Grundy Distinguishes Treewidth from Pathwidth

Rémy Belmonte
University of Electro-Communications, Chofu, Tokyo, 182-8585, Japan
remybelmonte@gmail.com

Eun Jung Kim
Université Paris-Dauphine, PSL University, CNRS, LAMSADE, 75016, Paris, France
eun-jung.kim@dauphine.fr

Michael Lampis
Université Paris-Dauphine, PSL University, CNRS, LAMSADE, 75016, Paris, France
michail.lampis@lamsade.dauphine.fr

Valia Mitsou
Université de Paris, IRIF, CNRS, 75205, Paris, France
vmitsou@irif.fr

Yota Otachi
Kumamoto University, Kumamoto, 860-8555, Japan
otachi@cs.kumamoto-u.ac.jp

Abstract

Structural graph parameters, such as treewidth, pathwidth, and clique-width, are a central topic of study in parameterized complexity. A main aim of research in this area is to understand the "price of generality" of these widths: as we transition from more restrictive to more general notions, which are the problems that see their complexity status deteriorate from fixed-parameter tractable to intractable? This type of question is by now very well-studied, but, somewhat strikingly, the algorithmic frontier between the two (arguably) most central width notions, treewidth and pathwidth, is still not understood: currently, no natural graph problem is known to be W-hard for one but FPT for the other. Indeed, a surprising development of the last few years has been the observation that for many of the most paradigmatic problems, their complexities for the two parameters actually coincide exactly, despite the fact that treewidth is a much more general parameter. It would thus appear that the extra generality of treewidth over pathwidth often comes "for free".

Our main contribution in this paper is to uncover the first natural example where this generality comes with a high price. We consider Grundy Coloring, a variation of coloring where one seeks to calculate the worst possible coloring that could be assigned to a graph by a greedy First-Fit algorithm. We show that this well-studied problem is FPT parameterized by pathwidth; however, it becomes significantly harder (W[1]-hard) when parameterized by treewidth. Furthermore, we show that Grundy Coloring makes a second complexity jump for more general widths, as it becomes para-NP-hard for clique-width. Hence, Grundy Coloring nicely captures the complexity trade-offs between the three most well-studied parameters. Completing the picture, we show that Grundy Coloring is FPT parameterized by modular width.

2012 ACM Subject Classification Mathematics of computing→Graph algorithms; Theory of Computation → Design and Analysis of Algorithms → Parameterized Complexity and Exact Algorithms

Keywords and phrases Treewidth, Pathwidth, Clique-width, Grundy Coloring

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:27
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

The study of the algorithmic properties of *structural graph parameters* has been one of the most vibrant research areas of parameterized complexity in the last few years. In this area we consider graph complexity measures ("graph width parameters"), such as treewidth, and attempt to characterize the class of problems which become tractable for each notion of width. The most important graph widths are often comparable to each other in terms of their generality. Hence, one of the main goals of this area is to understand which problems separate two comparable parameters, that is, which problems transition from being FPT for a more restrictive parameter to W-hard for a more general one[1]. This endeavor is sometimes referred to as determining the "price of generality" of the more general parameter.

The two most widely studied graph widths are probably treewidth and pathwidth, which have an obvious containment relationship to each other. Despite this, to the best of our knowledge, no natural problem is currently known to delineate their complexity border in the sense we just described. Our main contribution is exactly to uncover a natural, well-known problem which fills this gap. Specifically, we show that Grundy Coloring, the problem of ordering the vertices of a graph to maximize the number of colors used by the First-Fit coloring algorithm, is FPT parameterized by pathwidth, but W[1]-hard parameterized by treewidth. We then show that Grundy Coloring makes a further complexity jump if one considers clique-width, as in this case the problem is para-NP-complete. Hence, Grundy Coloring turns out to be an interesting specimen, nicely demonstrating the algorithmic trade-offs involved among the three most central graph widths.

Graph widths and the price of generality. Much of modern parameterized complexity theory is centered around studying graph widths, especially treewidth and its variants. In this paper we focus on the parameters summarized in Figure 1, and especially the parameters that form a linear hierarchy, from vertex cover, to tree-depth, pathwidth, treewidth, and clique-width. Each of these parameters is a strict generalization of the previous ones in this list. On the algorithmic level we would expect this relation to manifest itself by the appearance of more and more problems which become *intractable* as we move towards the more general parameters. Indeed, a search through the literature reveals that for each step in this list of parameters, several *natural* problems have been discovered which distinguish the two consecutive parameters (we give more details below). The one glaring exception to this rule seems to be the relation between treewidth and pathwidth.

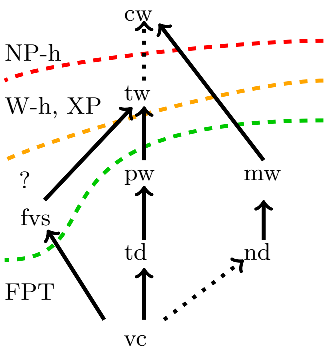
Treewidth is a parameter of central importance to parameterized algorithmics, in part because wide classes of problems (notably all MSO_2-expressible problems [19]) are FPT for this parameter. Treewidth is usually defined in terms of tree decompositions of graphs, which naturally leads to the equally well-known notion of pathwidth, defined by forcing the decomposition to be a path. On a graph-theoretic level, the difference between the two notions is well-understood and treewidth is known to describe a much richer class of graphs. In particular, while all graphs of pathwidth k have treewidth at most k, there exist graphs of constant treewidth (in fact, even trees) of unbounded pathwidth. Naturally, one would expect this added richness of treewidth to come with some negative algorithmic consequences in the form of problems which are FPT for pathwidth but W-hard for treewidth. Furthermore, since treewidth and pathwidth are probably the most studied parameters in our list, one

[1] We assume the reader is familiar with the basics of parameterized complexity theory, such as the classes FPT and W[1], as given in standard textbooks [22].

might expect the problems that distinguish the two to be the first ones to be discovered.

Nevertheless, so far this (surprisingly) does not seem to have been the case: on the one hand, FPT algorithms for pathwidth are DPs which also extend to treewidth; on the other hand, we give (in the appendix) a semi-exhaustive list of dozens of natural problems which are W[1]-hard for treewidth and turn out without exception to also be hard for pathwidth. In fact, even when this is sometimes not explicitly stated in the literature, the same reduction that establishes W-hardness by treewidth also does so for pathwidth. Intuitively, an explanation for this phenomenon is that the basic structure of such reductions typically resembles a $k \times n$ (or smaller) grid, which has both treewidth and pathwidth bounded by k.

Our main motivation in this paper is to take a closer look at the algorithmic barrier between pathwidth and treewidth and try to locate a natural (that is, not artificially contrived) problem whose complexity transitions from FPT to W-hard at this barrier. Our main result is the proof that Grundy Coloring is such a problem. This puts in the picture the last missing piece of the puzzle, as we now have natural problems that distinguish the parameterized complexity of any two consecutive parameters in our main hierarchy.



Parameter	Result	Ref
Clique-width	para-NP-hard	Theorem 22
Treewidth	W[1]-hard	Theorem 13
Pathwidth	FPT	Theorem 17
Modular Width	FPT	Theorem 23

In the figure, clique-width, treewidth, pathwidth, tree-depth, vertex cover, feedback vertex set, neighborhood diversity, and modular-width are indicated as cw, tw, pw, td, vc, fvs, nd, and mw respectively. Arrows indicate more general parameters. Dotted arrows indicate that the parameter may increase exponentially, (e.g. graphs of vc k have nd at most $2^k + k$).

Figure 1 Summary of considered graph parameters and results.

Grundy Coloring. In the Grundy Coloring problem we are given a graph $G = (V, E)$ and are asked to order V in a way that maximizes the number of colors used by the greedy (First-Fit) coloring algorithm. The notion of Grundy coloring was first introduced by Grundy in the 1930s, and later formalized in [18]. Since then, the complexity of Grundy Coloring has been very well-studied (see [1, 3, 15, 31, 44, 46, 52, 55, 73, 75, 78, 79, 80] and the references therein). For the natural parameter, namely the number of colors to be used, Grundy coloring was recently proved to be W[1]-hard in [1]. An XP algorithm for Grundy Coloring parameterized by treewidth was given in [75], using the fact that the Grundy number of any graph is at most $\log n$ times its treewidth. In [14] Bonnet et al. explicitly asked whether this can be improved to an FPT algorithm. They also observed that the problem is FPT parameterized by vertex cover. It appears that the complexity of Grundy Coloring parameterized by pathwidth was never explicitly posed as a question and it was not suspected that it may differ from that for treewidth. We note that, since the problem (as given in Definition 1) is easily seen to be MSO_1 expressible for a fixed Grundy number, it is FPT for all considered parameters if the Grundy number is also a parameter [20], so we intuitively want to concentrate on cases where the Grundy number is large.

Our results. Our results illuminate the complexity of Grundy Coloring parameterized by pathwidth and treewidth, as well as clique-width and modular-width. More specifically:

1. We show that Grundy Coloring is W[1]-hard parameterized by treewidth via a reduction from k-Multi-Colored Clique. The main building block of our reduction is the structure of binomial trees, which have treewidth one but unbounded pathwidth, which explains the complexity jump between the two parameters. As mentioned, an XP algorithm is known in this case [75], so this result is in a sense tight.
2. We show that Grundy Coloring is FPT parameterized by pathwidth. Our main tool here is a combinatorial lemma, which draws heavily from known combinatorial bounds on the performance of First-Fit coloring on intervals graphs [53, 65]. We use this lemma to show that on any graph the Grundy number is at most a linear function of the pathwidth.
3. We show that Grundy Coloring is para-NP-complete parameterized by clique-width, that is, NP-complete for graphs of constant clique-width (specifically, clique-width 6).
4. We show that Grundy Coloring is FPT parameterized by neighborhood diversity [56] and leverage this result to obtain an FPT algorithm by modular-width [38].

Our main interest is concentrated in the first two results, which achieve our goal of finding a natural problem distinguishing pathwidth from treewidth. The result for clique-width nicely fills out the picture by giving an intuitive view of the evolution of the complexity of the problem and showing that in a case where no non-trivial bound can be shown on the optimal value, the problem becomes hopelessly hard from the parameterized point of view.

Other related work. Let us now give a brief survey of "price of generality" results involving our considered parameters, that is, results showing that a problem is efficient for one parameter but hard for a more general one. In this area, the results of Fomin et al. [35], introducing the term "price of generality", have been particularly impactful. This work and its follow-ups [36, 37], were the first to show that four natural graph problems (Coloring, Edge Dominating Set, Max Cut, Hamiltonicity) which are FPT for treewidth, become W[1]-hard for clique-width. In this sense, these problems, as well as problems discovered later such as counting perfect matchings [21], SAT [68, 24], ∃∀-SAT [59], Orientable Deletion [45], and d-Regular Induced Subgraph [17], form part of the "price" we have to pay for considering a more general parameter. This line of research has thus helped to illuminate the complexity border between the two most important sparse and dense parameters (treewidth and clique-width), by giving a list of *natural* problems distinguishing the two. (An artificial MSO_2-expressible such problem was already known much earlier [20, 58]).

Let us now focus in the area below treewidth in Figure 1 by considering problems which are in XP but W[1]-hard parameterized by treewidth. By now, there is a small number of problems in this category which are known to be W[1]-hard even for vertex cover: List Coloring [32] was the first such problem, followed by CSP (for the vertex cover of the dual graph) [70], and more recently by (k, r)-Center, d-Scattered Set, and Min Power Steiner Tree [49, 48, 50] on weighted graphs. Intuitively, it is not surprising that problems W[1]-hard by vertex cover are few and far between, since this is a very restricted parameter. Indeed, for most problems in the literature which are W[1]-hard by treewidth, vertex cover is the only parameter (among the ones considered here) for which the problem becomes FPT.

A second interesting category are problems which are FPT for tree-depth ([66]) but W[1]-hard for pathwidth. Mixed Chinese Postman Problem was the first discovered problem of this type [43], followed by Min Bounded-Length Cut [26, 10], ILP [40], Geodetic Set [51] and unweighted (k, r)-Center and d-Scattered Set [49, 48].

Let us also mention in passing that the algorithmic differences of pathwidth and treewidth may also be studied in the context of problems which are hard for constant treewidth. Such problems also generally remain hard for constant pathwidth (examples are Steiner Forest

[42], Bandwidth [64], Minimum mcut [41]). One could also potentially try to distinguish between pathwidth and treewidth by considering the parameter dependence of a problem that is FPT for both. Indeed, for a long time the best-known algorithm for Dominating Set had complexity 3^k for pathwidth, but 4^k for treewidth. Nevertheless, the advent of fast subset convolution techniques [77], together with tight SETH-based lower bounds [60] has, for most problems, shown that the complexities on the two parameters coincide exactly.

Finally, let us mention a case where pathwidth and treewidth have been shown to be quite different in a sense similar to our framework. In [69] Razgon showed that a CNF can be compiled into an OBDD (Ordered Binary Decision Diagram) of size FPT in the pathwidth of its incidence graphs, but there exist formulas that always need OBDDs of size XP in the treewidth. Although this result does separate the two parameters, it is somewhat adjacent to what we are looking for, as it does not speak about the complexity of a decision problem, but rather shows that an OBDD-producing algorithm parameterized by treewidth would need XP time simply because it would have to produce a huge output in some cases.

2 Definitions and Preliminaries

For non-negative integers i, j, we use $[i, j]$ to denote the set $\{k \mid i \leq k \leq j\}$. Note that if $j > i$, then the set $[i, j]$ is empty. We will also write simply $[i]$ to denote the set $[1, i]$.

We give two equivalent definitions of our main problem.

▶ **Definition 1.** *A k-Grundy Coloring of a graph $G = (V, E)$ is a partition of V into k non-empty sets $V_1, \ldots, V_k$ such that: (i) for each $i \in [k]$ the set V_i induces an independent set; (ii) for each $i \in [k-1]$ the set V_i dominates the set $\bigcup_{i<j\leq k} V_j$.*

▶ **Definition 2.** *A k-Grundy Coloring of a graph $G = (V, E)$ is a proper k-coloring $c : V \to [k]$ that results by applying the First-Fit algorithm on an ordering of V, where each vertex is assigned the minimum color that is not assigned to any of its previously colored neighbors.*

The Grundy number of a graph G, denoted by $\Gamma(G)$, is the maximum k such that G admits a k-Grundy Coloring. In a given Grundy Coloring, if $u \in V_i$ (equiv. if $c(u) = i$) we will say that u was given color i. The Grundy Coloring problem is the problem of determining the maximum k for which a graph G admits a k-Grundy Coloring. It is not hard to see that a proper coloring is a Grundy coloring if and only if every vertex assigned color i has at least one neighbor assigned color j, for each $j < i$.

3 W[1]-Hardness for Treewidth

In this section we prove that Grundy Coloring parameterized by treewidth is W[1]-hard (Theorem 13). Our proof relies on a reduction from k-Multi-Colored Clique and initially establishes W[1]-hardness for a more general problem where we are given a target color for a set of vertices (Lemma 8); we then reduce this to Grundy Coloring. Interestingly, this intermediate problem turns out to be W[1]-hard even for pathwidth (Lemma 9), since our reduction uses the standard strategy of constructing a grid-like structure of dimensions $k \times n$. The reason this reductioni fails to prove that Grundy Coloring is W[1]-hard by pathwidth is that we use some gadgets to implement the targets and a support operation (which "pre-colors" some vertices) and for these gadgets we use trees of unbounded pathwidth. The results of Section 4 show that this is essential: our reduction *needs* some part that causes it to have high pathwidth, otherwise the Grundy number of the constructed graph would be bounded by the parameter, resulting in an instance that can be solved in FPT time.

Let us now present the different parts of our construction. We will make use of the structure of binomial trees T_i.

▶ **Definition 3.** *The binomial tree T_i with root r_i is a rooted tree defined recursively in the following way: T_1 consists simply of its root r_1; in order to construct T_i for $i > 1$, we construct one copy of T_j for all $j < i$ and connect r_j with r_i. An alternative equivalent definition of the binomial tree T_i, $i \geq 2$ is that we construct two trees T_{i-1} , T'_{i-1}, we connect their roots r_{i-1}, r'_{i-1} and select one of them as the new root r_i.*

▶ **Proposition 4.** *Let T_i be a binomial tree and $t < i$. There exist 2^{i-t-1} vertex-disjoint subtrees T_t in T_i, where no T_t contains the root r_i of T_i.*

▶ **Proposition 5.** $\Gamma(T_i) \leq i$. *Furthermore, for all $j \leq i$ there exists a Grundy coloring which assigns color j to the root of T_i.*

We now define a generalization of the Grundy coloring problem with target colors and show that it is W[1]-hard parameterized by treewidth. We later describe how to reduce this problem to Grundy Coloring such that the treewidth does not increase by a lot.

▶ **Definition 6** (Grundy Coloring with Targets). *We are given a graph $G(V, E)$, an integer $t \in \mathbb{N}$ called the target and a subset $S \subset V$. (For simplicity we will say that vertices of S have target t.) We say that G admits a* Target-achieving Grundy Coloring *if there exists a Grundy Coloring which assigns to all vertices of S color t.*

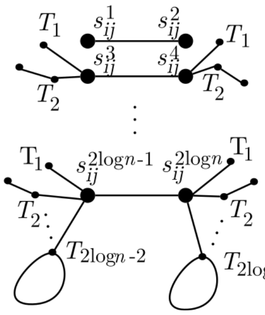
We will also make use of the following operation:

▶ **Definition 7** (Tree-support.). *Given a graph $G = (V, E)$, a vertex $u \in V$ and a set N of positive integers, we define the* tree-support *operation as follows: (a) for all $i \in N$ we add a copy of T_i in the graph; (b) we connect u to the root r_i of each of the T_i. We say that we* add supports *N on u. The trees T_i will be called the* supporting trees *or* supports *of u. Slightly abusing notation, we also call* supports *the numbers $i \in N$.*

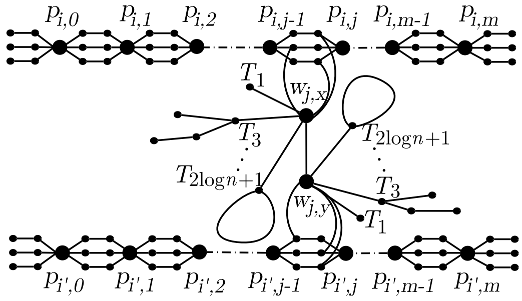
Intuitively, the tree-support operation ensures that vertex u may have at least one neighbor of color i for each $i \in N$ in a Grundy coloring, and thus increase the color u can take. Observe that adding supporting trees to a vertex does not increase the treewidth, but does increase the pathwidth (binomial trees have unbounded pathwidth).

Our reduction is from k-Multi-Colored Clique: given a k-multipartite graph $G = (V_1, V_2, \ldots, V_k, E)$, decide if for every $i \in [k]$ we can pick $u_i \in V_i$ forming a clique, where k is the parameter. We can also assume that $\forall i \in [k], |V_i| = n$, that n is a power of 2, and that $V_i = \{v_{i,0}, v_{i,1}, \ldots, v_{i,n-1}\}$. Furthermore, let $|E| = m$. We construct an instance of Grundy Coloring with Targets $G' = (V', E')$ and $t = 2\log n + 4$ (where all logarithms are base two) using the following gadgets:

Vertex selection $S_{i,j}$. See Figure 2a. This gadget consists of $2\log n$ vertices $S^1_{i,j} \cup S^2_{i,j} = \bigcup_{l\in[\log n]}\{s^{2l-1}_{i,j}\} \cup \bigcup_{l\in[\log n]}\{s^{2l}_{i,j}\}$, where for each $l \in [\log n]$ we connect vertex $s^{2l-1}_{i,j}$ to $s^{2l}_{i,j}$ thus forming a matching. Furthermore, for each $l \in [2, \log n]$, we add supports $[2l-2]$ to vertices $s^{2l-1}_{i,j}$ and $s^{2l}_{i,j}$. Observe that the vertices $s^{2l-1}_{i,j}$ and $s^{2l}_{i,j}$ together with their supports form a binomial tree T_{2l}. We construct $k(m+2)$ gadgets $S_{i,j}$, one for each $i \in [k], j \in [0, m+1]$. The vertex selection gadget $S_{i,1}$ encodes in binary the vertex that is selected in the clique from V_i. In particular, for each pair $s^{2l-1}_{i,1}, s^{2l}_{i,1}$, $l \in [\log n]$ either of these vertices can take the maximum color in an optimal grundy coloring of the binomial tree T_{2l} (that is, a coloring that gives the root of the binomial tree T_{2l} color $2l$). A selection corresponds to bit 0 or 1 for the l^{th} binary position. In order to ensure that for each $j \in [m]$ all (middle) $S_{i,j}$ encode the same vertex, we use propagators.



(a) Vertex Selection gadget $S_{i,j}$.



(b) Propagators $p_{i,j}$ and Edge Selection gadget W_j. We don't show the edge selection checkers and the supports of the $p_{i,j}$. In the example $B_x = 010$ and $B_y = 100$.

Figure 2 The gadgets. Figure 2a is an enlargment of Figure 2b between $p_{i,j-1}$ and $p_{i,j}$.

Propagators $p_{i,j}$. See Figure 2b. For $i \in [k]$ and $j \in [0, m]$, a propagator $p_{i,j}$ is a single vertex connected to all vertices of $S^2_{i,j} \cup S^1_{i,j+1}$. To each $p_{i,j}$, we also add supports $\{2\log n + 1, 2\log n + 2, 2\log n + 3\}$. The propagators have target $t = 2\log n + 4$.

Edge selection W_j. See Figure 2b. Let $j = (v_{i,x}, v_{i',y}) \in E$, where $v_{i,x} \in V_i$ and $v_{i',y} \in V_{i'}$. The gadget W_j consists of four vertices $w_{j,x}, w_{j,y}, w'_{j,x}, w'_{j,y}$. We call $w'_{j,x}, w'_{j,y}$ the *edge selection checkers*. We have the edges $(w_{j,x}, w_{j,y}), (w'_{j,x}, w_{j,x}), (w'_{j,y}, w_{j,y})$. Let us now describe the connections of these vertices with the rest of the graph. Let $B_x = b_1 b_2 \dots b_{\log n}$ be the binary representation of x. We connect $w_{j,x}$ to each vertex $s^{2l-b_l}_{ij}$, $l \in [\log n]$ (we do similarly for $w_{j,y}, S_{i',j}$, and B_y). We add to each of $w_{j,x}, w_{j,y}$ supports $\bigcup_{l\in[\log n+1]}\{2l-1\}$. We add to each of $w'_{j,x}, w'_{j,y}$ supports $[2\log n+3] \setminus \{2\log n+1\}$ and set for these two vertices target $t = 2\log n + 4$. We construct m such gadgets, one for each edge. We say that W_j *is activated* if at least one of $w_{j,x}, w_{j,y}$ receives color $2\log n + 3$.

Edge checkers $q_{i,i'}$. We construct $\binom{k}{2}$ of them, one for each pair $(i, i'), i < i' \in [k]$. The edge checker is a single vertex that is connected to all vertices $w_{j,x}$ for which j is an edge between V_i and $V_{i'}$. We add supports $[2\log n + 1]$ and a target of $t = 2\log n + 4$.

The edge checker plays the role of an "or" gadget: in order for it to achieve its target, at least one of its neighboring edge selection gadgets should be activated.

▶ **Lemma 8.** *G has a clique of size k if and only if G' has a target-achieving Grundy coloring.*

Proof. $\Rightarrow$) Suppose that G has a clique. We color the vertices of G' in the following order: First, we color the vertex selection gadget $S_{i,j}$, starting from the supports (which we color optimally) and then color the matchings as follows: let $v_{i,x}$ be the vertex that was selected in the clique from V_i and $b_1 b_2 \dots b_{\log n}$ be the binary representation of x; we color vertices $s^{2l-(1-b_l)}_{i,j}$, $l \in [\log n]$ with color $2l - 1$ and vertices $s^{2l-b_l}_{i,j}$, $l \in [\log n]$ will receive color $2l$. For the propagators, we color their supports optimally. Propagators have $2\log n + 3$ neighbors each, all with different colors, so they receive color $2\log n + 4$, thus achieving their targets.

Then, we color the edge-checkers $q_{i,i'}$ and the edge selection gadgets W_j that correspond to edges of the clique (that is, $j = (v_{i,x}, v_{i',y}) \in E$ and $v_{i,x} \in V_i$, $v_{i',y} \in V_{i'}$ are selected in the clique). We first color the supports of $q_{i,i'}, w_{j,x}, w_{j,y}$ optimally. From the construction, vertex $w_{j,x}$ is connected with vertices $s^{2l-b_l}_{i,j}$ which have already been colored $2l$, $l \in [\log n]$ and with supports $\bigcup_{l\in[\log n+1]}\{2l-1\}$, thus $w_{j,x}$ will receive color $2\log n + 2$. Similarly $w_{j,y}$ already has neighbors which are colored $[2\log n + 1]$, but also $w_{j,x}$, thus it will receive color $2\log n + 3$. These W_j will be activated. Since both $w_{j,x}, w_{j,y}$ connect to $q_{i,i'}$, the latter will be assigned color $2\log n+4$, thus achieving its target. As for $w'_{j,x}$ and $w'_{j,y}$, such a vertex has a neighbor with color c where $c=2\log n+2$ or $c=2\log n+3$. We therefore, color the support T_c in a way that gives its root color $2\log n+1$ and color the remaining supports optimally. This gives vertices $w'_{j,x}, w'_{j,y}$ color $t=2\log n+4$ achieving their target.

Finally, for the remaining W_j, we claim that we can assign to both $w_{j,x}, w_{j,y}$ a color that is at least as high as $2\log n+1$. Indeed, we assign to each supporting tree T_r of $w_{j,x}$ a coloring that gives its root the maximum color that is $\le r$ and does not appear in any neighbor of $w_{j,x}$ in the vertex selection gadget. We claim that in this case $w_{j,x}$ will have neighbors with all colors in $[2\log n]$, because in every interval $[2l-1,2l]$ for $l\in[\log n]$, $w_{j,x}$ has a neighbor with a color in the interval and a support tree T_{2l+1}. If $w_{j,x}$ has color $2\log n+1$ then we color the supports of $w'_{j,x}$ optimally and achieve its target, while if $w_{j,x}$ has color higher than $2\log n+1$, we achieve the target of $w'_{j,x}$ as in the previous paragraph.

$\Leftarrow$) Suppose that G' admits a coloring that achieves the target for all propagators, edge-checkers, and edge selection checkers. We will prove the following: 1) the coloring of the vertex selection gadgets is consistent throughout (this corresponds to a selection of k vertices of G); 2) that $\binom{k}{2}$ edge selection gadgets have been activated (that correspond to $\binom{k}{2}$ edges of G) and 3) if an edge selection gadget $W_j=\{w_{j,x}, w_{j,y}\}$ with $j=(v_{i,x}, v_{i',y})$ has been activated then the coloring of the vertex selection gadgets $S_{i,j}$ and $S_{i',j}$ corresponds to the selection of vertices $v_{i,x}$ and $v_{i',y}$ (selected vertices and edges form indeed a K_k in G).

1) Suppose that an edge selection checker $w'_{j,x}$ achieved its target. We claim that this implies that $w_{j,x}$ has color at least $2\log n+1$. Indeed, $w'_{j,x}$ has degree $2\log n+3$, so its neighbors must have all distinct colors in $[2\log n+3]$, but among the supports there are only 2 neighbors which may have colors in $[2\log n+1, 2\log n+3]$. Therefore, the missing color must come from $w_{j,x}$. We now observe that vertices from the vertex selection gadgets have color at most $2\log n$, because if we exclude from their neighbors the vertices $w_{j,x}$ (which we argued have color at least $2\log n+1$) and the propagators (which have target $2\log n+4$), these vertices have degree at most $2\log n-1$.

Suppose that a propagator $p_{i,j}$ achieves its target of $2\log n+4$. Since this vertex has a degree of $2\log n+3$, that means that all of its neighbors should receive all the colors in $[2\log n+3]$. As argued, colors $[2\log n+1, 2\log n+3]$ must come from the supports. Therefore, the colors $[2\log n]$ come from the neighbors of $p_{i,j}$ in the vertex selection gadgets.

We now note that, because of the degrees of vertices in vertex selection gadgets, only vertices $s_{i,j}^{2\log n}, s_{i,j+1}^{2\log n-1}$ can receive colors $2\log n, 2\log n-1$; from the rest, only $s_{i,j}^{2\log n-2}, s_{i,j+1}^{2\log n-3}$ can receive colors $2\log n-2, 2\log n-3$ etc. Thus, for each $l\in[\log n]$, if $s_{i,j}^{2l}$ receives color $2l-1$ then $s_{i,j+1}^{2l-1}$ should receive color $2l$ and vice versa. With similar reasoning, in all vertex selection gadgets we have that $s_{i,j}^{2l-1}, s_{i,j}^{2l}$, since they are neighbors, received the two colors $\{2l-1,2l\}$. As a result, the colors of $s_{i,j+1}^{2l-1}$, $s_{i,j}^{2l-1}$ (and thus the colors of $s_{i,j+1}^{2l}$, $s_{i,j}^{2l}$) are the same, therefore, the coloring is consistent, for all values of $j\in[m]$.

2) If an edge checker achieves its target of $2\log n+4$, then at least one of its neighbors from an edge selection gadget has received color $2\log n+3$. We know that each edge selection gadget only connects to a unique edge checker, so there should be $\binom{k}{2}$ edge selection gadgets which have been activated in order for all edge checkers to achieve their target.

3) Suppose that an edge checker $q_{i,i'}$ achieves its target. That means that there exists an edge selection gadget $W_j=\{w_{j,x}, w_{j,y}, w'_{j,x}, w'_{j,y}\}$ for which at least one of its vertices, say $w_{j,x}$ has received color $2\log n+3$. Let j be an edge connecting $v_{i,x}\in V_i$ to $v_{i',y}\in V_{i'}$. Since the degree of $w_{j,x}$ is $2\log n+4$ and we have already assumed that two of its neighbors ($q_{i,i'}$ and $w'_{j,x}$) have color $2\log n+4$, in order for it to receive color $2\log n+3$ all its other neighbors should receive all colors in $[2\log n+2]$. The only possible assignment is to give

colors $2l$, $l \in [\log n]$ to its neighbors from $S_{i,j}$ and color $2\log n + 2$ to $w_{j,y}$. The latter is, in turn, only possible if the neighbors of $w_{j,y}$ from $S_{i',j}$ receive all colors $2l$, $l \in [\log n]$. The above corresponds to selecting vertex $v_{i,x}$ from V_i and $v_{i',y}$ from $V_{i'}$. ◀

▶ **Lemma 9.** *Let G'' be the graph that results from G' if we remove all the tree-supports. Then G'' has pathwidth at most $\binom{k}{2} + 2k + 3$.*

We will now show how to implement the targets using the tree-filling operation below.

▶ **Definition 10** (Tree-filling). *Let $G = (V, E)$ be a graph and $S = \{s_1, s_2, \ldots, s_j\} \subset V$ a set of vertices with target t. The* tree-filling *operation is the following: (a) we add in G a binomial tree T_i, where $i = \lceil \log j \rceil + t + 1$; (b) for each $s \in S$ we find a disjoint copy of T_t in T_i, identify s with its root r_t, and delete all other vertices of the sub-tree T_t.*

Observe that i is calculated in a way that by Proposition 4 there always exist enough disjoint T_t sub-trees to perform the operation. The tree-filling operation might in general increase treewidth, but we will do it in a way that it only increases by a constant factor.

▶ **Lemma 11.** *Let $G = (V, E)$ be a graph of pathwidth w and $S = \{s_1, \ldots, s_j\} \subset V$ a subset of vertices having target t. Then there is a way to apply the tree-filling operation such that the resulting graph H has $tw(H) \leq 4w + 5$.*

Proof. **Construction of H.** Let $(\mathcal{P}, \mathcal{B})$ be a path-decomposition of G whose largest bag has size $w + 1$ and $B_1, B_2, \ldots, B_j \in \mathcal{B}$ distinct bags where $\forall i, s_i \in B_i$ (assigning a distinct bag to each s_i is always possible, as we can duplicate bags if necessary). We call those bags *important*. We define an ordering $o : S \to \mathbb{N}$ of the vertices of S that follows the order of the important bags from left to right, that is $o(s_i) < o(s_j)$ if B_i is on the left of B_j in $\mathcal{P}$. For simplicity, let us assume that $o(s_i) = i$ and that B_i is to the left of B_j if $i < j$.

We describe a recursive way to do the substitution of the trees in the tree-filling operation. Crucially, when $j > 2$ we will have to select an appropriate mapping between the vertices of S and the disjoint subtrees T_t in the added binomial tree T_i, so that we will be able to keep the treewidth of the new graph bounded.

- If $j = 1$ then $i = t + 1$. We add to the graph a copy of T_i, arbitrarily select the root of a copy of T_t contained in T_i, and perform the tree-filling operation as described.
- Suppose that we know how to perform the substitution for sets of size at most $\lceil j/2 \rceil$, we will describe the substitution process for a set of size j. We have $i = \lceil \log j \rceil + t + 1$ and for all j we have $\lceil \log \lceil j/2 \rceil \rceil = \lceil \log j \rceil - 1$. Split the set S into two (almost) equal disjoint sets S^L and S^R of size at most $\lceil j/2 \rceil$, where for all $s_i \in S^L$ and for all $s_j \in S^R$, $i < j$. We perform the tree-filling on each of these sets by constructing two binomial trees T^L_{i-1}, T^R_{i-1} and doing the substitution; then, we connect their roots and set the root of the left tree as the root r_i of T_i, thus creating the substitution of a tree T_i.

Small treewidth. We now prove that the new graph H that results from applying the tree-filling operation on G and S as described above has a tree decomposition $(\mathcal{T}, \mathcal{B}')$ of width $4w + 5$; in fact we prove by induction a stronger statement: if $A, Z \in \mathcal{B}$ are the left-most and right-most bags of $\mathcal{P}$, then there exists a tree decomposition $(\mathcal{T}, \mathcal{B}')$ of H of width $4w + 5$ with the added property that there exists $R \in \mathcal{B}'$ such that $A \cup Z \cup \{r_i\} \subset R$, where r_i is the root of the tree T_i.

For the base case, if $j = 1$ we have added to our graph a T_i of which we have selected an arbitrary sub-tree T_t, and identified the root r_t of T_t with the unique vertex of S that has a target. Take the path decomposition $(\mathcal{P}, \mathcal{B})$ of the initial graph and add all vertices of A (its first bag) and the vertex r_i (the root of T_i) to all bags. Take an optimal tree decomposition of T_i and add r_i to each bag, obtaining a decomposition of width 2. We add an edge between the bag of $\mathcal{P}$ that contains the unique vertex of S, and a bag of the decomposition of T_i that contains the selected r_t. We now have a tree decomposition of the new graph of width $2w + 2 < 4w + 5$. Observe that the last bag of $\mathcal{P}$ now contains all of A, Z and r_i.

For the inductive step, suppose we applied the tree-filling operation for a set S of size $j > 1$. Furthermore, suppose we know how to construct a tree decomposition with the desired properties (width $4w + 5$, one bag contains the first and last bags of the path decomposition $\mathcal{P}$ and r_i), if we apply the tree-filling operation on a target set of size at most $j - 1$. We show how to obtain a tree decompostition with the desired properties if the target set has size j.

By construction, we have split the set S into two sets S^L, S^R and have applied the tree-filling operation to each set separately. Then, we connected the roots of the two added trees to obtain a larger binomial tree. Observe that for $|S| = j > 1$ we have $|S^L|, |S^R| < j$.

Let us first cut $\mathcal{P}$ in two parts, in such a way that the important bags of S^L are on the left and the important bags of S^R are on the right. We call $A^L = A$ and Z^L the leftmost and rightmost bags of the left part and A^R, $Z^R = Z$ the leftmost and rightmost bags of the right part. We define as G^L (respectively G^R) the graph that contains all the vertices of the left (respectively right) part. Let r_i be the root of T_i and r_{i-1} the root of its subtree T_{i-1}. From the inductive hypothesis, we can construct tree decompositions $(\mathcal{T^L}, \mathcal{B^L})$, $(\mathcal{T^R}, \mathcal{B^R})$ of width $4w + 5$ for the graphs H^L, H^R that occur after applying tree-filling on G^L, S^L and G^R, S^R; furthermore, there exist $R^L \in \mathcal{B^L}, R^R \in \mathcal{B^R}$ such that $R^L \supseteq A \cup Z^L \cup \{r_i\}$ and $R^R \supseteq A^R \cup Z \cup \{r_{i-1}\}$.

We construct a new bag $R' = A \cup A^R \cup Z^L \cup Z \cup \{r_{i-1}, r_i\}$, and we connect R' to both R^L and R^R, thus combining the two tree-decompositions into one. Last we create a bag $R = A \cup Z \cup \{r_i\}$ and attach it to R'. This completes the construction of $(\mathcal{T}, \mathcal{B}')$.

Observe that $(\mathcal{T}, \mathcal{B}')$ is valid for H:

- $V(H) = V(H^L) \cup V(H^R)$, thus $\forall v \in V(H), v \in \mathcal{B^L} \cup \mathcal{B^R} \subset \mathcal{B}$.
- $E(H) = E(H^L) \cup E(H^R) \cup \{(r_{i-1}, r_i)\}$. We have that $r_{i-1}, r_i \in R' \in \mathcal{B}$. All other edges were dealt with in $\mathcal{T^L}, \mathcal{T^R}$.
- Each vertex $v \in V(H)$ that belongs in exactly one of H^L, H^R trivially satisfied the connectivity requirement: bags that contain v are either fully contained in $\mathcal{T^L}$ or $\mathcal{T^R}$. A vertex v that belongs in both H^L and H^R belongs in $Z^L \cap A^R$, hence in R'. Therefore, the sub-trees of bags that contain v in $\mathcal{T^L}, \mathcal{T^R}$, form a connected sub-tree in $\mathcal{T}$.

The width of $\mathcal{T}$ is $\max\{tw(H^L), tw(H^R), 4w + 5\} = 4w + 5$. ◀

The last thing that remains to do in order to complete the proof is to show the equivalence between achieving the targets and finding a Grundy coloring.

▶ **Lemma 12.** *Let G and G' be two graphs as described in Lemma 8 and let H be constructed from G' by using the tree-filling operation. Then G has a clique of size k iff $\Gamma(H) \geq \lceil \log(k(m+1) + \binom{k}{2} + 2m) \rceil + 2\log n + 5$. Furthermore, $tw(H) \leq 4\binom{k}{2} + 8k + 17$.*

▶ **Theorem 13.** Grundy Coloring *parameterized by treewidth is W[1]-hard.*

4 FPT for pathwidth

In this section, we show that, in contrast to treewidth, Grundy Coloring is FPT parameterized by pathwidth. We achieve this by providing an upper bound on the Grundy number of any graph as a function of its pathwidth. Pipelining this with the algorithm of [76], we obtain a dependency on pathwidth alone. In order to obtain our bound, we rely on the following result on the performance ratio of the first-fit coloring algorithm on interval graphs.

▶ **Theorem 14** ([65]). *First-Fit is 8-competitive for online coloring interval graphs.*

In other words, interval graphs satisfy $\Gamma(G) \leq 8 \cdot \chi(G)$. Since on for any interval graph G we have $\chi(G) = pw(G) + 1$, we immediately obtain the following:

▶ **Corollary 15.** *For every interval graph G, $\Gamma(G) \leq 8 \cdot (pw(G) + 1)$.*

▶ **Lemma 16.** *For every graph G, $\Gamma(G) \leq 8 \cdot (pw(G) + 1)$.*

Proof. For a contradiction, suppose there exists G such that $\Gamma(G) > 8 \cdot (pw(G) + 1)$, and let $c : V(G) \to \{1, \ldots, \Gamma(G)\}$ be a Grundy coloring using $\Gamma(G)$ colors. In addition, let G have the smallest possible number of vertices, i.e., there is no G' satisfying those conditions with $|V(G')| < |V(G)|$. This implies that, for every optimal path decomposition of G, there is no bag B and vertices $u, v \in B$ such that $c(u) = c(v)$. Indeed, if such vertices exist, adding the edge uv to G and contracting uv yields a new graph G' such that $pw(G') \leq pw(G)$, $\Gamma(G') \geq \Gamma(G)$ and $|V(G')| < |V(G)|$, contradicting the assumption that G is smallest possible. In addition, for any u, v such that $c(u) \neq c(v)$ and $v \notin N(u)$, adding edge uv to G does not decrease the Grundy number of G since c remains a valid Grundy coloring of the new graph. In particular, since, as previously observed, vertices in any bag of an optimal path decomposition of G all have pairwise different colors, turning every bag of such a decomposition into a clique does not decrease the Grundy number of G. More precisely, this yields a graph G' such that $pw(G') = pw(G)$ and $\Gamma(G') = \Gamma(G)$, where G' is an interval graph. Applying Corollary 15 we obtain $\Gamma(G) \leq \Gamma(G') \leq 8 \cdot (pw(G') + 1)$, contradiction. ◀

Combining Lemma 16 with the $O^*(2^{O(tw\Gamma(G))})$ algorithm of [76], we have:

▶ **Theorem 17.** Grundy Coloring *can be solved in time $O^*(2^{O(pw(G)^2)})$.*

Finally, note that there exist interval graphs that satisfy $\Gamma(G) \geq r \cdot pw(G)$, for any $r < 5$ [53], therefore, the constant in Lemma 16 cannot be improved below 5.

5 NP-hardness for Constant Clique-width

In this section we prove that Grundy Coloring is NP-hard even for constant clique-width via a reduction from 3-SAT. We use a similar idea of adding supports as in Section 3, but supports now will be cliques instead of binomial trees. The support operation is defined as:

▶ **Definition 18.** *Given a graph $G = (V, E)$, a vertex $u \in V$ and a set of positive integers S, we define the **support** operation as follows: for each $i \in S$, we add to G a clique of size i (using new vertices) and we connect one arbitrary vertex of each such clique to u.*

When applying the support operation we will say that we support vertex u with set S and we will call the vertices introduced supporting vertices. Intuitively, the support operation ensures that the vertex u may have at least one neighbor with color i for each $i \in S$.

We are now ready to describe our construction. Suppose we are given a 3CNF formula ϕ with n variables $x_1, \ldots, x_n$ and m clauses $c_1, \ldots, c_m$. We assume without loss of generality that each clause contains exactly three variables. We construct a graph $G(\phi)$ as follows:

1. For each $i \in [n]$ we construct two vertices x_i^P, x_i^N and the edge (x_i^P, x_i^N).

2. For each $i \in [n]$ we support the vertices x_i^P, x_i^N with the set $[2i-2]$. (Note that x_1^P, x_1^N have empty support).
3. For each $i \in [n], j \in [m]$, if variable x_i appears in clause c_j then we construct a vertex $x_{i,j}$. Furthermore, if x_i appears positive in c_j, we connect $x_{i,j}$ to $x_{i'}^P$ for all $i' \in [n]$; otherwise we connect $x_{i,j}$ to $x_{i'}^N$ for all $i' \in [n]$.
4. For each $i \in [n], j \in [m]$ for which we constructed a vertex $x_{i,j}$ in the previous step, we support that vertex with the set $(\{2k \mid k \in [n]\} \cup \{2i-1, 2n+1, 2n+2\}) \setminus \{2i\}$.
5. For each $j \in [m]$ we construct a vertex c_j and connect to all (three) vertices $x_{i,j}$ already constructed. We support the vertex c_j with the set $[2n]$.
6. For each $j \in [m]$ we construct a vertex d_j and connect it to c_j. We support d_j with the set $[2n+3] \cup [2n+5, 2n+3+j]$.
7. We construct a vertex u and connect it to d_j for all $j \in [m]$. We support u with the set $[2n+4] \cup [2n+5+m, 10n+10m]$.

This completes the construction. Before we proceed, let us give some intuition. Observe that we have constructed two vertices x_i^P, x_i^N for each variable. The support of these vertices and the fact that they are adjacent, allow us to give them colors $\{2i-1, 2i\}$. The choice of which gets the higher color encodes an assignment to variable x_i. The vertices $x_{i,j}$ are now supported in such a way that they can "ignore" the values of all variables except x_i; for x_i, however, $x_{i,j}$ "prefers" to be connected to a vertex with color $2i$ (since $2i-1$ appears in the support of $x_{i,j}$, but $2i$ does not). Now, the idea is that c_j will be able to get color $2n+4$ if and only if one of its literal vertices $x_{i,j}$ was "satisfied" (has a neighbor with color $2i$). The rest of the construction checks if all clause vertices are satisfied in this way.

▶ **Lemma 19.** *If ϕ is satisfiable then $G(\phi)$ has a Grundy coloring with $10n+10m+1$ colors.*

▶ **Lemma 20.** *If $G(\phi)$ has a Grundy coloring with $10n+10m+1$ colors, then ϕ is satisfiable.*

▶ **Lemma 21.** *The graph $G(\phi)$ has constant clique-width.*

▶ **Theorem 22.** *Given graph $G=(V,E)$, k-Grundy Coloring is NP-hard even when the clique-width of the graph $cw(G)$ is a constant.*

6 FPT for modular-width

In this section we show that Grundy Coloring is FPT parameterized by modular width. Recall that $G=(V,E)$ has modular width w if V can be partitioned into at most w modules, such that each module is a singleton or induces a graph of modular width w. Neighborhood diversity is the restricted version of this measure where modules are required to be cliques or independent sets. We sketch the main ideas of the algorithm (a full proof is in the appendix).

The first step is to show that Grundy Coloring is FPT parameterized by neighborhood diversity. Similarly to the standard Coloring algorithm for this parameter [56], we observe that, without loss of generality, all modules can be assumed to be cliques, and hence any color class has one of 2^w possible types. We would like to use this to reduce the problem to an ILP with 2^w variables, but unlike Coloring, the ordering of color classes matters. We thus prove that the optimal solution can be assumed to have a "canonical" structure where each color type only appears in consecutive colors. We then extend the neighborhood diversity algorithm to modular width using the idea that we can calculate the Grundy number of each module separately, and then replace it with an appropriately-sized clique.

▶ **Theorem 23.** *Let $G=(V,E)$ be a graph of modular-width w. The Grundy number of G can be computed in time $2^{O(w2^w)}n^{O(1)}$.*

References

1. Pierre Aboulker, Édouard Bonnet, Eun Jung Kim, and Florian Sikora. Grundy coloring & friends, half-graphs, bicliques. In *37th Symposium on Theoretical Aspects of Computer Science, STACS 2020, March 10-13, 2020, Montpellier, France*, LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2020.
2. Eric Angel, Evripidis Bampis, Bruno Escoffier, and Michael Lampis. Parameterized power vertex cover. *Discrete Mathematics & Theoretical Computer Science*, 20(2), 2018. URL: `http://dmtcs.episciences.org/4873`.
3. Júlio Araújo and Cláudia Linhares Sales. On the grundy number of graphs with few p_4's. *Discrete Applied Mathematics*, 160(18):2514–2522, 2012. URL: `https://doi.org/10.1016/j.dam.2011.08.016`, `doi:10.1016/j.dam.2011.08.016`.
4. Ashwin Arulselvan, Ágnes Cseh, Martin Groß, David F. Manlove, and Jannik Matuschke. Matchings with lower quotas: Algorithms and complexity. *Algorithmica*, 80(1):185–208, 2018. URL: `https://doi.org/10.1007/s00453-016-0252-6`, `doi:10.1007/s00453-016-0252-6`.
5. Haris Aziz, Serge Gaspers, Edward J. Lee, and Kamran Najeebullah. Defender stackelberg game with inverse geodesic length as utility metric. In Elisabeth André, Sven Koenig, Mehdi Dastani, and Gita Sukthankar, editors, *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, pages 694–702. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018. URL: `http://dl.acm.org/citation.cfm?id=3237486`.
6. Rémy Belmonte, Tesshu Hanaka, Ioannis Katsikarelis, Eun Jung Kim, and Michael Lampis. New results on directed edge dominating set. In Igor Potapov, Paul G. Spirakis, and James Worrell, editors, *43rd International Symposium on Mathematical Foundations of Computer Science, MFCS 2018, August 27-31, 2018, Liverpool, UK*, volume 117 of *LIPIcs*, pages 67:1–67:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. URL: `https://doi.org/10.4230/LIPIcs.MFCS.2018.67`, `doi:10.4230/LIPIcs.MFCS.2018.67`.
7. Rémy Belmonte, Tesshu Hanaka, Ioannis Katsikarelis, Michael Lampis, Hirotaka Ono, and Yota Otachi. Parameterized complexity of safe set. In Pinar Heggernes, editor, *Algorithms and Complexity - 11th International Conference, CIAC 2019, Rome, Italy, May 27-29, 2019, Proceedings*, volume 11485 of *Lecture Notes in Computer Science*, pages 38–49. Springer, 2019. URL: `https://doi.org/10.1007/978-3-030-17402-6_4`, `doi:10.1007/978-3-030-17402-6\_4`.
8. Rémy Belmonte, Michael Lampis, and Valia Mitsou. Parameterized (approximate) defective coloring. In Rolf Niedermeier and Brigitte Vallée, editors, *35th Symposium on Theoretical Aspects of Computer Science, STACS 2018, February 28 to March 3, 2018, Caen, France*, volume 96 of *LIPIcs*, pages 10:1–10:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. URL: `https://doi.org/10.4230/LIPIcs.STACS.2018.10`, `doi:10.4230/LIPIcs.STACS.2018.10`.
9. Oren Ben-Zwi, Danny Hermelin, Daniel Lokshtanov, and Ilan Newman. Treewidth governs the complexity of target set selection. *Discrete Optimization*, 8(1):87–96, 2011. URL: `https://doi.org/10.1016/j.disopt.2010.09.007`, `doi:10.1016/j.disopt.2010.09.007`.
10. Matthias Bentert, Klaus Heeger, and Dušan Knop. Length-bounded cuts: Proper interval graphs and structural parameters, 2019. `arXiv:1910.03409`.
11. Nadja Betzler, Robert Bredereck, Rolf Niedermeier, and Johannes Uhlmann. On bounded-degree vertex deletion parameterized by treewidth. *Discrete Applied Mathematics*, 160(1-2):53–60, 2012. URL: `https://doi.org/10.1016/j.dam.2011.08.013`, `doi:10.1016/j.dam.2011.08.013`.
12. Marzio De Biasi and Juho Lauri. On the complexity of restoring corrupted colorings. *J. Comb. Optim.*, 37(4):1150–1169, 2019. URL: `https://doi.org/10.1007/s10878-018-0342-2`, `doi:10.1007/s10878-018-0342-2`.
13. Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1-2):1–45, 1998.

14 Édouard Bonnet, Florent Foucaud, Eun Jung Kim, and Florian Sikora. Complexity of grundy coloring and its variants. *Discrete Applied Mathematics*, 243:99–114, 2018. URL: https://doi.org/10.1016/j.dam.2017.12.022, doi:10.1016/j.dam.2017.12.022.

15 Édouard Bonnet, Michael Lampis, and Vangelis Th. Paschos. Time-approximation trade-offs for inapproximable problems. *J. Comput. Syst. Sci.*, 92:171–180, 2018. URL: https://doi.org/10.1016/j.jcss.2017.09.009, doi:10.1016/j.jcss.2017.09.009.

16 Édouard Bonnet and Nidhi Purohit. Metric dimension parameterized by treewidth. *CoRR*, abs/1907.08093, 2019. URL: http://arxiv.org/abs/1907.08093, arXiv:1907.08093.

17 Hajo Broersma, Petr A. Golovach, and Viresh Patel. Tight complexity bounds for FPT subgraph problems parameterized by the clique-width. *Theor. Comput. Sci.*, 485:69–84, 2013. URL: https://doi.org/10.1016/j.tcs.2013.03.008, doi:10.1016/j.tcs.2013.03.008.

18 Claude A. Christen and Stanley M. Selkow. Some perfect coloring properties of graphs. *J. Comb. Theory, Ser. B*, 27(1):49–59, 1979. URL: https://doi.org/10.1016/0095-8956(79)90067-4, doi:10.1016/0095-8956(79)90067-4.

19 Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990. URL: https://doi.org/10.1016/0890-5401(90)90043-H, doi:10.1016/0890-5401(90)90043-H.

20 Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000. URL: https://doi.org/10.1007/s002249910009, doi:10.1007/s002249910009.

21 Radu Curticapean and Dániel Marx. Tight conditional lower bounds for counting perfect matchings on graphs of bounded treewidth, cliquewidth, and genus. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1650–1669. SIAM, 2016. URL: https://doi.org/10.1137/1.9781611974331.ch113, doi:10.1137/1.9781611974331.ch113.

22 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms.* Springer, 2015. URL: https://doi.org/10.1007/978-3-319-21275-3, doi:10.1007/978-3-319-21275-3.

23 Guilherme de C. M. Gomes, Carlos V. G. C. Lima, and Vinícius Fernandes dos Santos. Parameterized complexity of equitable coloring. *Discrete Mathematics & Theoretical Computer Science*, 21(1), 2019. URL: http://dmtcs.episciences.org/5464.

24 Holger Dell, Eun Jung Kim, Michael Lampis, Valia Mitsou, and Tobias Mömke. Complexity and approximability of parameterized max-csps. *Algorithmica*, 79(1):230–250, 2017. URL: https://doi.org/10.1007/s00453-017-0310-8, doi:10.1007/s00453-017-0310-8.

25 Michael Dom, Daniel Lokshtanov, Saket Saurabh, and Yngve Villanger. Capacitated domination and covering: A parameterized perspective. In Martin Grohe and Rolf Niedermeier, editors, *Parameterized and Exact Computation, Third International Workshop, IWPEC 2008, Victoria, Canada, May 14-16, 2008. Proceedings*, volume 5018 of *Lecture Notes in Computer Science*, pages 78–90. Springer, 2008. URL: https://doi.org/10.1007/978-3-540-79723-4_9, doi:10.1007/978-3-540-79723-4_9.

26 Pavel Dvorák and Dusan Knop. Parameterized complexity of length-bounded cuts and multicuts. *Algorithmica*, 80(12):3597–3617, 2018. URL: https://doi.org/10.1007/s00453-018-0408-7, doi:10.1007/s00453-018-0408-7.

27 Eduard Eiben, Robert Ganian, K. Kangas, and Sebastian Ordyniak. Counting linear extensions: Parameterizations by treewidth. *Algorithmica*, 81(4):1657–1683, 2019. URL: https://doi.org/10.1007/s00453-018-0496-4, doi:10.1007/s00453-018-0496-4.

28 Eduard Eiben, Robert Ganian, and Sebastian Ordyniak. A structural approach to activity selection. In Jérôme Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 203–209. ijcai.org, 2018. URL: https://doi.org/10.24963/ijcai.2018/28, doi:10.24963/ijcai.2018/28.

29 Rosa Enciso, Michael R. Fellows, Jiong Guo, Iyad A. Kanj, Frances A. Rosamond, and Ondrej Suchý. What makes equitable connected partition easy. In Jianer Chen and Fedor V. Fomin, editors, *Parameterized and Exact Computation, 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, September 10-11, 2009, Revised Selected Papers*, volume 5917 of *Lecture Notes in Computer Science*, pages 122–133. Springer, 2009. URL: `https://doi.org/10.1007/978-3-642-11269-0_10`, `doi:10.1007/978-3-642-11269-0\_10`.

30 Alina Ene, Matthias Mnich, Marcin Pilipczuk, and Andrej Risteski. On routing disjoint paths in bounded treewidth graphs. In *SWAT*, volume 53 of *LIPIcs*, pages 15:1–15:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

31 Paul Erdös, Stephen T. Hedetniemi, Renu C. Laskar, and Geert C. E. Prins. On the equality of the partial grundy and upper ochromatic numbers of graphs. *Discrete Mathematics*, 272(1):53–64, 2003. URL: `https://doi.org/10.1016/S0012-365X(03)00184-5`, `doi:10.1016/S0012-365X(03)00184-5`.

32 Michael R. Fellows, Fedor V. Fomin, Daniel Lokshtanov, Frances A. Rosamond, Saket Saurabh, Stefan Szeider, and Carsten Thomassen. On the complexity of some colorful problems parameterized by treewidth. *Inf. Comput.*, 209(2):143–153, 2011. URL: `https://doi.org/10.1016/j.ic.2010.11.026`, `doi:10.1016/j.ic.2010.11.026`.

33 Jirí Fiala, Petr A. Golovach, and Jan Kratochvíl. Parameterized complexity of coloring problems: Treewidth versus vertex cover. *Theor. Comput. Sci.*, 412(23):2513–2523, 2011. URL: `https://doi.org/10.1016/j.tcs.2010.10.043`, `doi:10.1016/j.tcs.2010.10.043`.

34 Krzysztof Fleszar, Matthias Mnich, and Joachim Spoerhase. New algorithms for maximum disjoint paths based on tree-likeness. In *ESA*, volume 57 of *LIPIcs*, pages 42:1–42:17. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

35 Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, and Saket Saurabh. Algorithmic lower bounds for problems parameterized with clique-width. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 493–502. SIAM, 2010. URL: `https://doi.org/10.1137/1.9781611973075.42`, `doi:10.1137/1.9781611973075.42`.

36 Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, and Saket Saurabh. Almost optimal lower bounds for problems parameterized by clique-width. *SIAM J. Comput.*, 43(5):1541–1563, 2014. URL: `https://doi.org/10.1137/130910932`, `doi:10.1137/130910932`.

37 Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. Clique-width III: hamiltonian cycle and the odd case of graph coloring. *ACM Trans. Algorithms*, 15(1):9:1–9:27, 2019. URL: `https://doi.org/10.1145/3280824`, `doi:10.1145/3280824`.

38 Jakub Gajarský, Michael Lampis, and Sebastian Ordyniak. Parameterized algorithms for modular-width. In Gregory Z. Gutin and Stefan Szeider, editors, *Parameterized and Exact Computation - 8th International Symposium, IPEC 2013, Sophia Antipolis, France, September 4-6, 2013, Revised Selected Papers*, volume 8246 of *Lecture Notes in Computer Science*, pages 163–176. Springer, 2013. URL: `https://doi.org/10.1007/978-3-319-03898-8_15`, `doi:10.1007/978-3-319-03898-8\_15`.

39 Robert Ganian, Fabian Klute, and Sebastian Ordyniak. On structural parameterizations of the bounded-degree vertex deletion problem. In Rolf Niedermeier and Brigitte Vallée, editors, *35th Symposium on Theoretical Aspects of Computer Science, STACS 2018, February 28 to March 3, 2018, Caen, France*, volume 96 of *LIPIcs*, pages 33:1–33:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. URL: `https://doi.org/10.4230/LIPIcs.STACS.2018.33`, `doi:10.4230/LIPIcs.STACS.2018.33`.

40 Robert Ganian and Sebastian Ordyniak. The complexity landscape of decompositional parameters for ILP. *Artif. Intell.*, 257:61–71, 2018. URL: `https://doi.org/10.1016/j.artint.2017.12.006`, `doi:10.1016/j.artint.2017.12.006`.

41 Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18(1):3–20, 1997. URL: `https://doi.org/10.1007/BF02523685`, `doi:10.1007/BF02523685`.

42 Elisabeth Gassner. The steiner forest problem revisited. *J. Discrete Algorithms*, 8(2):154–163, 2010. URL: https://doi.org/10.1016/j.jda.2009.05.002, doi:10.1016/j.jda.2009.05.002.

43 Gregory Z. Gutin, Mark Jones, and Magnus Wahlström. The mixed chinese postman problem parameterized by pathwidth and treedepth. *SIAM J. Discrete Math.*, 30(4):2177–2205, 2016. URL: https://doi.org/10.1137/15M1034337, doi:10.1137/15M1034337.

44 András Gyárfás and Jenö Lehel. On-line and first fit colorings of graphs. *Journal of Graph Theory*, 12(2):217–227, 1988. URL: https://doi.org/10.1002/jgt.3190120212, doi:10.1002/jgt.3190120212.

45 Tesshu Hanaka, Ioannis Katsikarelis, Michael Lampis, Yota Otachi, and Florian Sikora. Parameterized orientable deletion. In David Eppstein, editor, *16th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2018, June 18-20, 2018, Malmö, Sweden*, volume 101 of *LIPIcs*, pages 24:1–24:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. URL: https://doi.org/10.4230/LIPIcs.SWAT.2018.24, doi:10.4230/LIPIcs.SWAT.2018.24.

46 Frédéric Havet and Leonardo Sampaio. On the grundy and *b*-chromatic numbers of a graph. *Algorithmica*, 65(4):885–899, 2013. URL: https://doi.org/10.1007/s00453-011-9604-4, doi:10.1007/s00453-011-9604-4.

47 Ramin Javadi and Amir Nikabadi. On the parameterized complexity of sparsest cut and small-set expansion problems, 2019. arXiv:1910.12353.

48 Ioannis Katsikarelis, Michael Lampis, and Vangelis Th. Paschos. Structurally parameterized d-scattered set. In Andreas Brandstädt, Ekkehard Köhler, and Klaus Meer, editors, *Graph-Theoretic Concepts in Computer Science - 44th International Workshop, WG 2018, Cottbus, Germany, June 27-29, 2018, Proceedings*, volume 11159 of *Lecture Notes in Computer Science*, pages 292–305. Springer, 2018. URL: https://doi.org/10.1007/978-3-030-00256-5_24, doi:10.1007/978-3-030-00256-5_24.

49 Ioannis Katsikarelis, Michael Lampis, and Vangelis Th. Paschos. Structural parameters, tight bounds, and approximation for (k, r)-center. *Discrete Applied Mathematics*, 264:90–117, 2019. URL: https://doi.org/10.1016/j.dam.2018.11.002, doi:10.1016/j.dam.2018.11.002.

50 Chamanvir Kaur and Neeldhara Misra. On the parameterized complexity of spanning trees with small vertex covers. In *CALDAM*, volume 12016 of *Lecture Notes in Computer Science*, pages 427–438. Springer, 2020.

51 Leon Kellerhals and Tomohiro Koana. Parameterized complexity of geodetic set, 2020. arXiv:2001.03098.

52 Hal A. Kierstead and Karin Rebecca Saoub. First-fit coloring of bounded tolerance graphs. *Discrete Applied Mathematics*, 159(7):605–611, 2011. URL: https://doi.org/10.1016/j.dam.2010.05.002, doi:10.1016/j.dam.2010.05.002.

53 Hal A. Kierstead, David A. Smith, and William T. Trotter. First-fit coloring on interval graphs has performance ratio at least 5. *Eur. J. Comb.*, 51:236–254, 2016. URL: https://doi.org/10.1016/j.ejc.2015.05.015, doi:10.1016/j.ejc.2015.05.015.

54 Dusan Knop, Tomás Masarík, and Tomás Toufar. Parameterized complexity of fair vertex evaluation problems. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany.*, volume 138 of *LIPIcs*, pages 33:1–33:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. URL: https://doi.org/10.4230/LIPIcs.MFCS.2019.33, doi:10.4230/LIPIcs.MFCS.2019.33.

55 Guy Kortsarz. A lower bound for approximating grundy numbering. *Discrete Mathematics & Theoretical Computer Science*, 9(1), 2007. URL: http://dmtcs.episciences.org/391.

56 Michael Lampis. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, 64(1):19–37, 2012. URL: https://doi.org/10.1007/s00453-011-9554-x, doi:10.1007/s00453-011-9554-x.

57 Michael Lampis. Parameterized maximum path coloring. *Theor. Comput. Sci.*, 511:42–53, 2013. URL: https://doi.org/10.1016/j.tcs.2013.01.012, doi:10.1016/j.tcs.2013.01.012.

58 Michael Lampis. Model checking lower bounds for simple graphs. *Logical Methods in Computer Science*, 10(1), 2014. URL: `https://doi.org/10.2168/LMCS-10(1:18)2014`, `doi:10.2168/LMCS-10(1:18)2014`.

59 Michael Lampis and Valia Mitsou. Treewidth with a quantifier alternation revisited. In Daniel Lokshtanov and Naomi Nishimura, editors, *12th International Symposium on Parameterized and Exact Computation, IPEC 2017, September 6-8, 2017, Vienna, Austria*, volume 89 of *LIPIcs*, pages 26:1–26:12. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. URL: `https://doi.org/10.4230/LIPIcs.IPEC.2017.26`, `doi:10.4230/LIPIcs.IPEC.2017.26`.

60 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Known algorithms on graphs of bounded treewidth are probably optimal. *ACM Trans. Algorithms*, 14(2):13:1–13:30, 2018. URL: `https://doi.org/10.1145/3170442`, `doi:10.1145/3170442`.

61 Dániel Marx and Michal Pilipczuk. Everything you always wanted to know about the parameterized complexity of subgraph isomorphism (but were afraid to ask). In Ernst W. Mayr and Natacha Portier, editors, *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014), STACS 2014, March 5-8, 2014, Lyon, France*, volume 25 of *LIPIcs*, pages 542–553. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014. URL: `https://doi.org/10.4230/LIPIcs.STACS.2014.542`, `doi:10.4230/LIPIcs.STACS.2014.542`.

62 Kitty Meeks and Alexander Scott. The parameterised complexity of list problems on graphs of bounded treewidth. *Inf. Comput.*, 251:91–103, 2016. URL: `https://doi.org/10.1016/j.ic.2016.08.001`, `doi:10.1016/j.ic.2016.08.001`.

63 Kitty Meeks and Fiona Skerman. The parameterised complexity of computing the maximum modularity of a graph. In *IPEC*, volume 115 of *LIPIcs*, pages 9:1–9:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

64 Burkhard Monien. The bandwidth minimization problem for caterpillars with hair length 3 is np-complete. *SIAM Journal on Algebraic Discrete Methods*, 7(4):505–512, 1986.

65 N. S. Narayanaswamy and R. Subhash Babu. A note on first-fit coloring of interval graphs. *Order*, 25(1):49–53, 2008. URL: `https://doi.org/10.1007/s11083-008-9076-6`, `doi:10.1007/s11083-008-9076-6`.

66 Jaroslav Nesetril and Patrice Ossona de Mendez. Tree-depth, subgraph coloring and homomorphism bounds. *Eur. J. Comb.*, 27(6):1022–1041, 2006. URL: `https://doi.org/10.1016/j.ejc.2005.01.010`, `doi:10.1016/j.ejc.2005.01.010`.

67 André Nichterlein, Rolf Niedermeier, Johannes Uhlmann, and Mathias Weller. On tractable cases of target set selection. *Social Netw. Analys. Mining*, 3(2):233–256, 2013. URL: `https://doi.org/10.1007/s13278-012-0067-7`, `doi:10.1007/s13278-012-0067-7`.

68 Sebastian Ordyniak, Daniël Paulusma, and Stefan Szeider. Satisfiability of acyclic and almost acyclic CNF formulas. *Theor. Comput. Sci.*, 481:85–99, 2013. URL: `https://doi.org/10.1016/j.tcs.2012.12.039`, `doi:10.1016/j.tcs.2012.12.039`.

69 Igor Razgon. On obdds for cnfs of bounded treewidth. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*. AAAI Press, 2014. URL: `http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/7982`.

70 Marko Samer and Stefan Szeider. Constraint satisfaction with bounded treewidth revisited. *J. Comput. Syst. Sci.*, 76(2):103–114, 2010. URL: `https://doi.org/10.1016/j.jcss.2009.04.003`, `doi:10.1016/j.jcss.2009.04.003`.

71 Marko Samer and Stefan Szeider. Tractable cases of the extended global cardinality constraint. *Constraints*, 16(1):1–24, 2011. URL: `https://doi.org/10.1007/s10601-009-9079-y`, `doi:10.1007/s10601-009-9079-y`.

72 Stefan Szeider. Not so easy problems for tree decomposable graphs. *CoRR*, abs/1107.1177, 2011. URL: `http://arxiv.org/abs/1107.1177`, `arXiv:1107.1177`.

73 Zixing Tang, Baoyindureng Wu, Lin Hu, and Manouchehr Zaker. More bounds for the grundy number of graphs. *J. Comb. Optim.*, 33(2):580–589, 2017. URL: `https://doi.org/10.1007/s10878-015-9981-8`, `doi:10.1007/s10878-015-9981-8`.

74 Marc Tedder, Derek G. Corneil, Michel Habib, and Christophe Paul. Simpler linear-time modular decomposition via recursive factorizing permutations. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part I: Tack A: Algorithms, Automata, Complexity, and Games*, volume 5125 of *Lecture Notes in Computer Science*, pages 634–645. Springer, 2008. URL: `https://doi.org/10.1007/978-3-540-70575-8_52`, `doi:10.1007/978-3-540-70575-8\_52`.

75 Jan Arne Telle and Andrzej Proskurowski. Algorithms for vertex partitioning problems on partial k-trees. *SIAM J. Discrete Math.*, 10(4):529–550, 1997. URL: `https://doi.org/10.1137/S0895480194275825`, `doi:10.1137/S0895480194275825`.

76 Jan Arne Telle and Andrzej Proskurowski. Algorithms for vertex partitioning problems on partial k-trees. *SIAM J. Discrete Math.*, 10(4):529–550, 1997. URL: `https://doi.org/10.1137/S0895480194275825`, `doi:10.1137/S0895480194275825`.

77 Johan M. M. van Rooij, Hans L. Bodlaender, and Peter Rossmanith. Dynamic programming on tree decompositions using generalised fast subset convolution. In Amos Fiat and Peter Sanders, editors, *Algorithms - ESA 2009, 17th Annual European Symposium, Copenhagen, Denmark, September 7-9, 2009. Proceedings*, volume 5757 of *Lecture Notes in Computer Science*, pages 566–577. Springer, 2009. URL: `https://doi.org/10.1007/978-3-642-04128-0_51`, `doi:10.1007/978-3-642-04128-0\_51`.

78 Manouchehr Zaker. Grundy chromatic number of the complement of bipartite graphs. *Australasian J. Combinatorics*, 31:325–330, 2005. URL: `http://ajc.maths.uq.edu.au/pdf/31/ajc_v31_p325.pdf`.

79 Manouchehr Zaker. Results on the grundy chromatic number of graphs. *Discrete Mathematics*, 306(23):3166–3173, 2006. URL: `https://doi.org/10.1016/j.disc.2005.06.044`, `doi:10.1016/j.disc.2005.06.044`.

80 Manouchehr Zaker. Inequalities for the grundy chromatic number of graphs. *Discrete Applied Mathematics*, 155(18):2567–2572, 2007. URL: `https://doi.org/10.1016/j.dam.2007.07.002`, `doi:10.1016/j.dam.2007.07.002`.

A List of known problems which are W-hard for treewidth and for pathwidth

Here we give a list of problems found in the literature which are known to be W[1]-hard by treewidth. After reviewing the relevant works we have verified that all of the following problems are in fact shown to be W[1]-hard parameterized by pathwidth (and in many case by feedback vertex set and tree-depth), even if this is not explicitly claimed.

- Precoloring Extension and Equitable Coloring are shown to be W[1]-hard for both tree-depth and feedback vertex set in [32] (though the result is claimed only for treewidth). This is important, because Equitable Coloring often serves as a starting point for reductions to other problems. A second hardness proof for this problem was recently given in [23]. These two problems are FPT by vertex cover [33].
- Capacitated Dominating Set and Capacitated Vertex Cover are W[1]-hard for both tree-depth and feedback vertex set [25] (though again the result is claimed for treewidth).
- Min Maximum Out-degree on weighted graphs is W[1]-hard by tree-depth and feedback vertex set [72].
- General Factors is W[1]-hard by tree-depth and feedback vertex set [71].
- Target Set Selection is W[1]-hard by tree-depth and feedback vertex set [9] but FPT for vertex cover [67].
- Bounded Degree Deletion is W[1]-hard by tree-depth and feedback vertex set, but FPT for vertex cover [11, 39].
- Fair Vertex Cover is W[1]-hard by tree-depth and feedback vertex set [54].
- Fixing Corrupted Colorings is W[1]-hard by tree-depth and feedback vertex set [12] (reduction from Precoloring Extension).
- Max Node Disjoint Paths is W[1]-hard by tree-depth and feedback vertex set [30, 34].
- Defective Coloring is W[1]-hard by tree-depth and feedback vertex set [8].
- Power Vertex Cover is W[1]-hard by tree-depth but open for feedback vertex set [2].
- Majority CSP is W[1]-hard parameterized by the tree-depth of the incidence graph [24].
- List Hamiltonian Path is W[1]-hard for pathwidth [62].
- L(1,1)-Coloring is W[1]-hard for pathwidth, FPT for vertex cover [33].
- Counting Linear Extensions of a poset is W[1]-hard (under Turing reductions) for pathwidth [27].
- Equitable Connected Partition is W[1]-hard by pathwidth and feedback vertex set, FPT by vertex cover [29].
- Safe Set is W[1]-hard parameterized by pathwidth, FPT by vertex cover [7].
- Matching with Lower Quotas is W[1]-hard parameterized by pathwidth [4].
- Subgraph Isomorphism is W[1]-hard parameterized by the pathwidth of both graphs [61].
- Metric Dimension is W[1]-hard by pathwidth [16].
- Simple Comprehensive Activity Selection is W[1]-hard by pathwidth [28].
- Defensive Stackelberg Game for IGL is W[1]-hard by pathwidth (reduction from Equitable Coloring) [5].
- Directed (p,q)-Edge Dominating Set is W[1]-hard parameterized by pathwidth [6].
- Maximum Path Coloring is W[1]-hard for pathwidth [57].
- Unweighted k-Sparsest Cut is W[1]-hard parameterized by the three combined parameters tree-depth, feedback vertex set, and k [47].

- Graph Modularity is W[1]-hard parameterized by pathwidth plus feedback vertex set [63].

B W[1]-hardness for treewidth – Missing Proofs

Proposition 4. By induction in $i-t$. For $i-t=1$, T_i indeed contains one T_{i-1} that does not contain the root r_i. Let it be true that T_{i-1} contains 2^{i-t-2} subtrees T_t. Then T_i contains two trees T_{i-1} each of which contains 2^{i-t-2} T_j, thus 2^{i-t-1} in total. ◀

Proposition 5. The first part is trivial since in any graph G with maximum degree Δ we have $\Gamma(G) \leq \Delta+1$. In this case $\Gamma(T_i) \leq (i-1)+1=i$. For the second part, we first prove that there is a Grundy coloring which assigns color i to the root. This can be proven by strong induction: if for all $k<i$, there is a Grundy coloring which assigns color k to r_k for all $1 \leq k \leq i-1$, then under this coloring, r_i has at least one neighbor receiving color k for all $1 \leq k \leq i-1$, so it has to receive color i. To assign to the root a color $j<i$ we use the fact that (by inductive hypothesis) there is a coloring that assigns color $j-1$ to r_j, so in this coloring the root r_i will take color j. ◀

Lemma 9. We will use the equivalent definition of pathwidth as a node-searching game, where the robber is eager and invisible and the cops are placed on nodes [13]. We will use $\binom{k}{2}+2k+4$ cops to clean G'' as follows: We place $\binom{k}{2}$ cops on the edge checkers. Then, starting from $j=0$, we place $2k$ cops on the propagators $p_{i,0}, p_{i,1}$ for $i=1,\dots,k$, plus 2 cops on the edge selection vertices $w_{j,x}$, $w_{j,y}$ that correspond to edge j. We use the two additional cops to clean line by line the gadgets $S_{i,j}$. We then use one of these cops to clear $w'_{j,x}, w'_{j,y}$. We continue then to the next column $j=2$ by removing the k cops from the propagators $p_{i,1}$ and placing them to $p_{i,3}$. We continue for $j=3,\dots m-1$ until the whole graph has been cleaned. ◀

Lemma 12. We note that the number of vertices with targets in our construction is $m' = k(m+1)+\binom{k}{2}+2m$ (the propagators, edge selection checkers, and edge-checkers). From Lemma 8, it only suffices to show that $\Gamma(H) \geq \lceil \log m' \rceil + 2\log n + 5$ iff the vertices with targets achieve color $t = 2\log n + 4$.

For the forward direction, once vertices with targets get the desirable colors, the rest of the binomial tree of the tree-filling operation can be colored optimally, starting from its leaves all the way up to its roots, which will get color $i = \lceil \log m' \rceil + 2\log n + 5$.

For the converse direction, observe that the only vertices having degree higher than $2\log n + 4$ are the edge-checkers and the vertices of the binomial tree $H \setminus G'$. However, the edge-checkers connect to only one vertex of degree higher than $2\log n+4$, that in the binomial tree. Thus no vertex of G' can ever get a color higher than $2\log n+6$ and the only way that $\Gamma(H) \geq \lceil \log m' \rceil + 2\log n + 5$ is if the root of the binomial tree of the tree-filling operation (the only vertex of high enough degree) receives color $\lceil \log m' \rceil + 2\log n + 5$. For that to happen, all the support-trees of this tree should be colored optimally, which proves that the vertices with targets $2\log n+4$ having substituted support trees $T_{2\log n+4}$ should achieve their targets.

In terms of the treewidth of H we have the following: Lemma 9 says that G' once we remove all the supporting trees has pathwidth at most $\binom{k}{2}+2k+3$. Applying Lemma 11 we get that H where we have ignored the tree-supports from G' has treewidth at most $4\left(\binom{k}{2}+2k+3\right)+5$. Adding back the tree-supports does not increase its treewidth. ◀

C NP-hardness for clique-width – missing proofs

Lemma 19. Consider a satisfying assignment of ϕ. We first produce a coloring of the vertices x_i^P, x_i^N as follows: if x_i is set to True, then x_i^P is colored $2i$ and x_i^N is colored $2i-1$; otherwise x_i^P is colored $2i-1$ and x_i^N is colored $2i$. Before proceeding, let us also color the supporting vertices of x_i^P, x_i^N: each such vertex belongs to a clique which contains only one vertex with a neighbor outside the clique. For each such clique of size ℓ, we color all vertices of the clique which have no outside neighbors with colors from $[\ell-1]$ and use color ℓ for the remaining vertex. Note that the coloring we have produced so far is a valid Grundy coloring, since each vertex x_i^P, x_i^N has for each $c \in [2i-2]$ a neighbor with color c among its supporting vertices, allowing us to use colors $\{2i-1, 2i\}$ for x_i^P, x_i^N. In the remainder, we will use similar such colorings for all supporting cliques. We will only stress the color given to the vertex of the clique that has an outside neighbor, respecting the condition that this color is not larger than the size of the clique. Note that it is not a problem if this color is strictly smaller than the size of the clique, as we are free to give higher colors to internal vertices.

Consider now a clause c_j for some $j \in [m]$. Suppose that this clause contains the three variables $x_{i_1}, x_{i_2}, x_{i_3}$. Because we started with a satisfying assignment, at least one of these variables has a value that satisfies the clause, without loss of generality x_{i_3}. We therefore color $x_{i_1}, x_{i_2}, x_{i_3}$ with colors $2n+1, 2n+2, 2n+3$ respectively and we color c_j with color $2n+4$. We now need to show that we can appropriately color the supporting vertices to make this a valid Grundy coloring.

Recall that the vertex x_{i_3} has support $\{2, 4, \dots, 2n\} \setminus \{2i_3\} \cup \{2i_3 - 1, 2n+1, 2n+2\}$. For each $i' \neq i_3$ we observe that x_{i_3} is connected to a vertex (either $x_{i_3}^P$ or $x_{i_3}^N$) which has a color in $\{2i'-1, 2i'\}$, we are therefore missing the other color from this set. We consider the clique of size $2i'$ supporting $x_{i_3,j}$: we assign this missing color to the vertex of this clique that is adjacent to $x_{i_3,j}$. Note that the clique is large enough to color its remaining vertices with lower colors in order to make this a valid Grundy coloring. For i_3, we observe that, since x_{i_3} satisfies the clause, the vertex $x_{i_3,j}$ has a neighbor (either $x_{i_3}^P$ or $x_{i_3}^N$) which has received color $2i_3$; we use color $2i_3 - 1$ in the support clique of the same size. Similarly, we use colors $2n+1, 2n+2$ in the support cliques of the same sizes, and x_{i_3} has neighbors with colors covering all of $[2n+2]$.

For the vertex $x_{i_2,j}$ we proceed in a similar way. For $i' < i_2$ we give the support vertex from the clique of size $2i'$ the color from $\{2i'-1, 2i'\}$ which does not already appear in the neighborhood of $x_{i_2,j}$. For $i' \in [i_2, n-1]$ we take the vertex from the clique of size $2i'+2$ and give it the color of $\{2i'-1, 2i'\}$ which does not yet appear in the neighborhood of $x_{i_2,j}$. In this way we cover all colors in $[2n-2]$. We now observe that $x_{i_2,j}$ has a neighbor with color in $\{2n-1, 2n\}$ (either x_n^P or x_n^N); together with the support vertices from the cliques of sizes $2n+1, 2n+2$ this allows us to cover the colors $[2n-1, 2n+1]$. We use a similar procedure to cover the colors $[2n]$ in the neighborhood of $x_{i_1,j}$. Now, the $2n$ support vertices in the neighborhood of c_j, together with $x_{i_1,j}, x_{i_2,j}, x_{i_3,j}$ allow us to give that vertex color $2n+4$.

We now give each vertex d_j, for $j \in [m]$ color $2n+j+4$. This can be extended to a valid coloring, because d_j is adjacent to c_j, which has color $2n+4$, and the support of d_j is $[2n+j+3] \setminus \{2n+4\}$.

Finally, we give u color $10n+10m+1$. Its support is $[10n+10m] \setminus [2n+5, 2n+m+4]$. However, u is adjacent to all vertices d_j, whose colors cover the set $\{2n+4+j \mid j \in [m]\}$. ◀

Lemma 20. Consider a Grundy coloring of $G(\phi)$. We first assume without loss of generality that we consider a minimal induced subgraph of G for which the coloring remains valid, that is, deleting any vertex will either reduce the number of colors or invalidate the coloring. In particular, this means there is a unique vertex with color $10n + 10m + 1$. This vertex must have degree at least $10n + 10m$. However, there are only two such vertices in our graph: u and its support neighbor vertex in the clique of size $10n + 10m$. If the latter vertex has color $10n + 10m + 1$, we can infer that u has color $10n + 10m$: this color cannot appear in the clique because all its internal vertices have degree $10n + 10m - 1$, and one of their neighbors has a higher color. We observe now that exchanging the colors of u and its neighbor produces another valid coloring. We therefore assume without loss of generality that u has color $10n + 10m + 1$.

We now observe that in each supporting clique of u of size i the maximum color used is i (since u has the largest color in the graph). Similarly, the largest color that can be assigned to d_j is $2n+j+4$, because d_j has degree $2n+j+4$, but one of its neighbors (u) has a higher color. We conclude that the only way for the $10n + 10m$ neighbors of u to cover all colors in $[10n + 10m]$ is for each support clique of size i to use color i and for each d_j to be given color $2n + j + 4$.

Suppose now that d_j was given color $2n + j + 4$. This implies that the largest color that c_j may have received is $2n + 4$, since its degree is $2n + 4$, but d_j received a higher color. We conclude again that for the neighbors of d_j to cover $[2n + j + 3]$ it must be the case that each supporting clique used its maximum possible color and c_j received color $2n + 4$.

Suppose now that a vertex c_j received color $2n + 4$. Since d_j received a higher color, the remaining $2n + 3$ neighbors of this vertex must cover $[2n + 3]$. In particular, since the support vertices have colors in $[2n]$, its three remaining neighbors, say $x_{i_1,j}, x_{i_2,j}, x_{i_3,j}$ must have colors covering $[2n+1, 2n+3]$. Therefore, all vertices $x_{i,j}$ have colors in $[2n+1, 2n+3]$.

Consider now two vertices x_i^P, x_i^N, for some $i \in [n]$. We claim that the vertex which among these two has the lower color, has color at most $2i - 1$. To see this observe that this vertex may have at most $2i - 2$ neighbors from the support vertices that have lower colors and these must use colors in $[2i - 2]$ because of their degrees. Its neighbors of the form $x_{i,j}$ have color at least $2n + 1 > 2i - 1$, and its neighbor in $\{x_i^P, x_i^N\}$ has a higher color. Therefore, the smaller of the two colors used for $\{x_i^P, x_i^N\}$ is at most $2i - 1$ and by similar reasoning the higher of the two colors used for this set is at most $2i$. We now obtain an assignment for ϕ by setting x_i to True if x_i^P has a higher color than x_i^N and False otherwise (this is well-defined, since x_i^P, x_i^N are adjacent).

Let us argue why this is a satisfying assignment. Take a clause vertex c_j. As argued, one of its neighbors, say $x_{i_3,j}$ has color $2n + 3$. The degree of $x_{i_3,j}$, excluding c_j which has a higher color, is $2n + 2$, meaning that its neighbors must exactly cover $[2n + 2]$ with their colors. Since vertices x_i^P, x_i^N have color at most $2i$, the colors $[2n + 1, 2n + 2]$ must come from the support cliques of the same sizes. Now, for each $i \in [n]$ the vertex $x_{i_3,j}$ has exactly two neighbors which may have received colors in $\{2i - 1, 2i\}$. This can be seen by induction on i: first, for $i = n$ this is true, since we only have the support clique of size $2n$ and the neighbor in $\{x_n^P, x_n^N\}$. Proceeding in the same way we conclude the claim for smaller values of i. The key observation is now that the clique of size $2i_3 - 1$ cannot give us color $2i_3$, therefore this color must come from $\{x_{i_3}^N, x_{i_3}^P\}$. If the neighbor of $x_{i_3,j}$ in this set uses $2i_3$, this must be the higher color in this set, meaning that x_{i_3} has a value that satisfies c_j. ◀

Lemma 21. Let us first observe that the support operation does not significantly affect a graph's clique-width. Indeed, if we have a clique-width expression for $G(\phi)$ without the support vertices, we can add these vertices as follows: each time we introduce a vertex that must be supported we instead construct the (constant clique-width) graph induced by this vertex and its support and then rename all supporting vertices to a junk label that is never connected to anything else. It is clear that this can be done by (in the worst case) adding a constant number of new labels.

Let us then argue why the rest of the graph has constant clique-width. First, the graph induced by x_i^N, x_i^P, for $i \in [n]$ is a matching, which has constant clique-width. We construct this graph in a way that uses one label for the vertices x_i^N and another for x_i^P. We then introduce to the graph the clauses one by one: first the verticex $x_{i,j}$ (which are connected with an appropriate join to x_i^N or x_i^P), c_j and d_j. We do this in a way that all d_j have in the end the same label. Finally we introduce u and join it to all d_j vertices. ◀

D FPT for modular width

Recall that two vertices $u, v \in V$ of a graph $G = (V, E)$ are *twins* if $N(u) \setminus v = N(v) \setminus u$, and called *true* (respectively, *false*) twins if they are adjacent (respectively, non-adjacent). A *twin class* is a maximal set of vertices that are pairwise twins. It is easy to see that any twin class is either a clique or an independent set. We say that a graph $G = (V, E)$ has *neighborhood diversity* at most w if and only if V admits a partition into at most w vertex subsets, each of which consists of pairwise twins.

The main result of this section is that Grundy Coloring is FPT with respect to modular-width. The modular-width is upper bounded by the neighborhood diversity, and can be viewed as a generalization of the latter measure. We first prove that Grundy Coloring is FPT parameterized by neighborhood diversity, and then use this algorithm to establish the tractability result with respect to modular-width.

D.1 Neighborhood diversity

Let $G = (V, E)$ be a graph of neighborhood diversity w with a vertex partition $V = W_1 \dot\cup \ldots \dot\cup W_w$ into twin classes. It is obvious that in any Grundy Coloring of G, the vertices of a true twin class must have all distinct colors because forms a clique. Furthermore, it is not difficult to see that the vertices of a false twin class must be colored by the same color because all of its vertices have the same neighbors.

In fact, we can show that we can remove vertices from a false twin class without affecting the grundy number of the graph:

▶ **Lemma 24.** *Let $G = (V, E)$ be a graph of neighborhood diversity w with a vertex partition $V = W_1 \dot\cup \ldots \dot\cup W_w$ into twin classes. Let W_i be a false twin class having at least two distinct vertices $u, v \in W_i$. Then $G - v$ has k-Grundy coloring if and only if G has.*

Proof. The forward implication is trivial. To see the opposite direction, consider an arbitrary k-Grundy coloring of G, Any vertex whose color is higher than v and is adjacent with v must be to u as well. Since u and v have the same color, this implies that the same coloring restricted to $G - v$ is a k-Grundy coloring. ◀

Using Lemma 24, we can reduce every false twin class into a singleton vertex, thus from now on we may assume that every twin class is a clique (possibly a singleton). An immediate consequence is that that any color class of a Grundy coloring can take at most one vertex from each twin class. Furthermore, the colors of any two vertices from the same twin class are interchangeable. Therefore, a color class V_i of a Grundy coloring is precisely characterized by the set of twin classes W_j that V_i intersects. For a color class V_i, we call the set $\{j \in [w] : W_j \cap V_i \neq \emptyset\}$ as the *intersection pattern* of V_i.

Let $\mathcal{I}$ be the collection of all sets $I \subseteq [w]$ of indices such that W_i and W_j are non-adjacent for every distinct pairs $i, j \in [w]$. It is clear that the intersection pattern of any color class is a member of $\mathcal{I}$. It turns out that if $I \in \mathcal{I}$ appears as an intersection pattern for more than one color classes, then it can be assumed to appear on a consecutive set of colors.

▶ **Lemma 25.** *Let $G = (V, E)$ be a graph of neighborhood diversity w with a vertex partition $V = W_1 \dot\cup \ldots \dot\cup W_w$ into true twin classes. Let $V_1 \dot\cup \ldots \dot\cup V_k$ be a k-Grundy coloring of G and let $I_i \in \mathcal{I}$ be the set of indices j such that $V_i \cap W_j \neq \emptyset$ for each $i \in [w]$. If $I_i = I_{i'}$ for some $i' \geq i + 2$, then the coloring $V'_1 \dot\cup \ldots \dot\cup V'_k$ where*

$$V'_\ell = \begin{cases} V_{i'} & \text{if } \ell = i+1, \\ V_{\ell+1} & \text{if } i < \ell < i', \\ V_\ell & \text{otherwise} \end{cases}$$

(i.e. the coloring obtained by 'inserting' $V_{i'}$ in between V_i and V_{i+1}) is a Grundy coloring as well.

Proof. Consider an arbitrary i'' with $i + 1 < i'' \leq i'$. To establish the statement, it suffices to show that every vertex of $V'_{i''}$ has a neighbor in V'_{i+1} in the new coloring. Recall that $V'_{i''} = V_{i''-1}$ and for an arbitrary vertex $v \in V_{i''-1}$ has a neighbor in V_i, thus in W_j for some $j \in I_i$. From the fact that $I_{i'} = I_i$ and the construction of the new coloring, it follows that $W_j \cap V'_{i+1} = W_j \cap V_{i'} \neq \emptyset$ and v has a neighbor in V'_{i+1}. ◀

The following is a consequence of Lemma 25.

▶ **Corollary 26.** *Let $G = (V, E)$ be a graph of neighborhood diversity w with a vertex partition $V = W_1 \dot\cup \ldots \dot\cup W_w$ into true twin classes. If G admits a k-Grundy coloring, then there is a k-Grundy coloring $V_1 \dot\cup \ldots \dot\cup V_k$ such that for each $I \in \mathcal{I}$, the set of colors i for which I is an intersection pattern of V_i forms a (possibly empty) sub-interval of $[k]$.*

For a sub-collection $\mathcal{I}'$ of $\mathcal{I}$, we say that $\mathcal{I}'$ is *eligible* if there is an ordering $\preceq$ on $\mathcal{I}'$ such that for every $I, I' \in \mathcal{I}'$ with $I \succeq I'$, and for every $i \in I$, there exists $i' \in I'$ such that the twin classes W_i and $W_{i'}$ are adjacent. Clearly, a sub-collection of an eligible sub-collection of $\mathcal{I}$ is again eligible.

Now we are ready to present an fpt-algorithm, parameterized by the neighborhood diversity w, to compute the Grundy number. The algorithm consists in two steps: (i) guess a sub-collection $\mathcal{I}'$ of $\mathcal{I}$ which are used as intersection patterns by a Grundy coloring, and (ii) given $\mathcal{I}'$, we solve an integer linear program.

Let $\mathcal{I}'$ be a sub-collection of $\mathcal{I}$. For each $I \in \mathcal{I}'$, let x_I be an integer variable which is interpreted as the number of colors for which I appears as an intersection pattern. Now, the linear integer program $\texttt{ILP}(\mathcal{I}')$ for a sub-collection $\mathcal{I}'$ is given as the following:

$$\max \sum_{I \in \mathcal{I}'} x_I \qquad \text{s.t.} \qquad \sum_{I \in \mathcal{I}' : i \in I} x_I = |W_i| \qquad \forall i \in [w], \tag{1}$$

where each x_I takes a positive integer value.

▶ **Lemma 27.** *Let $G = (V, E)$ be a graph of neighborhood diversity w with a vertex partition $V = W_1 \dot\cup \ldots \dot\cup W_w$ into true twin classes. The maximum value of $\texttt{ILP}(\mathcal{I}')$ over all eligible $\mathcal{I}' \subseteq \mathcal{I}$ equals the Grundy number of G.*

Proof. We first prove that the maximum value over all considered `ILP`s are at least the Grundy number of G. Fix a Grundy coloring $V_1 \dot\cup \cdots \dot\cup V_k$ achieving the Grundy number while satisfying the condition of Corollary 26. Consider the sub-collection $\mathcal{I}'$ of $\mathcal{I}$ used as intersection patterns in the fixed Grundy coloring. It is obvious that $\mathcal{I}'$ is eligible. Let $\bar{x}_I$ be the number of colors for which I is an intersection pattern for each $I \in \mathcal{I}'$. It is straightforward to check that setting the variable x_I at value $\bar{x}_I$ satisfies the constraints of $\texttt{ILP}(\mathcal{I}')$. Therefore, the objective value of $\texttt{ILP}(\mathcal{I}')$ is at least the Grundy number.

To establish the opposite direction of inequality, let $\mathcal{I}'$ be an eligible sub-collection of $\mathcal{I}$ achieving the maximum ILP objective value. Notice that $\texttt{ILP}(\mathcal{I}')$ is feasible, and let x^*_I be the value taken by the variable x_I for each $I \in \mathcal{I}'$. Since $\mathcal{I}'$ is eligible, there exists an ordering $\preceq$ on $\mathcal{I}'$ such that for every $I, I' \in \mathcal{I}'$ with $I \succeq I'$, and for every $i \in I$, there exists $i' \in I'$ such that the twin classes W_i and $W_{i'}$ are adjacent. Now, we can define the coloring $V_1 \dot\cup \cdots \dot\cup V_\ell$ by taking the first (i.e. minimum element in $\preceq$) element I_1 of $\mathcal{I}'$ x^*_I times. That is, each of V_1 up to $V_{x^*_{I_1}}$ contains precisely one vertex of W_i for each $i \in I$. The succeeding element I_2 similarly yields the next $x^*_{I_2}$ colors, and so on. From the constraint of $\texttt{ILP}(\mathcal{I}')$, we know that the constructed coloring indeed partitions V. The eligibility of $\mathcal{I}'$ ensure that this is a Grundy coloring. Finally, observe that the number of colors in the constructed coloring equals the objective value of $\texttt{ILP}(\mathcal{I}')$. This proves that the latter value is the lower bound for the Grundy number. ◀

▶ **Theorem 28.** *Let $G = (V, E)$ be a graph of neighborhood diversity w. In time $2^{O(w2^w)}$, the Grundy number of G can be computed. Furthermore, a Grundy coloring achieving the Grundy number can be found in the same running time.*

Proof. We first compute the partition $V = W_1 \dot\cup \ldots \dot\cup W_w$ of G into twin classes in polynomial time. By Lemma 24, we may assume that each W_i is a true twin class by discard some vertices of G, if necessary. Next, we compute $\mathcal{I}$ and notice that $\mathcal{I}$ contains at most 2^w elements. For each eligible sub-collection of $\mathcal{I}'$ of $\mathcal{I}$, we can solve $\texttt{ILP}(\mathcal{I}')$ by Lenstra's algorithm which runs in time $O(n^{2.5n+o(n)})$, where n denotes the number of variables in a given linear integer program. As $\texttt{ILP}(\mathcal{I}')$ contains as many as $|\mathcal{I}'| \leq 2^w$ variables, this lead to an ILP solver running in time $2^{O(w2^w)}$. Iterating over all sub-collections $\mathcal{I}'$ of $\mathcal{I}$ and checking whether each one is eligible or not takes $O(2^{2^w} \cdot (2^w)!)$-time. Due to Lemma 27, we can correctly compute the Grundy number by solving $\texttt{ILP}(\mathcal{I}')$ for each eligible $\mathcal{I}'$. This proves the first part of the statement. The second part is trivial. ◀

D.2 Modular-width

Let $G = (V, E)$ be a graph. A *module* is a set $X \subseteq V$ of vertices such that $N(u) \setminus X = N(v) \setminus X$ for every $u, v \in X$, that is, their neighborhoods coincides outside of X. Clearly, a connected component is a module. Moreover, a connected component in the complement of G forms a module as well. It is known that if neither G nor its complement is disconnected, the collection of maximal module which are not V forms a partition of V. Moreover, from maximality of modules and that neither G nor its complement is disconnected, it is not difficult to see that such a partition is unique. Let $\mathcal{M} = M_1 \dot\cup \cdots \dot\cup M_k$ be such a partition of V. Then a *quotient graph* of G, denoted as $G/\mathcal{M}$, takes the maximal modules in $\mathcal{M}$ as the vertex set and two vertices are adjacent in $G/\mathcal{M}$ if and only if the corresponding modules are (fully) adjacent. Notice that in $G/\mathcal{M}$, every module is either a singleton or the entire vertex set.

Recall that a complete join of G_1 and G_2 is the graph obtained by taking a disjoint union of $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ and furthermore adding an edge between every vertex pair $u \in V_1$ and $v \in V_2$. All together, the notion of module points to a natural way for recursively decomposing a graph. Namely, for any graph with at least two vertices, it is known that exactly one of the three decomposition applies.

1. Disjoint union: if G is a disjoint union of G_1 and G_2, write $G = G_1 \oplus G_2$.
2. Complete join: if G is a complete join of G_1 and G_2, write $G = G_1 \otimes G_2$.
3. Prime[2]: if $\mathcal{M} = M_1 \dot\cup \cdots \dot\cup M_k$ is a nontrivial partition of V into maximal modules and $H = G/\mathcal{M}$, write $G = H[G[M_1], \ldots, G[M_k]]$.

Recursively applying one of the above decompositions till no longer possible, we obtain a canonical tree[3] T called a *modular decomposition tree* such that

- the root node represents G,
- each internal node representing a graph G' is labeled by the operator $\oplus$, $\otimes$, or the prime graph H, depending on the type of decomposition applied to G'. Its children represent the induced subgraph of G' that are operands of the said operator.
- the leaf set is bijectively mapped to V.

Finally, the *modular-width* of G defined as the maximum number of children over all internal nodes of a modular decomposition tree.

▶ **Lemma 29.** *Let $G = (V, E)$ be a graph. Then the following holds.*

$$\Gamma(G) = \begin{cases} \max\{\Gamma(G_1), \Gamma(G_2)\} & \textit{if } G = G_1 \oplus G_2 \\ \Gamma(G_1) + \Gamma(G_2) & \textit{if } G = G_1 \otimes G_2 \\ \Gamma(H[G']) & \textit{if } G = H[G[M_1], \ldots, G[M_k]], \end{cases}$$

where G' is the graph obtained from $G = H[G[M_1], \ldots, G[M_k]]$ by replacing $G[M_i]$ by a clique on $\Gamma(G[M_i])$ vertices for each $i \in [k]$ and maintaining a full adjacency between i-th and j-th cliques whenever the quotient graph H indicates an adjacency between M_i and M_j.

Proof. When $G = G_1 \oplus G_2$, it is trivial to see that $\Gamma(G) = \max\{\Gamma(G_1), \Gamma(G_2)\}$. If $G = G_1 \otimes G_2$, then fix a Grundy coloring of G_1 and G_2 achieving $\Gamma(G_1)$ and $\Gamma(G_2)$ respectively, By reassigning color $i + \Gamma(G_1)$ to the vertices of G_2 with color i, we obtain a new coloring of G. Obviously, it is a Grundy coloring using the claimed number of colors.

Now suppose that $G = H[G[M_1], \ldots, G[M_k]]$ and notice that G' has neighborhood diversity k with i-th clique replacing the module M_i being a true twin class for each $i \in [k]$. We will first prove that $\Gamma(G') \le \Gamma(G)$. Fix a $\Gamma(G')$-Grundy coloring $V_1 \dot\cup \cdots \dot\cup V_{|\Gamma(G')|}$ of G', and for each $i \in [k]$, let $V_1^i \dot\cup \cdots \dot\cup V^i_{\Gamma(G[M_i])}$ be a Grundy coloring of $G[M_i]$ using $\Gamma(G[M_i])$ colors. In the Grundy coloring of G', the vertices of i-clique gets mutually distinct colors and thus the number of colors taken by some vertex of i-th clique is precisely $\Gamma(G[M_i])$. Let σ_i be the ordering of colors (from low to high) that appear in some vertex in the i-th clique of G'. It is trivial to verify that the following coloring of G is proper and a Grundy coloring with $\Gamma(G')$ colors, thus proving that $\Gamma(G') \le \Gamma(G)$.

---

[2] A graph in which every module is either a singleton or the entire vertex set is called a *prime graph*. When neither $\oplus$ nor $\otimes$ applies, the quotient graph of G is a prime graph, which prompts the name.

[3] An avid reader may notice that our definition of modular decomposition slight deviates from the standard one. In the standard definition, the node labeled by $\oplus$ (resp. $\otimes$) renders all connected component of G (resp. $\bar{G}$) to be represented in its children, therefore allowing such nodes to have more than one children, see [74].

In each module M_i and for each color $j \in [\Gamma(G')]$, the vertices of V_j^i gets the color $\sigma_i(j)$.

To prove that $\Gamma(G') \geq \Gamma(G)$, fix a $\Gamma(G)$-Grundy coloring $V_1 \dot\cup \cdots \dot\cup V_{\Gamma(G)}$ of G.

▷ Claim 30. The number of colors used by a module M_i is at most $\Gamma(G[M_i])$ for each i, that is, $|\{j \in [\Gamma(G)] : V_j \cap M_i \neq \emptyset\}| \leq \Gamma(G[M_i])$.

Proof. We claim that the number of colors used by a module M_i is at most $\Gamma(G[M_i])$ for each i. Suppose the contrary. Then there exists two colors $c < c'$ and a vertex v of the module M_i colored by c' such that v's neighbors in color c are all belong to $V \setminus M_i$. Indeed, if there is no such color pair and a vertex, then the collection of sets $V_1 \cap M_i, \cdots, V_{\Gamma(G)} \cap M_i$ contain more than $\Gamma(G[M_i])$ non-empty sets. Such a collection provides a Grundy coloring for $G[M_i]$ using more than $\Gamma(G[M_i])$, a contradiction. However, any neighbor u of v outside the module M_i is a neighbor of every vertex in M_i. As the color class V_c intersects with M_i, this means that V_c is not independent, a contradiction. ◀

Let us color the vertices of G'. By the previous claim, the following coloring can be performed by giving each vertex of G' at most one color. That is, for each module M_i,

if color c appears in M_i, precisely one vertex from the i-th clique of G' gets color c.

All the vertices of G' which did not receive any color is removed and let G'' be the resulting induced subgraph of G'. It is easy to see that the constructed coloring of G'' is a Grundy coloring, and consequently it holds that $\Gamma(G') \geq \Gamma(G'') \geq \Gamma(G)$. This completes the proof. ◀

With Lemma 29 and using the result of Subsection D.1, we have a standard bottom-up algorithm for computing the Grundy number.

▶ **Theorem 31.** *Let $G = (V, E)$ be a graph of modular-width w. In time $2^{O(w2^w)}$, the Grundy number of G can be computed. Furthermore, a Grundy coloring achieving the Grundy number can be found in the same running time.*

Proof. Consider a modular decomposition tree T of G, which can be computed in linear time, for example [74]. For each tree node t representing a vertex set $X \subseteq V$, we can compute the Grundy number of $G[X]$ assuming that the Grundy number on the graphs represented by its children are known. Namely, if t is labeled by either $\oplus$ or $\otimes$, the Grundy number of $G[X]$ can be obtained by either taking the maximum or the sum of the two Grundy numbers on its children. If $G[X]$ is labeled by a quotient graph H, then note that H has at most w vertices. By Lemma 29, computing the Grundy number of $G[X]$ is equivalent to computing the Grundy number of a graph whose neighborhood diversity is at most w. The latter can be done in time $2^{O(w2^w)}$ by Theorem 28. As the leaf nodes represent singleton graphs, clearly the Grundy number can be computed on the leaves. Repeatedly computing the Grundy number in a bottom-to-top manner, we can compute the Grundy number of G within the claimed running time. We omit a tedious proof on how to construct an actual $\Gamma(G)$-Grundy coloring. ◀