Discrete Applied Mathematics 243 (2018) 99–114

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Complexity of Grundy coloring and its variants

Édouard Bonnet [a,*,1], Florent Foucaud [b], Eun Jung Kim [c], Florian Sikora [c]

[a] Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), Hungary
[b] Université Blaise Pascal, CNRS, LIMOS, France
[c] Université Paris-Dauphine, PSL Research University, CNRS, LAMSADE, Paris, France

ARTICLE INFO

Article history:
Received 6 December 2016
Received in revised form 12 December 2017
Accepted 16 December 2017
Available online 23 February 2018

Keywords:
Grundy coloring
Computational complexity
Weak Grundy
Connected Grundy
Exact algorithm
Parameterized complexity

ABSTRACT

The Grundy number of a graph is the maximum number of colors used by the greedy coloring algorithm over all vertex orderings. In this paper, we study the computational complexity of Grundy Coloring, the problem of determining whether a given graph has Grundy number at least k. We also study the variants Weak Grundy Coloring (where the coloring is not necessarily proper) and Connected Grundy Coloring (where at each step of the greedy coloring algorithm, the subgraph induced by the colored vertices must be connected).

We show that Grundy Coloring can be solved in time $O^*(2.443^n)$ and Weak Grundy Coloring in time $O^*(2.716^n)$ on graphs of order n. While Grundy Coloring and Weak Grundy Coloring are known to be solvable in time $O^*(2^{O(wk)})$ for graphs of treewidth w (where k is the number of colors), we prove that under the Exponential Time Hypothesis (ETH), they cannot be solved in time $O^*(2^{o(w \log w)})$. We also describe an $O^*(2^{2^{O(k)}})$ algorithm for Weak Grundy Coloring, which is therefore FPT for the parameter k. Moreover, under the ETH, we prove that such a running time is essentially optimal (this lower bound also holds for Grundy Coloring). Although we do not know whether Grundy Coloring is in FPT, we show that this is the case for graphs belonging to a number of standard graph classes including chordal graphs, claw-free graphs, and graphs excluding a fixed minor. We also describe a quasi-polynomial time algorithm for Grundy Coloring and Weak Grundy Coloring on apex-minor graphs. In stark contrast with the two other problems, we show that Connected Grundy Coloring is NP-complete already for $k = 7$ colors.

1. Introduction

A *k-coloring* of a graph G is a surjective mapping $\varphi : V(G) \to \{1, \ldots, k\}$ (we say that vertex v is colored with $\varphi(v)$). A k-coloring φ is *proper* if any two adjacent vertices receive different colors in φ. The *chromatic number* $\chi(G)$ of G is the smallest k such that G has a proper k-coloring. Determining the chromatic number of a graph is one of the most fundamental problems in graph theory. Given a graph G and an ordering $\sigma = v_1, \ldots, v_n$ of $V(G)$, the *first-fit coloring algorithm* colors the vertices from v_1 to v_n in the order imposed by σ, and the vertex v_i is colored with the smallest positive integer that is *not* present among the colors of the neighbors of v_i which are in $\{v_1, \ldots, v_{i-1}\}$ (in other words, the neighbors of v_i which are already colored). The *Grundy number* $\Gamma(G)$ is the largest k such that G admits a first-fit coloring (for some ordering)

* Corresponding author.
E-mail addresses: edouard.bonnet@ens-lyon.fr (É. Bonnet), florent.foucaud@gmail.com (F. Foucaud), eunjung.kim@dauphine.fr (E.J. Kim), florian.sikora@dauphine.fr (F. Sikora).

1 Current affiliation: ENS Lyon, LIP, France.

https://doi.org/10.1016/j.dam.2017.12.022
0166-218X/

using k colors. First-fit is presumably the simplest heuristic to compute a proper coloring of a graph. In this sense, the Grundy number gives an algorithmic upper bound on the performance of any heuristic for the chromatic number. This notion was first studied by Grundy in 1939 in the context of digraphs and games [4,18], and formally introduced 40 years later by Christen and Selkow [9]. It was independently defined under the name *ochromatic number* by Simmons [36] (the two concepts were proved to be equivalent in [14]). Many works have studied the first-fit algorithm in connection with on-line coloring algorithms, see for example [32]. A natural relaxation of this concept is the *weak Grundy number*, introduced by Kierstead and Saoub [26], where the obtained coloring is not asked to be proper. A more restricted concept is the one of *connected Grundy number*, introduced by Benevides et al. [3], where the algorithm is given an additional "local" restriction on the feasible vertex orderings that can be considered: at each step of the first-fit algorithm, the subgraph induced by the colored vertices must be connected.

The goal of this paper is to advance the study of the computational complexity of determining the Grundy number, the weak Grundy number and the connected Grundy number of a graph.

Let us introduce the problems formally. Let G be a graph and let $\sigma = v_1, \dots, v_n$ be an ordering of $V(G)$. A k-coloring $\varphi : V(G) \to \{1, \dots, k\}$ of G is a *first-fit coloring with respect to* σ if for every vertex v_i, the two following conditions hold: (1) for every color (i.e., positive integer) c with $c < \varphi(v_i)$, there is a $j < i$ such that v_i and v_j are adjacent and $\varphi(v_j) = c$, and (2) there is no $j < i$ such that v_i and v_j are adjacent and $\varphi(v_i) = \varphi(v_j)$. A k-coloring is a *Grundy coloring* if it is a first-fit coloring with respect to *some* vertex ordering σ. A k-coloring is a *weak Grundy coloring* if it satisfies the condition (1) with respect to some vertex ordering σ. A vertex ordering $\sigma = v_1, \dots, v_n$ is *connected* if for every i, $1 \leqslant i \leqslant n$, the subgraph induced by $\{v_1, \dots, v_i\}$ is connected. A k-coloring is a *connected Grundy coloring* if it is a Grundy coloring with respect to a connected vertex ordering. We note that a (connected) Grundy coloring is a proper coloring, and a weak Grundy coloring is not necessarily proper. Observe that a (connected) Grundy coloring is uniquely defined by its ordering σ, while it is not the case for the weak Grundy coloring.

The maximum number of colors used, taken among all (weak, connected, respectively) Grundy colorings, is called the (*weak*, *connected*, respectively) Grundy number and is denoted $\Gamma(G)$ ($\Gamma'(G)$ and $\Gamma_c(G)$, respectively). In this paper, we study the complexity of computing these invariants.

GRUNDY COLORING
Input: A graph G, an integer k.
Question: Do we have $\Gamma(G) \geqslant k$?

WEAK GRUNDY COLORING
Input: A graph G, an integer k.
Question: Do we have $\Gamma'(G) \geqslant k$?

CONNECTED GRUNDY COLORING
Input: A graph G, an integer k.
Question: Do we have $\Gamma_c(G) \geqslant k$?

Note that $\chi(G) \leqslant \Gamma(G) \leqslant \Delta(G)+1$, where $\chi(G)$ is the chromatic number and $\Delta(G)$ is the maximum degree of G. However, the difference $\Gamma(G) - \chi(G)$ can be (arbitrarily) large, even for bipartite graphs. For example, the Grundy number of the tree of Fig. 1 is 4, whereas its chromatic number is 2. Note that this is not the case for Γ_c for bipartite graphs, since $\Gamma_c(G) \leqslant 2$ for any bipartite graph G [3]. However, the difference $\Gamma_c(G) - \chi(G)$ can be (arbitrarily) large even for planar graphs [3].

Previous results. GRUNDY COLORING remains NP-complete on bipartite graphs [22] and their complements [38] (and hence claw-free graphs and P_5-free graphs), on chordal graphs [35], and on line graphs [21]. Certain graph classes admit polynomial-time algorithms. There is a linear-time algorithm for GRUNDY COLORING on trees [23]. This result was extended to graphs of bounded treewidth by Telle and Proskurowski [37], who proposed a dynamic programming algorithm running in time $k^{O(w)}2^{O(wk)}n = O(n^{3w^2})$ for graphs of treewidth w (in other words, their algorithm is in FPT for parameter $k + w$ and in XP for parameter w).[2] A polynomial-time algorithm for GRUNDY COLORING on P_4-laden graphs, which contains all cographs as a subfamily, was given in [2].

Note that GRUNDY COLORING admits a polynomial-time algorithm when the number k of colors is fixed [39], in other words, it is in XP for parameter k.

GRUNDY COLORING has polynomial-time constant-factor approximation algorithms for inputs that are interval graphs [20,32], complements of chordal graphs [20], complements of bipartite graphs [20] and bounded tolerance graphs [26].

2 The first running time is not explicitly stated in [37] but follows from their meta-theorem. The second one is deduced by the authors of [37] from the first one by upper-bounding k by $w \log_2 n + 1$.

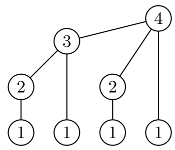

Fig. 1. The binomial tree T_4, where numbers denote the color of each vertex in a first-fit coloring with largest number of colors.

However, there is a constant $c > 1$ such that approximating Grundy Coloring within c in polynomial time is impossible unless NP ⊆ RP [27] (a result extended to chordal graphs under the assumption P ≠ NP in the unpublished manuscript [17]). It is not known whether a polynomial-time $o(n)$-factor approximation algorithm exists.

When parameterized by the order of the graph minus the number of colors, Grundy Coloring was shown to be in FPT by Havet and Sempaio [22].

Weak Grundy Coloring was not studied as much as Grundy Coloring, but many results that hold for Grundy Coloring are applicable to Weak Grundy Coloring. Weak Grundy Coloring was shown to be NP-hard to approximate within some constant factor $c > 1$, even on chordal graphs [17]. Furthermore, in [37] an algorithm for Weak Grundy Coloring running in time $2^{O(wk)}n = O(n^{3w^2})$ for graphs of treewidth w was given (in [37], Weak Grundy Coloring was called Iterated Dominating Set).

Connected Grundy Coloring was introduced by Benevides et al. [3], who proved it to be NP-complete, even for chordal graphs and for co-bipartite graphs.

Our results. We give two exact algorithms for Grundy Coloring and Weak Grundy Coloring running in time $O^*(2.443^n)$ and $O^*(2.716^n)$, respectively. It was previously unknown if any $O^*(c^n)$-time algorithms exist for these problems (with c a constant). Denoting by w the treewidth of the input graph, it is not clear whether the $O^*(2^{O(wk)})$-time algorithms for Grundy Coloring and Weak Grundy Coloring of [37] can be improved, for example to algorithms of running time $O^*(k^{O(w)})$ or $O^*(f(w))$ (the notation O^* neglects polynomial factors). In fact we show that an $O^*(k^{O(w)})$-time algorithm for Grundy Coloring would also have running time $O^*(2^{O(w \log w)})$.

As a lower bound, we show that assuming the Exponential Time Hypothesis (ETH), an $O^*(2^{o(w \log w)})$-time algorithm for Grundy Coloring or Weak Grundy Coloring does not exist (where w is the feedback vertex set number of the input graph). In particular, the exponent n cannot be replaced by the feedback vertex set number (or treewidth) in our $O^*(2.443^n)$-and $O^*(2.716^n)$-time algorithms.

We prove that on apex-minor-free graphs, quasi-polynomial time algorithms, of running time $n^{O(\log^2 n)}$, exist for Grundy Coloring and Weak Grundy Coloring.

We also show that Weak Grundy Coloring can be solved in FPT time $O^*(2^{2^{O(k)}})$ using the color coding technique. Under the ETH, we show that this is essentially optimal: no $O^*(2^{2^{o(k)}}2^{o(n+m)})$-time algorithm for graphs with n vertices and m edges exists. The latter lower bound also holds for Grundy Coloring.

We also study the parameterized complexity of Grundy Coloring parameterized by the number of colors, showing that it is in FPT for graphs including chordal graphs, claw-free graphs, and graphs excluding a fixed minor.

Finally, we show that Connected Grundy Coloring is computationally much harder than Grundy Coloring and Weak Grundy Coloring when viewed through the lens of parameterized complexity. While for the parameter "number of colors", Grundy Coloring is in XP and Weak Grundy Coloring is in FPT, we show that Connected Grundy Coloring is NP-complete even when $k = 7$, that is, it does not belong to XP unless P = NP. Note that the known NP-hardness proof of [3] for Connected Grundy Coloring was only for an unbounded number of colors.

Structure of the paper. We start with some preliminary definitions, observations and lemmas in Section 2. Our positive algorithmic results are presented in Section 3, and our algorithmic lower bounds are presented in Section 4. We conclude the paper in Section 5.

2. Preliminaries

Graphs and sets. For any two integers $x < y$, we set $[x, y] := \{x, x+1, \ldots, y-1, y\}$, and for any positive integer x, $[x] := [1, x]$. $V(G)$ denotes the set of vertices of a graph G and $E(G)$ its set of edges. For any $S \subseteq V(G)$, $E(S)$ denotes the subset of edges of $E(G)$ having both endpoints in S, and $G[S]$ denotes the subgraph of G induced by S; that is, graph $(S, E(S))$. If $H \subseteq V(G)$, $G - H$ denotes the graph $G[V(G) \setminus H]$. As a slight abuse of notation, if H is an (induced) subgraph of G, we also denote by $G - H$ the graph $G[V(G) \setminus V(H)]$. For any vertex $v \in V(G)$, $N(v) := \{w \in V(G) | vw \in E(G)\}$ denotes the set of neighbors of v in G. For any subset $S \subseteq V(G)$, $N(S) = \bigcup_{v \in S} N(v) \setminus S$. The *distance-$k$* neighborhood of v is the set of vertices at distance at most k from v.

Computational complexity. A decision problem is said to be *fixed-parameter tractable* (or in the class FPT) w.r.t. parameter k if it can be solved in time $f(k) \cdot |I|^c$ for an instance I, where f is a computable function and c is a constant (see for example the books [12,33] for details). The class XP contains those problems solvable in time $|I|^{f(k)}$, where f is a computable function.

The *Exponential Time Hypothesis* (ETH) is a conjecture by Impagliazzo et al. asserting that there is no $2^{o(n)}$-time algorithm for 3-SAT on instances with n variables [24]. The ETH, together with the sparsification lemma [24], even implies that there is no $2^{o(n+m)}$-time algorithm solving 3-SAT. Many algorithmic lower bounds have been proved under the ETH, see for example [29].

Minors. A *minor* of a graph G is a graph that can be obtained from G by (i) deletion of vertices or edges (ii) contraction of edges (removing an edge and merging its endpoints into one). Given a graph H, a graph G is H-*minor-free* if H is not a minor of G.

An *apex graph* is a graph obtained from a planar graph G and a single vertex v, and by adding arbitrary edges between v and G. A graph is said to be *apex-minor-free* if it is H-minor-free for some apex graph H.

Tree-decompositions. A *tree-decomposition* of a graph G is a pair $(\mathcal{T}, \mathcal{X})$, where $\mathcal{T}$ is a tree and $\mathcal{X} := \{X_t : t \in V(\mathcal{T})\}$ is a collection of subsets of $V(G)$ (called *bags*), and they must satisfy the following conditions: (i) $\bigcup_{X \in V(T)} = V(G)$, (ii) for every edge $uv \in E(G)$, there is a bag of $\mathcal{T}$ that contains both u and v, and (iii) for every vertex $v \in V(G)$, the set of bags containing v induces a connected subtree of $\mathcal{T}$.

The maximum size of a bag X_t over all tree nodes t of $\mathcal{T}$ minus one is called the *width* of $\mathcal{T}$. The minimum width of a tree-decomposition of G is the *treewidth* of G. The notion of tree-decomposition has been used extensively in algorithm design, especially via dynamic programming on the tree-decomposition.

Grundy coloring. Let G be a graph, $\sigma = v_1, \ldots, v_n$, an ordering of its vertices, $\varphi : V(G) \to [k]$, the first-fit coloring of G with respect to σ, and j, the smallest index such that $\varphi(v_j) = k$. Informally, finishing the Grundy coloring of $v_{j+1}, \ldots, v_n$ is irrelevant in asserting that $\Gamma(G) \geqslant k$. Indeed, this bound is established as soon as we color vertex v_j. We formalize this idea that a potentially much smaller induced subgraph of the input graph (here, $G[\{v_1, \ldots, v_j\}]$) might be a relevant certificate, via the notion of *witnesses* and *minimal witnesses*.[3]

In a graph G, a *witness achieving color* k, or simply a k-*witness*, is an induced subgraph G' of G, such that $\Gamma(G') \geqslant k$. Such a k-witness is *minimal* if no proper induced subgraph of G' has Grundy number at least k.

Observation 1. *For any graph G, $\Gamma(G) \geqslant k$ if and only if G admits a minimal k-witness.*

We can also notice that, in any Grundy k-coloring (that is, Grundy coloring achieving color k) of a minimal k-witness, exactly one vertex is colored with k. Otherwise, it would contradict the minimality.

If k is not specified, we assume that the witness achieves the largest possible color: a (minimal) *witness* is a (minimal) witness achieving color $\Gamma(G)$. A *colored* (minimal) (k-)witness is a (minimal) (k-)witness together with a Grundy k-coloring of its vertices, that can be given equivalently by the coloring function φ, or the ordering σ, or a partition $W_1 \uplus \ldots \uplus W_k$ of the vertices into color classes (namely, the vertices of W_i are colored with i).

We will now observe that minimal k-witnesses have at most 2^{k-1} vertices. To that end, we define a family of rooted trees sometimes called *binomial trees*. If, for each $i \in [l]$, t_i is a tree rooted at v_i, $v[t_1, \ldots, t_l]$ denotes the tree rooted at v obtained by adding v to the disjoint union of the t_i's and linking it to all the v_i's. Then, the *ith child* of v is v_i and is denoted by $v(i)$. We say that v is the *parent* of v_i. We may also say that v is the *parent* of the tree t_i. The set of binomial trees $(T_k)_{k\geqslant 1}$ is a family of rooted trees defined as follows (see Fig. 1 for an illustration):

- T_1 consists only of one vertex (incidentally the root), and
- $\forall k \geqslant 1$, $T_{k+1} = v[T_1, T_2, \ldots, T_k]$.

The binomial tree T_k can be seen as the dependencies between the vertices of a minimal k-witness colored by a coloring φ. More concretely, any vertex colored with color $i \leqslant k$ needs to have in its neighborhood $i-1$ vertices colored with each color from 1 to $i-1$. Say, we label the root of T_k with the unique vertex colored k. And then, in a top-town manner, we label, for each $j \in [\varphi(v) - 1]$, the jth child of a vertex labeled by v, by a neighbor of v colored with j. Each vertex of the minimal k-witness should appear at least once as a label of T_k, for the sake of minimality. Besides, the number of vertex of T_k is 2^{k-1}. This leads to the following observations:

Observation 2. *A minimal k-witness W has radius at most k. More precisely, W is entirely included in the distance-k neighborhood of the vertex colored with k in a Grundy k-coloring of W.*

Observation 3. *A minimal k-witness has at most 2^{k-1} vertices.*

Observation 4. *The color of a vertex of degree d in any Grundy coloring is at most $d+1$.*

By Observations 1 and 3, Grundy Coloring can be solved by checking if one of the $\binom{n}{2^{k-1}}$ induced subgraphs on 2^{k-1} vertices, has Grundy number k. This shows that, parameterized by the number k of colors, the problem is in XP:

Corollary 1 (*Zaker [39]*)**.** Grundy Coloring *can be solved in time $f(k)n^{2^{k-1}}$.*

[3] Witnesses were called *atoms* by Zaker [39] and *critical* in [19].

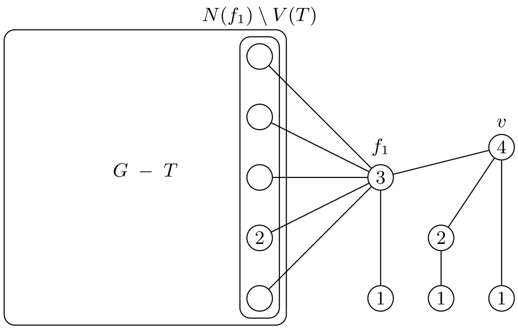

Fig. 2. A simple instantiation of Lemma 2 with $s = 4$, $p = 1$, and $a_1 = 2$.

We now come back to binomial trees and show two lemmas that will be very helpful to prove the hardness results of the paper.

Lemma 1. *The Grundy number of T_k is k. Moreover, there are exactly two Grundy colorings achieving color k, and a unique Grundy coloring if we impose that the root v is colored k.*

Proof. As hinted before, the tree T_k is a minimal k-witness with the largest number of vertices, so $\Gamma(T_k) \geqslant k$. The easiest Grundy k-coloring of T_k consists of coloring all the leaves with color 1. Now, if one removes all the leaves of T_k, one gets a binomial tree T_{k-1}, whose leaves can all be colored 2, and so forth, up to coloring v with color k (see Fig. 1). As the degree of T_k is $k-1$, by Observation 4, $\Gamma(T_k) \leqslant k$ also holds.

What remains to be seen is that the Grundy k-coloring of T_k is unique up to deciding which of v and $v(k-1)$ gets color k and which gets color $k-1$. There are only two vertices of degree $k-1$ in T_k: v and $v(k-1)$. Therefore, only v or $v(k-1)$ can potentially be colored with k, by Observation 4. As T_k rooted at v is isomorphic to T_k rooted at $v(k-1)$, we can assume that v will be the vertex colored k. We show by strong induction that there is only one Grundy coloring of T_k where the root has color k. Obviously, there is a unique Grundy coloring of T_1. For any integer $k \geqslant 2$, if we impose that the root v is colored k, the $k-1$ children of v have to be colored with all the integers of $[k-1]$. As for each $i \in [k-1]$, T_i has maximum degree $i-1$, the color of $v(i)$ is at most i. First, color $k-1$ can only come from $v(k-1)$. But, now that the color of this vertex is imposed, color $k-2$ can only come from $v(k-2)$. Finally, the only possibility is to color $v(i)$ with color i for each $i \in [k-1]$. By the induction hypothesis, there is a unique such Grundy coloring for each subtree. □

Subtrees and dominant subtrees. The subtree $t[x]$ rooted at vertex x of a tree t rooted at v, is the tree induced by all the vertices y of t such that the simple path from v to y goes through x. The rooted tree t' is a subtree of t, if there exists a vertex x of t such that $t' = t[x]$. The *number* of rooted subtree t' of a rooted tree t is the number of vertices x of t such that $t' = t[x]$. In a binomial tree T_k, the number of T_l (for $l \in [k-1]$) is 2^{k-l-1}. For any $l \in [k-1]$, we say that a subtree T_l of T_k is *dominant*, if its root is the child of the root of a T_{l+1}. In other words, a dominant subtree is the largest among its siblings. The *dominant subtree* of a vertex of a binomial tree is the largest subtree rooted at one of its children. In a binomial tree T_k, the number of dominant T_l (for $l \in [k-2]$) is by definition the number of T_{l+1}, that is 2^{k-l-2}.

Although the statement of the next lemma is rather technical, its underlying idea is fairly simple. If one removes some well-chosen subtree T_{a_i} from a binomial tree T_s rooted at v, and connects the parent f of this removed subtree to the rest of a graph G, then in order to color v with color s, one would have to color with a_i at least one of the neighbors of f outside T_s (see Fig. 2). Using this as a gadget, we will be able to make sure that at least one vertex of a specific vertex-subset is colored with a specific color. We prove the more general result when multiple subtrees are removed.

Lemma 2. *Let $a_1, \ldots, a_p < s$ be integers. Let G be a graph, and let T be an induced subgraph of G such that the following hold.*

- *T can be obtained from T_s (rooted at v), a set $T_{a_1}, \ldots, T_{a_p}$ of pairwise disjoint dominant subtrees. Let $F = \{f_1, \ldots, f_p\}$ be the set of parents of those subtrees.*
- *We have $N(V(G) \setminus V(T)) \cap V(T) = F$, that is, only F links T to the rest of G.*

Then, the following conditions on Grundy colorings of G are equivalent.

- *(i) There is a Grundy coloring such that v is colored s.*
- *(ii) There is a Grundy coloring of an induced subgraph of $G - T$ such that, for each $i \in [p]$, at least one vertex of $N(f_i) \setminus V(T)$ is colored a_i and no vertex of $N(f_i) \setminus V(T)$ is colored $a_i + 1$.*

Proof. (ii) ⇒ (i). Assume that there is a Grundy coloring of an induced subgraph of $G - T$ such that, for each $i \in [p]$, at least one vertex of $N(f_i) \setminus V(T)$ is colored a_i and no vertex of $N(f_i) \setminus V(T)$ is colored $a_i + 1$. We extend this Grundy coloring by coloring T as we would optimally color T_s. By Lemma 1, vertex v will be colored with s.

(i) $\Rightarrow$ (ii). Now, suppose that there is a Grundy coloring where vertex v receives color s. By the same induction as in the second part of the proof of Lemma 1, each vertex f_i has to be colored $a_i + 1$. The degree of the neighbors of f_i within T is bounded by a_i, hence they cannot be colored with color a_i (unless one first colors f_i with a smaller color, but this would be a contradiction). Thus, the color a_i in the neighborhood of f_i has to come from a vertex of $G - T$. Moreover, no vertex in $N(f_i) \setminus V(T)$ can be colored a_{i+1}, otherwise f_i cannot get this color. Summing up, there is a Grundy coloring of an induced subgraph of $G - F$ such that, for each $i \in [p]$, at least one vertex of $N(f_i) \setminus V(T)$ is colored a_i and no vertex of $N(f_i) \setminus V(T)$ is colored $a_i + 1$. As F separates $T - F$ from $G - T$, coloring vertices in $T - F$ is not helpful to color vertices in $N(f_i) \setminus V(T)$. Hence, there is a Grundy coloring of an induced subgraph of $G - \mathbf{T}$ such that, for each $i \in [p]$, at least one vertex of $N(f_i) \setminus V(T)$ is colored a_i and no vertex of $N(f_i) \setminus V(T)$ is colored $a_i + 1$. □

Weak Grundy and connected Grundy colorings. We can naturally extend the notion of witnesses to the Weak Grundy Coloring problem. It turns out that everything we observed or showed so far for Grundy Coloring, namely Observations 1, 2, 3, 4, and Lemmas 1 and 2 (where condition (ii) is replaced by the simpler condition: *(ii') There is a weak Grundy coloring of an induced subgraph of $G - T$ such that, for each $i \in [p]$, at least one vertex of $N(f_i)$ is colored a_i*), are also valid when it comes to weak Grundy colorings.

For Connected Grundy Coloring, again, we can similarly define a notion of witness. Though, as we will see, the size of minimal k-witnesses for the connected version cannot be bounded by a function of k. Here, the only statements that remain valid are Observations 1 and 4. To illustrate the different behavior of this variant, the connected Grundy number of any binomial tree T_k is 2, as it is for every bipartite graph with at least one edge [3].

3. Positive results

We now present the positive algorithmic results of this paper.

3.1. *Exact algorithms for* Grundy Coloring *and* Weak Grundy Coloring

A straightforward way to solve Grundy Coloring is to enumerate all possible orderings of the vertex set and to check whether the greedy algorithm uses at least k colors. This is a $\Theta(n!)$-time algorithm. A natural question is whether there is a faster exact algorithm. Such algorithms for Coloring based on dynamic programming have been long known, see for example Lawler [28], but no c^n algorithm for Grundy Coloring, for any constant c, was previously known. We now give such an algorithm.

As a preparatory lemma, we remark that a colored minimal k-witness can be seen as a set of nested independent dominating sets, in the following sense.

Lemma 3 (*[19]*)**.** *Let G be a graph and let G' be a colored k-witness with the partition into color classes $W_1 \uplus \cdots \uplus W_k$. Then, W_i is an independent set which dominates the set $\bigcup_{j\in[i+1,k]} W_j$. In particular, W_1 is an independent dominating set of $V(G')$.*

Proof. As a Grundy coloring is a proper coloring, W_i is an independent set. If a vertex $v \in W_h$ (with $h > i$) has no neighbor in W_i, then v is colored with a color at most i, a contradiction. So, W_i should dominate W_h. □

We rely on two observations: (a) in a colored witness, every color class W_i is an independent dominating set in $G[\bigcup_{j\geqslant i} W_j]$ (Lemma 3), and (b) any independent dominating set is a maximal independent set (and vice versa). The algorithm is obtained by dynamic programming over subsets, and uses an algorithm which enumerates all maximal independent sets.

Theorem 5. Grundy Coloring *can be solved in time $O^*(2.4423^n)$.*

Proof. Let $G = (V, E)$ be a graph. We present a dynamic programming algorithm to compute $\Gamma(G)$. For simplicity, given $S \subseteq V$, we denote the Grundy number of the induced subgraph $G[S]$ by $\Gamma(S)$. We recursively fill a table $\Gamma^*(S)$ over the subset lattice $(2^V, \subseteq)$ of V in a bottom-up manner starting from $S = \emptyset$. The base case of the recursion is $\Gamma^*(\emptyset) = 0$. The recursive formula is given as

$$\Gamma^*(S) = \max\{\Gamma^*(S \setminus X) + 1 \mid X \subseteq S \text{ is an independent dominating set of } G[S]\}.$$

Now let us show by induction on $|S|$ that $\Gamma^*(S) = \Gamma(S)$ for all $S \subseteq V$. The assertion trivially holds for the base case. Consider a nonempty subset $S \subseteq V$; by induction hypothesis, $\Gamma^*(S') = \Gamma(S')$ for all $S' \subset S$. Let X be a subset of S achieving $\Gamma^*(S) = \Gamma^*(S \setminus X) + 1$ and X' be the set of the color class 1 in the ordering achieving the Grundy number $\Gamma(S)$.

Let us first see that $\Gamma^*(S) \leqslant \Gamma(S)$. By induction hypothesis we have $\Gamma^*(S \setminus X) = \Gamma(S \setminus X)$. Consider a vertex ordering σ on $S \setminus X$ achieving $\Gamma(S \setminus X)$. Augmenting σ by placing all vertices of X at the beginning of the sequence yields a (set of) vertex ordering(s). Since X is an independent set, the first-fit algorithm gives color 1 to all vertices in X, and since X is also a dominating set for $S \setminus X$, no vertex of $S \setminus X$ receives color 1. Therefore, the first-fit algorithm on such ordering uses $\Gamma(S \setminus X) + 1$ colors. We deduce that $\Gamma(S) \geqslant \Gamma(S \setminus X) + 1 = \Gamma^*(S \setminus X) + 1 = \Gamma^*(S)$.

To see that $\Gamma^*(S) \geqslant \Gamma(S)$, we first observe that $\Gamma(S \setminus X') \geqslant \Gamma(S) - 1$. Indeed, the use of the optimal ordering of S ignoring vertices of X' on $S \setminus X'$ yields the color $\Gamma(S) - 1$. We deduce that $\Gamma(S) \leqslant \Gamma(S \setminus X') + 1 = \Gamma^*(S \setminus X') + 1 \leqslant \Gamma^*(S \setminus X) + 1 = \Gamma^*(S)$.

As an independent dominating set is a maximal independent set, we can estimate the computation of the table by restricting X to the family of maximal independent sets of $G[S]$. On an n-vertex graph, one can enumerate all maximal independent sets in time $O(1.4423^n)$ [31]. Thus, filling the table by increasing size of set S takes:

$$\sum_{i=0}^{n} \binom{n}{i} \cdot 1.4423^i = (1 + 1.4423)^n. \quad \square$$

A similar dynamic programming gives a slightly worse running time for Weak Grundy Coloring.

Theorem 6. Weak Grundy Coloring *can be solved in time* $O^*(2.7159^n)$.

Proof. Now, we fill the table:

$$\Gamma^w(S) = \max\{\Gamma^w(S \setminus X) + 1 \mid X \subseteq S \text{ is a minimal dominating set of } G[S]\}.$$

In a colored witness $W_1 \uplus \cdots \uplus W_k$ of Weak Grundy Coloring, for any $i \in [k]$, W_i (is no longer necessarily an independent set and) dominates $\bigcup_{j\in[i+1,k]} W_j$. To establish that, for any $S \subseteq V$, $\Gamma^w(S) = \Gamma'(S)$, we need to transform any colored witness $W_1 \uplus \cdots \uplus W_k$ (with $k \leqslant 2$) into a colored witness $W'_1 \uplus \cdots \uplus W'_k$ on the same induced subgraph G', also achieving color k, but with the additional property that W'_1 is a minimal dominating set of G'. Actually, in order to obtain that property we only need to transfer some vertices of W_1 to W_2. We can choose $W'_1 \subseteq W_1$ to be any minimal dominating set of G'. Then, we set $W'_2 = W_2 \cup (W_1 \setminus W'_1)$. For any $i \in [3, k]$, we just set $W'_i = W_i$. As W'_1 is a dominating set of G', the partition $W'_1 \uplus \cdots \uplus W'_k$ is indeed a colored witness. Enumerating all the minimal dominating sets of a graph on i vertices can be done in time $O^*(1.7159^i)$ [16], hence the running time of our algorithm. $\square$

We leave it as an open question to improve the running time of those algorithms. We note that the *fast subset convolution* technique [5], which is commonly used to design exponential-time algorithms, does not seem to be directly applicable here.

3.2. *Quasi-polynomial algorithms for* Grundy Coloring *and* Weak Grundy Coloring *on apex-minor-free graphs*

We will now show that the XP algorithms of [37] for Grundy Coloring and Weak Grundy Coloring imply the existence of quasi-polynomial-time algorithms for these problems on apex-minor-free graphs (such as planar graphs).

The following result of Chang and Hsu [8] will be used:

Theorem 7 (*[8]*). *Let G be a graph on n vertices for which every subgraph H has at most $d|V(H)|$ edges. Then $\Gamma(G) \leqslant \log_{d+1/d}(n) + 2$.*

In fact, we note that the bound of Theorem 7 also holds for the weak Grundy number, indeed the proof of [8] is still valid for this case.

A class of graphs has *bounded local treewidth* if for any of its members G, the treewidth of G is upper-bounded by a function of the diameter of G. The following result was proved by Demaine and Hajiaghayi [11]:

Theorem 8 (*[11]*). *For every apex graph H, the class of H-minor-free graphs has bounded local treewidth. More precisely, there is a function f such that any H-minor-free graph G of diameter D has treewidth at most $f(H)D$.*

In fact, it was proved by Eppstein [13] that a graph has bounded local treewidth if and only if it is apex-minor-free.

Theorem 9. Grundy Coloring *and* Weak Grundy Coloring *can be solved in time* $n^{O(\log^2 n)}$ *on apex-minor-free graphs of order n.*

Proof. We first consider Grundy Coloring. Any H-minor-free graph of order n has at most $f(H)n$ edges [30] for some function f; hence, by Theorem 7, we have $k \leqslant \Gamma(G) \leqslant c \log n$ for some constant c (otherwise, we have a NO-instance). As noted in Observation 2, any minimal k-witness is included in some distance-k neighborhood of G. Hence, we apply the $O(n^{3w^2})$-time algorithm of [37] for graphs of treewidth at most w: for every vertex v of G, apply it to the distance-k neighborhood of v. This is a subgraph of diameter at most $2k = O(\log n)$, and by Theorem 8 it has treewidth $w = O(\log n)$ as well. Hence $O(n^{3w^2}) = n^{O(\log^2 n)}$.

The same argumentation also works for Weak Grundy Coloring. Indeed, as pointed out before, the bound of Theorem 7 also holds for the weak Grundy number. Moreover, there is also an algorithm running in time $O(n^{3w^2})$ for Weak Grundy Coloring [37] (where the problem is called Iterated Dominating Removal). $\square$

In the light of Theorem 9, it is natural to ask whether Grundy Coloring can be solved in polynomial time on apex-minor-free graphs (or planar graphs)? Note that by Theorem 9, an NP-hardness result for Grundy Coloring on apex-minor-free graphs would contradict the ETH.

3.3. Weak Grundy Coloring *parameterized by k is in* FPT

We recall that Weak Grundy Coloring is NP-complete [17]. In this subsection, we show that Weak Grundy Coloring has an $O^*(2^{2^{O(k)}})$-time algorithm (Theorem 10). We will later show that this running time is essentially optimal under the ETH (Theorem 13).

Theorem 10. Weak Grundy Coloring *can be solved in time* $O^*(2^{2^{O(k)}})$*, where k is the number of colors.*

Proof. Let G be the input graph. We use the randomized color-coding technique of Alon et al. [1]. Let us first uniformly randomly color the vertices of G with integers between 1 and k, and denote by col the function giving the color of a vertex according to this random coloring. Then, we apply a pruning step, removing all vertices which violate the property of a weak Grundy coloring. That is, we remove each vertex v such that $\mathrm{col}(v) = c$ if $\exists c' < c, \neg\exists u \in N(v), \mathrm{col}(u) = c'$. Equivalently, we keep only the vertices v such that $\forall c \in [\mathrm{col}(v) - 1], \exists u \in N(v), \mathrm{col}(u) = c$. Note that is well possible that a vertex satisfying the condition at first, no longer satisfies it at a later point, after some of its neighbors are removed. Therefore, we apply the pruning until all the vertices satisfy the condition. If there is still a vertex colored with k after this pruning step, then, by construction, there is a weak Grundy coloring achieving color k in G (by coloring first the vertices v such that $\mathrm{col}(v) = 1$, then the vertices v such that $\mathrm{col}(v) = 2$, and so on, up to k).

If there is no weak Grundy (minimal) k-witness, this computation always rejects. Otherwise, it accepts only if a witness is well-colored by the random coloring. By Observation 3, a weak Grundy k-witness (as a Grundy k-witness) has size at most 2^{k-1}. At worst, there is a unique weak Grundy witness of size 2^{k-1} admitting a unique coloring. The probability to find this witness in one trial is $\frac{1}{k^{2^{k-1}}}$. Therefore, by repeating the previous step $\log(\frac{1}{\varepsilon})k^{2^{k-1}}$ times, we find a solution with probability at least $1 - \varepsilon$, for any $\varepsilon > 0$. Overall, the running time is $O(k^{2^{k-1}}(n+m)n) = O^*(2^{2^{O(k)}})$. □

We observe that the algorithm of Theorem 10 can be derandomized using so-called *universal coloring families* [1].

Unfortunately, the approach used to prove Theorem 10 does not work for Grundy Coloring because we have no guarantee that the color classes are independent sets.

3.4. Grundy Coloring *parameterized by k is in* FPT *on special graph classes*

For each fixed k, Grundy Coloring can be solved in polynomial time [39] and thus Grundy Coloring parameterized by the number k of colors is in XP. However (unlike Weak Grundy Coloring, as seen in Theorem 10), it is unknown whether Grundy Coloring is in FPT when parameterized by k. We will next show that it is indeed the case when restricting the instances to H-minor-free, chordal and claw-free graphs.

Theorem 11. Grundy Coloring *parameterized by the number of colors is in* FPT *for the class of graphs excluding a fixed graph H as a minor.*

Proof. By Observation 1, G contains a minimal k-witness H as an induced subgraph if and only if $\Gamma(G) \geqslant k$. By Observation 3, a minimal k-witness has at most 2^{k-1} vertices. So, the number of minimal k-witnesses (up to isomorphism) is bounded by a function of k. Besides, H-Induced Subgraph Isomorphism is in FPT when parameterized by $|V(H)|$ on graphs excluding H as a minor [15]. Therefore, one can check if $\Gamma(G) \geqslant k$ by solving H-Induced Subgraph Isomorphism for all minimal k-witnesses H. □

We have the following corollary of the algorithm of Telle and Proskurowski [37]. Note that Grundy Coloring is NP-complete on chordal graphs [35].

Theorem 12. *Let $\mathcal{C}$ be a graph class for which every member G satisfies $tw(G) \leqslant f(\Gamma(G))$ for some function f. Then,* Grundy Coloring *parameterized by the number of colors is in* FPT *on $\mathcal{C}$. In particular,* Grundy Coloring *is in* FPT *on chordal graphs.*

Proof. Since Grundy Coloring is in FPT for parameter combination of the number of colors and the treewidth [37], the first claim is immediate. Moreover $\omega(G) \leqslant \Gamma(G)$, hence if $tw(G) \leqslant f(\omega(G))$ we have $tw(G) \leqslant f(\Gamma(G))$. For any chordal graph G, $tw(G) = \omega(G) - 1$ [6]. □

The following shows that, unlike the classical Coloring problem, which remains NP-hard on degree 4 graphs, Grundy Coloring is FPT when parameterized by the maximum degree $\Delta(G)$.

Proposition 1 (*[35]*). Grundy Coloring *is in* FPT *when parameterized by the maximum degree $\Delta(G)$.*

Proof. By Observation 2, one can enumerate every distance-k neighborhood of each vertex, test every k-coloring of this neighborhood, and check if it is a valid Grundy k-coloring. Every such neighborhood has size at most $\Delta^{k+1} \leqslant \Delta^{\Delta+2}$ since by Observation 4, $k \leqslant \Delta + 1$. Finally, there are at most k^a k-colorings of a set of a elements. Therefore, Grundy Coloring can be solved in time $O\left(nk^{\Delta^{k+1}}\right) = n\Delta^{\Delta^{O(\Delta)}}$ for graphs of maximum degree Δ. □

We have the following corollary of Proposition 1. Note that Grundy Coloring is NP-complete on claw-free graphs [38].

Corollary 2. *Let $\mathcal{C}$ be a graph class for which every member G satisfies $\Delta(G) \leqslant f(\Gamma(G))$ for some function f. Then,* Grundy Coloring *parameterized by the number of colors is in* FPT *for graphs in $\mathcal{C}$. In particular, this holds for the class of claw-free graphs.*

Proof. The first part directly follows from Proposition 1. For the second part, consider a claw-free graph G and a vertex v of degree $\Delta(G)$ in G. Since G is claw-free, the subgraph induced by the neighbors of v has independence number at most 2, and hence $\Gamma(G) \geqslant \chi(G) \geqslant \chi(N(v)) \geqslant \frac{\Delta(G)}{2}$. □

4. Negative results

In this section, we present our algorithmic lower bounds.

4.1. A lower bound for Weak Grundy Coloring and Grundy Coloring under the ETH

We now present two similar reductions that (under the ETH) rule out algorithms for Weak Grundy Coloring and Grundy Coloring with a running time that is sub-double-exponential in k and sub-exponential in the instance size. In particular, this shows that the FPT algorithm for Weak Grundy Coloring of Theorem 10 has a near-optimal running time, assuming the ETH.

The property "$k \leqslant 1 + w \log n$" (which also holds for weak Grundy colorings [37]), entails that a running time $O^*(2^{2^{o(\frac{k}{w})}})$ is in fact subexponential-time $2^{o(n)}$. Therefore, if a subexponential-time algorithm (in the number of vertices) is proven unlikely, we would immediately obtain the conditional lower bound of $O^*(2^{2^{o(\frac{k}{w})}})$. Though, it is unclear whether the reductions from the literature on Grundy colorings allow to rule out a subexponential-time algorithm for Grundy Coloring (or Weak Grundy Coloring) under ETH. More importantly, what we prove next in Theorem 13 is a stronger lower bound, since the treewidth disappears in the denominator of the second exponent.

Theorem 13. *If* Weak Grundy Coloring *or* Grundy Coloring *is solvable in time $O^*(2^{2^{o(k)}}2^{o(n+m)})$ on graphs with n vertices and m edges, then the ETH fails.*

Proof. We first give the reduction for Weak Grundy Coloring.

In Monotone 3-NAE-SAT, being given a 3-CNF formula without negation, one is asked to find a truth assignment such that every clause contains a true literal and a false literal. The Monotone 3-NAE-SAT problem (also called Positive 3-NAE-SAT) with n variables and m clauses, is not solvable in time $2^{o(n+m)}$, unless the ETH fails (see for instance Lemma 3.12. in [25]). More precisely, authors of [25] present a reduction from 3-SAT to Monotone 3-NAE-SAT producing instances with $O(n+m)$ variables and clauses, which together with the Sparsification Lemma of Impagliazzo et al. [24] gives the claimed result.

We now build from an instance of Monotone 3-NAE-SAT $\mathcal{C} = \{C_1, \dots, C_m\}$ over the variables $X = \{x_1, \dots, x_n\}$, an equivalent instance of Weak Grundy Coloring with $O(n + m)$ vertices and clauses, and $k := \lceil \log m \rceil + 5$.

We remove, from a binomial tree T_k, rooted at r, m dominant subtrees T_3. This is possible since the number of such subtrees is $2^{k-3-2} = 2^{\lceil \log m \rceil + 5 - 3 - 2} = 2^{\lceil \log m \rceil} \geqslant m$. We call T the tree that we obtain by this process. We denote by $f_1, \dots, f_m$ the parents of those removed subtrees, and we link, for each $j \in [m]$, f_i to a new vertex $v(C_j)$ representing the clause C_j. For each $i \in [n]$, we add a star $K_{1,n}$, whose center is denoted by c and whose leaves are denoted by $v(x_i)$, and that represents the variables. We link each vertex $v(x_i)$ to vertex $v(C_j)$ if variable x_i appears in clause C_j. This ends the construction of the graph G.

Let us first show that $\Gamma'(G) = k$ if and only if r can be colored k. By Observation 4, the only vertices that can (potentially) be colored with color k are r, $r(k-1)$, c, and the $v(x_i)$'s. We already remarked that if $r(k-1)$ can be colored k, then, so does r (Lemma 1). What remains to prove is that neither c nor any of the $v(x_i)$'s cannot be colored k. The neighbors of a vertex $v(x_i)$ are c and some vertices $v(C_j)$, whose degree is bounded by 4 (recall that the clauses contain at most three variables). Thus, $v(x_i)$ can have in its neighborhood at most six distinct colors, and its color can be at most 7. Similarly, the neighbors of c are the $v(x_i)$'s, so the color of vertex c can be at most 8. We can assume that $\lceil \log m \rceil > 3$ (and, $k > 8$) since otherwise the instance is of constant size. Therefore, $\Gamma'(G) = k$ if and only if r can be colored k, which means that, by applying Lemma 2 with induced subtree T, we have $\Gamma'(G) = k$ if and only if $v(C_j)$ can be colored 3, for each $j \in [m]$, without first coloring any of the f_j's.

Now, suppose that $\mathcal{C}$ is satisfiable. Let ψ be a satisfying truth assignment of $\mathcal{C}$. Then, we can color each vertex $v(C_j)$ with color 3 in the following way. We first color c with color 1. Then, for each $i \in [n]$, we color $v(x_i)$ with 1 if x_i is set to false by ψ, and with 2 if it is set to true. Recall that the weak Grundy coloring does not need to be proper. As each clause C_j has at least one variable x_{i_1} set to true and at least one variable x_{i_2} set to false, $v(C_j)$ has in its neighborhood a vertex $v(x_{i_1})$ colored 2 and a vertex $v(x_{i_2})$ colored 1. Hence, $v(C_j)$ can be colored 3; moreover, we have not colored any vertex f_j, and we are done.

Conversely, suppose that $v(C_j)$ can be colored 3, for each $j \in [m]$, without coloring first any of the f_j's. Then, in the neighborhood of each $v(C_j)$ deprived of the f_j's, there should be one vertex $v(x_{i_1})$ colored 2 and one vertex $v(x_{i_2})$ colored 1. Therefore, the truth assignment ψ setting x_i to true if $v(x_i)$ has been colored 2 and to false if $v(x_i)$ has been colored 1 or has not been colored, satisfies $\mathcal{C}$.

In conclusion, we showed that $\Gamma'(G) = k$ if and only if $\mathcal{C}$ is satisfiable. The number N of vertices of the graph G is bounded by $n+1+2^{\lceil \log m \rceil + 4} \leqslant n+32m+1 = O(n+m)$. The number of edges M is bounded by $n+3m+2^{\lceil \log m \rceil + 4} \leqslant n+35m = O(n+m)$

Thus, solving Weak Grundy Coloring in time $O^*(2^{2^{o(k)}}2^{o(N+M)}) = O^*(2^{o(m)}2^{o(n+m)}) = O^*(2^{o(n+m)})$ would solve Monotone 3-NAE-SAT in subexponential-time, disproving the ETH.

For Grundy Coloring, we use a similar reduction by replacing the star $K_{1,n}$ encoding the variables by a matching of n edges $v(\neg x_i)v(x_i)$ where $v(\neg x_i)$ is a new vertex having only one neighbor: $v(x_i)$. Then, the proof carries over: in a Grundy coloring, one could color $v(x_i)$ with color 1 or 2, by first coloring $v(\neg x_i)$ with color 1. $\square$

The behavior shown by Weak Grundy Coloring is rare, and up to our knowledge, the only other known example for which an $O^*(2^{2^{O(k)}})$ is optimal under the ETH (with k the natural parameter) is the Edge Clique Cover problem [10]. For the Edge Clique Cover problem, where one wants to cover all the edges of a graph by a minimum number k of cliques, only an algorithm running in time $O^*(2^{2^{o(k)}}2^{o(n)})$ would disprove ETH. The number of edges in the produced instance of Edge Clique Cover has to be superlinear. Indeed, otherwise the maximum clique would be of constant size, and the parameter k would be at least linear in the number of vertices n, when it should in fact be logarithmic in n. Therefore, Weak Grundy Coloring seems to be the first problem for which an $O^*(2^{2^{o(k)}}2^{o(n+m)})$-algorithm is shown to be unlikely, while an $O^*(2^{2^{O(k)}})$-algorithm exists.

4.2. *Lower bound on the treewidth dependency for* Grundy Coloring *and* Weak Grundy Coloring

Let us recall that the algorithm for Grundy Coloring and Weak Grundy Coloring running in time $n^{O(w^2)}$ of Telle and Proskurowski comes from a $2^{O(wk)}n$-algorithm and the fact that $k \leqslant w \log n + 1$ [37].

An interesting observation is that an algorithm for Grundy Coloring or Weak Grundy Coloring running in time $O^*(k^{O(w)}) = O^*(2^{O(w \log k)})$, where w is the treewidth of the input graph, would imply an FPT algorithm for the parameter treewidth alone.

Observation 14. *If* Grundy Coloring *or* Weak Grundy Coloring *can be solved in time $O^*(k^{O(w)})$ on instances of treewidth w, then it can be solved in time $O^*(2^{O(w \log w)})$.*

Proof. Since, as mentioned before, $k \leqslant w \log n + 1$ [37] and using the fact that $\forall x, y > 0, (\log x)^y \leqslant y^{2y}x$, we have $O^*(k^{O(w)}) = O^*(w^{O(w)}(\log n)^{O(w)}) = O^*(w^{O(w)}) = O^*(2^{O(w \log w)})$. $\square$

Note that there are k^w possible k-colorings of a bag of size w, hence an algorithm for Grundy Coloring or Weak Grundy Coloring running in time $O^*(k^{O(w)})$ could be based on dynamic programming over a tree decomposition (and would greatly improve over the running time of the algorithm of [37]). Although we do not know whether such an algorithm exists, we now show that, assuming the ETH, one cannot get a significantly better running time (even when replacing the treewidth by the larger parameter "feedback vertex set number"). The reduction has some similarities with the reduction from Theorem 13, but it is more involved since we need to additionally lower the value of the treewidth.

Theorem 15. *If* Grundy Coloring *or* Weak Grundy Coloring *is solvable in time $O^*(2^{o(w \log w)})$ on graphs with treewidth (even, feedback vertex set number) at most w, then the ETH fails.*

Proof. We describe the proof for Grundy Coloring, but the same proof also works for Weak Grundy Coloring.

We build from an instance of SAT an equivalent instance of Grundy Coloring with subexponentially many vertices and sublinear feedback vertex set number. We rely on the grouping technique (similarly to [29]) that uses the fact that the number of permutations over a slightly sublinear number of elements is still exponential. We also make multiple applications of Lemma 2.

Let $\mathcal{C} = \{C_1, \ldots, C_m\}$ be the m clauses of an instance of SAT over the set of variables $X = \{x_1, \ldots, x_n\}$. Let q be a positive integer that we will fix later. We partition arbitrarily X into q sets $X_1, \ldots, X_q$ called *groups*, each of size at most $\lceil \frac{n}{q} \rceil$. A *group assignment* is a truth assignment of the variables of X_h for some $h \in [q]$. A group assignment *satisfies* a clause if it sets to true at least one of its literals (even if some variables of the clause are not instantiated). By potentially adding dummy variables, we may assume that $|X_h| = \lceil \frac{n}{q} \rceil$, for each $h \in [q]$. We also fix an arbitrary order of the variables within each group X_h, so that an assignment of X_h can be seen as a word of $\{0, 1\}^{\lceil \frac{n}{q} \rceil}$. Let $t = \lceil 3n/(q \log \frac{n}{q}) \rceil$ and recall that $\mathfrak{S}_t$ is the symmetric group. We fix an arbitrary one-to-one function $\zeta : \{0, 1\}^{\lceil \frac{n}{q} \rceil} \to \mathfrak{S}_t$ mapping a group assignment to a permutation over t elements. Such a function exists, since $|\mathfrak{S}_t| = t! > (\frac{t}{3})^t \geqslant 2^{3n(\log \frac{n}{q} - \log\log \frac{n}{q})/(q \log \frac{n}{q})} > 2^{\lceil n/q \rceil}$. Finally, we set $s = \lceil \log m \rceil + 2t + 4$.

We now describe the instance graph G of Grundy Coloring. We remove, from a binomial tree T_s rooted at r, m (arbitrary) dominant subtrees T_{t+2}. This is possible since, in T_s, there are $2^{s-t-4} = 2^{\lceil \log m \rceil + t} \geqslant m$ dominant trees T_{t+2}. We denote by $f_1, \ldots, f_m$ the m parents of those m removed subtrees. We call T the tree that we have obtained so far. For each clause C_j ($j \in [m]$) and for each group assignment τ (of some group X_h) satisfying C_j, we add a vertex $v(j, \tau)$ that we link to f_j. We denote by I_j the set of vertices $v(j, \cdot)$. Vertex $v(j, \tau)$ also becomes the *root* of a binomial tree T_{t+2} from which we remove the dominant subtree of each of its children (except for the child $v(j, \tau)(1)$ which is a leaf and therefore has no dominant subtree). We call that tree $T(j, \tau)$. Now, for each group X_h ($h \in [q]$), we add a clique $S_h = \{s_h^1, \ldots, s_h^t\}$ on t vertices. For each vertex $v(j, \tau)$, if τ is an assignment of the group X_h (for some $h \in [q]$) and $\sigma = \zeta(\tau)$, we link $v(j, \tau)(p + 1)$ to $s_h^{\sigma(p)}$, for each $p \in [t]$.

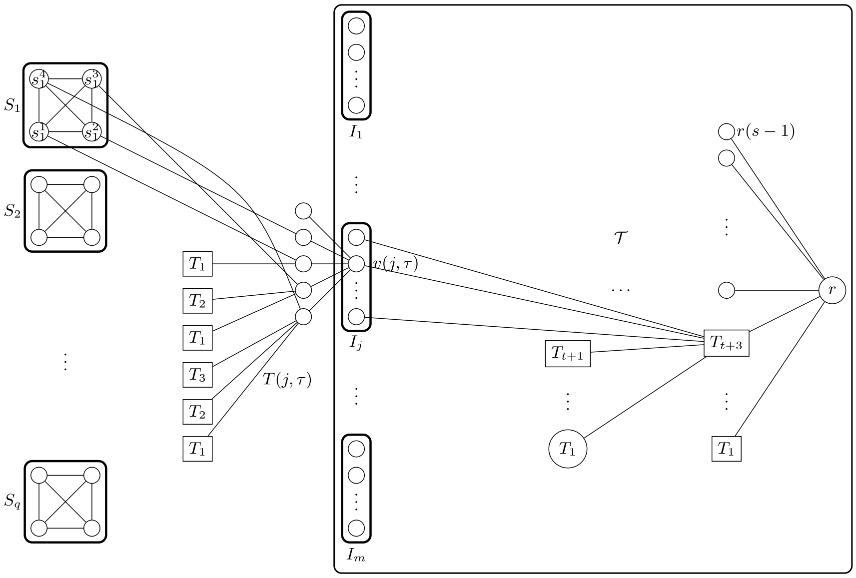

Fig. 3. A sample of the construction of graph G. The edges incident to the rectangular boxes containing T_is are only incident to the root of the tree. For the sake of readability, only one $T(j, \tau)$ is represented. Here, $v(j, \tau)$ represents an assignment of group X_1 mapped to the permutation $\sigma = (12)$.

This ends the construction of graph G (see Fig. 3 for an illustration). The number N of vertices of G is upper-bounded by $mq2^{\lceil \frac{n}{q} \rceil}2^{t+1} + 2^{s-1} + qt = O(mq2^{2t+\frac{n}{q}})$. The set $\bigcup_{h\in[q]} S_h$ is a feedback vertex set of G of size qt.

We now show that $\mathcal{C}$ is satisfiable if and only if $\Gamma(G) \geqslant s$. The proof goes as follows:

(1) $\Gamma(G) \geqslant s$ if and only if r, the root of T, can be colored with color s;

(2) by Lemma 2 on T, this is equivalent to color a vertex in each I_j with color $t + 2$;

(3) by Lemma 2 applied to the set of all trees $T(j, \tau)$, this is equivalent to a property $(\mathcal{P})$ (that we will define later) on the coloring of the cliques S_h's;

(4) $\mathcal{C}$ is satisfiable implies $(\mathcal{P})$;

(5) $(\mathcal{P})$ implies $\mathcal{C}$ is satisfiable.

First, we show the equivalence (1) that $\Gamma(G) \geqslant s$ if and only if r can be colored s. Assume that $\Gamma(G) \geqslant s$ (the other implication is trivial). By Observation 4, the only vertices (besides r) whose degree are (or at least may be) sufficient to be colored s are $r(s-1)$ (but we already noticed in Lemma 1 that $r(s-1)$ can be colored s if and only if this is the case for r) and the vertices of the cliques S_h's. The vertices in $N(S_h)$ have degree at most $t + 1$, hence their color can be at most $t + 2$. Thus, the number of distinct colors that a vertex of S_h can see in its neighborhood is at most $t + 2 + (t - 1) = 2t - 1$. Hence, its color cannot exceed $2t$, which is strictly smaller than s. Hence, r (or $r(s-1)$) has to be the vertex colored s.

To see that (2) holds, observe that, by Lemma 2 applied to the induced tree T and the set of parents $F = \{f_1, \dots, f_m\}$ of removed subtrees, $\Gamma(G) \geqslant s$ if and only if there is a Grundy coloring of $G - T$ coloring at least one vertex of I_j with color $t+2$, without coloring any vertex of any I_j with color $t + 3$. This latter condition can be omitted, since, in order to color a vertex with color $t + 2$, first coloring other vertices with color $t + 3$ or more is not helpful. We can now remove T from the graph G and equivalently ask if one can color with $t + 2$ at least one vertex in each set I_j ($j \in [m]$) in this new graph G'.

For each $j \in [m]$, for every vertex $v(j, \tau) \in I_j$, we apply Lemma 2 with the induced tree $T(j, \tau)$ and the set of parents $\{v(j, \tau)(2), \dots, v(j, \tau)(t + 1)\}$: $v(j, \tau)$ can be colored with color $t+2$ if and only if $s_h^{\sigma(p)}$ can be colored with p, for each $p \in [t]$, without coloring first any vertex of $T(j, \tau)$ (where $\sigma = \zeta(\tau)$ and τ is an assignment of the group X_h). As S_h is a t-clique, receiving each color from 1 to t cannot benefit from coloring vertices of $N(S_h)$ first. Thus, we will assume that all the S_h's are colored first.

We call $(\mathcal{P})$ the property:
$\forall j \in [m]$, $\exists v(j, \tau) \in I_j$, such that $\forall p \in [t]$, $s_h^{\zeta(\tau)(p)}$ has color p.

So far, we have shown that $\Gamma(G) \geqslant s$ if and only if $(\mathcal{P})$ holds. We now show that $(\mathcal{P})$ holds if and only if $\mathcal{C}$ is satisfiable.

Assume $\mathcal{C}$ is satisfiable. Let ψ be a satisfying global assignment. Let τ_h be the projection of ψ to X_h for each $h \in [q]$, and $\sigma_h = \zeta(\tau_h)$. We color the cliques S_h's such that $s_h^{\sigma_h(p)}$ is colored p, for each $p \in [t]$; that is, we first color $s_h^{\sigma_h(1)}$, then $s_h^{\sigma_h(2)}$, and so on, up to $s_h^{\sigma_h(t)}$. Now, for each clause C_j, there is a literal of C_j which is set to true by ψ. Say, this literal is on a variable of X_h for some $h \in [q]$. Then, the group assignment τ_h satisfies C_j. Therefore, for each $j \in [m]$, vertex $v(j, \tau_h) \in I_j$ exists and $\forall p \in [t]$, $s_h^{\zeta(\tau_h)(p)}$ has color p.

Assume now that the S_h's has been colored first, and such that $(\mathcal{P})$ holds. For each $h \in [q]$, let σ_h be the permutation of $\mathfrak{S}_t$ such that $\sigma_h(p)$ is defined as the index in S_h of the vertex colored p. This is well-defined since the vertices of the clique S_h

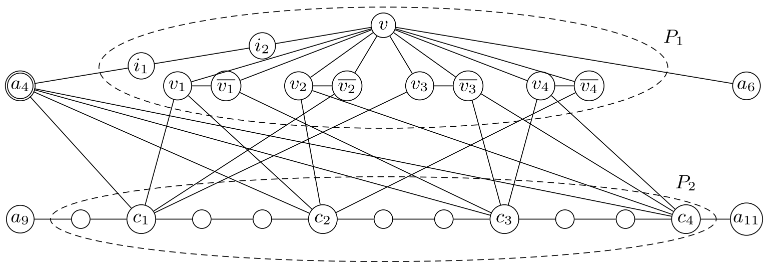

Fig. 4. P_1 and P_2 for the instance $\{x_1 \vee \neg x_2 \vee x_3\}$, $\{x_1 \vee x_2 \vee \neg x_4\}$, $\{\neg x_1 \vee x_3 \vee x_4\}$, $\{x_2 \vee \neg x_3 \vee x_4\}$.

have each color of $[t]$ exactly once. Now, let τ_h be the assignment of the group X_h such that $\sigma_h = \zeta(\tau_h)$. Group assignment τ_h is unique since ζ is one-to-one, and exists for $(\mathcal{P})$ to hold. Let ψ be the global assignment whose projection to each X_h is τ_h. By $(\mathcal{P})$, for each $j \in [m]$, there is vertex $v(j, \tau) \in I_j$ such that $\forall p \in [t]$, $s_h^{\zeta(\tau)(p)}$ has color p, for some $h \in [q]$. This vertex has in fact to be $v(j, \tau_h)$ since clique S_h has been colored such that $s_h^{\sigma_h(p)}$ has color p. So, the group assignment τ_h satisfies C_j. Therefore, ψ is a satisfying global assignment, hence $\mathcal{C}$ is satisfiable.

Suppose there is an algorithm solving Grundy Coloring on graphs with N vertices and feedback vertex set w in time $2^{o(w\log w)}N^c$ for some constant c. Recall that in G, we have $N = O(mq2^{2t+\frac{n}{q}})$ and $w \leqslant qt$. Assuming the ETH, there is a constant $s_3 > 0$ such that SAT (even 3-SAT) is not solvable in time $O(2^{s_3 n})$. Setting $q = \lceil \frac{2c}{s_3} \rceil$, one can solve SAT in time $O(2^{qt\log(qt)}(mq)^c 2^{2t} 2^{\frac{cn}{q}}) = O(2^{o(3n(\frac{\log n-\log\log n+\log\log q+\log 3}{\log n-\log q}))}(mq)^c 2^{\frac{6n}{q\log n/q}} 2^{\frac{s_3 n}{2}}) = O((mq)^c 2^{o(n)} 2^{o(n)} 2^{\frac{s_3 n}{2}})$ that is $O(2^{s_3 n})$, contradicting the ETH. $\square$

Note that in reduction of the proof of Theorem 15, we had $2^{o(s\log s)} = 2^{o(n+m)}$, so we even proved that Grundy Coloring and Weak Grundy Coloring cannot be solved in time $O^*(2^{o(k\log k)}2^{o(w\log w)})$ unless the ETH fails (where k is the number of colors).

4.3. Connected Grundy Coloring *is NP-hard for $k = 7$ colors*

Minimal connected Grundy k-witnesses, contrary to minimal Grundy k-witnesses (Observation 3), have arbitrarily large order: for instance, the cycle C_n of order n ($n > 4$, n odd) has a Grundy 3-witness of order 4, but its unique *connected* Grundy 3-witness is of order n: the whole cycle.

Observe that $\Gamma_c(G) \leqslant 2$ if and only if G is bipartite. Hence, Connected Grundy Coloring is polynomial-time solvable for any $k \leqslant 3$. However, we will now show that the problem is already NP-hard when $k = 7$, contrary to Grundy Coloring and Weak Grundy Coloring which are polynomial-time solvable whenever k is a constant (Corollary 1 and Theorem 10). Thus, in the terminology of parameterized complexity, Connected Grundy Coloring is para-NP-hard.

Theorem 16. Connected Grundy Coloring *is* NP-*hard even for* $k = 7$.

Proof. We give a reduction from 3-SAT 3-OCC, an NP-complete restriction of 3-SAT where each variable appears in at most three clauses [34], to Connected Grundy Coloring with $k = 7$. We first give the intuition of the reduction. The construction consists of a tree-like graph of constant order (resembling binomial tree T_6) whose root is adjacent to two vertices of a K_6 (this constitutes W) and contains three special vertices a_4, a_{21}, and a_{24} (which will have to be colored with colors 1, 3, and 2 respectively), a connected graph P_1 which encodes the variables and a path P_2 which encodes the clauses. One in every three vertices of P_2 is adjacent to a_4, a_{21} and a_{24}. To achieve color 7, we will need to color those vertices with color strictly greater than 3. This will be possible if and only if the assignment corresponding to the coloring of P_1 satisfies all the clauses.

We now formally describe the construction. Let $\phi = (X = \{x_1, \ldots, x_n\}, \mathcal{C} = \{C_1, \ldots, C_m\})$ be an instance of 3-SAT 3-OCC where no variable appears always as the same literal. $P_1 = (\{i_1, i_2, v\} \cup \{v_i, \overline{v_i} \mid i \in [n]\}, \{\{i_1, i_2\}, \{i_2, v\}\} \cup \{\{v, v_i\} \cup \{v, \overline{v_i}\} \cup \{v_i, \overline{v_i}\} \mid i \in [n]\})$ consists of n triangles sharing the vertex v. $P_2 = (\{p_j \mid j \in [3m-1]\}, \{\{p_j, p_{j+1}\} \mid j \in [3m-2]\})$ consists of a path of length $3m-1$. For each $j \in [m]$ and $i \in [n]$, $c_j \overset{def}{=} p_{3j-1}$ is adjacent to v_i if x_i appears positively in C_j, and is adjacent to $\overline{v_i}$ if x_i appears negatively in C_j. For each $j \in [m]$, c_j is adjacent to a_4, a_{21}, and a_{24}.

Intuitively, setting a literal to true consists of coloring the corresponding vertices with 3. Therefore, a clause C_j is satisfied if c_j has a 3 among its neighbors. To actually satisfy a clause, one has to color c_j with 4 or higher. Thus, c_j must also see a 2 in its neighborhood. We will show that the unique way of doing so is to color p_{3j-2} with 2, so all the clauses have to be checked along the path P_2.

We give, in Fig. 5, a coloring of P_1 corresponding to a truth assignment of the instance SAT formula. One can check that when going along P_2 all the c_j's are colored with color 4.

The constant gadget W is depicted in Fig. 6. The waves between a_4 and a_6 and between a_9 and a_{11} correspond, respectively, to the gadgets encoding the variables (P_1) and the clauses (P_2) described above and drawn in Fig. 4. A connected Grundy

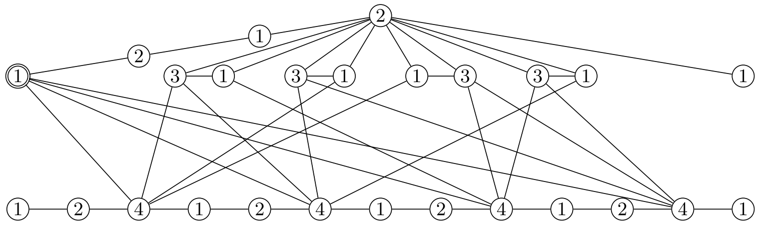

Fig. 5. A connected Grundy coloring such that all the c_j's are colored with color at least 4.

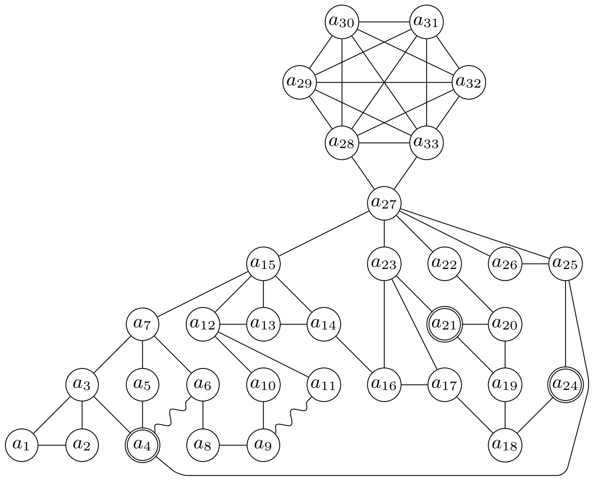

Fig. 6. The constant gadget. The doubly circled vertices are adjacent to all the c_j's ($j \in [m]$).

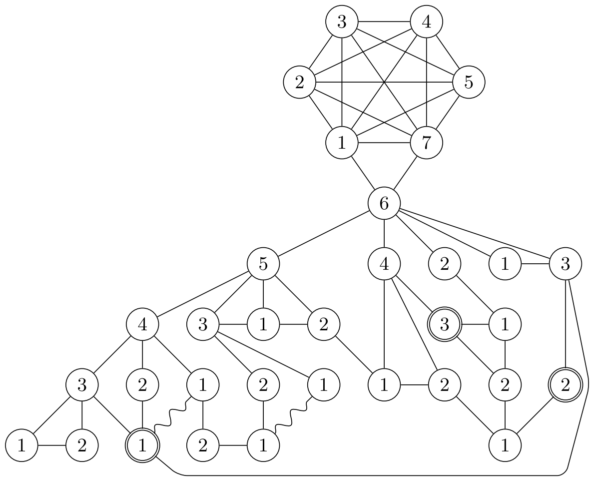

Fig. 7. A connected Grundy coloring of the constant gadget achieving color 7. The order is given by the sequence $(a_i)_{1 \leqslant i \leqslant 33}$.

coloring achieving color 7 is given in Fig. 7 provided that going from a_9 to a_{11} can be done without coloring any vertex c_j with color 2 or less.

In the following claims, we use extensively Observation 4 which states that a vertex with degree d gets color at most $d+1$. We observe that coloring a vertex z of degree d with color $d+1$ is only useful if the ultimate goal is to achieve color $d+1$. Indeed, for z to be colored with color $d+1$, all its neighbors have first to be colored (by each color from 1 to d), which means that z cannot be used thereafter. Moreover, if one wants to color a neighbor y of a vertex x in order to color x with a higher color, y cannot receive a color greater than its degree $d(y)$. Hence, the only vertices that could achieve color k are vertices of degree at least $k-1$ having at least one neighbor of degree at least $k-1$.

We call *doubly circled vertices* the special vertices a_4, a_{21} and a_{24} (they are doubly circled in the figures).

Claim 16.1. *To achieve color* 7, a_{27} *needs to be colored with color* 6 *(while for all* $i \in [28, 33]$, a_i *is still uncolored).*

Proof. One can achieve color 7 only in a vertex of degree at least 6 which has a neighbor of degree at least 6. There are $m+7$ vertices of degree at least 6: a_{28} and a_{33} (of degree 6), a_{27} (of degree 7), all the c_j's (of degree 8), v (of degree $2n+2$), a_{24} (of degree $m+2$), a_{21} (of degree $m+3$), and a_4 (of degree $m+4$).

As each vertex c_j is adjacent to a_4, a_{21} and a_{24}, we need to investigate the possibility of coloring with color 7, a vertex c_j, a_4, a_{21}, or a_{24}. A vertex c_j has two neighbors of degree 2 (p_{3j-2} and p_{3j}; or p_{3m-2} and a_{11} in the special case of c_m), three neighbors of degree at most 4 (the three vertices corresponding to the literals of C_j) since a literal has at most two occurrences, and three vertices of degree more than $m+2$ (a_4, a_{21}, and a_{24}). So, if no doubly circled vertex is colored yet, a vertex c_j can be colored with a color at most 5. And if some doubly circled vertices are already colored but with always the same color, a vertex c_j can be colored with a color at most 6 (when the shared color of the doubly circled vertices is 5).

Let us show that the three doubly circled vertices a_4, a_{21}, and a_{24} cannot take two different colors both greater than or equal to 5. Indeed, suppose that two of those three vertices are colored with colors p and q such that $p < q$ and $p, q \geqslant 5$. The doubly circled vertex colored with color q must have a vertex colored p in its neighborhood, but that color p cannot come from a c_j (since the vertex colored p is adjacent to the c_j's). Thus, this color p must come from another neighbor. But, among all the neighbors of the doubly circled vertices which are not a vertex c_j, no vertex is of degree at least 5, a contradiction.

From the last two paragraphs, we conclude that none of the vertices a_4, a_{21}, a_{24}, and the c_j's can receive color 7.

The only other pairs of adjacent vertices both of degree at least 6 are the pairs of the triangle formed by a_{27}, a_{28} and a_{33}. We observe that a_{27} is a cut-vertex whose removal disconnects the clique K_6 from the rest of the graph. Hence, in a connected Grundy coloring, a_{27} cannot get a color higher than 6 since its degree in one part of this cut is 2 and in the other part its degree is 5. Vertex a_{33} (or by symmetry a_{28}) can be colored with color 7, but then a_{27} has to be colored with color 6 otherwise it will lack a vertex colored 6 in its neighborhood. The conclusion is that the only way to achieve color 7 is to color a_{27} with color 6. □

Claim 16.2. *Vertices* a_{26}, a_{22}, a_{25}, a_{23}, a_{15} *must receive color* 1, 2, 3, 4, 5 *respectively.*

Proof. By Claim 16.1, a_{27} must be colored with color 6 before the clique K_6 is colored. Thus, the five neighbors of a_{27} which are not in the clique K_6 must get all the colors from 1 to 5. Among those neighbors, the only vertex with degree 5 is a_{15}, so this vertex must get color 5. Vertices a_{23} and a_{25} both have degree 4 but for connectivity reasons a_{26} cannot be colored before a_{25}, so a_{25} cannot get a color higher than 3. Thus, a_{23} must get color 4. Vertex a_{22} can bring a 1 or a 2 to a_{27} while the pair (a_{25}, a_{26}) can only bring the combinations $(1, 2)$, $(2, 1)$ or $(3, 1)$. Thus, the unique way to bring 1, 2 and 3 to a_{27} is that a_{25} is colored 3, a_{26} is colored 1, and a_{22} is colored 2. □

Claim 16.3. *Vertex* a_7 *must receive color* 4.

Proof. By Claim 16.2, a_{15} has to receive color 5, so one of its four neighbors (apart from a_{27}) must receive color 4. Only a_7 and a_{12} have degree 4. But a_{12} cannot be colored 4 since its three neighbors a_{10}, a_{11}, and a_{13} (apart from a_{15}) have only one neighbor which is neither a_{12} nor a_{15}, so none of these vertices can bring color 3 to a_{12}. □

Claim 16.4. *Vertex* a_3 *must receive color* 3.

Proof. By Claim 16.3, a_7 must be colored 4. Thus, one of its three neighbors a_3, a_5, and a_6 (apart from a_{15}) must receive color 3. Vertices a_3 and a_6 have two neighbors apart from a_7. But if a_6 is colored with color 3, then a_4 must be colored 3 to let colors 1 and 2 available for a_3 and a_5. In that case, a_3 and a_5 would both receive color 1. Another attempt is to color a_1 (or a_2) with 1, a_3 with 2 but then a_4 has to be colored 1 and a_5 can no longer be colored 1. Hence, only a_3 can be colored with 3. □

Claim 16.4 has further consequences: we must start the connected Grundy coloring by giving colors 1 and 2 to a_1 and a_2. The only follow-up, for connectivity reasons, is then to color a_3 with color 3 and a_4 with color 1. Thus, vertices a_5 and a_6 have to be colored with colors 2 and 1 respectively (so that a_7 can be colored 4). As, by Claim 16.2, a_{25} must receive color 3, a_{24} must receive color 2 (since a_4 has already color 1), so a_{18} must be colored 1.

Claim 16.5. *Vertex* a_{21} *must receive color* 3.

Proof. By Claim 16.2, a_{23} must get color 4, so its three neighbors apart from a_{27} must receive colors 1, 2 and 3. As a_{20} must be colored 1 (in order to color a_{22} with color 2), a_{21} will be colored 2 or 3. Suppose a_{21} is colored 2. Then, $\{a_{16}, a_{17}\}$ must be colored 1 and 3. Vertex a_{17} cannot be colored 1 since a_{18} must get color 1, so a_{16} must get color 1 and a_{17}, color 3. In that case, a_{17} lacks a vertex colored 2 in its neighborhood, and therefore cannot be colored 3. So, a_{21} has to be colored 3 and a_{19} has to be colored 2 (since a_{20} has to get color 1). □

A further consequence of Claim 16.5 is that a_{16} must be colored 2 and a_{17} must be colored 1 (the reverse being impossible, since a_{18} has to be colored 1). More importantly, we have now established that all the colored c_j's (for each $j \in [m]$) have to be colored with color 4 or higher. Indeed, we recall that the three doubly circled vertices (adjacent to all the c_j's) a_4, a_{21}, and

a_{24} must respectively get color 1, 3, and 2. In particular, after having colored a_1 up to a_4, we cannot short-cut to P_2 since it will color a c_j with 2, so we have to color i_1 with 2, i_2 with 1, and v with 2. As v must be colored with color 2, none of the vertices encoding the literals can have color 2, so, again, we cannot short-cut from P_1 to P_2 otherwise, we would color a c_j with 2. Then, we can partly (or entirely) color P_1 but we have to color a_6 with 1, a_8 with 2, and a_9 with 1. As a_9 is forced to get color 1, a_{10} has to give a 2 to a_{12} and a_{11} is therefore forced to give color 1 to a_{12}.

Claim 16.6. *The unique way of coloring a_{11} with color 1 without coloring any vertex c_j with color 1, 2, or 3 is to color all the c_j's for each $j \in [m]$.*

Proof. We recall that the first four vertices to be colored are a_1, a_2, a_3, When going along the path from a_9 to a_{11}, the only vertex colored 2 which can be in the neighborhood of c_j is p_{3j-2}. Indeed, we recall that the vertices encoding literals cannot be colored 2 since they are all adjacent to v which is colored 2. By induction, as the only way to color vertex p_{3j-2} with color 2 before c_j is colored, is to color c_{j-1}, we have to color all the vertices in the path P_2. □

We remark that opposite literals are adjacent, so for each $i \in [n]$, only one of v_i and $\overline{v_i}$ can be colored with color 3. We interpret coloring v_i with 3 as setting x_i to true and coloring $\overline{v_i}$ with 3 as setting x_i to false.

Claim 16.7. *To color each c_j ($j \in [m]$) of the path P_2 with a color at least 4, the SAT formula must be satisfiable.*

Proof. Each c_j must have a vertex colored 3 in its neighborhood, but this vertex cannot be a_{21} since this vertex cannot be colored yet. We recall that a_{21} will be colored after a_{11} is colored. Thus, the vertex colored 3 can only belong to a set $\{v_i, \overline{v_i}\}$ encoding a literal l_i such that l_i is in C_j. Indeed, the neighbors p_{3j-2} and p_{3j} are of degree 2 and a_4 is already colored 1. Hence, there must be an assignment of the variables such that all the clauses of $\mathcal{C}$ are satisfied. As one cannot color both v_i and $\overline{v_i}$ with color 3, the coloring of P_1 does constitute a feasible assignment. □

So, to achieve color 7 in a connected Grundy coloring, the SAT formula must be satisfiable. The reverse direction consists of completing the coloring by giving a_{13} color 1 and a_{14} color 2, as shown in Figs. 5 and 7. □

5. Concluding remarks and questions

To conclude this article, we suggest some questions which might be useful as a guide for further studies.

We have given two $O^*(c^n)$ exact algorithms for Grundy Coloring and Weak Grundy Coloring with c a constant, but we do not know whether such an algorithm exists for Connected Grundy Coloring.

There is a gap between the $O^*(2^{O(wk)})$ algorithm of [37] and the lower bound of Theorem 15. Is Grundy Coloring in FPT when parameterized by the treewidth w? A simpler question is whether there is a better $O^*(f(k, w))$ algorithm (as noted in Observation 14, if $f(k, w) = k^{O(w)}$ we directly obtain an FPT algorithm for parameter w). It could also be simpler to first determine whether Grundy Coloring is in FPT when parameterized by the feedback vertex set number (it is easy to see that it is in FPT when parameterized by the vertex cover number).

Grundy Coloring (parameterized by the number of colors) is in XP, and we showed it to be in FPT on many important graph classes. Yet, the central question whether it is generally in FPT or W[1]-hard remains unsolved. A perhaps more accessible research direction is to settle this question on bipartite graphs.

It would also be interesting to determine the (classic) complexity of Grundy Coloring on interval graphs and chordal bipartite graphs (the latter question being asked in [35]). Also, we saw that the algorithm of [37] implies a quasi-polynomial algorithm for planar (even apex-minor-free) graphs, making it unlikely to be NP-complete on this class. Is there a polynomial-time algorithm for such graphs?

We also recall that the exact polynomial-time approximation complexity of Grundy Coloring and Weak Grundy Coloring is unknown; it is known that they admit no PTAS [17,27], but no $o(n)$-factor polynomial-time approximation algorithm is known. Recently, it was proved that for any $r > 1$, Grundy Coloring can be r-approximated in time $O^*(c^{n \log r/r})$ for some constant c, where n is the graph's order [7]. The approximation complexity of Connected Grundy Coloring has not yet been studied.

Regarding Connected Grundy Coloring, we showed that it remains NP-complete even for $k = 7$. As Connected Grundy Coloring is polynomial-time solvable for $k \leqslant 3$, its complexity status for $k = 4, 5, 6$ remains open. It would also be interesting to study Connected Grundy Coloring on restricted graph classes.

Acknowledgment

We thank Sundar Vishwanathan for sending us the paper [17].

References

[1] N. Alon, R. Yuster, U. Zwick, Color-Coding, J. ACM 42 (4) (1995) 844–856.
[2] J. Araujo, C. Linhares Sales, On the Grundy number of graphs with few P_4's, Discrete Appl. Math. 160 (18) (2012) 2514–2522.
[3] F. Benevides, V. Campos, M. Dourado, S. Griffiths, R. Morris, L. Sampaio, A. Silva, Connected Greedy Colourings, LATIN 2014, in: LNCS, vol. 8392, Springer, 2014, pp. 433–441.

[4] C. Berge, Graphs and Hypergraphs, North Holland, 1973.
[5] A. Björklund, T. Husfeldt, P. Kaski, M. Koivisto, Fourier meets MöBius: fast subset convolution, in: Proc. 39th Annual ACM Symposium on Theory of Computing, in: STOC '07, ACM, 2007, pp. 67–74.
[6] H.L. Bodlaender, A tourist guide through treewidth, Acta Cybernet. 11 (1–2) (1993) 1–21.
[7] É. Bonnet, M. Lampis, V.T. Paschos, Time-approximation trade-offs for inapproximable problems, in: N. Ollinger, H. Vollmer (Eds.), 33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, OrlÉAns, France, in: LIPIcs, vol. 47, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016, pp. 22:1–22:14.
[8] G.J. Chang, H.-C. Hsu, First-fit chromatic numbers of d-degenerate graphs, Discrete Math. 312 (12–13) (2012) 2088–2090.
[9] C.A. Christen, S.M. Selkow, Some perfect coloring properties of graphs, J. Combin. Theory Ser. B 27 (1) (1979) 49–59.
[10] M. Cygan, M. Pilipczuk, M. Pilipczuk, Known algorithms for edge clique cover are probably optimal, SIAM J. Comput. 45 (1) (2016) 67–83.
[11] E.D. Demaine, M. Hajiaghayi, Equivalence of local treewidth and linear local treewidth and its algorithmic applications, in: Proc. 15th Annual ACM-SIAM Symposium on Discrete Algorithms, in: SODA '04, SIAM, 2004, pp. 840–849.
[12] R.G. Downey, M.R. Fellows, Fundamentals of Parameterized Complexity, Springer, 2013.
[13] D. Eppstein, Diameter and treewidth in minor-closed graph families, Algorithmica 27 (3) (2000) 275–291.
[14] P. Erdős, W.R. Hare, S.T. Hedetniemi, R. Laskar, On the equality of the Grundy and ochromatic numbers of a graph, J. Graph Theory 11 (2) (1987) 157–159.
[15] J. Flum, M. Grohe, Fixed-parameter tractability, definability, and model-checking, SIAM J. Comput. 31 (1) (2001) 113–145.
[16] F.V. Fomin, F. Grandoni, A.V. Pyatkin, A.A. Stepanov, Combinatorial bounds via measure and conquer: Bounding minimal dominating sets and applications, ACM Trans. Algorithms 5 (1) (2008). http://dx.doi.org/10.1145/1435375.1435384.
[17] N. Goyal, S. Vishwanathan, NP-completeness of undirected Grundy numbering and related problems. Manuscript, 1997.
[18] P.M. Grundy, Mathematics and games, Eureka 2 (1939) 6–8.
[19] A. Gyárfás, Z. Király, J. Lehel, Online 3-chromatic graphs - II critical graphs, Discrete Math. 177 (1–3) (1997) 99–122.
[20] A. Gyárfás, J. Lehel, Online and first fit colorings of graphs, J. Graph Theory 12 (2) (1988) 217–227.
[21] F. Havet, A.K. Maia, M.-L. Yu, Complexity of greedy edge-colouring. Research report RR-8171, INRIA, 2012.
[22] F. Havet, L. Sampaio, On the Grundy and b-chromatic numbers of a graph, Algorithmica 65 (4) (2013) 885–899.
[23] S.M. Hedetniemi, S.T. Hedetniemi, T. Beyer, A linear algorithm for the Grundy (coloring) number of a tree, Congr. Numer. 36 (1982) 351–363.
[24] R. Impagliazzo, R. Paturi, F. Zane, Which problems have strongly exponential complexity?, J. Comput. System Sci. 63 (4) (2001) 512–530.
[25] K. Jansen, F. Land, K. Land, Bounding the running time of algorithms for scheduling and packing problems, SIAM J. Discrete Math. 30 (1) (2016) 343–366.
[26] H.A. Kierstead, K.R. Saoub, First-Fit coloring of bounded tolerance graphs, Discrete Appl. Math. 159 (7) (2011) 605–611.
[27] G. Kortsarz, A Lower bound for approximating Grundy numbering, Discrete Math. Theoret. Comput. Sci. 9 (1) (2007).
[28] E.L. Lawler, A note on the complexity of the chromatic number problem, Inform. Process. Lett. 5 (3) (1976) 66–67.
[29] D. Lokshtanov, D. Marx, S. Saurabh, Known algorithms on graphs of bounded treewidth are probably optimal, in: Proc. 22nd Annual ACM-SIAM Symposium on Discrete Algorithms, in: SODA '11, SIAM, 2011, pp. 777–789.
[30] W. Mader, Homomorphieeigenschaften und mittlere Kantendichte von Graphen, Math. Ann. 174 (4) (1967) 265–268.
[31] J. Moon, L. Moser, On cliques in graphs, Israel J. Math. 3 (1) (1965) 23–28.
[32] N.S. Narayanaswamy, R.S. Babu, A note on first-fit coloring of interval graphs, Order 25 (1) (2008) 49–53.
[33] R. Niedermeier, Invitation to Fixed-Parameter Algorithms, Oxford University Press, 2006. http://dx.doi.org/10.1093/acprof:oso/9780198566076.001.0001.
[34] C.H. Papadimitriou, Computational Complexity, Addison-Wesley, 1994.
[35] L. Sampaio, Algorithmic Aspects of Graph Colouring Heuristics, (Ph.D. thesis,), University of Nice-Sophia Antipolis, 2012.
[36] G.J. Simmons, The ochromatic number of graphs, Graph Theory Newsl. 11 (6) (1982) 2–3.
[37] J.A. Telle, A. Proskurowski, Algorithms for vertex partitioning problems on partial k-Trees, SIAM J. Discrete Math. 10 (4) (1997) 529–550.
[38] M. Zaker, The Grundy chromatic number of the complement of bipartite graphs, Australas. J. Combin. 31 (2005) 325–329.
[39] M. Zaker, Results on the Grundy chromatic number of graphs, Discrete Math. 306 (23) (2006) 3166–3173.