computable function $f : \{0,1\}^* \to \{0,1\}^*$ such that for all $x \in \{0,1\}^*$, $x \in L_1$ if and only if $f(x) \in L_2$; the function f is called *reduction function* and the algorithm A that computes f is called *reduction algorithm*. A language $L \subseteq \{0,1\}^*$ is NP-*hard* if $L' \leq_p L$ for every $L' \in \mathsf{NP}$. A language is NP-*complete* if it is in NP and is NP-hard. Given a boolean formula with variables $x_1, \ldots, x_n$, a *literal* is an occurrence of a variable x_i or its negation $\overline{x_i}$. A *clause* is the OR of one or more literals, such as $x_1 \vee \overline{x_2} \vee \overline{x_3}$. A boolean formula φ with boolean variables $x_1, \ldots, x_n$ is in *conjunctive normal form (CNF)* if it is expressed as an AND of clauses, and it is in 3-*conjunctive normal form (*3*-CNF)* if each clause contains exactly three distinct literals. A *truth assignment* of a boolean formula φ of variables $X = \{x_1, \ldots, x_n\}$ is an assignment of values TRUE and FALSE such that φ is true.

Proposition 2.2. If any NP-complete problem is polynomial-time solvable, then $\mathsf{NP} = \mathsf{P}$.

Proof. Suppose that an *NP*-complete problem Q is in P and let A_1 be the polynomial-time algorithm that decides Q and L be any problem in NP. Since Q is an NP-complete problem, we know that $L \leq_p Q$. In this sense, let A_2 the reduction algorithm that computes f. We then construct an algorithm B that decides L' in polynomial time. For a given input $x \in \{0,1\}^*$, algorithm B runs A_2 in order to transform x into $f(x)$ and then uses A_1 to decide whether $f(x) \in Q$, taking the output from A_1 as its output. Furthermore, B is a polynomial-time algorithm, since both A_1 and A_2 are. $\square$

A *parameterized problem* is a language $L \subset \Sigma^* \times \mathbb{N}$, where Σ is a fixed and finite alphabet. For an instance $(x,k) \in \Sigma^* \times \mathbb{N}$, k is called the *parameter*, and its *size* is defined as $|(x,k)| = |x| + k$, where $|x|$ is the number of elements in x. For example, an instance of CLIQUE parameterized by the solution size is a pair (G,k), where G is an undirected graph and k is a positive integer. A *kernel* for a parameterized problem Q is an algorithm A that, given an instance (I,k) of Q, outputs an equivalent instance (I',k') of Q; moreover, we require the existence of a computable function g such that whenever (I',k') is the output for an instance (I,k) it holds that $|I'| + k' \leq g(k)$. If the upper bound given by g is polynomial, we say that Q admits a *polynomial kernel*.

A parameterized problem $L \subset \Sigma^* \times \mathbb{N}$ is called *fixed-parameter tractable* (FPT) if there exists an algorithm A, called a *fixed-parameter algorithm*, a computable function $f : \mathbb{N} \to \mathbb{N}$ and a constant c such that, given $(x,k) \in \Sigma^* \times \mathbb{N}$, the algorithm A correctly decides whether $(x,k) \in L$ in time bounded by $\mathcal{O}(f(k)|(x,k)|^c)$. The complexity class containing all fixed-parameter tractable problems is called FPT. A parameterized problem $L \subset \Sigma^* \times \mathbb{N}$ is called *slice-wise polynomial* (XP) if there exists an algorithm A and two computable functions $f, g : \mathbb{N} \to \mathbb{N}$ such that given $(x,k) \in \Sigma^* \times \mathbb{N}$, the algorithm A correctly decides whether $(x,k) \in L$ in time bounded by $f(k)|(x,k)|^{g(k)}$. The complexity class containing all slice-wise polynomial problems is called XP.

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, outputs an instance (x',k') of B such that

- (x,k) is an yes-instance of A if and only if (x',k') is an yes-instance of B;
- $k' \leq g(k)$ for some computable function g, and
- the running time is $f(k)|x|^{\mathcal{O}(1)}$ for some computable function f.

We now define the parameterized problem INDEPENDENT SET as follows:

INDEPENDENT SET

Input:	A graph G.
Parameter:	A positive integer k..
Question:	There is an induced subgraph $X \subseteq G$ such that $|X| \leq k$ and $E(X) = \emptyset$?

By simplification, in order to prove that a parameterized problem Q is W[1]-*hard*, one constructs a parameterized reduction from INDEPENDENT SET to Q. We now define the following (not parameterized) problem:

3-SAT

Input:	A boolean formula φ in 3-CNF.
Question:	There is an truth assignment for φ?

The current status of research on satisfiability problems suggest that is hard to obtain an algorithm for 3-SAT running in time $2^{o(n)}$. In this sense, let δ_3 be the infimum of the set of constants c for which there exists an algorithm solving 3-SAT in $\mathcal{O}^*(2^{cn})$ steps. In this sense, the *Exponential Time Hypothesis (ETH)* is a conjecture that states that $\delta_3 > 0$. Intuitively, ETH states that any algorithm for 3-SAT must search through an exponential number of alternatives.

3 STATE OF THE ART

Subfall coloring has not been studied in other works apart from the paper (Dunbar *et al.*, 2000), that introduces fall colorings. In this chapter, we do a literature review of fall colorings and present the existing results on subfall colorings. Some authors in the literature have also used the name *independent* and *dominating vertex partition* instead of fall coloring, calling the colors classes *disjoint* and *dominating sets*, as in (Heggernes; Telle, 1998). We separate the works into 3 sections. In Section 3.1, we present results about fall spectrum of some graph classes. In Section 3.2, we present the computational complexity results about fall colorings and the parameter ψ_f. In Section 3.3, we present further interesting results related to the fall coloring problem that do not fit in the first two sections. Finally, in Section 3.4, we show the results on subfall colorings presented in (Dunbar *et al.*, 2000).

3.1 Fall spectrum

In order to present the results found in the literature, we need to establish some definitions. We say that a graph G is *uniquely k-colorable* if $V(G)$ has only one partition into k independent sets. We say that a graph G is a *k-tree* if it is a complete graph with $k+1$ vertices or there exists a vertex $u \in V(G)$ such that $N(u)$ is a clique with exactly k vertices and $G-u$ is a k-tree.

In (Cockayne; Hedetniemi, 1976), the authors established the fall spectrum of some graphs, which we compile in the following proposition:

Proposition 3.1 (Cockayne; Hedetniemi, 1976)**.** The following holds:

1. The complete graph on n vertices satisfies $\text{Fall}(K_n) = \{n\}$, for every positive integer n;
2. $2 \in \text{Fall}(G)$ whenever G is a connected bipartite graph;
3. $\text{Fall}(G) = \{k\}$ for complete k-partite graphs;
4. $\text{Fall}(G) \subseteq \{2,3\}$ for cycles of length multiple of 3;
5. $k \in \text{Fall}(G)$ for uniquely k-colorable G;
6. $k+1 \in \text{Fall}(G)$ for k-trees;
7. $\text{Fall}(G) = \{2,k\}$ whenever G is obtained from a complete bipartite graph $K_{k,k}$ by removing the edges of a perfect matching, for every integer $k \geq 2$;
8. $k+\ell \in \text{Fall}(G+H)$ if G has fall k-coloring and H has fall ℓ-coloring.

In order to give an example on proofs about fall colorings, we now prove items 3, 5 and 7 separately. For item 3, we have:

Proof of Proposition 3.1 - 3. Being G_1 a complete k-partite graph and $v \in V(G_1)$, we know that v is adjacent to all vertices belonging to other parts than its own part. Therefore, letting $P = \{V_1, V_2, \ldots, V_k\}$ be the k-partition, we take the coloring f of $V(G)$ such that all vertices of the set V_i are colored with the color i, where $i \in \{1, \ldots, k\}$. Clearly f is a fall coloring of $V(G)$, since each vertex colored with color i is adjacent to all vertices colored with color $i, j \neq i$, where $j \in \{1, \ldots, n\}$. □

Proof of Proposition 3.1 - 5. Let G_2 be a uniquely k-colorable graph. If this happens, we know that for each k-coloring f of $V(G)$, the sets of vertices $X_1, \ldots, X_k$ such that $x \in X_i$ if and only if $f(x) = i$ produce the same partitioning of $V(G)$ into k independent sets. Indeed, being $P = \{V_1, V_2, \ldots, V_k\}$ such partition of $V(G)$, we know that if $v \in V_i$, $i \in \{1, \ldots, k\}$, then it must happen, for every $j \in \{1, \ldots, k\} \setminus i$, that $N(v) \cap V_j \neq \emptyset$. In fact, if there are $v \in V_i$ and $\ell \neq i$ such that $N(v) \cap V_\ell = \emptyset$, then $P' = \{V_1, \ldots, V_i \setminus \{v\}, \ldots, V_\ell \cup \{v\}, \ldots, V_k\}$ is a partition of $V(G)$ different from P, where we suppose without loss of generality that $\ell > i$. Thus, every vertex v is adjacent to at least one vertex colored with a color different from $f(v)$. □

Before showing a proof for Item 7, we define the following concept: given a function $f : X \to Y$ and a subset $S \subseteq X$, we denote the image of S by $f(S)$. Formally, $f(S) = \{d \in D \mid \exists x \in S (f(x) = d)\}$.

Proof of Proposition 3.1 - 7. First, denote by $K'_{n,n}$ the graph obtained from a complete bipartite graph $K_{k,k}$ by removing the edges of a perfect matching. We prove this item by induction. As basis, take $K'_{2,2}$. Hence, without loss of generality, we may assume that $V(K'_{2,2}) = \{v_1, v_2, v_3, v_4\}$ and $E(K'_{2,2}) = \{v_1v_3, v_2v_4\}$. Clearly, any 2-coloring f of $V(K'_{2,2})$ is a fall coloring. In addition, there is no fall k-coloring with $k \neq 2$, since every vertex has only one neighbor and any 1-coloring is not proper. Now we prove that $n \in \text{Fall}(K'_{n,n})$ for every positive integer n. Let A and B be its partitions, $a_1, a_2, \ldots, a_n \in A$ and $b_1, b_2, \ldots, b_n \in B$ be its vertices such that $a_ib_i \notin E(K'_{n,n})$ for each $1 \leq i \leq n$. Let f be a n-coloring of G such that $f(a_i) = f(b_i) = i$ for every $1 \leq i \leq n$ and note that $a_ib_j \in E(K'_{n,n})$ for every $1 \leq i, j \leq n$ with $i \neq j$, thus every a_i is a b-vertex. Analogously, every b_i is a b-vertex. Therefore, f is a fall n-coloring of $K'_{n,n}$.

It remains to show that there is no fall k-coloring for $2 < k < n$. Indeed, for there to exist such a fall k-coloring h of $K'_{n,n}$, there must exist at least two distinct vertices $a_1, a_2 \in A$

colored with the same color $\ell \in \{1,\dots,k\}$, since the number of vertices in each partition is greater than the number of colors. Moreover, since we must have $h(B) = \{1,\dots,k\}$ for the vertices of A to be b-vertices, there exists a vertex $b \in B$ such that $h(b) = \ell$. Since $K'_{n,n}$ is the graph $K_{n,n}$ minus a perfect matching, at least one among a_1, a_2 is neighbor of b, so h cannot be proper. This being so, we have $\text{Fall}(K'_{n,n}) = \{2,n\}$ for any natural n. □

In (Balakrishnan; Kavaskar, 2010), the authors proved the following:

Theorem 3.1 (Balakrishnan; Kavaskar, 2010)**.** Let G be a graph with $\delta(G) \geq |V(G)| - 2$. Then $\text{Fall}(G)$ is nonempty.

In (Lauri; Mitillos, 2020), the authors establish the following property for maximal outerplanar graphs:

Theorem 3.2 (Lauri; Mitillos, 2020)**.** Let G be a maximal outerplanar graph with at least 3 vertices. Then $\text{Fall}(G) = \{3\}$.

Before we present the next theorem, we need a new definition. Given a graph G, the *Mycielskian* of G, denoted by $M(G)$, is the graph $M(G)$ obtained by the following construction. Starting with G, take C as a copy of $V(G)$ and let $V(M(G)) = V(G) \cup C \cup \{u^*\}$. Given a vertex $v \in V(G)$, denote by v' its copy in C. Add all the edges between v' and each neighbor of v in $V(G)$ and add the edges u^*v' for each $v' \in C$. In (Shaebani, 2009), the author demonstrates the following result for Mycieslkian of graphs:

Theorem 3.3 (Shaebani, 2009)**.** $\text{Fall}(M(G)) = \emptyset$, for every graph G.

The *Kneser graph* $K(k,n)$ is the graph whose vertices correspond to the subsets of k elements of a set of n elements, with two vertices u and v being adjacent if and only if the corresponding subsets are disjoint. In (Shaebani, 2019), the author continues the study of the fall spectrum, this time for the Kneser graph $K(n,2)$, for all $n \geq 2$:

Theorem 3.4 (Shaebani, 2019)**.** For every natural $n \geq 2$, it holds that:

$$\text{Fall}(KG(n,2)) = \begin{cases} \{1\}, & \text{if } n \in \{2,3\}; \\ \{2\}, & \text{if } n = 4; \\ \{\frac{n(n-1)}{6}\}, & \text{if } n \geq 5,\ n = 1 \text{ or } 3 \pmod 6; \\ \{\frac{(n-1)(n-2)}{6} + 1\}, & \text{if } n \geq 5,\ n = 2 \text{ or } 4 \pmod 6; \\ \emptyset, & \text{if } n \geq 5,\ n = 0 \text{ or } 5 \pmod 6. \end{cases}$$

In the next two subsections, we show some results found in the literature on the fall spectrum of cartesian, categorical and lexicographic product of graphs.

3.1.1 Cartesian products

With respect to fall coloring results for the Cartesian products of graphs, in (Dunbar *et al.*, 2000) the authors established properties for $\mathrm{Fall}(G)$ when G is a product of specific classes of graphs, such as product of paths (i.e., $P_i \square P_j$), product of paths with cycles $(C_i \square P_j)$ and product of cycles $(C_i \square C_j)$. We present them in the following proposition:

Proposition 3.2 (Dunbar *et al.*, 2000)**.** Let P_i and P_j be paths on i and j vertices, respectively, and let C_k and C_ℓ be cycles on k and ℓ vertices, with $i, j \geq 2$ being positive integers and $k, \ell \geq 3$. Then:

1. $\mathrm{Fall}(P_i \square P_j) = \{2\}$;
2. We have that $2 \in \mathrm{Fall}(C_k \square P_j)$ if and only if $k \geq 4$ is even;
3. If $k = 3m$ for some m, it holds that $3 \in \mathrm{Fall}(C_{3m} \square P_j)$;
4. If $k = 4m$ and $j = 2n$, for some m and n, we have that $4 \in \mathrm{Fall}(C_{4m} \square P_{2n})$;
5. $2 \in \mathrm{Fall}(C_k \square C_\ell)$ if and only if both $k \geq 4$ and $\ell \geq 4$ are even;
6. $3 \in \mathrm{Fall}(C_k \square C_\ell)$ if and only if at least one between k and ℓ is multiple of 3;
7. If $k = 4m$ and $\ell = 2n$ for some m and n, then $4 \in \mathrm{Fall}(C_{4m} \square C_{2n})$;
8. $5 \in \mathrm{Fall}(C_k \square C_\ell)$ if and only if both k and ℓ are multiple of 5.

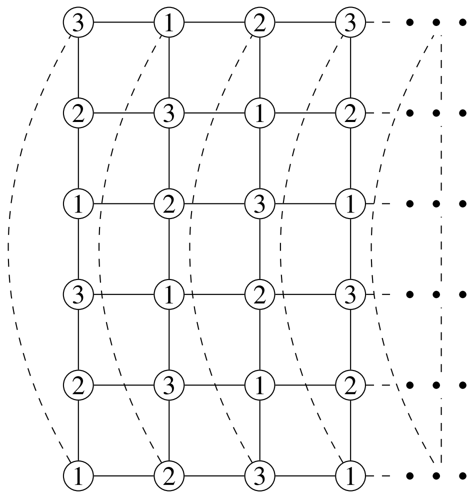
In order to present the reader we now prove items 1, 3 and 5.

Proof of Proposition 3.2 - 1. Let G be the Cartesian product $P_i \square P_j$ and label the vertices of G as $a_{n,m} = (v_n, u_m)$, being $P_i = \{v_1, \dots, v_i\}$, $P_j = \{u_1, \dots, u_j\}$, $n \in \{1, \dots, i\}$ and $m \in \{1, \dots, j\}$. Since G is bipartite and has no isolated vertex, we have $2 \in \mathrm{Fall}(G)$ and $1 \notin \mathrm{Fall}(G)$. Furthermore, let us prove that G has no fall k-coloring for $k > 2$. By contradiction, let f be a fall k-coloring of G, for some $k \geq 3$. Since $d_G(a_{1,1}) = 2$, and $a_{1,1}$ is a b-vertex with respect to f, then $k = 3$ and colors $1, 2, 3$ occur in $a_{1,1}, a_{1,2}$ and $a_{2,1}$. Without loss of generality, suppose that the vertices $a_{1,1}, a_{1,2}$ and $a_{2,2}$ are colored with the colors $1, 2$ and 3, respectively. Then $a_{2,2}$ must be given the color 1 for the coloring to be proper. If $i \leq 2$, then $a_{2,1}$ cannot be a b-vertex, since its two neighbors would receive color 1, thus $i \geq 3$. Then, $a_{3,1}$ must receive the color 2 so that $a_{2,1}$ is b-vertex. Similarly, $a_{3,2}$ must receive color 3 for it to be a proper coloring. Again, if $i \leq 3$, then $a_{3,1}$ cannot be a b-vertex, since its two neighbors would receive color 3, thus suppose $i \geq 4$.

Then $a_{4,1}$ must receive color 1 for $a_{3,1}$ to be b-vertex. Continuing this process, we have that the color of $a_{i-1,2}$ is different from the color of $a_{i,1}$ and therefore $a_{i,2}$ must receive the same color as $a_{i-1,1}$ for the coloring to be proper. But that being the case, $a_{i,1}$ cannot be b-vertex. $\square$

Proof of Proposition 3.2 - 3. First note that there exists fall 3-coloring of C_{3m}. Let $V(C_{3m} = \{v_1, v_2, \ldots, v_{3m}\}$ such that $v_1 v_{3m} \in E(C_{3m})$ and $v_i v_{i+1} \in E(C_{3m})$ for each $1 \leq i \leq 3m$. It suffices to take the coloring f such that $f(v_i) = i \pmod 3$ if $i \equiv 1 \pmod 3$ or $i \equiv 2 \pmod 3$ and $f(v_i) = 3$ if $i \equiv 0 \pmod 3$)., which is fall 3-coloring of C_{3m}, because, for every positive integer i, it holds that $i \neq i+1 \pmod 3$ and $i \neq i+2 \pmod 3$, so every vertex v_i with $1 < i < 3m$ is b-vertex. For vertex v_1, notice that $f(v_1) = 1$, $f(v_2) = 2$ and $f(v_{3m}) = 3$. Therefore, since v_1 is adjacent to v_2 and v_{3m}, v_1 is b-vertex. On the other hand, v_{3m} is b-vertex since it is neighboring v_{3m-1}, which is colored with the color 2. Therefore, $3 \in \text{Fall}(C_{3m})$. Figure 15 shows how to extend such a fall 3-coloring of C_{3m} to a fall 3-coloring of $(C_{3m} \square P_j)$ when $m = 2$.

Figure 15 – Example of fall 3-coloring of $C_6 \square P_j$.



Source: prepared by the author.

$\square$

Proof of Proposition 3.2 - 5. For Item 5, notice that $2 \in \text{Fall}(C_k \square C_\ell)$ if and only if $C_k \square C_\ell$ is bipartite. However, for $C_k \square C_\ell$ to be bipartite, it is necessary that $k, \ell \geq 4$ and that neither is an odd cycle, that is, it is necessary that k and ℓ are even. $\square$

In the same paper, the authors settled the following result for the fall spectrum of products of fall colorable graphs, which generalizes Items 1, 2, 3 and 4 of Proposition 3.2:

Theorem 3.5 (Dunbar *et al.*, 2000)**.** Let G and H be fall k-colorable and fall ℓ-colorable graphs, respectively, with $k \geq \ell$. Then, $k \in \text{Fall}(G \square H)$.

In (Laskar; Lyle, 2009), the authors proved some additional properties for the fall spectrum of other specific classes of graphs, such as the cartesian product of trees, cartesian product of complete graphs, and hypercubes (which are cartesian products of K_2) and cartesian products of hypercubes with a fall colorable graph. We state the results in the following proposition:

Proposition 3.3 (Laskar; Lyle, 2009)**.** Let K_n and K_m be complete graphs on n and m vertices, with $n \leq m$ and T_1, T_2 be trees. Then, it holds that:

1. $\text{Fall}(T_1 \square T_2) = \{2\}$;
2. $m \in \text{Fall}(K_n \square K_m)$;
3. $3 \in \text{Fall}(K_2 \square G)$ if and only if $3 \in \text{Fall}(G)$.

Notice that Item 3 gives us, as a corollary, the following important result:

Corollary 3.1 (Laskar; Lyle, 2009)**.** Let G be any graph and $Q_n = \square_{i=1}^{n} K_2$ the hypercube of dimension $n \geq 2$. Then, $3 \in \text{Fall}(Q_n \square G)$ if, and only if, $3 \in \text{Fall}(G)$. Moreover, $3 \notin \text{Fall}(Q_n)$ for any natural n.

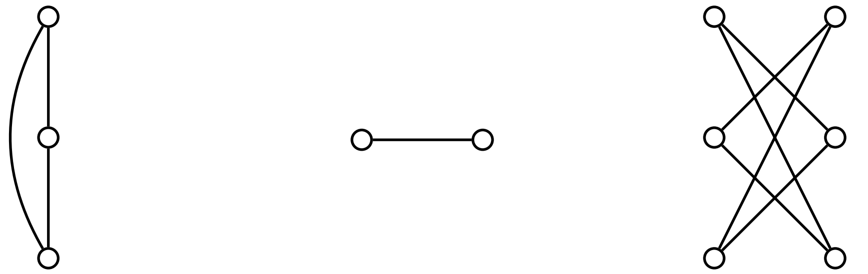
Such corollary is interesting because, still in (Laskar; Lyle, 2009), the authors partially answered the fifth question raised in the seminal article (Dunbar *et al.*, 2000): "What fall colorings do n-cubes have?" by means of the following theorem:

Theorem 3.6 (Laskar; Lyle, 2009)**.** For every positive integer n, the hypercube Q_n has no fall 3-coloring. However, for every $3 \neq k \geq 2$, there exists n such that $k \in \text{Fall}(Q_n)$.

3.1.2 Categorical and Lexicographic products

Before showing the following results, we first recall the definitions of both categorical and lexicographic products presented in Chapter 2. The *categorical product* (also known as: *direct product*, *tensor product*, *Kronecker product* and *conjunction*) of two graphs G and H is the graph $G \times H$ with vertex set $V(G) \times V(H)$ specified by putting (u,v) adjacent to (u',v') if and only if $u' \in E(G)$ and $vv' \in E(H)$. Figure 16 shows an example.

Figure 16 – Categorical product of the graphs C_3 and P_2.

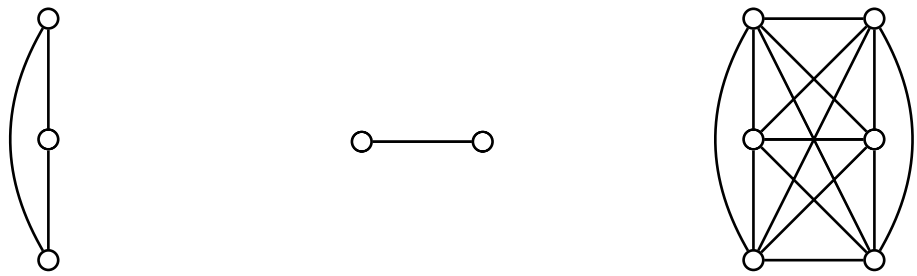


(a) K_3 (b) K_2 (c) $K_3 \times K_2$

Source: prepared by the author

The *lexicographic product* of two graphs G and H is the graph $G[H]$ with vertex set $V(G) \times V(H)$ specified by putting (u,v) adjacent to (u',v') if and only if either $uu' \in E(G)$ or $u = u'$ and $vv' \in E(H)$. Figure 17 shows an example.

Figure 17 – Lexicographic product of the graphs C_3 and P_2.



(a) C_3 (b) P_2 (c) $C_3[P_2]$

Source: prepared by the author

In (Dunbar *et al.*, 2000) the authors proved the following result for categorical product of two complete graphs:

Theorem 3.7 (Dunbar *et al.*, 2000)**.** If $r \geq 2$ and $s \geq 2$ are distinct and positive integers, then $\mathrm{Fall}(K_r \times K_s) = \{r, s\}$.

Proof. Let $V(K_r) = \{a_1, a_2, \ldots, a_r\}$ and $V(K_s) = \{b_1, b_2, \ldots, b_s\}$ and notice that vertices $u = (a_i, b_j)$ and $y = (a_n, b_m)$ are adjacent if and only if $i \neq n$ and $j \neq m$, by definition. Moreover, for each $1 \leq i \leq r$, let $A_i = \{(a_i, b_j) \mid 1 \leq j \leq s\}$ and, for each $1 \leq j \leq s$, let $B_j = \{(a_i, b_j) \mid 1 \leq i \leq r\}$. Finally, let I be a maximal independent set of $K_r \times K_s$ such that, for some $1 \leq i \leq r$, we have that $|I \cap A_i| \geq 2$ and assume that $n \neq m$ and that $(a_i, b_n), (a_i, b_m) \in I$. This being so, the set $\{(a_i, b_n), (a_i, b_m)\}$ dominates the set $V(K_r \times K_s) \setminus A_i$ and, since I is independent, it follows that $I \subseteq A_i$. On the other hand, we have that A_i is independent on $K_r \times K_s$ and, since I is dominant and a maximal independent set, we have that $A_i \subseteq I$. Hence, we have $I = A_i$. Similarly, we have that if I is a maximal independent set of $K_r \times K_s$ with $|I \cap B_j| \geq 2$, then $I = B_j$. Since we have that $\gamma(K_r \times K_s) \geq 2$, we know that $|I| \geq 2$ and therefore let $u, v \in I$. Since I is independent, u and v are not adjacent, so we must have $u, v \in A_i$ for some $i \in [r]$ or $u, v \in B_j$ for some $j \in [s]$, which would give us $I = A_i$ and $I = B_j$, respectively. Therefore, the only way to partition $K_r \times K_s$ into maximal independent sets is by partitions with sets of type $\{A_1, A_2, \ldots, A_r\}$ of partitions with sets of type $\{B_1, B_2, \ldots, B_s\}$, i.e., $K_r \times K_s$ can have only fall r-colorings or fall s-colorings. □

Note, however, that this result cannot be extended to the categorical product of more than two complete graphs, given that the graph $K_2 \times K_3 \times K_4$ has the colors $2, 3$ and 4 in its fall spectrum, but also has a fall 6-coloring. Indeed, the following partition of $K_2 \times K_3 \times K_4$ into 6 sets is a partition such that each set is independent and dominant: $\{(1,1,1),(1,2,2),(2,1,2),(2,2,1)\}$, $\{(1,2,1),(1,3,2),(2,2,2),(2,3,1)\}$, $\{(1,1,2),(1,3,1),(2,1,1),(2,3,2)\}$, $\{(1,1,3),(1,2,4),$ $(2,1,4),(2,2,3)\}$, $\{(1,2,3),(1,3,4),(2,1,4),(2,3,3)\}$ and $\{(1,1,4),(1,3,3),(2,1,3),(2,3,4)\}$. However, it holds that for categorical product of complete graphs K_i, for any i, the value i belongs to the fall achromatic spectrum of the product. Moreover, in (Shaebani, 2009), the author continues his studies on fall colorings of the categorical product of graphs, initially by generalizing the property presented earlier via the following theorem:

Theorem 3.8 (Shaebani, 2009)**.** Let n be any natural and $G_1, \ldots, G_n$ be arbitrary graphs. Then, for each $1 \leq i \leq n$, it holds that $\text{Fall}(G_i) \subseteq \text{Fall}(\times_{i=1}^{n} G_i)$.

Which gives us, directly, the following:

Corollary 3.2 (Valencia-Pabon, 2010)**.** Let $t \geq 3$ and $n_1, n_2, \ldots, n_{t+1}$ be such that $n_i \geq 2$, for all $i \in [t+1]$, and let $S \subseteq [t+1]$. If $k \in \text{Fall}(\times_{i \in S} K_{n_i})$, then $k \in \text{Fall}(\times_{i=1}^{t+1} K_{n_i})$.

Having established such a generalization, the author presented another interesting result, answering the following question in (Dunbar *et al.*, 2000): "Under what conditions does the categorical product of a set of complete graphs K_r, for every integer r in some specified set S, have a fall k-coloring for some integer k not in S?".

Theorem 3.9 (Shaebani, 2009)**.** Let $n \geq 3$, $S = \{k_1, \ldots, k_n\} \subseteq \mathbb{N}$, with $1 < k_1 < k_2 < \ldots < k_n$ and at least one k_i even. Then it holds that $S \subsetneq \text{Fall}(\times_{i=1}^{n} K_{k_i})$ and, furthermore, $\text{Fall}(\times_{i=1}^{n} K_{k_i})$ contains a natural greater than k_n.

Finally, for the lexicographic product of graphs, in (Shaebani, 2009) it is established the first fall spectrum results in the literature, as well as establishing some inequalities for the fall achromatic number of lexicographic products and for which graphs the inequalities are strictly satisfied. For the results on chromatic fall spectrum, we have:

Theorem 3.10 (Shaebani, 2009)**.** Let G and H be any graphs, $k \in \text{Fall}(G[H])$ and f be a fall k-coloring of $G[H]$. Then, for every $x \in V(G)$, f restricted to $V(H_x)$ is a fall coloring of H_x, where H_x is the subgraph of $G[H]$ induced by $\{x\} \times V(H)$.

Proof. Let $x \in V(G)$ and (x,y) be an arbitrary vertex of H_x colored with the color i and $S_x = f(V(H_x))$. Then, for every $j \in S_x \setminus \{i\}$, there exists a vertex (u,v) of $G[H]$ adjacent to (x,y) colored with color j. Notice that we must have $u = x$, otherwise, since $j \in S_x$, there would exist a vertex $(x,z) \in V(H_x)$ colored with color j. But this cannot occur, since (x,y) is adjacent to (u,v) and $x \neq a$, so we would have the edge $xu \in E(G)$; this being true, (u,v) and (x,z) would be vertices of color j adjacent in $G[H]$. Thus, we have that $u = x$ and $(u,v) \in V(H_x)$. Hence, S_x forms a fall $|S_x|$-coloring of H_x. □

Finally, the author proved the following result for lexicographic product of fall colorable graphs G and H:

Theorem 3.11 (Shaebani, 2009)**.** Let G and H be graphs such that $\text{Fall}(G) \neq \emptyset$ and $\text{Fall}(H) \neq \emptyset$. Then, it holds that $\{\sum_{i=1}^{s} k_i \mid \exists s \in \text{Fall}(G)\ \forall 1 \leq i \leq s\ (k_i \in \text{Fall}(H))\} \subseteq \text{Fall}(G[H])$.

3.2 Computational Complexity

In this section, we present results found in the literature on the computational complexity of the following decision problem:

FALL k-COLORING

Input:	A graph $G = (V(G), E(G))$ and a positive integer k.
Question:	$k \in \text{Fall}(G)$?

We also show the results found on the computational complexity of the following optimization problem:

FALL ACHROMATIC NUMBER

Input:	A graph $G = (V(G), E(G))$ such that $\text{Fall}(G) \neq \emptyset$.
Question:	What is the value of $\psi_f(G)$?

In (Dunbar *et al.*, 2000) and (Heggernes; Telle, 1998), independently, the authors have established the complexity of FALL k-COLORING, for every fixed $k \geq 3$. In order to present the reduction, we define the following concepts. Given a set of variables $X = \{x_1, x_2, \ldots, x_n\}$ and a set of clauses $C = \{C_1, C_2, \ldots, C_m\}$ such that each clause has exactly three literals from the variables in X, a *Not-All-Equal (NAE) assignment* is an assignment of TRUE and FALSE values to the variables such that for every clause C_i there is at least one literal assigned TRUE and one literal assigned FALSE. We have the following problem, which NP-completeness is proved in (Schaefer, 1978):

NOT-ALL-EQUAL-3SAT

Input:	A boolean formula ϕ in 3-CNF.
Question:	Does there exist a Not-All-Equal assignment of X?

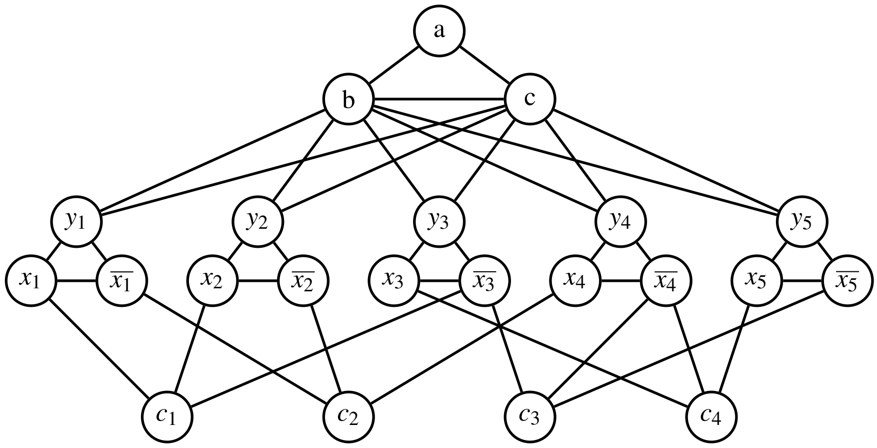
For the complexity of FALL k-COLORING, we have:

Theorem 3.12 (Dunbar *et al.*, 2000; Heggernes; Telle, 1998)**.** Let G be any graph and $k \geq 3$ an integer. It is NP-Complete to decide whether $k \in \text{Fall}(G)$.

Proof. First, note that if the problem of, given a graph G, deciding whether $k \in \text{Fall}(G)$ is NP-complete, then the problem of deciding whether a graph G has a fall $(k+1)$-coloring is also NP-complete. Indeed, given a graph G, let G' be the graph obtained from G by the addition of a universal vertex u, i.e. such that $V(G') = V(G) \cup \{u\}$ and $E(G') = E(G) \cup \{uv \mid v \in V(G)\}$. Thus, G' has a fall $(k+1)$-coloring if and only if G has a fall k-coloring.

Note that we can check whether a k-coloring f of a graph G is a fall k-coloring in polynomial time by checking, for each vertex $v \in V(G)$ if, for every color $1 \leq i \leq k$ such that $f(v) \neq i$, v is adjacent to at least one vertex colored i. Thus, FALL k-COLORING is in NP. We now establish now a polynomial reduction from the NP-complete problem NOT-ALL-EQUAL-3SAT. We transform I to an instance $(G_I, 3)$ of FALL 3-COLORING problem. For a representation of the construction of the graph G_I described in the sequel, see Figure 18.

Figure 18 – The graph G_I obtained from a NOT-ALL-EQUAL-3SAT instance, where $c_1 = x_1 \vee x_2 \vee \overline{x_3}$, $c_2 = \overline{x_1} \vee \overline{x_2} \vee x_4$, $c_3 = \overline{x_3} \vee \overline{x_4} \vee x_5$ and $c_4 = x_3 \vee \overline{x_4} \vee x_5$.



Source: prepared by the author

Initialize G_I with k disjoint copies of the graph K_3 and label the vertices of each copy as $\{y_i, x_i, \overline{x_i}\}$. Label the vertices of another copy of K_3 with $\{a,b,c\}$ and add edges by_i and cy_i for every $i = 1,2,\ldots,k$. Corresponding to each clause $C_i \in C$, add a single vertex c_i to the graph G_I. Finally, join each vertex c_i to the three vertices corresponding to the literals in the clause C_i. The construction can be accomplished in polynomial time, since $|V(G_I)| = m + 3(n+1)$. All that remains to show is that I is satisfiable by a NAE assignment if and only if G_I has a fall 3-coloring. Assume first that I has a satisfying truth NAE assignment $f : X \to \{T,F\}$. Color the vertices $\{a,b,c\}$ in G_I with the colors $\{1,2,3\}$ respectively. Next color every vertex y_i and every vertex c_i with color 1. Finally, for $i = 1,2,\ldots,k$, if $f(x_i) = T$, color the associated vertex x_i with color 2. Otherwise, color $\overline{x_i}$ with color 2. Color all remaining vertices with color 3. We argue that this is a proper coloring, and every vertex in a triangle is a b-vertex: note that vertices a, b and c are colored with distinct colors, thus they are b-vertices. Vertices y_i are colored with the same color of a and are adjacent to vertices b and c. In addition, $f(x_i) \neq f(\overline{x_i})$ and x_i and $\overline{x_i}$ are colored with colors 2 and 3 and are adjacent to y_i. It remains to prove that each vertex c_i is a b-vertex. Since f is a satisfying truth NAE assignment, every clause C_i has at least one TRUE literal and at least one FALSE literal. Thus every vertex c_i is a b-vertex.

Assume next that G_I has a fall 3-coloring f. Then, without loss of generality, we may assume the vertex a is colored 1. Since a is a b-vertex, b and c must have colors 2 and 3, which forces y_i to have color 1, for every $i \in \{1,\ldots,n\}$. Since each color class is an independent set, this means the vertices x_i and $\overline{x_i}$, for $i \in \{1,\ldots,n\}$, must have distinct colors in $\{2,3\}$. Thus, each c_i must have color 1, for $i \in \{1,\ldots,n\}$. We define a function $f : X \to \{T,F\}$ by letting

$f(X_i) = T$ if and only if x_i has color 2. Since the coloring of G_I is a fall 3-coloring, each vertex c_i is adjacent to at least one vertex with color 2 and one vertex with color 3. Thus the function f is a satisfying truth NAE assignment for the instance I of NOT-ALL-EQUAL-3SAT. □

While in (Dunbar *et al.*, 2000) the authors use a reduction from NOT-ALL-EQUAL-3-SAT, in (Heggernes; Telle, 1998) they made a reduction from the known problem k-EDGE-COLORING on k-regular graphs. Having established the complexity of the problem for general graphs, it is natural to try to establish the complexity for specific classes of graphs.

Note that $2 \in \text{Fall}(G)$ if and only if G is bipartite without isolated vertices, which can be verified in $O(n+m)$ time, where $n = |V(G)|$ and $m = |E(G)|$. In (Laskar; Lyle, 2009), the authors showed, also with a reduction from NOT-ALL-EQUAL-3SAT, the following result for bipartite graphs:

Theorem 3.13 (Laskar; Lyle, 2009)**.** Let G be a bipartite graph. Deciding whether $3 \in \text{Fall}(G)$ is NP-complete.

Complementing this result, in (Lauri; Mitillos, 2020), the authors proved the following:

Theorem 3.14 (Lauri; Mitillos, 2020)**.** For every $k \geq 3$, it is NP-complete to decide whether $k \in \text{Fall}(G)$, where G is a bipartite graph.

Continuing the study of problem complexity for bipartite graphs, in (Silva, 2019) the author demonstrated the following theorem, making a reduction from EDGE COLORING of 3-regular graphs:

Theorem 3.15 (Silva, 2019)**.** Let G be a bipartite graph with $|E(G)| \geq 1$. It is NP-complete to decide if $|\text{Fall}(G)| > 1$.

Notice that the assumption that G has at least one edge is relevant because, otherwise, G would be an independent set, so we would have $\text{Fall}(G) = \{1\}$, and therefore $\text{Fall}(G)$ always has size 1. For even more restrictive cases of bipartite graphs, in (Campos *et al.*, 2021) the authors proved:

Theorem 3.16 (Campos *et al.*, 2021)**.** Let G be a bipartite graph with diameter less than or equal to 4. Deciding whether $3 \in \text{Fall}(G)$ is NP-complete.

Another interesting case is the class of complement of bipartite graphs, since for graphs belonging to this class the cardinality of each color class is at most two, for every proper k-coloring. As such, in (Shaebani, 2009) the following result was proved:

Theorem 3.17 (Shaebani, 2009)**.** Let G be a bipartite graph. Then $\mathrm{Fall}(\overline{G}) \subseteq \{\chi(\overline{G})\}$ and it is possible to decide in polynomial time whether $\mathrm{Fall}(\overline{G}) \neq \emptyset$.

For bipartite planar graphs, in (Lauri; Mitillos, 2020) the authors showed the following:

Theorem 3.18 (Lauri; Mitillos, 2020)**.** Let G be a bipartite planar graph such that $\Delta(G) = 6$. Then it is NP-complete to decide whether $3 \in \mathrm{Fall}(G)$.

Also in (Silva, 2019), the author established more properties for the fall spectrum of some relevant classes of graphs. For chordal graphs, the author made a reduction from the EDGE COLORING of 3-regular graphs to prove the following result:

Theorem 3.19 (Silva, 2019)**.** Let G be a chordal graph such that $\chi(G) = \delta(G) + 1$. It is NP-complete to decide whether $\mathrm{Fall}(G) \neq \emptyset$.

We say that a graph G is P_4*-sparse* if every 5 vertices of $V(G)$ induce at most one P_4 and this concept was firstly introduced in (Jamison; Olariu, 1992). Furthermore, we say that a graph G is a *spider* if its vertex set can be partitioned into disjoint sets S, K and R such that:

1. $|S| = |K| \geq 2$. S is an independent set and K is a clique;
2. Every vertex in R is adjacent to all the vertices in K and no vertex in S;
3. There exists a bijection $f : S \to K$ such that either:

$$N_G(s) \cap K = K - \{f(s)\}, \text{ for all vertices } s \in S, \text{ in which case we the spider is called } \textit{fat}, \text{ or}$$

$$N_G(s) \cap K = \{f(s)\}, \text{ for all vertices } s \in S, \text{ in which case it is called } \textit{thin}.$$

It is proved in (Giakoumakis; Vanherpe, 1997) that a graph G is P_4 sparse if and only if the quotient graph of each neighborhood node of its modular decomposition tree is isomorphic to a spider H.

Finally, for the class of P_4-sparse graphs, which contains the class of cographs, (Silva, 2019) established the following:

Theorem 3.20 (Silva, 2019)**.** Let G be a P_4-sparse graph. If G is not a cograph and $\text{Fall}(G) \neq \emptyset$, then $\text{Fall}(G) = \{\chi(G)\}$. If G is a cograph, then $\text{Fall}(G) \subseteq \{\chi(G)\}$ and it is possible to decide in polynomial time whether $\text{Fall}(G) \neq \emptyset$.

Proof. First, we prove that if G is a P_4-sparse graph such that $\text{Fall}(G) \neq \emptyset$, then G is either a cograph or a fat spider with empty head. Let G be the spider (C,S,R). If G is not a fat spider, we know that $|C| \geq 3$ and that $d(u) = 1$ for every $u \in S$. This contradicts the fact that $\text{Fall}(G) \neq \emptyset$, since $\chi(G) \geq \omega(G) \geq 3$, but $\delta(G)+1 = 2$ and $\psi_f(G) \leq \delta(G)+1$. Also, if G is a fat spider, then R must be empty since the colors in R cannot appear in C, i.e., the vertices of S could not be b-vertices, as S is an independent set. Finally, note that for every fall coloring f of G, and every $u \in S$ with non-neighbor u' in C, we must have $f(u) = f(u')$ as otherwise u would not be a b-vertex. Therefore, every fall coloring is also an $\chi(G)$-coloring, and clearly an $\chi(G)$-coloring is a fall coloring, i.e., we get that $\text{Fall}(G) = \{\chi(G)\}$.

Now, suppose that G is a cograph. We prove that $\text{Fall}(G) \subseteq \{\chi(G)\}$ by induction on $n = |V(G)|$. If $n = 1$, then trivially we have that $\text{Fall}(G) = \{\chi(G)\}$. Suppose, hence, that $\text{Fall}(G') \subseteq \{\chi(G')\}$ for every cograph G' with $k < n$ vertices. Recall that, as presented in Chapter 2, for every cograph G there exist two cographs G_1 and G_2 such that G is either the union or join of G_1 and G_2. Therefore, let G be a cograph with $|V(G)| = n$ and let G_1 and G_2 be the cographs such that G is either the union or join of G_1 and G_2. We prove now that:

$$\text{Fall}(G) = \begin{cases} \{k+l \mid k \in \text{Fall}(G_1) \wedge l \in \text{Fall}(G_2)\}, & \text{if } G \text{ is the join of } G_1 \text{ and } G_2; \\ \{\text{Fall}(G_1) \cap \text{Fall}(G_2)\}, & \text{if } G \text{ is the union of } G_1 \text{ and } G_2. \end{cases}$$

In the former case, if f_1 and f_2 are fall colorings of G_1 and G_2 with k_1 and k_2 colors, respectively, then a fall coloring of G with k_1+k_2 colors can be obtained by using distinct colors in f_1 and f_2. On the other hand, if f is a fall coloring of G, then the colors used in G_1 and G_2 must be distinct, what implies that each vertex in G_i must be a b-vertex in f restricted to $V(G_i)$, for $i = 1$ and $i = 2$. Now, if G is the union of G_1, G_2 and f is a b-coloring of G with k colors, then f restricted to $V(G_i)$ is a b-coloring of G_i with k colors, for $i = 1$ and $i = 2$. Also $\text{Fall}(G_1) \cap \text{Fall}(G_2) \subseteq \text{Fall}(G)$ clearly holds.

Now, when G is the join of G_1 and G_2, then the equation above gives us that $\text{Fall}(G) = \{k+l | k \in \text{Fall}(G_1) \wedge l \in \text{Fall}(G_2)\}$, which by induction hypothesis is contained in $\{\chi(G_1)+\chi(G_2)\}$, which equals $\{\chi(G)\}$. If G is the union of G_1 and G_2, then by induction hypothesis $\text{Fall}(G) \subseteq \{\chi(G_1)\} \cap \{\chi(G_2)\}$ which is non-empty if and only if $\chi(G_1) = \chi(G_2)$, in which case we know that $\chi(G)$ also equals $\chi(G_1)$. Observe that the proof gives a polynomial algorithm to decide whether $\mathrm{Fall}(G) \neq \emptyset$. $\square$

For 3-regular graphs, in (Lauri; Mitillos, 2020), the authors showed the following:

Theorem 3.21 (Lauri; Mitillos, 2020)**.** Let G be a 3-regular graph. It is NP-complete to decide whether $4 \in \mathrm{Fall}(G)$.

Following the study of the complexity problem for regular graphs, still in (Lauri; Mitillos, 2020), the authors established:

Theorem 3.22 (Lauri; Mitillos, 2020)**.** Let G be a p-regular graph, with $p \in \{2k-2, 2k-1\}$ for some positive integer $k \geq 3$. Then it is NP-complete to decide whether $p \in \mathrm{Fall}(G)$.

Furthermore, in (Barth *et al.*, 2009), the authors proved that, given a graph G with n vertices, there is no polynomial-time $n^{1-\varepsilon}$-approximation algorithm for any real number $\varepsilon > 0$ to compute $\psi_f(G)$, unless $\mathsf{P} = \mathsf{NP}$. Recall that a graph G such that $\mathrm{Fall}(G) \neq \emptyset$ is *fall continuous* if, being ℓ the least positive integer such that G admits a fall ℓ-coloring, it holds that $\mathrm{Fall}(G) = \{\ell, \ldots, \psi_f(G)\}$. The authors also proved the following result:

Theorem 3.23 (Barth *et al.*, 2009)**.** Let G be a graph such that $\mathrm{Fall}(G) \neq \emptyset$. It is NP-complete to decide whether G is fall continuous.

Again in (Lauri; Mitillos, 2020), the authors settled the following theorem for bounded treewidth graphs:

Theorem 3.24 (Lauri; Mitillos, 2020)**.** Let G be a graph of bounded treewidth. Then, computing $\mathrm{Fall}(G)$ can be done in polynomial time.

In the same paper, the authors further showed the following property for kernelization:

Theorem 3.25 (Lauri; Mitillos, 2020)**.** Let G be a graph. Decide whether $k \in \mathrm{Fall}(G)$ parameterized by treewidth does not admit a polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP}/\mathsf{poly}$.

In addition, we have the following:

Theorem 3.26 (Lauri; Mitillos, 2020)**.** Deciding whether $3 \in \mathrm{Fall}(G)$, where G is planar can be done in $2^{\mathcal{O}(\sqrt{|V(G)|})}$ time and this is the best possible, unless ETH fails.

3.3 Further results

Previously, we grouped similar results for which one can find several related contributions in the literature. In this section, we present results on fall colorings of graphs which did not fit in the previous sections, but are important contributions to the state of the art regarding this problem. Recall that, as defined in Chapter 2, the minimum positive integer such a graph G admits a fall k-coloring is denoted by $\chi_f(G)$. In (Balakrishnan; Kavaskar, 2010), the authors answered positively the following question raised in (Dunbar *et al.*, 2000): "Can the difference between $\chi(G)$ and $\chi_f(G)$ be arbitrarily large?". Later, in (Kaul; Mitillos, 2019), the authors also answered the same question. They proved:

Theorem 3.27 (Balakrishnan; Kavaskar, 2010; Kaul; Mitillos, 2019)**.** For any positive integers $3 \leq a \leq b$ there exists an infinite family of graphs $\{G_i\}_{i \in \mathbb{Z}_+}$ such that $\chi(G_i) = a$ and $\chi_f(G_i) = b$.

Again, recall that $\psi(G)$ is the maximum positive integer k such that a graph G admits a complete k-coloring, as defined in Chapter 1. In addition, as defined in Chapter 2, $b(G)$ is the maximum positive integer k such that the graph G admits a b-coloring using k colors, Γ is the maximum positive integer k such that the graph G admits a first-fit coloring using k colors and $\partial\Gamma(G)$ is the maximum positive integer k such that the graph G admits a partial Grundy coloring using k colors. Moreover, in (Balakrishnan; Kavaskar, 2011), the authors have provided another answer to the following question, also raised in (Dunbar *et al.*, 2000): "Does there exist a graph G such that $\chi(G) < \chi_f(G) < \psi_f(G) < b(G) < \Gamma(G) < \psi(G)$?"

Theorem 3.28 (Balakrishnan; Kavaskar, 2011)**.** There exists an infinite family of graphs G such that $\chi(G) < \chi_f(G) < \psi_f(G) < b(G) < \partial\Gamma(G) < \psi(G)$.

Further, in (Balakrishnan *et al.*, 2012), the authors observe the following result regarding the cardinality of the set of graphs with empty and non-empty fall spectrum:

Proposition 3.4 (Balakrishnan *et al.*, 2012)**.** The set of connected graphs G such that $\text{Fall}(G) \neq \emptyset$ and the set of connected graphs G' with $\text{Fall}(G') = \emptyset$ are infinite.

For the next result, we need further definitions. A *wheel* of order n is a graph obtained by adding to a cycle on $n-1$ vertices a central vertex that is adjacent to each vertex on the cycle; it is denoted by W_n. A *helm* of order n is a graph obtained from a wheel by adding a leaf to each vertex in the cycle. A *Flower graph* Fl_n is a graph obtained from a helm by adding, for each leaf, an edge between the leaf and the central vertex. A *Sunflower graph* SFl_n is a graph obtained by replacing each edge of the rim of a wheel graph W_n by a triangle such that two triangles share a common vertex if and only if the corresponding edges in W_n are adjacent in W_n. In (Kalpana; Vijayalakshmi, 2018), the authors proved the following properties for the complement of Flower and Sunflower graphs.

Theorem 3.29 (Kalpana; Vijayalakshmi, 2018)**.** Let Fl_n and SFl_n be the flower and sunflower graphs, respectively. Let $\chi_f(\overline{Fl_n}) = \psi_f(\overline{Fl_n}) = n$ and $\chi_f(\overline{SFl_n}) = \psi_f(\overline{SFl_n}) = 2n$.

A graph G is *minimal k-fall-imperfect* if $\psi_f(G) > \chi(G) = k$ and every proper subgraph of G is fall perfect. This concept was firstly introduced in (Silva, 2019). The following holds:

Proposition 3.5 (Silva, 2019)**.** Let G be a minimal k-fall imperfect graph. Then $\psi_f(G) = k+1$ and $\psi_f(G-u) = k$ for all $u \in V(G)$.

In the same work, the author showed the following characterization for bipartite, fall perfect graphs:

Theorem 3.30 (Silva, 2019)**.** Let G be a bipartite graph. Then G is fall perfect if and only if G has no C_{6k} as an induced subgraph, for all $k > 0$.

In (Kaul; Mitillos, 2019), the authors demonstrated a number of results about the existence of fall colorings, one of them being:

Theorem 3.31 (Kaul; Mitillos, 2019)**.** Let G be a k-colorable graph with $\delta(G) > \frac{k-2}{k-1}|V(G)|$, for $2 \leq k \leq |V(G)|$. Then every k-coloring of G is also a fall k-coloring of G.

Furthermore, in the same paper, we have the following result that shows that the lower bound on the minimum degree cannot be improved:

Theorem 3.32 (Kaul; Mitillos, 2019)**.** Let G be a k-colorable graph, $\frac{k-2}{k-1}|V(G)|$-regular, with $2 \leq k \leq |V(G)|$. Then every k-color of G is either a fall k-coloring or can be converted to a fall $(k-1)$-coloring by merging two color classes. Moreover, there always exists G as described such that $k-1 \in \text{Fall}(G)$ and $k \notin \text{Fall}(G)$.

In (Jaffke *et al.*, 2023), the authors prove the following result for fall coloring using as parameter the module-width:

Theorem 3.33 (Jaffke *et al.*, 2023)**.** There is an algorithm that decides if a graph G has a fall k-coloring in time $n^{2^{\mathcal{O}(w)}}$, where n denotes the number of vertices of the input graph, and w denotes the module-width of a given rooted branch decomposition of the input graph.

Finally, in the same paper, the authors prove the following result:

Proposition 3.6 (Jaffke *et al.*, 2023)**.** Deciding whether a graph on n vertices has a fall k-coloring parameterized by the module-width w of the input graph is W[1]-hard and cannot be solved in time $n^{2^{o(w)}}$, unless ETH fails.

Note that the last result gives a lower bound based on the Exponential Time Hipothesis (ETH), and therefore it proves that the algorithm running in time $n^{2^{\mathcal{O}(w)}}$ is optimal, unless ETH fails.

3.4 Subfall Colorings

In this section, we present the results on subfall colorings presented in (Dunbar *et al.*, 2000). We emphasize that, apart from this work, there are no other works on subfall colorings present in the literature. In the seminal paper, it is first proved the following:

Proposition 3.7 (Dunbar *et al.*, 2000)**.** The difference between $\psi_f(G)$ and $\psi_{fs}(G)$ can be arbitrarily large.

Proof. Consider the graph G which consists of a complete bipartite graph $K_{n,n}$ minus the edges in a perfect matching, with an additional vertex adjacent to one vertex of $K_{n,n}$. Since $\psi_f(G) \leq \delta(G)+1$, it holds that $\psi_f(G) = 2$, while $\psi_{fs}(G) = n$. □

Finally, the authors showed:

Theorem 3.34 (Dunbar *et al.*, 2000)**.** For every tree T, $\psi_{fs}(T) = 2$.

While few results on the problem have been proven, the authors raised two questions about the subfall achromatic number of graphs. The first one, with respect to the NP-completeness of the problem of deciding whether a graph G admits a subfall k-coloring, is partially answered in Theorem 4.1. The second question, asking whether $\psi_{fs}(G) \leq b(G)$ for every graph G, was answered in the negative in Theorem 4.3.

4 OUR CONTRIBUTIONS

Recall that, given a graph G, a subfall k-coloring of G is a fall k-coloring of some induced subgraph $H \subseteq G$; and $\psi_{fs}(G)$ is the maximum integer k such that a graph G has a subfall k-coloring. In this chapter, we present our results on subfall coloring and its optimization parameter ψ_{fs}. In Section 4.1, we show that deciding whether a graph G has a subfall k-coloring is an NP-complete problem for every fixed $k \geq 4$, and provide a characterization for graphs that are subfall 3-colorable. In Section 4.2, we provide an FPT algorithm to compute ψ_{fs} when parameterized by the treewidth of the input graph. We also show how to adapt this algorithm in order to decide whether a graph has a fall k-coloring when parameterized by the treewidth; and b-coloring using k colors when parameterized by the treewidth plus the number of colors. In Section 4.3, we show some properties of the optimization parameter ψ_{fs}, such as its continuity and its relationship to similar parameters. In Section 4.4, we introduce the subfall achromatic index of graphs and provide some results. All the results found in Sections 4.1 and 4.3 were presented in *VI Encontro de Teoria da Computação* (ETC) in 2021 and the results found in Section 4.2 were presented in *VII Encontro de Teoria da Computação* (ETC) in 2022.

4.1 Computational Complexity of subfall k-coloring

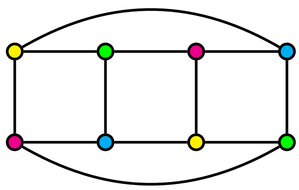
In this section, we present our complexity results. Note that every graph with non-empty edge set has a subfall coloring with 2 colors. Moreover, every graph with a subfall 2-coloring has non-empty edge set, since its subgraph that admits a fall 2-coloring must have an edge, otherwise the vertices would have no neighbors and, therefore, would not be b-vertices. So, we study the complexity of subfall k-coloring for $k \geq 3$. By applying a result in (Lauri; Mitillos, 2019), we first obtain:

Theorem 4.1. Deciding whether a connected graph G has a subfall k-coloring is NP-complete for every fixed $k \geq 4$.

Proof. To see that the problem is in NP, just observe that, given a coloring $f : V(H) \to \{1,2,3,\ldots,k\}$ of a subgraph $H \subseteq G$, one can verify in polynomial time whether f defines a fall k-coloring of H. We split the proof into two cases: $k = 4$ and $k \geq 5$.

For $k = 4$, we make a reduction from fall 4-coloring of 3-regular graphs, which is known to be NP-complete (Lauri; Mitillos, 2019), in which we use the same instance. Clearly, if G has a fall 4-coloring, then G has a subfall 4-coloring. On the other hand, suppose H is an induced subgraph of G that has a fall 4-coloring, f. Because f uses 4 colors and each vertex of H is a b-vertex in f, it follows that $d_H(u) \geq 3$ for every $u \in V(H)$. But now, since G is connected, if $V(H) \neq V(G)$, it means that some $u \in V(H)$ has a neighbor in $V(G-H)$. This is a contradiction since G is 3-regular, and this would imply that $d_H(u) < 3$. Therefore we get $V(H) = V(G)$ as we wanted to prove. Figure 19 shows an example of a 3-regular graph that admits a fall 4-coloring. Note that deleting any vertex reduces the minimum degree of the graph, and thus ψ_f by Proposition 2.1.

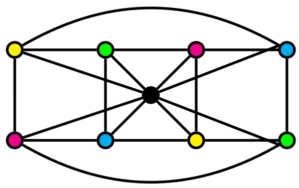
Figure 19 – Example of a 3-regular graph G with a (sub)fall 4-coloring.



Source: prepared by the author

For $k \geq 5$, we construct the following graph G': take a 3-regular graph G and a complete graph C on $k-4$ vertices, and add all edges joining $V(G)$ and $V(C)$. We show that G' has a subfall k-coloring if and only if G has a subfall 4-coloring. If G has a subfall 4-coloring, we can obtain a subfall k-coloring of G' just by coloring C with $(k-4)$ new colors. Conversely, let $H' \subseteq G'$ be an induced subgraph that has a fall k-coloring. Let $H \subseteq G$ be equal to H' restricted to G, i.e. $H = G[V(H') \cap V(G)]$, and $H_c \subseteq H'$ be equal to H' restricted to C. Because $|V(H_c)| \leq k-4$ and no color in H_c can be used in H and vice-versa, we get that H must be fall $(k-k')$-colorable, where $k' = |\{f(v) \mid v \in V(C) \cap V(H')\}|$. Note that $k' \leq k-4$ implies $k-k' \geq 4$, while H being a subgraph of a 3-regular graph implies that $\Delta(H) \leq 3$, giving us $k-k' \leq 4$, and hence $k-k' = 4$. Thus, we have that H must be fall 4-colorable and G has a subfall 4-coloring. Figure 19 shows an example of a 3-regular graph that admits a fall 4-coloring; note that deleting any vertex reduces the minimum degree of the graph.

Figure 20 – Example of construction of G' for $k = 5$, using G as in Figure 19.



Source: prepared by the author

□

From Theorem 3.21 and Theorem 4.1, we obtain the following corollary:

Corollary 4.1. It is NP-complete to decide whether $p_1 = p_2$, where $p_1 \in \{\psi_f(G), \psi_{fs}(G)\}$ and $p_2 \in \{\delta(G)+1, \mathrm{col}(G)+1\}$.

Note that, for $k = 3$, we could not settle the complexity of the problem, which seems challenging. Instead, we present the following characterization of subfall 3-colorable graphs, that might help in its solution:

Theorem 4.2. A graph G has subfall 3-coloring if and only if it contains a cycle C_{3k} as induced subgraph, for some positive integer k.

Proof. Note that, if G contains an induced cycle C isomorphic to C_{3k}, for some $k \in \mathbb{N}$, then a 3-coloring of the vertices of C, in consecutive order, with $1,2,3$ is a fall 3-coloring of C. Conversely, let f be a subfall 3-coloring of G, and let $H \subseteq G$ be the subgraph of G colored by f. We will construct a cycle of length multiple of 3 from H by these steps: let $v \in V(H)$ be a vertex with $f(v) = 1$. Because every vertex of H is a b-vertex, we get that v is adjacent to at least one vertex u colored with 2. Now, we construct a path (v_1, ..., v_ℓ), with $v_1 = v$ and $v_2 = u$, by taking next a neighbor of v_ℓ colored with the least possible color distinct from $f(v_{\ell-1})$ and $f(v_{\ell-2})$. Mote that it exists because every vertex in H is a b-vertex with respect to f, and f uses 3 colors. Since H is finite, eventually these steps will lead us to an already iterated vertex, at which point we stop before adding a repeated vertex. Let $P = (v_1, ..., v_q)$ be the constructed path and let $v_j \in V(P)$ be the neighbor of v_q in P of color distinct from v_{q-1}.

We first prove that f restricted to $C = (v_j, ..., v_q)$, which we denote by f', is a fall 3-coloring of C. Note that, by the choices in the construction of P, we have that all vertices are b-vertices in f', with the exception of possibly v_j. Hence, we only need to prove that

$f(v_q) \neq f(v_{j+1})$. To see this, denote the colors alternating in C starting from v_j by a,b,c. We know that the immediately subsequent vertices in the cyclic order of vertices of color b are of color c. Therefore, since the successor of v_q in the cyclic order is v_j and $f(v_j) = a$, we must have $c = f(v_q) \neq b = f(v_{j+1})$. It remains to argue that C is an induced cycle. In fact, we argue that if C is any cycle such that f restricted to C is a fall 3-coloring of C, then either C has no chords, or C contains a smaller cycle with the same property. Indeed, write $C = (v_1,\dots,v_q)$, suppose $f(v_1) = 1$, and let v_1v_j be a chord in C. One can verify that either $f(v_j) = 2$ and f restricted to $(v_1,v_j,v_{j+1},\dots,v_q)$ is a fall 3-coloring, or $f(v_j) = 3$ and f restricted to $(v_1,v_2,\dots,v_j)$ is a fall 3-coloring. □

Given the NP-completeness of deciding whether $\psi_{fs}(G) \geq k$ for general graphs G, it is natural to investigate the complexity of the problem restricted to specific graph classes. We investigate now two of the most studied graph classes: chordal graphs and cographs. In (Silva, 2019), the author proves that chordal graphs and cographs are fall perfect. Furthermore, chordal graphs and cographs are both hereditary classes of perfect graphs. The aforementioned result and the following one give us that computing $\psi_{fs}(G)$ can be done in polynomial time on chordal graphs and cographs, since computing $\omega(G)$ in these classes can be done in polynomial time.

Theorem 4.3. Let $\mathscr{G}$ be a hereditary class of graphs which are perfect and fall perfect. Then $\psi_{fs}(G) = \omega(G)$, for every $G \in \mathscr{G}$.

Proof. Let $G \in \mathscr{G}$. We know that $\psi_{fs}(G) \geq \omega(G)$ for every graph G since any proper coloring of the maximum clique is a subfall coloring of G. In addition, because G is fall perfect, we get that $\mathrm{Fall}(H) \subseteq \{\chi(H)\}$ for every $H \subseteq G$. In other words, $\psi_{fs}(G) \leq \max_{H\subseteq G} \chi(H) \leq \chi(G) = \omega(G)$, where the last equality holds because G is a perfect graph. □

4.2 Parameterized complexity of subfall k-coloring

In this section, we present our parameterized complexity results. We give an explicit dynamic-programming algorithm that, given a graph G and an integer $k \geq 3$, decides whether G admits a subfall k-coloring. The algorithm runs in FPT time when parameterized by the treewidth of G, for a fixed constant $k \geq 3$. Later, we show how to adapt our algorithm to an algorithm that decides whether a graph G has a fall k-coloring in FPT time when parameterized by the treewidth of G. We also show how to adapt our algorithm to an algorithm that decides whether a graph G admits a b-coloring using k colors. For this problem, the algorithm runs in FPT time when parameterized by the treewidth of G plus the number of colors. We mention that there already exist some results on these problems in the literature. In (Telle; Proskurowski, 1997), the authors give a general algorithm for solving locally checkable vertex partitioning problems, category in which both fall coloring and subfall coloring problems fall in; the algorithm runs in FPT time when parameterized by the treewidth of the graph. In (Jaffke *et al.*, 2023), the authors show an algorithm that decides whether a graph G has a b-coloring using k colors running in FPT time when parameterized by clique-width, and in (Jaffke *et al.*, 2022) the authors prove that deciding whether a graph G admits a b-coloring with k colors with $\text{tw}(G)$ as unique parameter is W[t]-hard for every $t \geq 1$.

Before we construct our explicit FPT algorithms found in this section, we present the following observation stating that the treewidth of a graph G is an upper bound for both fall and subfall achromatic numbers of G. This observation is very important throughout the section, since it allows us to use only the treewidth of the graph as a parameter for the algorithms for (sub)fall k-coloring, instead of using as $\text{tw}(G)+k$ as a parameter.

Proposition 4.1. For every graph G, $\psi_f(G) \leq \text{tw}(G)+1$ and $\psi_{fs}(G) \leq \text{tw}(G)+1$.

Proof. Since it holds that $\psi_f(G) \leq \delta(G)+1$, by 2.1, and that $\delta(G) \leq \text{tw}(G)$, it follows that $\psi_f(G) \leq \text{tw}(G)+1$. Furthermore, we have that $\text{tw}(H) \leq \text{tw}(G)$ for every subgraph $H \subseteq G$. Thus, it holds that $\text{col}(G) \leq \text{tw}(G)$ and, therefore, $\psi_{fs}(G) \leq \text{tw}(G)+1$, again by 2.1. □

In order to be able to present our explicit FPT algorithms found in this section, we provide some further definitions. Let $(\mathscr{X},T)$ be a nice tree decomposition of a graph G. For a node t of T, we denote by G_t the subgraph of G induced by the vertices $\bigcup_{t' \in V(T_t)} X_{t'}$, where T_t is the subtree of T rooted at t. Also, given a k-coloring f of X_t, we say that a k-coloring f' of G_t *extends* f to G_t if $f'(u) = f(u)$ for every $u \in X_t$.

We are now able to present our dynamic-programming algorithm. The general idea for our table is that, for a node $t \in V(T)$, we compute whether there exists a subset S of X_t such that there exists a k-coloring f and a subgraph H of G_t such that f can be extended to a *partial* fall k-coloring of H, where by partial we mean that every vertex of H is a b-vertex, with the exception of possibly some vertices of S. For this, we need to keep track of which colors are being used and which colors are missing in the neighborhood of each vertex of the bag X_t. Formally, given a node $t \in V(T)$, where $X_t = \{v_1, v_2, \ldots, v_q\}$, we define the table related to t as follows: for each subset $S = \{v_1, \ldots, v_p\} \subseteq X_t$, each proper k-coloring f of X_t, and each

$\mathscr{M} = \{M_0, M_1, M_2, \ldots, M_p\}$, with $M_i \subseteq [k]$ for every i, we say that $c_t(S, f, \mathscr{M}) = 1$ if and only if there exist $H \subseteq G_t$ and a k-coloring f' that extends f to H that satisfy:

1. Every $u \in V(H) \setminus X_t$ is a b-vertex in f';
2. For every $v_i \in S$, we have $f'(N_H[v_i]) = [k] \setminus M_i$ (i.e., M_i are the missing colors for v_i);
3. $f'(V(H)) = [k] \setminus M_0$; and
4. $V(H) \cap X_t = S$.

Otherwise, $c_t(S, f\mathscr{M}) = 0$. The correctness of such a procedure can be expressed in the following lemma:

Lemma 4.1. Let r be the root of a nice tree decomposition of G, with $\mathscr{M}_\emptyset = \{M_0\}$ where $M_0 = \emptyset$. Then G has a subfall k-coloring if and only if $c_r(\emptyset, \emptyset, \mathscr{M}_\emptyset) = 1$.

Proof. Since r is the root node of a nice tree decomposition, we know that $X_r = \emptyset$, thus we have $S = \emptyset$. If $c_r(\emptyset, \emptyset, \mathscr{M}_\emptyset) = 1$, then there exist a subgraph $H \subseteq G$ and a k-coloring f such that every vertex $u \in V(H)$ is a b-vertex in f and $f(V(H)) = [k]$, i.e., f is a fall k-coloring of H. Thus, G has a subfall k-coloring.

On the other hand, if G has a subfall k-coloring, then there exist a subgraph $H \subseteq G$ such that H has a fall k-coloring f. Therefore, f extends the (empty) coloring of X_r to H and trivially satisfies all the four conditions. □

We now show how to compute $c_t(f, S, \mathscr{M})$ based on each type of node of a given nice tree decomposition. For leaf nodes, we have:

Lemma 4.2. Let t be a leaf node (and hence $X_t = \{v_1\}$ for some $v_1 \in V(G)$). Also, let $S \subseteq X_t$ and $f : S \to [k]$. Then,

$$c_t(S, f, \mathscr{M}) = 1 \quad \text{if and only if} \quad v_1 \notin S, \mathscr{M} = \{M_0\} \text{ and } M_0 = [k],$$
$$\text{or } v_1 \in S, \mathscr{M} = \{M_0, M_1\} \text{ and } M_0 = M_1 = [k] \setminus \{f(v_1)\}.$$

Clearly, deciding $c_t(S, f, \mathscr{M})$ can be done in time $\mathscr{O}(1)$. Before showing how to compute $c_t(S, f, M)$ for join nodes, we need a new definition. Given a join node t, where $X_t = \{v_1, \ldots, v_p\}$, with children t_1, t_2, and given $\mathscr{M} = \{M_0, \ldots, M_p\}$, $\mathscr{M}' = \{M'_0, \ldots, M'_p\}$ and $\mathscr{M}'' = \{M''_0, \ldots, M''_p\}$, we say that $\mathscr{M}', \mathscr{M}''$ *combine into* $\mathscr{M}$ if $M'_i \cap M''_i = M_i$, for every i. For join nodes, we have:

Lemma 4.3. Let t be a join node with children t_1, t_2, and $S = \{v_1, \ldots, v_p\} \subseteq X_t$. Also, let $f : S \to [k]$ and $\mathscr{M} = \{M_0, \ldots, M_p\}$. Then,

$$c_t(S, f, \mathscr{M}) = 1 \quad \text{if and only if} \quad \text{there exist } \mathscr{M}', \mathscr{M}'' \text{ that combine into } \mathscr{M} \text{ such that } c_{t_1}(S, f, \mathscr{M}') = c_{t_2}(S, f, \mathscr{M}'') = 1.$$

Proof. Suppose $c_t(S, f, \mathscr{M}) = 1$ and then take the subgraph $H \subseteq G_t$ and a k-coloring f' of H as given in the definition of the table. Let $H_1 = G_{t_1}[V(H)]$ and f_1 be equal to f' restricted to $V(H_1)$. Since for every $u \in V(H_1) \setminus X_{t_1}$ we have $N_H[u] \subseteq V(H_1)$, Condition 1 gives us that u is a b-vertex in f_1. Furthermore, for every $v_i \in S$, by letting $M'_i = [k] \setminus f_1(N_H[v_i])$ we get that $f_1(N_{H_1}[v_i]) = [k] \setminus M'_i$; and letting $M'_0 = [k] \setminus f_1(V(H_1))$ gives us that $f_1(V(H_1)) = [k] \setminus M'_0$. Since $X_t = X_{t_1}$, we have $V(H) \cap X_t = V(H_1) \cap X_{t_1}$ and, thus, f_1 also satisfies Conditions 2-4. Therefore, by letting $\mathscr{M}' = \{M'_0, \ldots, M'_p\}$, we get that $c_{t_1}(S, f, \mathscr{M}') = 1$. We define $\mathscr{M}''$, f_2 and H_2 analogously with relation to X_{t_2} and also get $c_{t_2}(S, f, \mathscr{M}'') = 1$. It remains to show that $M'_i \cap M''_i = M_i$, for every $i \in \{0, \ldots, p\}$. For this, first consider $d \in M_i$. By Condition 2, we get that d is not a color in $f'(N_H[v_i])$, which implies that d is not a color in both $f_1(N_{H_1}[v_i])$ and $f_2(N_{H_2}[v_i])$; it follows that $M_i \subseteq M'_i \cap M''_i$ by the definition of M'_i and M''_i. Observe that the reverse argument also applies, and hence $M'_i \cap M''_i \subseteq M_i$. A similar argument can be applied when $i = 0$ by using Condition 3.

Now, suppose that there exist $\mathscr{M}', \mathscr{M}''$ that combine into $\mathscr{M}$ such that $c_{t_1}(S, f, \mathscr{M}') = c_{t_2}(S, f, \mathscr{M}'') = 1$. By definition, there exist subgraphs $H_1 \subseteq G_{t_1}$ and $H_2 \subseteq G_{t_2}$, as well as k-colorings f_1, f_2 of $V(H_1)$ and $V(H_2)$, respectively, that extend f and satisfy Conditions 1-4. Note that if a vertex $v_i \in V(H)$ is an element of $V(H_1)$ and $V(H_2)$, we have $v_i \in S$ and $f_1(v_i) = f_2(v_i) = f(v_i)$, since both f_1 and f_2 extend the same coloring f. Thus, by letting $H = G_t[V(H_1) \cup V(H_2)]$, we construct the following coloring $f' : V(H) \to [k]$:

$$f'(u) = \begin{cases} f_1(u), & \text{if } u \in V(H_1) \text{ and } u \notin V(H_2) \\ f_2(u), & \text{if } u \in V(H_2) \text{ and } u \notin V(H_1) \\ f(u), & \text{if } u \in V(H_1) \cap V(H_2) = S. \end{cases}$$

We already argued that f' extends f. We need now to prove that f' satisfies Conditions 1-4 with relation to the entry $c_t(S, f, \mathscr{M})$. Since $X_t = X_{t_1} = X_{t_2}$, if $u \in V(H) \setminus X_t$, then $u \in V(H_1) \setminus X_{t_1}$ or $u \in V(H_2) \setminus X_{t_2}$. If $u \in V(H_1) \setminus X_{t_1}$, we know that $f'(u) = f_1(u)$ and, since u is a b-vertex in f_1, it is also a b-vertex in f'. An analogous argument applies if $u \in V(H_2) \setminus X_{t_2}$, hence f' satisfies Condition 1. By construction, if $v_i \in S$, we have $f'(N_H[v_i]) = f_1(N_{H_1}[v_i]) \cup f_2(N_{H_2}[v_i])$.

Because f_1 and f_2 satisfy Condition 2 with relation to entries $c_{t_1}(S, f, \mathscr{M}')$ and $c_{t_2}(S, f, \mathscr{M}'')$, we have

$$f'(N_H[v_i]) = ([k] \setminus M_i') \cup ([k] \setminus M_i'') = [k] \setminus (M_i' \cap M_i'').$$

Because $M_i' \cap M_i'' = M_i$, f' satisfies Condition 2. Again, a similar argument can be applied to conclude Condition 3 as $f'(H) = f_1(V(H_1)) \cup f_2(V(H_2))$. Finally, since $V(H_1) \cap X_{t_1} = V(H_2) \cap X_{t_2} = S$ and $X_{t_1} = X_{t_2} = X_t$, Condition 4 follows immediately. □

Note that the above computation can be done in time $\mathscr{O}(3^{k \cdot \mathrm{tw}(G)})$. Indeed, for each $v_i \in S \subseteq X_t$ and each color $c \in [k] \setminus M_i$, we can either put c in $M_i' \setminus M_i''$, or we can put c in $M_i'' \setminus M_i'$, or we can leave c out of both sets. Now we treat forget nodes.

Lemma 4.4. Let t be a forget node with child t', and let $S = \{v_1, \ldots, v_p\} \subseteq X_t$. Also, denote by v_{p+1} the forgotten vertex (i.e., the vertex in $X_{t'} \setminus X_t$), and consider $f : S \to [k]$ and $\mathscr{M} = \{M_0, \ldots, M_p\}$. Then,

$$c_t(S, f, \mathscr{M}) = 1 \quad \text{if and only if} \quad \begin{array}{l} \text{either } c_{t'}(S, f, \mathscr{M}) = 1, \text{ or} \\ \text{there exists } c \in [k] \text{ such that} \\ c_{t'}(S \cup \{v_{p+1}\}, f_c, \mathscr{M}') = 1. \end{array}$$

where $f_c(v_i) = f(v_i)$ for every $i \in [p]$, $f_c(v_{p+1}) = c$, and $\mathscr{M}' = \{M_0, \ldots, M_p, \emptyset\}$.

Proof. Suppose $c_t(S, f, \mathscr{M}) = 1$. Then, by definition, there exists a k-coloring f' of H that extends f and satisfies Conditions 1-4. We have two cases:

- If $v_{p+1} \notin V(H)$, then v_j is not colored by f', since $S = V(H) \cap X_t$, which gives us that $S = V(H) \cap (X_t \cup \{v_{p+1}) = V(H) \cap X_{t'}$, and thus $c_{t'}(S, f, \mathscr{M}) = 1$.
- If $v_{p+1} \in V(H)$, then let $c = f'(v_{p+1})$ and note that, since $S = V(H) \cap X_t$, we have that $V(H) \cap X_{t'} = S \cup \{v_{p+1}\}$. Now, let f_c be equal to f' restricted to $S \cup \{v_{p+1}\}$ and $\mathscr{M}' = \{M_0, \ldots, M_p, \emptyset\}$. It thus remains to prove that Conditions 1-4 hold for S, f' on t' and $\mathscr{M}'$. Because f' satisfies Condition 1 for t, and since $V(H) \setminus X_{t'} \subset V(H) \setminus X_t$, we get that f' also satisfies Condition 1 for t'. Additionally, Condition 2 still holds for every $v_i \in S \cup \{v_{p+1}\}$ since $M_i' = M_i$, $i \in [p]$. As for v_{p+1}, since f' satisfies Condition 1 and $v_{p+1} \in V(H) \setminus X_t$, we know that v_{p+1} is a b-vertex in f'; hence $f'(N_{H'}[v_{p+1}]) = [k] = [k] \setminus \emptyset$ and Condition 2 also holds for v_{p+1}. Finally, Condition 3 holds since $H \subseteq G_t = G_{t'}$ and, thus, $f(V(H) \cap V(G_{t'})) = f(V(H)) = [k] \setminus M_0$, and Condition 4 holds because $V(H) \cap X_t = S$ and $v_{p+1} \in H$, we have that $V(H) \cap X_{t'} = V(H) \cap (X_t \cup \{v_{p+1}\}) = S \cup \{v_{p+1}\}$.

Now, suppose that either $c_{t'}(S, f, \mathscr{M}) = 1$ or $c_{t'}(S \cup \{v_{p+1}\}, f_c, \mathscr{M}') = 1$. Again, we split the proof in cases:

- If $c_{t'}(S, f, \mathscr{M}) = 1$, then v_{p+1} is not colored by f', where f' extends f to $H \subseteq G_{t'}$, which implies that $v_{p+1} \notin V(H)$. In that case, H is a subset of G_t such that f' satisfies Conditions 1-3 immediately. For Condition 4, note that because $v_{p+1} \notin V(H)$, we have that $V(H) \cap X_t = V(H) \cap X_{t'} = S$.
- If $c_{t'}(S \cup \{v_{p+1}\}, f_c, \mathscr{M}') = 1$, then v_{p+1} is colored by f_c and we have $f_c(v_{p+1}) = c$ and $M'_{p+1} = \emptyset$ by hypothesis. Then, let $\mathscr{M} = \{M_0, \dots, M_p\}$ and we will prove that f' satisfies Conditions 1-3 for S, t and $\mathscr{M}$. Because every vertex in $V(H) \setminus X_{t'}$ is a b-vertex in f' and, since $M'_{p+1} = \emptyset$, v_{p+1} is also a b-vertex in f' and, then, Condition 1 is satisfied for t. Conditions 2 and 3 are trivially satisfied since they hold for f' because $S \subset S \cup \{v_{p+1}\}$ and by the definition of $\mathscr{M}'$. Finally, Condition 4 holds since we know that

$$V(H) \cap X_t = (V(H) \cap X_{t'}) \setminus \{v_{p+1}\} = \big(S \cup \{v_{p+1}\}\big) \setminus \{v_{p+1}\} = S.$$

□

This can be computed in time $\mathscr{O}(k)$, because we simply need to construct $\mathscr{M}'$ (which takes constant time), and f_c for each c (which takes $\mathscr{O}(k)$ time). Before we present the next table, we need a new definition. Let t be an introduce node t, t' be its child such that $X_t = X_{t'} \cup \{v_j\}$. Given $\mathscr{M} = (M_0, \dots, M_p)$, $S = \{v_1, \dots, v_p$ and $f : S \to [k]$, we say that $\mathscr{M}' = \{M'_0, \dots, M'_{p-1}\}$ *agrees* with $\mathscr{M}$ if either $v_j \notin S$ and $\mathscr{M}' = \mathscr{M}$, or $v_j = v_p$ and $M'_i = M_i$ for every i such that $v_i \notin N(v_j)$ and $M'_i \in \{M_i, M_i \setminus \{f(v_p)\}\}$ for every i such that $v_i \in N(v_p)$ or $i = 0$. Finally, for introduce nodes, we have:

Lemma 4.5. Let t be an introduce node with $X_t = \{v_1, \dots, v_j\}$, t' be its child, with $X_t = X_{t'} \cup \{v_j\}$, and $S = \{v_1, \dots, v_p\} \subseteq X_t$. Also, let $f : S \to [k]$, $\mathscr{M} = \{M_0, \dots, M_p\}$. Then,

$c_t(S, f, \mathscr{M}) = 1$ if and only if (1) Either $v_j \notin S$ and $c_{t'}(S, f, \mathscr{M}) = 1$; or
(2) $v_j \in S$ and $M_j = [k] \setminus f(N_S[v_j])$,
there exists $\mathscr{M}'$ that agrees with $\mathscr{M}$ and
f such that $c_{t'}(S', g, \mathscr{M}') = 1$, where
g equals f restricted to $S' = X_{t'} \cap S$.

Proof. First, suppose $c_t(S, f, \mathscr{M}) = 1$ and note that, if $v_j \notin S$, we have $S \subset X_{t'}$ and, then, $c_{t'}(S, f, \mathscr{M}) = 1$ holds immediately. If $v_j \in S$, then we get $M_j = [k] \setminus f(N_S[v_j])$ by Condition 2, and the fact that $N_{G_t} \subseteq X_t$. Now, let $H' = H - \{v_j\}$ and $S' = X_{t'} \cap S = S \setminus \{v_j\}$. By definition, we have a k-coloring f^* of H that extends f and satisfies Conditions 1-3. Let f' be equal to f^* restricted to $V(H')$ and let g be equal to f restricted to S'. Because f^* extends f, we know that f' also extends g. Now let $M'_0 = [k] \setminus f'(V(H'))$ and, for each $v_i \in S'$, let $M'_i = [k] \setminus f'(N_{H'}[v_i])$; also let $\mathscr{M}' = \{M'_0, \ldots, M'_{p-1}\}$. Since $S' = S \setminus \{v_j\}$ and by the construction of H', we have that $V(H') \cap X_{t'} = S'$, i.e., Condition 4 is satisfied for t'. It follows directly from the definition of $\mathscr{M}'$ and the fact that f^* satisfies Condition 1 that $c_{t'}(S', g, \mathscr{M}') = 1$; so it remains to prove that $\mathscr{M}'$ agrees with $\mathscr{M}$ and f. Observe first that if $v_i \notin N(v_j)$, then $N_{H'}[v_i] = N_H[v_i]$, and since Condition 2 holds for f^*, we get that $M'_i = M_i$. Additionally, if $v_i \in N(v_j)$, we know that the only color that appears in $f^*(N_H[v_i])$ but might not appear in $f'(N_{H'}[v_i])$ is exactly the color of v_j, i.e., $M'_i \in \{M_i, M_i \setminus \{f(v_j)\}\}$. A similar argument holds when $i = 0$, and hence we get that $\mathscr{M}'$ agrees with f and $\mathscr{M}$, as we wanted to prove.

For the converse, again, we can suppose that $v_j \in S$, since otherwise we have $c_{t'}(S, f, \mathscr{M}) = 1$ and, hence, $c_t(S, f, \mathscr{M}) = 1$. As in the hypothesis, let $M_j = [k] \setminus f(N_S[v_j])$ and $\mathscr{M}'$ be such that $\mathscr{M}'$ agrees with $\mathscr{M}$ and $c_{t'}(S', g, \mathscr{M}') = 1$, where $S' = S \cap X_{t'}$ and g equals f restricted to S'. From $c_{t'}(S', g, \mathscr{M}') = 1$, let $H' \subseteq G_{t'}$ together with f^* a k-coloring of $V(H')$ that extends g and satisfies Conditions 1-3. We prove that f' obtained from f^* by coloring v_j with $f(v_j)$ satisfies Conditions 1-4 with respect to t and $\mathscr{M}$; it thus follows that $c_t(S, f, \mathscr{M}) = 1$. First note that Condition 1 clearly holds since $V(H') \setminus X_{t'} = V(H) \setminus X_t$ and f' is equal to f^* when restricted to $V(H') \setminus X_{t'}$. Now let $i \in \{0, \ldots, p\}$. If $i = j$, then $N_H[v_j] = N_S[v_j]$ and Condition 2 follows since $M_j = [k] \setminus f(N_S[v_j])$ by hypothesis. If $i \notin \{0, j\}$, then Condition 2 on f^* tells us that $f^*(N_{H'}[v_i]) = [k] \setminus M'_i$. If $v_i \notin N(v_j)$, then $N_H[v_i] = N_{H'}[v_i]$ and, since $\mathscr{M}'$ agrees with $\mathscr{M}$ and f, we have $M'_i = M_i$ and Condition 2 also holds for f' on H and M'_i. And if $v_i \in N(v_j)$, then $f'(N_H[v_i]) = f^*(N_{H'}[v_i]) \cup \{f(v_j)\}$. Since f^* satisfies Condition 2, we get that $f'(N_H[v_i]) = [k] \setminus (M'_i \cup \{f(v_j)\}) = [k] \setminus M_i$, as we wanted. Finally, if $i = 0$, then Condition 3 on f^* and the fact that $\mathscr{M}'$ agrees with $\mathscr{M}$ and f give us that $f'(V(H)) = f^*(V(H')) \cup \{f(v_j)\} = ([k] \setminus M'_0) \cup \{f(v_j)\} = [k] \setminus (M'_0 \cup \{f(v_j)\}) = [k] \setminus M_0$. □

By the above lemma, in order to compute an entry $c_t(S, f, \mathscr{M})$, we need to investigate all the possibles $\mathscr{M}'$ that agree with $\mathscr{M}$. Since there are 2 choices for M'_i for every i, this gives us a total of $\mathscr{O}^*(2^{\text{tw}})$ possible choices for $\mathscr{M}'$.

The lemmas above define an algorithm running in FPT time when parameterized by the treewidth of G, which is summarized by the following theorem:

Theorem 4.4. Deciding whether a graph G has a subfall k-coloring can be done in FPT time when parameterized by $\text{tw}(G)$, with complexity $\mathscr{O}^*((6^{\text{tw}(G)} \cdot 2\,\text{tw}(G))^{\text{tw}(G)})$.

Proof. By (Bodlaender, 1993), we know that construct a tree decomposition of width tw can be done in FPT time when parameterized by the treewidth. Furthermore, given a tree decomposition of width w, one can compute a nice tree decomposition of width at most w in FPT time as well (see (Cygan *et al.*, 2015)).

Now, given a graph G, we first compute a tree decomposition of width $\text{tw}(G)$ and then we construct a nice tree decomposition $(T, \mathscr{X})$ of width $\text{tw}(G)$. Then, for each node starting from the leaves, compute the table's values. By the previous lemmas, the time to compute the table $c_t(S, f, \mathscr{M})$ at each node is dominated by the time to compute the table $c_t(S, f, \mathscr{M})$ for when t is a join node, which is $\mathscr{O}^*(3^{k \cdot \text{tw}(G)})$, as said. The algorithm returns the value $c_r(\emptyset, \emptyset, \mathscr{M})$, where r is the root of the nice tree decomposition. Correctness follows from Lemmas 4.1 to 4.5, where G has subfall k-coloring if $c_r(\emptyset, \emptyset, \mathscr{M}) = 1$ and G is not subfall k-colorable otherwise. By letting x be the number of subsets S, y be the number of k colorings of a subset S and z be the number of sets $\mathscr{M}$, we have that the complexity of the algorithm is equal to $\mathscr{O}^*(x \cdot y \cdot z \cdot 3^{k \cdot \text{tw}(G)}) = \mathscr{O}^*(2^{\text{tw}(G)} \cdot k^{\text{tw}(G)} \cdot 2^{k \cdot \text{tw}(G)} \cdot 3^{k \cdot \text{tw}(G)})$. Since $k \leq \text{col}(G) + 1 \leq \text{tw}(G) + 1$, we have that the complexity of the algorithm is $\mathscr{O}^*((6^{\text{tw}(G)} \cdot 2\,\text{tw}(G))^{\text{tw}(G)})$.

Because the complexity of computing the tables for the join nodes is worse than the complexity of computing a nice tree decomposition of a graph G, the latter complexity is already included in $\mathscr{O}^*((6^{\text{tw}(G)} \cdot 2\,\text{tw}(G))^{\text{tw}(G)})$. □

Due to the similarity between subfall coloring with fall coloring and b-coloring, it is natural to think of how to adapt the table to a node t of the decomposition tree for each of these colorings. Indeed, such an modification can be done, which gives an FPT algorithm with parameter $\text{tw}(G)$ for the fall coloring and an FPT algorithm with parameter $\text{tw}(G) + k$ for b-coloring. We explain how to make such changes in the two following subsections.

4.2.1 Fall Coloring

In this subsection, we show how to convert the FPT algorithm for deciding whether a graph G has a subfall k-coloring shown above to an algorithm that decides if G has a fall k-coloring in FPT time. Firstly, for a node $t \in V(T)$, where $X_t = \{v_1, v_2, \ldots, v_p\}$, we define the table related to t as follows: for each proper k-coloring f of X_t, and for each $\mathscr{M} = \{M_0, \ldots, M_p\}$, with $M_i \subseteq [k]$ for every i, we say that $c_t(f, \mathscr{M}) = 1$ if and only if there exists a coloring f' that extends f to G_t and satisfies:

1. Every $u \in V(G_t) \setminus X_t$ is a b-vertex in f';
2. For every $v_i \in X_t$, we have $f'(N_{G_t}[v_i]) = [k] \setminus M_i$; and
3. $f'(V(G_t)) = [k] \setminus M_0$.

Below, we present the analogous of Lemmas 4.1-4.5. We refrain from presenting the formal proofs as they are quite similar to the previous ones.

Lemma 4.6. Let r be the root of a nice tree decomposition of G, with $\mathscr{M}_\emptyset = \{M_0\}$ where $M_0 = \emptyset$. Then G has a fall k-coloring if and only if $c_r(\emptyset, \mathscr{M}_\emptyset) = 1$.

Furthermore, the four following lemmas show the adaptation of the tables related to each type of node:

Lemma 4.7. Let t be a leaf node (and hence $X_t = \{v_1\}$ for some $v_1 \in V(G)$). Also, let $f : \{v_1\} \to [k]$, and $\mathscr{M} = \{M_0, M_1\}$. Then,

$$c_t(f, \mathscr{M}) = 1 \quad \text{if and only if} \quad M_0 = [k] \text{ and } M_1 = [k] \setminus \{f(v_1)\}.$$

Lemma 4.8. Let t be a join node with children t_1, t_2, and $X_t = \{v_1, \ldots, v_p\}$. Also, let $f : X_t \to [k]$ and $\mathscr{M} = \{M_0, \ldots, M_p\}$. Then,

$$c_t(f, \mathscr{M}) = 1 \quad \text{if and only if} \quad \begin{array}{l} \text{there exist } \mathscr{M}', \mathscr{M}'' \text{ that combine into } \mathscr{M} \\ \text{such that } c_{t_1}(f, \mathscr{M}') = c_{t_2}(f, \mathscr{M}'') = 1. \end{array}$$

Lemma 4.9. Let t be a forget node with child t', and let $X_t = \{v_1, \ldots, v_p\}$. Also, denote by v_{p+1} the forgotten vertex (i.e., the vertex in $X_{t'} \setminus X_t$), and consider $f : X_t \to [k]$ and $\mathscr{M} = \{M_0, \ldots, M_p\}$. Then,

$$c_t(f, \mathscr{M}) = 1 \quad \text{if and only if} \quad \begin{array}{l} \text{there exists } c \in [k] \text{ such that } c_{t'}(f_c, \mathscr{M}') = 1, \text{ where} \\ M'_{p+1} = \emptyset, f_c(v_{p+1}) = c, \text{ and } M'_i = M_i \text{ for all} \\ i \in \{1, \ldots, p\}, \text{ and } f_c(v_i) = f(v_i), \text{ for all } i \in [p]. \end{array}$$

Before we present the table for an introduce node, we need to modify a previous definition. Let t be an introduce node such that $X_t = \{v_1, \ldots, v_p\}$ and t' be its child such that $X_t = X_{t'} \cup \{v_p\}$. Given $\mathscr{M} = (M_0, \ldots, M_p)$ and $f : X_t \to [k]$, we say that We say that $\mathscr{M}' = \{M'_0, \ldots, M'_{p-1}\}$ *agrees* with $\mathscr{M}$ and f if $M'_i = M_i$ for every $v_i \notin N(v_p)$, and $M'_i \in \{M_i, M_i \setminus \{f(v_p)\}\}$ for every i such that $v_i \in N(v_p)$ or $i = 0$.

Lemma 4.10. Let t be an introduce node with $X_t = \{v_1,\dots,v_p\}$, t' be its child, with $X_t = X_{t'} \cup \{v_p\}$. Also, let $f : X_t \to [k]$ and $\mathscr{M} = \{M_0,\dots,M_p\}$. Then,

$$c_t(f,\mathscr{M}) = 1 \quad \text{if and only if} \quad \text{there exists } \mathscr{M}' \text{ that agrees with } \mathscr{M} \text{ and } c_{t'}(g,\mathscr{M}') = 1, \text{ where } g \text{ equals } f \text{ restricted to } X_{t'}.$$

Theorem 4.5. Deciding whether a graph G has a fall k-coloring can be done in FPT time when parameterized by $\mathrm{tw}(G)$, with complexity $\mathscr{O}^*((6^{\mathrm{tw}(G)} \cdot \mathrm{tw}^{\mathrm{tw}(G)})$.

4.2.2 b-Coloring

Now, we present an algorithm that decides whether a graph G has a b-coloring using exactly k colors. This algorithm runs in FPT time when parameterized by $\mathrm{tw}(G) + k$. Note that the parameter is not the same as it is for the previous algorithms, since we use only $\mathrm{tw}(G)$ for the previous ones. Furthermore, since it is proved in (Jaffke *et al.*, 2023) that the problem of deciding whether a graph G admits a b-coloring using exactly k colors is W[1]-hard when parameterized by $\mathrm{tw}(G)$. For a node $t \in V(T)$, where $X_t = \{v_1, v_2,\dots,v_p\}$, we define the table related to t as follows: for each proper k-coloring f of X_t, and for each $\mathscr{M} = \{M_0,M_1,M_2,\dots,M_p\}$, with $M_i \subseteq [k]$ for every i, we say that $c_t(f,\mathscr{M}) = 1$ if and only if there exists k-coloring f' of G_t that extends f and satisfies:

1. For every $c \in [k] \setminus M_0$, there exists $u \in V(G_t) \setminus X_t$ such that u is a b-vertex of color c in f'; and
2. For every $v_i \in X_t$, we have $f'(N_{G_t}[v_i]) = [k] \setminus M_i$.

Note that now, even with the same definition for the sets in $\mathscr{M}$, the set M_0 has a slightly different meaning, since instead of tracking the colors that did not appear in the graph, it tracks the colors that do not have a b-vertex. Again, we present the appropriate lemmas and refrain from proving them, as the arguments are very similar to the ones in Section 4.2.

Lemma 4.11. Let r be the root of a nice tree decomposition of G, with $\mathscr{M}_\emptyset = \{M_0\}$ where $M_0 = \emptyset$. Then G has a b-coloring using k colors if and only if $c_r(\emptyset,\mathscr{M}_\emptyset) = 1$.

Lemma 4.12. Let t be a leaf node (and hence $X_t = \{v_1\}$ for some $v_1 \in V(G)$). Also, let $f : \{v_1\} \to [k]$, and $\mathscr{M} = \{M_0,M_1\}$. Then,

$$c_t(f,\mathscr{M}) = 1 \quad \text{if and only if} \quad M_0 = [k] \text{ and } M_1 = [k] \setminus \{f(v_1)\}.$$

Lemma 4.13. Let t be a join node with children t_1, t_2, and $X_t = \{v_1, \ldots, v_p\}$. Also, let $f : X_t \to [k]$ and $\mathscr{M} = \{M_0, \ldots, M_p\}$. Then,

$$c_t(f, \mathscr{M}) = 1 \quad \text{if and only if} \quad \begin{array}{l} \text{there exist } \mathscr{M}', \mathscr{M}'' \text{ that combine into } \mathscr{M} \\ \text{such that } c_{t_1}(f, \mathscr{M}') = c_{t_2}(f, \mathscr{M}'') = 1. \end{array}$$

The first item below covers the possibility of v_{p+1} not being a b-vertex, while the second item covers the case when v_{p+1} is a b-vertex of its color.

Lemma 4.14. Let t be a forget node with $X_t = \{v_1, \ldots, v_p\}$, and t' be its child, with $X_t = X_{t'} \setminus \{v_{p+1}\}$. Also, let $f : X_t \to [k]$ and $\mathscr{M} = \{M_0, \ldots, M_p\}$. Then, $c_t(f, \mathscr{M}) = 1$ if and only if one of the following holds:

- There exists $c \in [k]$ and there exists $M'_{p+1} \subseteq [k]$ such that $M'_{p+1} \neq \emptyset$ and $c_{t'}(f_c, \mathscr{M}') = 1$, where $\mathscr{M}' = \{M'_0, \ldots, M'_{p+1}\}$. $M'_i = M_i$ for each $i \in \{0, \ldots, p\}$, $f_c(u) = f(u)$ if $u \in X_t$ and $f_c(v_p) = c$;
- There exists $c \in [k] \setminus M_0$ such that $c_{t'}(f_c, \mathscr{M}') = 1$, where $M'_i = M_i$ for every $i \in [p]$, $M'_0 = M_0 \setminus \{c\}$, $M'_{p+1} = \emptyset$, $\mathscr{M}' = \{M'_0, \ldots, M'_{p+1}\}$, and $f_c(v_i) = f(v_i)$, for all $i \in [p]$.

For the following lemma, we use the same definition as the one for Lemma 4.10.

Lemma 4.15. Let t be an introduce node with $X_t = \{v_1, \ldots, v_p\}$, t' be its child, with $X_t = X_{t'} \cup \{v_p\}$. Also, let $f : X_t \to [k]$, $\mathscr{M} = \{M_0, \ldots, M_p\}$. Then,

$$c_t(f, \mathscr{M}) = 1 \quad \text{if and only if} \quad \begin{array}{l} \text{there exists } \mathscr{M}' = \{M'_0, \ldots, M'_{p-1}\} \text{ that agrees with } \mathscr{M} \text{ and} \\ \text{is such that } c_{t'}(f, \mathscr{M}') = 1, \text{ where } M'_0 = M_0 \text{ if } v_p \text{ is not} \\ \text{a b-vertex of its color and } M'_0 = M_0 \setminus f(v_p) \text{ otherwise.} \end{array}$$

Theorem 4.6. Deciding whether a graph G has a b-coloring using k colors can be done in FPT time when parameterized by $\text{tw}(G) + k$, with complexity $\mathscr{O}^*((6^k \cdot k^{\text{tw}(G)})$.

4.3 Continuity, relation to other parameters and other properties

In this section, we show some properties of subfall coloring. Among them, one of the most important differences between fall coloring and subfall coloring: the continuity of the latter. In fact, there are graphs that do not even admit a fall coloring, which does not happen for subfall colorings. Moreover, there are graphs, for positive integers k, m and n, with $k < m < n$, that admit fall k-coloring and fall n-coloring, but do not admit fall m-coloring, which shows the discontinuity of the fall spectrum of graphs. See Figure 21 for a graph obtained from the graph shown in Figure 3 by adding a new vertex adjacent to all other vertices. This modification changes the fall spectrum of the graph in a way that if the graph shown in Figure 3 admits a fall k-coloring, G admits a fall $(k+1)$-coloring, because the new vertex is adjacent to each other vertex and thus will always be a b-vertex.

Figure 21 – Graph G such that $\{3,5\} \subseteq \text{Fall}(G)$, but $4 \notin \text{Fall}(G)$.

Source: prepared by the author

Proposition 4.2. Let G be a graph. Then G has a subfall k-coloring, for every $k \in [\psi_{fs}(G)]$.

Proof. Since any vertex with color 1 define a subfall 1-coloring, we have $1 \in [\psi_{fs}(G)]$. So, letting $k = \psi_{fs}(G)$, consider an induced subgraph $H \subseteq G$ with a fall k-coloring f of H. Observe that, to obtain a subfall $(k-1)$-coloring of G, we can just take the induced subgraph $H' \subseteq H$ whose set of vertices equals $V(H) - f^{-1}(k)$ and the coloring $f' : V(H') \to \{1,\ldots,k-1\}$ such that f' equals f restricted to $V(H')$. Indeed, observe that f' is a fall coloring of $H' \subseteq G$. We can repeat the steps inductively to obtain a subfall j-coloring, for every $j \in [k-1]$. □

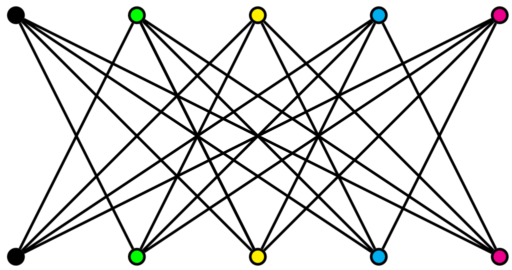
A natural concept of graph theory is to investigate possible relations between graph parameters. In (Dunbar *et al.*, 2000), the authors established that $\psi_f(G) \leq b(G)$, that $\psi_f(G) \leq \Gamma(G)$ and that $\psi_f(G) \leq \psi_{fs}(G)$. Moreover, in (Zaker, 2020), the author proved that $b(G)$ and $\Gamma(G)$ are not related. Below, we analyze the relation between each of the aforementioned parameters and $\psi_{fs}(G)$, as well as between $\chi(G)$ and $\psi_{fs}(G)$.

Proposition 4.3. The following statements are true:

a. There exists G_1 such that $\psi_{fs}(G_1) < \chi(G_1)$. Additionally, for every positive integer k, there exists G_2 such that $\psi_{fs}(G_2) - \chi(G_2) = k$;
b. For each positive integer k, there exist graphs G_1 and G_2 such that: $b(G_1) - \psi_{fs}(G_1) = \psi_{fs}(G_2) - b(G_2) = k$;
c. For every graph G, we have $\psi_{fs}(G) \leq \Gamma(G)$.

Proof. a. Let G_1 be the cycle on five vertices. We know that $\chi(G_1) = 3$, while $\psi_{fs}(G_1) = 2$ by Proposition 3.1 and Theorem 4.2, giving us $\chi(G_1) > \psi_{fs}(G_1)$. For the other inequality, let G_2 be obtained from the complete bipartite graph $K_{k+2,k+2}$ by removing a perfect matching. Then, we have $\chi(G_2) = 2$. Also, by giving colors 1 through $k+2$ to the vertices within the same part, making sure to give the same color to the endpoints of the removed perfect matching, we get a fall coloring of G_2 with $k+2$ colors, i.e., $\psi_{fs}(G_2) \geq k+2$. But since $\Delta(G_2) = k+1$ and $\psi_{fs}(G) \leq \Delta(G)+1$ for every G, we also get $\psi_{fs}(G_2) \leq k+2$, thus giving $\psi_{fs}(G_2) - \chi(G_2) = k$.

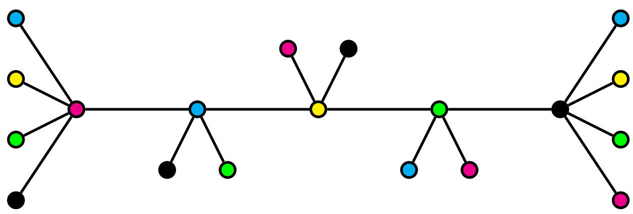
Figure 22 – Subfall 5-coloring of the graph G_2 constructed above, with $k = 3$.



Source: prepared by the author

b. First, let G'_1 be obtained from the path $(v_1, v_2, \ldots, v_k, v_{k+1}, v_{k+2})$ by adding k pendant vertices to the vertices v_1 and v_{k+2} and $k-1$ pendant vertices to each vertex v_i with $i \in \{2, 3, \ldots, k+1\}$ (in total we add $k \cdot (k+1)$ new vertices). By (Dunbar *et al.*, 2000), we know that every tree T is such that $\psi_{fs}(T) = 2$, giving us $\psi_{fs}(G'_1) = 2$. We can get a b-coloring $f : V(G) \to [k+2]$ just by coloring the vertices v_i with the color i, $i \in [k+2]$, then coloring the pendant vertices of each v_i with the colors $[k+2] \setminus f(N(v_i))$, with every two pendant vertices of v_i colored with distinct colors. Again, since $\Delta(G'_1) = k+1$ and $b(G) \leq \Delta(G)+1$ for every G, we get that $b(G_1) = k+2$, and hence $b(G'_1) - \psi_{fs}(G'_1) = k$.

Figure 23 – b-coloring using 5 colors of the graph G'_1 constructed above, with $k = 3$.

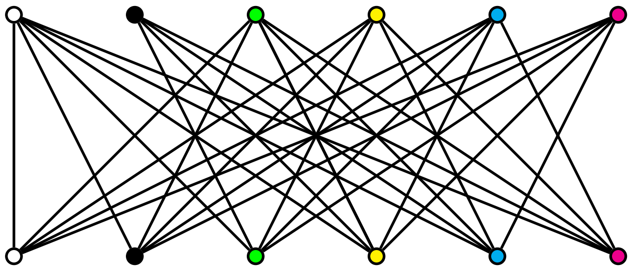


Source: prepared by the author

Now, let G'_2 be the complete bipartite graph $K_{k+3,k+3}$ minus a matching of size $k+2$. Since G'_2 is bipartite, any proper 2-coloring of G_2 is also a b-coloring. In order to show that

$b(G_2') = 2$, we will show that no proper coloring f of G_2' using more than two colors can be a b-coloring. Let $u, v \in V(G_2')$ be the vertices with $k+3$ neighbors and $w \in V(G_2')$ a neighbor of v, $w \neq u$. Since f must be proper, we have that $f(x) \neq f(u)$ for every $x \in N(u)$. Since $N(w) \subseteq N(u)$, if $f(w) \neq f(u)$, w cannot be a b-vertex, since it would not be adjacent to any vertex of color $f(u)$. For the neighbors of u it is analogous. Then, G_2' cannot have more than one b-vertex when f uses more than 2 colors; it follows that $b(G_2') = 2$. On the other hand, the graph G_2 from the previous item is a subgraph of G_2', thus giving $\psi_{fs}(G_2') = k+2$.

Figure 24 – Subfall 5-coloring of the graph G_2' constructed above, with $k = 3$, where the subgraph is induced by the colored vertices.



Source: prepared by the author

c. Let G be any graph and H a subgraph of G such that $\psi_f(H) = \psi_{fs}(G)$. Note that every b-vertex is also a Grundy vertex, immediately implying $\psi_f(H) \leq \Gamma(H)$. But since any Grundy k-coloring of H can be extended to a Grundy coloring of G with at least k colors (it suffices to greedily color the uncolored vertices), we get that $\Gamma(H) \leq \Gamma(G)$, giving us the inequality $\psi_{fs}(G) = \psi_f(H) \leq \Gamma(G)$, as wanted. □

4.4 Subfall achromatic index

In this section, we introduce the edge version of subfall k-coloring and establish some bounds and results for it for general graphs and some subclasses of graphs. Firstly, we denote the *fall achromatic index* and *subfall achromatic index* of a graph G by $\psi_f'(G) = \psi_f(L(G))$ and $\psi_{fs}'(G) = \psi_{fs}(L(G))$, respectively, where $L(G)$ is the line graph of G. It is well known that every planar graph has minimum degree at most 5 and every outerplanar graph has minimum degree at most 2 (see (West, 2001)). These two observations combined imply that every planar graph G is such that $\text{col}(G) \leq 5$ and every outerplanar graph G is such that $\text{col}(G) \leq 2$.

Proposition 2.1, which states that $\psi_f(G) \leq \delta(G) + 1$, immediately gives us that the subfall achromatic number of G cannot be higher than its coloring number plus one, as stated in the following proposition.

Proposition 4.4. For every graph G, we have that $\psi_{fs}(G) \leq \mathrm{col}(G)+1$.

Proof. For any subgraph $H \subseteq G$, we have that $\psi_f(H) \leq \delta(H)+1$. By definition, $\mathrm{col}(G)$ equals the maximum $\delta(H)$ among all subgraphs $H \subseteq G$, thus $\psi_{fs}(G) \leq \mathrm{col}(G)+1$. □

Therefore we get the following as a corollary of Proposition 4.4:

Corollary 4.2. For every planar graph G, $\psi_{fs}(G) \leq 6$. Furthermore, if G is also outerplanar, then $\psi_{fs}(G) \leq 3$.

Note that $\delta(L(G)) \leq \delta(G)+\Delta(G)-2$ holds, since any edge incident to a vertex of degree $\delta(G)$ can have at most $\Delta(G)-1$ adjacent edges that are incident at its other endpoint. With this fact, together with Proposition 2.1, we obtain the following inequality for the fall achromatic index of G:

Proposition 4.5. For every graph G, $\psi'_f(G) \leq \delta(G)+\Delta(G)-1$.

Which implies the following proposition:

Proposition 4.6. For every graph G, we have:

$$\Delta(G) \leq \psi'_{fs}(G) \leq \max_{H\subseteq G}(\delta(H)+\Delta(H)-1) \leq \Delta(G)+\mathrm{col}(G)-1.$$

Proof. Note that $L(G)$ always has a clique of size $\Delta(G)$. Indeed, let $v \in V(G)$ be a vertex such that $d_G(v) = \Delta(G)$. Since every two distinct edges e_1, e_2 such that e_1 and e_2 share an endpoint are adjacent in $L(G)$ and v is an endpoint of exactly $\Delta(G)$ edges, all the edges incident to v are pairwise adjacent, which forms a clique of size $\Delta(G)$ in $L(G)$. Thus, $\Delta(G) \leq \psi'_{fs}(G)$.

For the other inequality, by definition, we have that $\psi_{fs}(G) = \max_{H\subseteq G} \psi_f(H)$. Since $\psi'_f(H) \leq \delta(H)+\Delta(H)-1$ holds for every graph H. Then, we have that $\psi'_{fs}(G) = \max_{H\subseteq G} \psi'_f(H) \leq \max_{H\subseteq G}(\delta(H)+\Delta(H)-1)$. Finally, since $\max_{H\subseteq G}\Delta(H) = \Delta(G)$ and $\max_{H\subseteq G}\delta(H) = \mathrm{col}(G)$, we have that $\psi'_{fs}(G) \leq \Delta(G)+\mathrm{col}(G)-1$, as we wanted to prove. □

Again, since $\mathrm{col}(G) \leq 5$ if G is planar, and $\mathrm{col}(G) \leq 2$ if G is outerplanar, combined with the above proposition, we obtain a Vizing-like theorem for the subfall achromatic index of planar and outerplanar graphs, as stated below.

Corollary 4.3. For every planar graph G, $\Delta(G) \leq \psi'_{fs}(G) \leq \Delta(G)+4$. Furthermore, if G is also outerplanar, then $\Delta(G) \leq \psi'_{fs}(G) \leq \Delta(G)+1$.

5 CONCLUDING REMARKS

Subfall colorings were introduced almost 25 years ago, in (Dunbar *et al.*, 2000), being a variation of fall colorings. However, there was no other work on subfall colorings in the literature, while there are some papers on fall colorings. In this sense, in Chapter 3 we present a literature review of the work done on fall colorings, as it is the coloring most related to subfall coloring. In Chapter 4, we present our contributions to the state of art in the study of subfall colorings. Results found in Section 4.1 and Section 4.3 were presented in (Andrade; Silva, 2021) at *VI Encontro de Teoria da Computação* (ETC) in 2021. Results found in Section 4.2 were presented in (Andrade; Silva, 2022) at *VII Encontro de Teoria da Computação* (ETC) in 2022.

In Section 4.1, we answer the sixth questions raised in (Dunbar *et al.*, 2000), settling the NP-completeness of deciding whether a graph G has a subfall coloring. In Section 4.2, we give an explicit algorithm for deciding whether a graph G has a subfall k-coloring that runs in FPT with $\mathrm{tw}(G)$ as parameter. We also show explicitly how to adapt such algorithm to decide whether a graph G has a fall k-coloring with parameter $\mathrm{tw}(G)$ and how to adapt the algorithm to decide whether a graph G has a b-coloring using k colors with parameter $k+\mathrm{tw}(G)$. In Section 4.3, we answer the seventh question raised in (Dunbar *et al.*, 2000) in the negative, proving that the parameters $\psi_{fs}(G)$ and $b(G)$ are incomparable. We also give relations between ψ_{fs} and other related parameters, as well as proving basic properties of the parameter $\psi_{fs}(G)$. Finally, in Section 4.4, we define the subfall chromatic index and prove upper and lower bounds for general graphs, planar and outerplanar graphs. In the table below, we summarize the state of the art of the results found on the complexity of computing the fall achromatic number in some graph classes, while comparing it to the results on the complexity of computing the subfall achromatic number established in this dissertation.

A natural recommendation for future research is to study the behavior of both subfall chromatic number and the existence of subfall k-colorings with respect to graph products, specially Cartesian, Categorical and Lexicographic products, thus making a comparison with the results obtained until then for fall coloring in these same graph products, listed in Chapter 3. Moreover, as Table 1 shows, the complexity of deciding the subfall achromatic number for bipartite graphs, co-bipartite graphs and planar graphs would be a natural future research. Another natural recommendation is to establish the complexity of deciding whether a graph G admits a subfall k-coloring when parameterized by other parameters than the treewidth of the input graph.

Table 1 – Comparison on the complexity results of computing the fall and subfall achromatic numbers

Class	ψ_f	ψ_{fs}
General graphs	NP-complete (Dunbar *et al.*, 2000) even if $k = 3$	NP-complete when $k \geq 4$ by Theorem 4.1
Bipartite	NP-complete (Laskar; Lyle, 2009)	?
Chordal	NP-complete (Silva, 2019)	P by Theorem 4.3
Co-bipartite	P (Shaebani, 2009)	?
Cograph	P (Silva, 2019)	P by Theorem 4.3
k-regular	NP-complete (Lauri; Mitillos, 2020)	NP-complete when $k \geq 4$ by Theorem 4.1 ? when $k = 3$
Planar	NP-complete (Lauri; Mitillos, 2020)	?
General graphs	FPT param. by tw (Telle; Proskurowski, 1997) W[1]-hard param. by mw (Jaffke *et al.*, 2023)	FPT param. by tw by Theorem 4.4 ? param. by mw

Source: prepared by the author.

Note that, in Proposition 4.3, we show the existence of a graph G such that $\chi(G) > \psi_{fs}(G)$ and that the subfall-achromatic number can be arbitrarily larger than the chromatic number, even if the graph is not fall colorable. This naturally leads to another open question: given integer k, does there exist any graph G such that $\chi(G) - \psi_{fs}(G) = k$? In other words, does there there exist of a graph G such that the chromatic number is arbitrarily larger than the subfall chromatic number? By Theorem 4.2, such a graph must be C_{3n}-free for every n. This seems to be very challenging since C_3-free graphs with arbitrarily large chromatic numbers were not know until 1954, when the first construction was given by Tutte, under pseudonym (Descartes, 1954), and constructions forcing high girth appeared only around 10 years after Paul Erdős presented his groundbreaking probabilistic proof in (Erdős, 1959). Furthermore, the iterated Mycielskian of a C_3-free graph, which is a well-known construction of graphs with large chromatic number preserving the property of being a C_3-free graph, is not C_{3n}-free for every n. In this sense, even studying the behavior of subfall colorings in iterated Mycielskian is interesting, since they are not fall colorable by Theorem 3.3.

Moreover, recall that deciding whether a graph has a subfall 2-coloring can be done in linear time, since it is enough that the graph has at least one edge. Note that Theorem 4.1 shows that deciding whether a graph has a subfall k-coloring is NP-complete for each $k \geq 4$. We could not settle the complexity of deciding whether a graph G has a subfall 3-coloring, but we characterize all such graphs as the graphs that have a C_{3n} as induced subgraph for any integer n, in Theorem 4.2. In order to decide the existence of subfall 3-colorings for graphs, this characterization shows that it is enough to search for induced cycles of the form C_{3n} in the graph.

We mention two similar works in the literature, (Chudnovsky *et al.*, 2005; Chudnovsky *et al.*, 2020), where the authors show algorithms that run in polynomial time for detecting induced cycles of even and odd length, respectively. Finally, another natural direction is to investigate the complexity of deciding if $\psi'_{fs}(G) = \Delta(G)$ when G is a planar or outerplanar graph.

BIBLIOGRAPHY

ANDRADE, D.; SILVA, A. On the Complexity of Subfall Coloring of Graphs. In: **Anais do VI Encontro de Teoria da Computação**. Porto Alegre, RS, Brasil: SBC, 2021. p. 70–73. ISSN 2595-6116.

ANDRADE, D.; SILVA, A. (Sub)Fall Coloring and B-Coloring Parameterized by Treewidth. In: **Anais do VII Encontro de Teoria da Computação**. Porto Alegre, RS, Brasil: SBC, 2022. p. 69–72. ISSN 2595-6116.

APPEL, K.; HAKEN, W. Every planar map is four colorable. Part I: Discharging. **Illinois Journal of Mathematics**, Duke University Press, [*S.l.*], v. 21, n. 3, p. 429 – 490, 1977.

APPEL, K.; HAKEN, W. **Every planar map is four colorable**. [*S.l.*]: American Mathematical Society, 1989. v. 98.

BALAKRISHNAN, R.; KAVASKAR, T. Fall coloring of graphs I. **Discussiones Mathematicae Graph Theory**, [*S.l.*], v. 30, n. 3, p. 385–391, 2010. ISSN 2083-5892.

BALAKRISHNAN, R.; KAVASKAR, T. Color Chain of a Graph. **Graphs and Combinatorics**, Springer, [*S.l.*], v. 27, p. 487–493, 2011.

BALAKRISHNAN, R.; KAVASKAR, T.; SO, W. The energy of the Mycielskian of a regular graph. **The Australasian Journal of Combinatorics**, Citeseer, [*S.l.*], v. 52, p. 163–172, 2012.

BARTH, D.; COHEN, J.; FAIK, T. **Non approximability and non-continuity of the fall coloring graph problem**. [*S.l.*]: [*S. n.*], 2009.

BODLAENDER, H. L. Dynamic programming on graphs with bounded treewidth. In: . Berlin, Heidelberg: Springer Berlin Heidelberg, 1988. p. 105–118. ISBN 978-3-540-39291-0.

BODLAENDER, H. L. A linear time algorithm for finding tree-decompositions of small treewidth. In: **Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing**. New York, NY, USA: Association for Computing Machinery, 1993. (STOC ’93), p. 226–234. ISBN 0897915917.

BROWN, J. R. Chromatic Scheduling and the Chromatic Number Problem. **Management Science**, INFORMS, [*S.l.*], v. 19, n. 4, p. 456–463, 1972. ISSN 00251909, 15265501.

CAMPOS, V. A.; GOMES, G. C.; IBIAPINA, A.; LOPES, R.; SAU, I.; SILVA, A. Coloring Problems on Bipartite Graphs of Small Diameter. **The Electronic Journal of Combinatorics**, The Electronic Journal of Combinatorics, [*S.l.*], v. 28, n. 2, p. P2–14, 2021.

CHUDNOVSKY, M.; KAWARABAYASHI, K.-i.; SEYMOUR, P. Detecting even holes. **Journal of Graph Theory**, Wiley Online Library, [*S.l.*], v. 48, n. 2, p. 85–111, 2005.

CHUDNOVSKY, M.; SCOTT, A.; SEYMOUR, P.; SPIRKL, S. Detecting an odd hole. **Journal of the ACM (JACM)**, Association for Computing Machinery, [*S.l.*], v. 67, n. 1, p. 1–12, 2020. ISSN 0004-5411.

COCKAYNE, E.; HEDETNIEMI, S. Disjoint independent dominating sets in graphs. **Discrete Mathematics**, Elsevier, [*S.l.*], v. 15, n. 3, p. 213–222, 1976.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Introduction to Algorithms, Third Edition**. 3rd. ed. [*S.l.*]: The MIT Press, 2009. ISBN 0262033844.

COURNIER, A.; HABIB, M. A new linear algorithm for Modular Decomposition. In: TISON, S. (Ed.). **Trees in Algebra and Programming — CAAP'94**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994. p. 68–84. ISBN 978-3-540-48373-1.

CYGAN, M.; FOMIN, F. V.; KOWALIK, L.; LOKSHTANOV, D.; MARX, D.; PILIPCZUK, M.; PILIPCZUK, M.; SAURABH, S. **Parameterized Algorithms**. 1st. ed. [*S.l.*]: Springer Publishing Company, Incorporated, 2015. ISBN 3319212745.

DESCARTES, B. Solution to advanced problem No. 4526. **American Mathematical Monthly**, [*S.l.*], v. 61, p. 352, 1954.

DUNBAR, J.; HEDETNIEMI, S.; HEDETNIEMI, S.; JACOBS, D.; KNISELY, J.; LASKAR, R.; RALL, D. Fall Colorings of Graphs. **Journal of Combinatorial Mathematics and Combinatorial Computing**, [*S.l.*], v. 33, p. 257–273, 2000.

ERDŐS, P. Graph Theory and Probability. **Canadian Journal of Mathematics**, Cambridge University Press, [*S.l.*], v. 11, p. 34–38, 1959.

ERDŐS, P.; HEDETNIEMI, S. T.; LASKAR, R. C.; PRINS, G. C. On the equality of the partial grundy and upper ochromatic numbers of graphs. **Discrete Mathematics**, Elsevier, [*S.l.*], v. 272, n. 1, p. 53–64, 2003. ISSN 0012-365X. In Honor of Frank Harary.

ERDŐS, P.; RUBIN, A. L.; TAYLOR, H. Choosability in graphs. **Congressus numerantium**, [*S.l.*], v. 26, n. 4, p. 125–157, 1979.

EULER, L. Solutio problematis ad geometriam situs pertinentis. **Commentarii academiae scientiarum Petropolitanae**, [*S.l.*], p. 128–140, 1741.

GIAKOUMAKIS, V.; VANHERPE, J.-M. On extended P_4-reducible and extended P_4-sparse graphs. **Theoretical Computer Science**, Elsevier, [*S.l.*], v. 180, n. 1, p. 269–286, 1997. ISSN 0304-3975.

GRÜNBAUM, B. Acyclic colorings of planar graphs. **Israel Journal of Mathematics**, Springer, [*S.l.*], v. 14, n. 4, p. 390–408, 1973.

GRUNDY, P. M. Mathematics and games. **Eureka**, [*S.l.*], v. 2, p. 6–8, 1939.

GUAN, D.; XUDING, Z. A coloring problem for weighted graphs. **Information Processing Letters**, Elsevier, [*S.l.*], v. 61, n. 2, p. 77–81, 1997.

HARARY, F.; HEDETNIEMI, S. The achromatic number of a graph. **Journal of Combinatorial Theory**, Elsevier, [*S.l.*], v. 8, n. 2, p. 154–161, 1970. ISSN 0021-9800.

HARARY, F.; HEDETNIEMI, S.; PRINS, G. An interpolation theorem for graphical homomorphisms. **Portugaliae mathematica**, Sociedade Portuguesa de Matemática, [*S.l.*], v. 26, n. 4, p. 453–462, 1967.

HEGGERNES, P.; TELLE, J. A. Partitioning graphs into generalized dominating sets. **Nordic Journal of Computing**, [*S.l.*], v. 5, n. 2, p. 128–142, 1998. ISSN 1236-6064.

IRVING, R. W.; MANLOVE, D. F. The b-chromatic number of a graph. **Discrete Applied Mathematics**, Elsevier, [*S.l.*], v. 91, n. 1, p. 127–141, 1999.

JAFFKE, L.; LIMA, P. T.; LOKSHTANOV, D. b-coloring parameterized by clique-width. **Theory of Computing Systems**, Springer, [*S.l.*], p. 1–33, 2023.

JAFFKE, L.; LIMA, P. T.; SHARMA, R. *b*-Coloring Parameterized by Pathwidth is XNLP-complete. **arXiv preprint arXiv:2209.07772**, [*S.l.*], 2022. Disponível em: https://doi.org/10.48550/arXiv.2209.07772. Acesso em 05 set. 2024.

JAMISON, B.; OLARIU, S. A tree representation for P_4-sparse graphs. **Discrete Applied Mathematics**, Elsevier, [*S.l.*], v. 35, n. 2, p. 115–129, 1992. ISSN 0166-218X.

KALPANA, M.; VIJAYALAKSHMI, D. Fall coloring and b-coloring of graphs. **Journal of Physics: Conference Series**, IOP Publishing, [*S.l.*], v. 1139, n. 1, p. 012045, 2018.

KAUL, H.; MITILLOS, C. On Graph Fall-Coloring: Existence and Constructions. **Graphs and Combinatorics**, Springer, [*S.l.*], v. 35, n. 6, p. 1633–1646, 2019.

LARSEN, M.; PROPP, J.; ULLMAN, D. The fractional chromatic number of Mycielski's graphs. **Journal of Graph Theory**, Wiley Online Library, [*S.l.*], v. 19, n. 3, p. 411–416, 1995.

LASKAR, R.; LYLE, J. Fall colouring of bipartite graphs and cartesian products of graphs. **Discrete Applied Mathematics**, Elsevier, [*S.l.*], v. 157, n. 2, p. 330–338, 2009.

LAURI, J.; MITILLOS, C. Complexity of Fall Coloring for Restricted Graph Classes. **Theory of Computing Systems**, Springer, [*S.l.*], v. 11638, p. 352–364, 2019.

LAURI, J.; MITILLOS, C. Complexity of Fall Coloring for Restricted Graph Classes. **Theory of Computing Systems**, Springer, [*S.l.*], v. 64, n. 7, p. 1183–1196, 2020. ISSN 1432-4350.

MCCONNELL, R. M.; SPINRAD, J. P. Linear-time Modular Decomposition and Efficient Transitive Orientation of Comparability Graphs. In: **Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms**. Arlington, Virginia, USA: Society for Industrial and Applied Mathematics, 1994. (SODA '94), p. 536–545. ISBN 0898713293.

PANOLAN, F.; PHILIP, G.; SAURABH, S. On the Parameterized Complexity of B-Chromatic Number. **Journal of Computer and System Sciences**, Elsevier, [*S.l.*], v. 84, p. 120–131, 2017.

RAO, M. **Décompositions de graphes et algorithmes efficaces**. Tese (Doutorado) – Université Paul Verlaine - Metz, [*S.l.*], 2006.

ROBERTSON, N.; SANDERS, D. P.; SEYMOUR, P.; THOMAS, R. Efficiently Four-Coloring Planar Graphs. In: **Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing**. New York, NY, USA: Association for Computing Machinery, 1996. (STOC '96), p. 571–575. ISBN 0897917855.

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

SHAEBANI, S. On Fall Colorings of Graphs. **arXiv preprint arXiv:0909.2769**, [*S.l.*], 2009. Disponível em: https://doi.org/10.48550/arXiv.0909.2769. Acesso em 05 set. 2024.

SHAEBANI, S. A note on fall colorings of Kneser graphs. **Transactions on Combinatorics**, University of Isfahan, [*S.l.*], v. 8, n. 3, p. 13–15, 2019. ISSN 2251-8657.

SILVA, A. Graphs with small fall-spectrum. **Discrete Applied Mathematics**, Elsevier, [*S.l.*], v. 254, p. 183–188, 2019. ISSN 0166-218X.

TELLE, J. A.; PROSKUROWSKI, A. Algorithms for Vertex Partitioning Problems on Partial k-Trees. **SIAM Journal on Discrete Mathematics**, SIAM, [*S.l.*], v. 10, n. 4, p. 529–550, 1997.

VALENCIA-PABON, M. Idomatic partitions of direct products of complete graphs. **Discrete Mathematics**, Elsevier, [*S.l.*], v. 310, n. 5, p. 1118–1122, 2010. ISSN 0012-365X.

WEST, D. **Introduction to Graph Theory**. [*S.l.*]: Prentice Hall, 2001. (Featured Titles for Graph Theory). ISBN 9780130144003.

ZAKER, M. A note concerning the Grundy and *b*-chromatic number of graphs. **arXiv preprint arXiv:2003.14233**, [*S.l.*], 2020. Disponível em: https://doi.org/10.48550/arXiv.2003.14233. Acesso em 05 set. 2024.

INDEX