b-Coloring Parameterized by Clique-Width

Lars Jaffke ✉
University of Bergen, Norway

Paloma T. Lima ✉
University of Bergen, Norway

Daniel Lokshtanov ✉
University of California Santa Barbara, CA, USA

Abstract

We provide a polynomial-time algorithm for b-Coloring on graphs of constant clique-width. This unifies and extends nearly all previously known polynomial-time results on graph classes, and answers open questions posed by Campos and Silva [Algorithmica, 2018] and Bonomo et al. [Graphs Combin., 2009]. This constitutes the first result concerning structural parameterizations of this problem. We show that the problem is FPT when parameterized by the vertex cover number on general graphs, and on chordal graphs when parameterized by the number of colors. Additionally, we observe that our algorithm for graphs of bounded clique-width can be adapted to solve the Fall Coloring problem within the same runtime bound. The running times of the clique-width based algorithms for b-Coloring and Fall Coloring are tight under the Exponential Time Hypothesis.

2012 ACM Subject Classification Mathematics of computing → Graph coloring

Keywords and phrases b-Coloring, clique-width, vertex cover, structural parameterization

Digital Object Identifier 10.4230/LIPIcs.STACS.2021.43

Related Version *Full Version*: https://arxiv.org/abs/2003.04254

1 Introduction

This paper settles open questions regarding the complexity of the b-Coloring problem on graph classes and initiates the study of its structural parameterizations. A *b-coloring* of a graph G with k colors is a partition of the vertices of G into k independent sets such that each of them contains a vertex that has a neighbor in all of the remaining ones. The *b-chromatic number* of G, denoted by $\chi_b(G)$, is the maximum integer k such that G admits a b-coloring with k colors. This notion was introduced by Irving and Manlove [29] to describe the behavior of the following color-suppressing heuristic for the Graph Coloring problem. We start with some proper coloring of the input graph G and try to iteratively suppress one of its colors. That is, for a given color c, we consider each vertex v of color c, and check if there is another color $c' \neq c$ available that does not appear in its neighborhood. If so, we assign vertex v the color c', observing that the coloring remains proper, and repeat this process for the remaining vertices of color c. If successful, we remove the color c from all vertices of G and decrease the number of colors by one. Once no color can be supressed by this procedure, the coloring at hand is a b-coloring of G, and in the worst case, this heuristic produces a coloring with $\chi_b(G)$ many colors.

Since then, the b-Coloring and b-Chromatic Number problems which given a graph G and an integer k ask whether G has a b-coloring with k colors and whether $\chi_b(G) \geq k$, respectively, have received considerable attention in the algorithms and complexity communities.[1] While these problems have been shown to be NP-complete in the general

[1] We would like to remark that the b-Coloring and b-Chromatic Number problems are not as closely related as the Graph Coloring and Chromatic Number problems: If a graph G has a b-coloring with k colors, then $\chi_b(G) \geq k$, but $\chi_b(G) \geq k$ does not imply the existence of a b-coloring with k colors.

case [29], as well as on bipartite graphs [32], co-bipartite graphs [6], chordal graphs [24], and line graphs [7], a lot of effort has been put into devising polynomial-time algorithms for these problems in various other classes of graphs. These include trees [29], claw-free block graphs [10], tree-cographs [6], and graphs with few P_4s, such as cographs and P_4-sparse graphs [5], P_4-tidy graphs [45], and $(q, q-4)$-graphs for constant q [9]. A common property shared by these graph classes is that they all have bounded *clique-width*.[2]

The main contribution of this work is an algorithm that solves b-COLORING (and b-CHROMATIC NUMBER) in polynomial time on graphs of constant clique-width. Besides unifying the above mentioned polynomial-time cases, this extends the tractability landscape of these problems to larger graph classes, and answers two open problems stated in the literature.

Over a decade ago, Bonomo et al. [5] asked whether their polynomial-time result for P_4-sparse graphs can be extended to distance-hereditary graphs. Havet et al. [24] answered the question negatively by providing an NP-completeness proof for chordal distance-hereditary graphs. We observe, however, that their proof has a flaw and while it does prove the claimed statement for chordal graphs, it unfortunately fails to do so for distance-hereditary graphs. Our polynomial-time algorithm for graphs of bounded clique-width in fact provides a positive answer to Bonomo et al.'s question, as distance-hereditary graphs have clique-width at most 3 [23]. In recent years, even subclasses of distance-hereditary graphs have received significant attention, for instance in the work of Campos and Silva [10]: they provide a polynomial-time algorithm for claw-free block graphs, and ask whether this result can be generalized to block graphs. Our algorithm provides a positive answer to this question as well. Moreover, it extends the known algorithm for $(q, q-4)$-graphs [9] (for constant q) to all (q, t)-graphs for constants q and t with $q \geq 4$, $t \geq 0$, and either $q \leq 6$ and $t \leq q-4$, or $q \geq 7$ and $t \leq q-3$, by a theorem to due Makowsky and Rotics [36].

Our algorithm runs in time $n^{2^{\mathcal{O}(w)}}$, where n denotes the number of vertices of the input graph which is given together with a clique-width w-expression. As consequences of results due to Fomin et al. [20] and Fomin et al. [21], we observe that b-COLORING parameterized by clique-width is W[1]-hard, and that the exponential dependence on w in the degree of the polynomial cannot be avoided unless the Exponential Time Hypothesis (ETH) fails. Concretely, an algorithm running in time $n^{2^{o(w)}}$ would refute ETH.

From the point of view of parameterized complexity, Panolan et al. [38] showed that b-CHROMATIC NUMBER parameterized by the number of colors is W[1]-hard. However, this problem may even be harder, since so far no XP-algorithm is known. Recently, Aboulker et al. [1] showed that the more restrictive b-CHROMATIC CORE problem parameterized by the number of colors (which has a brute-force XP-algorithm, see e.g. [18]) remains W[1]-hard.

It is therefore natural to ask which additional restrictions can be imposed to obtain parameterized tractability results. For instance, an open problem posed by Sampaio [41] (see also [43]) asks whether b-COLORING parameterized by the number of colors is FPT on chordal graphs. We answer this question in the affirmative, via Courcelle's Theorem [11] for bounded treewidth graphs. Other restricted cases that have been considered in the literature target specific numbers of colors that depend on the input graph. The DUAL b-COLORING problem, which asks if an input n-vertex graph has a b-coloring with $n-k$ colors, is FPT parameterized by k [25]. Moreover, deciding if a graph G has a b-coloring with $k = \Delta(G) + 1$

[2] To the best of our knowledge, the only polynomial-time result for graphs of unbounded clique-width so far concerns graphs of large girth. In particular, Campos et al. [8] showed that b-CHROMATIC NUMBER is polyomial-time solvable on graphs of girth at least 7.

colors, which is an upper bound on $\chi_b(G)$, is FPT parameterized by k [38, 41], while the case $k = \Delta(G)$ is XP and for every fixed $p \geq 1$, the case $k = \Delta(G) - p$ is NP-complete for $k = 3$ [30].

Another novelty aspect of our XP-algorithm parameterized by clique-width is that it is the first result about *structural parameterizations* of the b-Coloring and b-Chromatic Number problems. In all previously known polynomial-time cases the algorithms only work if the input graph has some prescribed structure. Our algorithm works on all graphs, albeit with a prohibitively slow runtime on graphs of large clique-width. In this vein, we round off our work with an FPT-result for another lead player among structural parameterizations, the *vertex cover number* of a graph; a parameter often referred to as the *Drosophila* of parameterized complexity.

Fall Coloring. A *fall coloring* is a special type of b-coloring where *every* vertex needs to have at least one neighbor in all color classes except its own. In other words, it is a partition of the vertex set of a graph into independent dominating sets. As a standalone notion, fall coloring has been introduced by Dunbar et al. [17]. However, since the corresponding Fall Coloring problem falls in the category of locally checkable vertex partitioning problems, it has been shown in earlier work of Telle and Proskurowski [44] to be FPT parameterized by the treewidth of the input graph, and by Heggernes and Telle [26] to be NP-complete for fixed number of colors. Fall Coloring remains hard further restricted to bipartite [33, 34, 42], chordal [42], or planar [34] graphs. On the other hand, even with unbounded number of colors, it is known to be solvable in polynomial time on strongly chordal graphs [35, 22], threshold graphs and split graphs [37]. In all of these cases, one simply checks whether the chromatic number of the input graph is equal to its minimum degree plus one. To the best of our knowledge, these are the only known polynomial-time cases. We adapt our algorithm for b-Coloring on graphs of bounded clique-width to solve Fall Coloring, and therefore show that the latter problem is as well solvable in time $n^{2^{\mathcal{O}(w)}}$, where w denotes the clique-width of a given decomposition of the input graph. By a simple reduction, we show that Fall Coloring is also W[1]-hard in this parameterization and that an $n^{2^{o(w)}}$-time algorithm for it would refute ETH.

Vertex Coloring Problems Parameterized by Clique-Width. We briefly touch on differences in the complexities of vertex coloring problems of graphs when parameterized by clique-width. While the standard Graph Coloring problem, asking for a proper coloring of the input graph, is XP-time solvable parameterized by clique-width [19, 46], some of its generalizations are NP-complete on graphs of constant clique-width. In the List Coloring problem we are given a graph G and for each of its vertices v a list $L(v)$ of colors, and the question is whether G has a proper coloring such that each vertex is assigned a color from its list. This problem is NP-complete on the (not disjoint) union of two complete graphs [31], and such graphs clearly have constant clique-width. In the related Precoloring Extension problem, we are given a graph, some of whose vertices already received a color, and the question is whether this coloring can be extended to a proper coloring of the entire graph. The following standard reduction from List Coloring, starting with a graph that is the union of two complete graphs, shows that this variant is NP-complete on graphs of constant clique-width as well. Take the graph G together with the lists $L(\cdot)$, and construct a graph H by adding to G, for each vertex $v \in V(G)$ and each color $c \notin L(v)$, a new vertex x_v^c which is adjacent only to v and assigned color c. It is not difficult to see that this precoloring of H can be extended to the remainder of its vertices if and only if G has a list coloring using the lists $L(\cdot)$. Moreover, adding pendant vertices to a graph does not increase its clique-width.

Belmonte et al. [3] recently showed that the Grundy Coloring problem, which asks for a linear order of the vertices that maximizes the number of colors used by the greedy coloring heuristic, is NP-complete on graphs of constant clique-width. This nicely contrasts our XP-algorithm for *b*-Coloring, since both the *b*-Coloring and the Grundy Coloring problems are rooted in the theoretical analysis of graph coloring heuristics.

Sketch of the algorithm. Let us discuss how we obtain our XP-algorithm parameterized by clique-width. First, we consider a branch decomposition of the input graph G of bounded *module-width* w which is equivalent to clique-width and has the following property. At each node t of the branch decomposition we have a subgraph G_t of G whose vertex set can be partitioned into at most w equivalence classes with respect to their neighborhood outside of G_t. For the purpose of our dynamic programming algorithm, it suffices to describe colorings by the way each of their color classes interacts with these equivalence classes. In the Graph Coloring problem, it is enough to describe a color class according to its intersection with the equivalence classes of G_t alone [19, 46] (see also [21]). For the *b*-Coloring problem, we additionally have to ensure that eventually, each color class indeed has a *b*-vertex. The challenge is to do so without explicitly remembering which color classes a vertex has already seen in its neighborhood – this would result in prohibitively large tables. We overcome this difficulty by a symmetry breaking trick that instead stores, for each color class, a *demand* to the future neighbors of the equivalence classes which – if fulfilled – guarantees that the *other* color classes can have *b*-vertices in the end.

Due to space restrictions, proofs of statements marked "♣" as well as several discussions are deferred to the full version.

2 Preliminaries

We use standard terminology and assume the reader to be familiar with basic notions in graph theory and parameterized complexity and refer to the books [4, 15] and [14, 16], respectively, for introductions; or to the attached full version. To avoid confusion, we clarify some notation. All graphs considered here are simple and finite. For a graph G we denote by $V(G)$ and $E(G)$ the vertex set and edge set of G, respectively. For a set of vertices $S \subseteq V(G)$, the *subgraph of G induced by S* is $G[S]$. A graph is called *subcubic* if all its vertices have degree at most three. A graph G is *connected* if for all 2-partitions (X, Y) of $V(G)$ with $X \neq \emptyset$ and $Y \neq \emptyset$, there is a pair $x \in X$, $y \in Y$ such that $xy \in E(G)$. A *connected component* of a graph is a maximal connected subgraph. In a tree T, the vertices of degree one are called the *leaves* of T, denoted by $\mathrm{L}(T)$, and the vertices in $V(T) \setminus \mathrm{L}(T)$ are the *internal vertices* of T. The *length* of a path is the number of its edges. For a graph G and a pair of vertices $u, v \in V(G)$, we denote by $\mathrm{dist}_G(u, v)$ the length of the shortest path between u and v in G. A graph G is called *distance-hereditary* if for each connected induced subgraph H of G, and each pair of vertices $u, v \in V(H)$, $\mathrm{dist}_H(u, v) = \mathrm{dist}_G(u, v)$. A tree T is called a *caterpillar* if it contains a path $P \subseteq T$ such that all vertices in $V(T) \setminus V(P)$ are adjacent to a vertex in P.

Let Ω be a set and $\sim$ an equivalence relation over Ω. For an element $x \in \Omega$ the *equivalence class of x*, denoted by $[x]$, is the set $\{y \in \Omega \mid x \sim y\}$. We denote the set of all equivalence classes of $\sim$ by $\Omega/\sim$.

The *Exponential Time Hypothesis (ETH)* is the following conjecture about the 3-SAT problem, which given a boolean formula ϕ in conjunctive normal form with clauses of size at most three asks whether there is a truth assignment to its variables that lets ϕ evaluate to true.

▶ **Conjecture** (ETH [27, 28]). *There is no algorithm that solves each instance of* 3*-SAT on* n *variables in time* $2^{o(n)}$.

Clique-Width, branch decompositions, and module-width. We first define clique-width, introduced by Courcelle, Engelfriet, and Rozenberg [12], and then the equivalent measure of *module-width* that we will use in our algorithm. The reason why we choose module-width over clique-width is that at each node of the decomposition it captures information that is very useful for coloring problems:

We keep the definition of clique-width slightly informal and refer to [12, 13] for more details. Let G be a graph. The *clique-width* of G, denoted by $\mathsf{cw}(G)$, is the minimum number of labels $\{1, \ldots, k\}$ needed to obtain G using the following four operations: (1) Create a new graph consisting of a single vertex labeled i. (2) Take the disjoint union of two labeled graphs G_1 and G_2. (3) Add all edges between pairs of vertices of label i and label j. (4) Relabel every vertex labeled i to label j.

▶ **Definition 1** (Branch decomposition). *Let G be a graph. A* branch decomposition *of G is a pair $(T, \mathcal{L})$ of a subcubic tree T and a bijection $\mathcal{L}\colon V(G) \to \mathrm{L}(T)$. If T is a caterpillar, then $(T, \mathcal{L})$ is called* linear branch decomposition. *If T is rooted, then we call $(T, \mathcal{L})$ a* rooted branch decomposition. *In this case, for $t \in V(T)$, we denote by T_t the subtree of T rooted at t, and we define $V_t := \{v \in V(G) \mid \mathcal{L}(v) \in \mathrm{L}(T_t)\}$, $\overline{V_t} := V(G) \setminus V_t$, and $G_t := G[V_t]$.*

▶ **Definition 2** (Module-width, [39, 40]). *Let G be a graph, and $(T, \mathcal{L})$ be a rooted branch decomposition of G. For each $t \in V(T)$, let $\sim_t$ be the equivalence relation on V_t defined as: $\forall u, v \in V_t\colon u \sim_t v \Leftrightarrow N_G(u) \cap \overline{V_t} = N_G(v) \cap \overline{V_t}$. The* module-width *of $(T, \mathcal{L})$ is* $\mathsf{mw}(T, \mathcal{L}) := \max_{t \in V(T)} |V_t/{\sim_t}|$. *The* module-width of G, *denoted by* $\mathsf{mw}(G)$, *is the minimum module width over all rooted branch decompositions of G.*

Let $(T, \mathcal{L})$ be a rooted branch decomposition of a graph G and let $t \in V(T)$ be a node with children r and s. We now describe an operator associated with t that tells us how the graph G_t is formed from its subgraphs G_r and G_s, and how the equivalence classes of $\sim_t$ are formed from the equivalence classes of $\sim_r$ and $\sim_s$. Concretely, we associate with t a bipartite graph H_t on bipartition $(V_r/{\sim_r}, V_s/{\sim_s})$ such that:

1. $E(G_t) = E(G_r) \cup E(G_s) \cup F$, where $F = \{uv \mid u \in V_r, v \in V_s, \{[u], [v]\} \in E(H_t)\}$, and
2. there is a partition $\mathcal{P} = \{P_1, \ldots, P_h\}$ of $V(H_t)$ such that $V_t/{\sim_t} = \{Q_1, \ldots, Q_h\}$, where for $1 \le i \le h$, $Q_i = \bigcup_{Q \in P_i} Q$. For each $1 \le i \le h$, we call P_i the *bubble* of the resulting equivalence class $\bigcup_{Q \in P_i} Q$ of $\sim_t$.

As auxiliary structures, for $p \in \{r, s\}$, we let $\eta_p\colon V_p/{\sim_p} \to V_t/{\sim_t}$ be the map such that for all $Q_p \in V_p/{\sim_p}$, $Q_p \subseteq \eta_p(Q_p)$, i.e. $\eta_p(Q_p)$ is the equivalence class of $\sim_t$ whose bubble contains Q_p. We call (H_t, η_r, η_s) the *operator* of t.

▶ **Theorem 3** (Rao, Thm. 6.6 in [39]). *For any graph G, $\mathsf{mw}(G) \le \mathsf{cw}(G) \le 2 \cdot \mathsf{mw}(G)$, and given a decomposition of bounded clique-width, a decomposition of bounded module-width, and vice versa, can be constructed in time $\mathcal{O}(n^2)$, where $n = |V(G)|$.*

Colorings. Let G be a graph. An ordered partition $\mathcal{C} = (C_1, \ldots, C_k)$ of $V(G)$ is called a *coloring* of G (with k colors). (Observe that for $i \in \{1, \ldots, k\}$, C_i may be empty.) For $i \in \{1, \ldots, k\}$, we call C_i the *color class* i, and say that the vertices in C_i *have color* i. $\mathcal{C}$ is called *proper* if each C_i is an independent set in G. The *restriction* of a coloring $\mathcal{C} = (C_1, \ldots, C_k)$ to a vertex set $S \subseteq V(G)$, is $\mathcal{C}|_S := (C_1 \cap S, \ldots, C_k \cap S)$. In this case we

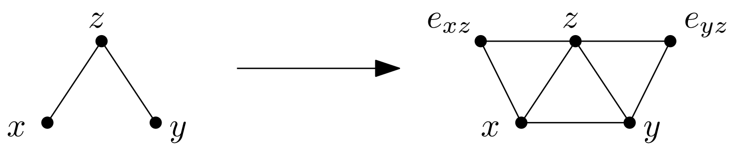

Figure 1 A gem created following the reduction in [24].

say conversely that $\mathcal{C}$ *extends* $\mathcal{C}|_S$. A proper coloring $(C_1, \ldots, C_k)$ is called a *b-coloring*, if for all $i \in \{1, \ldots, k\}$, there is a vertex $v_i \in C_i$, called *b-vertex of color i*, such that for all $j \in \{1, \ldots, k\} \setminus \{i\}$, $N_G(v_i) \cap C_j \neq \emptyset$.

b-Coloring

Input: Graph G, integer k
Question: Does G have a *b*-coloring with k colors?

Distance-hereditary graphs and chordal graphs. In their work on P_4-sparse graphs, Bonomo et al. [5] asked whether *b*-Coloring is polynomial-time solvable on the class of distance-hereditary graphs. Havet et al. [24] claimed to answer this question in the negative way, showing that *b*-Coloring is NP-complete on chordal distance-hereditary graphs. Their proof, however, contains a flaw and the graph constructed in their reduction, even though indeed chordal, fails to be distance-hereditary. In what follows, we briefly describe their reduction and argue that the graph constructed is not distance-hereditary. The reduction presented in [24] is from 3-Edge Coloring restricted to the class of 3-regular graphs. Given an instance G for 3-Edge Coloring with $V(G) = \{v_1, \ldots, v_n\}$, they construct a graph H as follows. The vertex set of H contains a copy of $V(G)$ plus one vertex associated with each edge of G. We denote by e_{xy} the vertex corresponding to the edge xy. The vertices of $V(G)$ form a clique in H, the vertices corresponding to edges form an independent set, and for each edge $xy \in E(G)$, the vertex e_{xy} is adjacent to the copy of x and y in H. The connected component of H induced by these vertices is therefore a split graph. Finally, they add three disjoint copies of $K_{1,n+2}$ to H. It is thus easy to see that H is a chordal graph. However, let xz and yz be two edges of G sharing one endpoint. Then the subgraph of H induced by $\{x, y, z, e_{xz}, e_{yz}\}$ is isomorphic to a gem (see Figure 1). As shown by Bandelt and Mulder [2], distance-hereditary graphs are gem-free graphs. This shows that the graph H is not a distance-hereditary graph.

Via monadic second order logic and Courcelle's Theorem [11], we can show the following result for chordal graphs.

▶ **Proposition 4 (♣).** *b-Coloring parameterized by k is* FPT *on chordal graphs.*

3 Parameterized by Clique-Width

In this section, we consider the *b*-coloring problem parameterized by the clique-width of the input graph. We will work with decompositions of bounded *module-width*, which is equivalent for our purposes, see Theorem 3.

The main contribution of this section is an algorithm that given a graph G on n vertices and one of its rooted branch decompositions of module-width w, and an integer k, decides whether G has a *b*-coloring with k colors in time $n^{2^{\mathcal{O}(w)}}$. Before we proceed, we observe that *b*-Coloring is W[1]-hard in this parameterization, and that the exponential dependence on w of the degree of the polynomial in the runtime is probably difficult to avoid.

▶ **Proposition 5** (♣). *The b-*Coloring *problem on graphs on n vertices parameterized by their module-width w is* W[1]-hard and cannot be solved in time $n^{2^{o(w)}}$, unless ETH fails. Moreover, the hardness holds even when a linear branch decomposition of width w is provided.

3.1 Outline of the Algorithm

Throughout the following, we are given a graph G and one of its rooted branch decompositions $(T, \mathcal{L})$ of module-width $w = \mathsf{mw}(T, \mathcal{L})$ and we want to find a b-coloring of G with k colors, if it exists. In particular, our algorithm will find a b-coloring $\mathcal{C}$ together with a set of *witness b-vertices*, containing precisely one b-vertex for each color class of $\mathcal{C}$, if it exists. This will be done via dynamic programming along T, and for each node $t \in V(T)$, the partial solutions associated with t are partial b-colorings of G_t.

▶ **Definition 6** (Partial b-Coloring). *Let G be a graph and $k \in \mathbb{N}$. For an induced subgraph H of G, a* partial b-coloring *of H is a pair $(\mathcal{C}, B)$ of a proper coloring $\mathcal{C} = (C_1, \ldots, C_k)$ of H and a subset $B \subseteq V(H)$ such that for all $i \in [k]$, $|C_i \cap B| \le 1$. We call the vertices in B the* partial b-vertices*.*

To obtain an efficient algorithm, we require a compact representation of the partial b-colorings of each subgraph G_t associated with a node $t \in V(T)$. To that end, we introduce the notion of a *t-signature* of a partial b-coloring. Two partial b-colorings with the same t-signature will be interchangeable for the sake of our algorithm, therefore the number of table entries at each node t will be bounded by the number of t-signatures.

Let $(\mathcal{C}, B)$ be a partial b-coloring of G_t. For $(\mathcal{C}, B)$ to be extended to a b-coloring $(\mathcal{C}', B')$ of the entire graph G, we have to ensure that two things happen for each color class $C \in \mathcal{C}$:

(I) The extension of C in $\mathcal{C}'$ is an independent set in G.

(II) There is a witness b-vertex in B' for the extension of C in $\mathcal{C}'$.

The t-signature has to represent a partial b-coloring faithfully enough so that we can keep track of all the ways in which the above two conditions can be satisfied for each of its color classes 'in the future'. At the same time, its definition has to enable us to significantly compress the information about partial b-colorings of G_t. This happens in the following way. We categorize color classes of partial b-colorings of G_t according to *t-types*. If two color classes C_1, C_2 of a partial b-coloring $(\mathcal{C}, B)$ have the same t-type, then the above two conditions can be satisfied for C_1 and C_2 by extensions of $(\mathcal{C}, B)$ in the exact same ways. This allows us to forget about the "names" of the color classes in a partial b-coloring, but instead to only remember for each t-type how many color classes with that type there are. This is precisely the information that is stored in a t-signature.

Now, if we can bound the number of t-types by some function of the module-width w, say $f(w)$, then the number of t-signatures is upper bounded by $k^{f(w)} \le n^{f(w)}$. (There are at most k colors, so in particular there are at most k colors with a given t-type.) This translates directly to an upper bound on the number of table entries in the dynamic programming algorithm, which, up to some constants in the degree of the polynomial, bounds the runtime of the resulting algorithm.

Let us discuss the information that goes into the definition of a t-type. Let C be a color class in a partial b-coloring $(\mathcal{C}, B)$ of G_t. To keep track of which vertices from $\overline{V_t}$ can be added to C without introducing a coloring conflict, it suffices to store which equivalence classes of $\sim_t$ have vertices in C,[3] since all vertices in a given equivalence class have the same neighbors in $\overline{V_t}$. This way we can ensure that condition (I) is satisfied.

[3] This is similar to the algorithm of Wanke for Graph Coloring on graphs of bounded NLC-width [46].

To verify if condition (II) is satisfied we have to store some information about the partial b-vertices. Naturally, we record whether or not B contains a partial b-vertex of C, but we need to store more information. Suppose that B contains the partial b-vertex v of C. In a straightforward approach, we would simply keep track of the color classes that already appear in the neighborhood of v. This way we could easily decide at which point during the execution of the algorithm, a partial b-vertex turns into a b-vertex. However, this results in prohibitively large table entries, since there are 2^{k-1} subsets of colors that we would have to consider, which for our purpose is no better than 2^n.

We overcome this issue with the following symmetry breaking trick: We do *not* record which color classes the partial b-vertex of C already sees/still needs to see. Instead, we record for which equivalence classes $Q \in V_t/{\sim_t}$ we need to add a *future neighbor of* Q, i.e. a vertex from $N(Q) \cap \overline{V_t}$, to C, such that the partial b-vertex from *some other color* C' sees color C in its neighborhood. More concretely, suppose that some equivalence class $Q \in V_t/{\sim_t}$ contains the partial b-vertex $w \in B$ of another color class $C' \neq C$, such that w has no neighbor of color C in V_t. For w to become a b-vertex of its color, color class C must be extended with a neighbor of w in the future, i.e. in $\overline{V_t}$. The neighborhood of w in $\overline{V_t}$ is precisely $N_G(Q) \cap \overline{V_t}$, therefore we can concisely model this situation as color class C requiring to contain a vertex among the future neighbors of Q. In this situation, we say that

color class C has demand to the future neighbors of Q.

The t-type records for each equivalence class Q of $\sim_t$, if a color class contains vertices of Q, or if it has demand to the future of Q, or none of the two. Note that if a color class both contains a vertex from Q and has demand to the future of Q, we already know that we can disregard the corresponding partial b-coloring: In the corresponding color class, we cannot add any future neighbors of Q without creating a coloring conflict, and if we do not add a future neighbor of Q, then there is some color class whose partial b-vertex will never become a b-vertex. Now, if we have a partial b-coloring in which every color class has a partial b-vertex, and all demands have been fulfilled, meaning that there is no color class that has demand to the future of some equivalence class of $\sim_t$, then we know that we actually have a b-coloring. Moreover, the number of t-types is $2^{\mathcal{O}(w)}$, so the resulting algorithm runs in time $n^{2^{\mathcal{O}(w)}}$.

3.2 t-Types and t-Signatures

In this section we introduce the basic concepts that we alluded to in the above description, namely the notion of a *t-type* and of a *t-signature*, where t is some node in the given branch decomposition. A t-type is meant to capture the necessary information of a color class in a partial b-coloring of G_t. However, we cannot give the definition of a t-type as a property of a vertex set alone: a color class C may have demand to the future of an equivalence class, which is because there is a partial b-vertex of *another* color $C' \neq C$ that has no neighbor of color C yet. Therefore, we first give the definition of a t-type abstractly, i.e. absent of any partial b-coloring or color class, and then define what it means for a color class to be of a certain t-type *within a partial b-coloring.* This is illustrated in Figure 2.

The t-type is a pair of a bit that is meant to tell us whether or not a coloring contains a partial b-vertex of that color, and a map that tells us for each equivalence class, whether there is a vertex of the color in the equivalence class (via the value cont), or if the color has demand to the future neighbors of the equivalence class (via the value dem), or none of the two (via the value none).

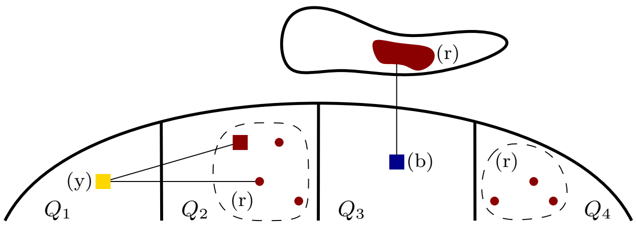

Figure 2 Illustration of the definition of a color class being of a certain *t-type* inside a partial b-coloring of G_t. The large square vertices are partial b-vertices for their color. The type of the red (r) color in the coloring is as follows. Since it has a b-vertex (the one in Q_2), we have that $\xi = 1$. Since Q_2 and Q_4 have red vertices, $\phi(Q_2) = \phi(Q_4) = \mathsf{cont}$. Q_1 and Q_3 do not have red vertices. Q_1 contains the b-vertex of color yellow (y), but this vertex already has a red neighbor. Therefore, $\phi(Q_1) = \mathsf{none}$. Finally, Q_3 has the b-vertex of color blue (b), and this vertex does not have a red neighbor yet. Therefore, there has to be a red vertex among the future neighbors of Q_3. Hence, $\phi(Q_3) = \mathsf{dem}$.

▶ **Definition 7** (t-Type)**.** *Let G be a graph with rooted branch decomposition $(T, \mathcal{L})$ and let $t \in V(T)$. A* t-type *is a pair (ϕ, ξ) of a map $\phi\colon Q_t/{\sim_t} \to \{\mathsf{none}, \mathsf{cont}, \mathsf{dem}\}$ and a bit $\xi \in \{0, 1\}$. We denote the set of all t-types by types_t.*

▶ **Observation 8.** *Let $(T, \mathcal{L})$ be a rooted branch decomposition of module-width $w = \mathsf{mw}(T, \mathcal{L})$. For each $t \in V(T)$, $|\mathsf{types}_t| = 2 \cdot 3^{|V_t/{\sim_t}|} \le 2 \cdot 3^w$.*

▶ **Definition 9.** *Let G be a graph with rooted branch decomposition $(T, \mathcal{L})$ and let $t \in V(T)$. Let $(\mathcal{C}, B)$ be a partial b-coloring of G_t, let $C \in \mathcal{C}$ be a color class, and let $\tau = (\phi, \xi) \in \mathsf{types}_t$ be a t-type. We say that C* has t-type τ in $(\mathcal{C}, B)$ *if*

(i) *$\xi = |C \cap B|$ and*

(ii) *for each $Q \in V_t/{\sim_t}$,*

  (a) *if $Q \cap C \neq \emptyset$, and $\nexists v \in (B \setminus C) \cap Q$ such that $N(v) \cap C = \emptyset$, then $\phi(Q) = \mathsf{cont}$;*

  (b) *if $Q \cap C = \emptyset$ and $\exists v \in (B \setminus C) \cap Q$ such that $N(v) \cap C = \emptyset$, then $\phi(Q) = \mathsf{dem}$; and*

  (c) *if $Q \cap C = \emptyset$, and $\nexists v \in (B \setminus C) \cap Q$ such that $N(v) \cap C = \emptyset$, then $\phi(Q) = \mathsf{none}$.*

The reader may have observed that (ii) does not cover all the possibilities. The situation that is not covered is when $Q \cap C \neq \emptyset$ and there is some $v \in (B \setminus C) \cap Q$ such that $N(v) \cap C = \emptyset$. A priori, we can of course not exclude this as a possibility, but there is a simple reason that partial b-colorings that contain a color class in which this situation arises can be disregarded: For the vertex v to become a b-vertex for its color, we have to add a future neighbor of Q to C; but since Q already contains a vertex from C this means that the resulting set is not independent anymore.

▶ **Definition 10** (t-Signature)**.** *Let G be a graph with rooted branch decomposition $(T, \mathcal{L})$, and let $t \in V(T)$. A* t-signature *is a map $\mathsf{sig}_t\colon \mathsf{types}_t \to \{0, 1, \ldots, k\}$ with $\sum_{\tau \in \mathsf{types}_t} \mathsf{sig}_t(\tau) = k$.*

The following bound on the number of t-signatures immediately follows from Observation 8: for each t-type, the function takes one of $k + 1 \le n + 1$ values.

▶ **Observation 11.** *Let G be a graph on n vertices and $(T, \mathcal{L})$ be one of its branch decompositions of module-width $w = \mathsf{mw}(T, \mathcal{L})$. For each $t \in V(T)$, there are at most $n^{2^{\mathcal{O}(w)}}$ many t-signatures.*

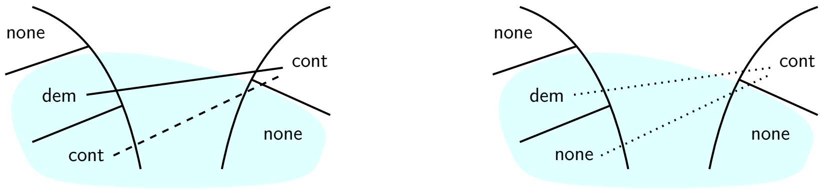

Figure 3 Illustration of Definition 13. The shaded area shows a bubble and the labels on the equivalence classes correspond to type labelings. For the left hand side, note that between a pair of classes that are both labeled "cont", there can be no edge in the operator. Moreover, since the bubble contains a class labeled cont and one labeled dem, the demand of the latter has to be fulfilled at this node, i.e. there has to be an edge from this class to a "cont"-class. The right side shows the situation when the "cont"-class in the bubble is changed to "none", in which case the dotted edges may or may not be present in the operator.

▶ **Definition 12.** *Let G be a graph with rooted branch decomposition $(T, \mathcal{L})$, and let $t \in V(T)$. Let furthermore* sig_t *be a t-signature and $(\mathcal{C}, B)$ a partial b-coloring in G_t. We say that* sig_t represents $(\mathcal{C}, B)$ *if for each t-type $\tau \in \mathsf{types}_t$, there are precisely $\mathsf{sig}_t(\tau)$ color classes in $(\mathcal{C}, B)$ that have t-type τ in $(\mathcal{C}, B)$. We call a partial b-coloring $(\mathcal{C}, B)$ of G_t* representable *if and only if there is a t-signature that represents it.*

3.3 Compatibility

Let $t \in V(T) \setminus \mathrm{L}(T)$ be an internal node of the given rooted branch decomposition, let r and s be its children, and let (H_t, η_r, η_s) be the operator of t. In our algorithm, we want to combine information about partial b-colorings of G_r and G_s to obtain information about partial b-colorings of G_t. We will try to obtain a color class of a partial b-coloring of G_t by taking the union of a color class C_r of a partial b-coloring of G_r and a color class C_s of a partial b-coloring of G_s.

However, in some cases this is not possible. For instance, when C_r contains vertices from some equivalence class $Q_r \in V_r/{\sim_r}$ and C_s contains vertices from some equivalence class $Q_s \in V_s/{\sim_s}$, and in the graph H_t of the operator of t, we have that $Q_rQ_s \in E(H_t)$. Then, in G_t all edges between the set Q_r and Q_s are present which means that $C_r \cup C_s$ is not an independent set in G_t. Another condition is necessary to ensure that several demands that *have to be* met at node t are indeed met. Let $C_t = C_r \cup C_s$ and suppose there is an equivalence class $Q_t \in V_t/{\sim_t}$ that contains a vertex of C_t. Suppose furthermore that there is another equivalence class $Q_r \in V_r/{\sim_r}$ contained in the bubble of Q_t such that C_r has demand to the future neighbors of Q_r. Then, this demand must be fulfilled by a neighbor of Q_r in C_s for otherwise, the equivalence class Q_t both contains vertices of C_t and C_t has demand to the future neighbors of Q_t. The resulting partial b-coloring would not be representable. We illustrate the notion of compatibility in Figure 3.

▶ **Definition 13** (Compatible types)**.** *Let G be a graph with rooted branch decomposition $(T, \mathcal{L})$. Let furthermore $t \in V(T) \setminus \mathrm{L}(T)$ with children r and s, and let (H_t, η_r, η_s) be the operator of t. Let $(\phi_r, \xi_r) \in \mathsf{types}_r$ and $(\phi_s, \xi_s) \in \mathsf{types}_s$. We say that (ϕ_r, ξ_r) and (ϕ_s, ξ_s) are* compatible *if the following conditions hold.*

(i) $\xi_r + \xi_s \leq 1$.

(ii) *There is no pair $Q_r \in V_r/{\sim_r}$, $Q_s \in V_s/{\sim_s}$ such that $Q_rQ_s \in E(H_t)$ and $\phi_r(Q_r) = \phi_s(Q_s) = \mathsf{cont}$.*

(iii) *For each $Q \in V_t/{\sim_t}$ such that there exists a $p \in \{r, s\}$ and a $Q_p \in \eta_p^{-1}(Q)$ with $\phi_p(Q_p) = \mathsf{cont}$, the following holds.*

(a) *For all* $Q_r \in \eta_r^{-1}(Q)$ *with* $\phi_r(Q_r) = \mathsf{dem}$*, there is a* $Q_s \in V_s/{\sim_s}$ *with* $\phi_s(Q_s) = \mathsf{cont}$ *and* $Q_rQ_s \in E(H_t)$*.*
(b) *Similarly, for all* $Q_s \in \eta_s^{-1}(Q)$ *with* $\phi_s(Q_s) = \mathsf{dem}$*, there is a* $Q_r \in V_r/{\sim_r}$ *with* $\phi_r(Q_r) = \mathsf{cont}$ *and* $Q_sQ_r \in E(H_t)$*.*

Given a pair of a color class C_r of a partial b-coloring of G_r and a color class C_s of a partial b-coloring of G_s whose types in the respective colorings are compatible, $C_r \cup C_s$, considered as a color class in a partial b-coloring of G_t, has a fixed type, which can formally be constructed as follows.

▶ **Definition 14** (Merge Type)**.** *Let G be a graph with rooted branch decomposition $(T, \mathcal{L})$. Let furthermore $t \in V(T) \setminus \mathrm{L}(T)$ with children r and s, and let (H_t, η_r, η_s) be the operator of t. Let $\rho = (\phi_r, \xi_r) \in \mathsf{types}_r$ and $\sigma = (\phi_s, \xi_s) \in \mathsf{types}_s$ be a pair of compatible types. The* merge type *of ρ and σ, denoted by $\mu(\rho, \sigma)$, is the following t-type (ϕ_t, ξ_t).*
(i) $\xi_t = \xi_r + \xi_s$*.*
(ii) *For each* $Q \in V_t/{\sim_t}$*:*
(a) *If for some* $p \in \{r, s\}$*,* $\exists Q_p \in \eta_p^{-1}(Q)$ *with* $\phi_p(Q_p) = \mathsf{cont}$*, then* $\phi_t(Q) = \mathsf{cont}$*.*
(b) *If (iia) does not apply and for some* $p \in \{r, s\}$*,* $\exists Q_p \in \eta^{-1}(Q)$ *with* $\phi_p(Q_p) = \mathsf{dem}$ *and for all* $Q_pQ_o \in E(H_t)$ *we have that* $\phi_o(Q_o) \neq \mathsf{cont}$*, then* $\phi_t(Q) = \mathsf{dem}$*.*
(c) *If neither (iia) nor (iib) applies, then* $\phi_t(Q) = \mathsf{none}$*.*

Towards a notion of compatibility of signatures, we first define a structure we call *merge skeleton*. Given a node $t \in V(T)$ with children r and s, the merge skeleton is an edge-labeled bipartite graph whose vertices are the r-types and the s-types, with the merge type of a compatible pair of types $\rho \in \mathsf{types}_r$, $\sigma \in \mathsf{types}_s$ written on the edge $\rho\sigma$. Such an edge is meant to represent the fact that taking the union of a color class C_r that has r-type ρ in a partial b-coloring of G_r with a color class C_s that has s-type σ in a partial b-coloring of G_s results in a color class of t-type $\mu(\rho, \sigma)$ in a partial b-coloring of G_t.

▶ **Definition 15** (Merge skeleton)**.** *Let G be a graph and $(T, \mathcal{L})$ one of its rooted branch decompositions. Let $t \in V(T) \setminus \mathrm{L}(T)$ with children r and s. The* merge skeleton *of r and s is an edge-labeled bipartite graph $(\mathfrak{J}, \mathfrak{m})$ where*
- $V(\mathfrak{J}) = \mathsf{types}_r \cup \mathsf{types}_s$*,*
- *for all* $\rho \in \mathsf{types}_r$*,* $\sigma \in \mathsf{types}_s$*,* $\rho\sigma \in E(\mathfrak{J})$ *if and only if* ρ *and* σ *are compatible, and*
- $\mathfrak{m} \colon E(\mathfrak{J}) \to \mathsf{types}_t$ *is such that for all* $\rho\sigma \in E(\mathfrak{J})$*,* $\mathfrak{m}(\rho\sigma)$ *is the merge type of* ρ *and* σ*.*

Now, any pair of an r-signature sig_r and an s-signature sig_s can "flesh out" the merge skeleton $(\mathfrak{J}, \mathfrak{m})$ of r and s, in the following sense. We can consider the union of sig_r and sig_s as a map labeling the vertices of $\mathfrak{J}$. Then, an edge-labeling $\mathfrak{n}$ of $\mathfrak{J}$ with integers from $\{0, 1, \ldots, k\}$, such that for each vertex of $\mathfrak{J}$, the sum over its incident edges e of $\mathfrak{n}(e)$ is equal to its vertex label, produces a t-signature sig_t. We can read off how many color classes of each type there are from the edge labeling $\mathfrak{n}$.

▶ **Definition 16** (Compatible signatures)**.** *Let $(T, \mathcal{L})$ be a rooted branch decomposition. Let furthermore $t \in V(T) \setminus \mathrm{L}(T)$ with children r and s. Let sig_t be a t-signature, let sig_r be an r-signature and sig_s be a s-signature. We say that $(\mathsf{sig}_t, \mathsf{sig}_r, \mathsf{sig}_s)$ is* compatible *if there is a triple $(\mathfrak{J}, \mathfrak{m}, \mathfrak{n})$ such that $(\mathfrak{J}, \mathfrak{m})$ is the merge skeleton of r and s, and $\mathfrak{n} \colon E(\mathfrak{J}) \to \{0, 1, \ldots, k\}$ is a map with the following properties.*
(i) *For all* $p \in \{r, s\}$ *and all* $\pi \in \mathsf{types}_p$*,* $\sum_{e \in E(\mathfrak{J}) \colon \pi \in e} \mathfrak{n}(e) = \mathsf{sig}_p(\pi)$*.*
(ii) *For all* $\tau \in \mathsf{types}_t$*,* $\sum_{e \in E(\mathfrak{J}) \colon \mathfrak{m}(e) = \tau} \mathfrak{n}(e) = \mathsf{sig}_t(\tau)$*.*

▶ **Lemma 17** (♣). *Let G be a graph on n vertices and let $(T, \mathcal{L})$ be one of its rooted branch decomposition of module-width $w = \mathsf{mw}(T, \mathcal{L})$. Let $t \in V(T) \setminus \mathrm{L}(T)$ with children r and s. Let sig_t be a t-signature, sig_r be an r-signature, and sig_s be an s-signature. One can decide in time $n^{2^{\mathcal{O}(w)}}$ whether or not $(\mathsf{sig}_t, \mathsf{sig}_r, \mathsf{sig}_s)$ is compatible.*

3.4 Merging and Splitting Partial b-Colorings

In this section we state the lemmas that show that the notions introduced above work as desired, and the technical lemmas we prove here will be the cornerstone of the correctness proof of the resulting algorithm.

▶ **Lemma 18** (♣). *Let G be a graph with rooted branch decomposition $(T, \mathcal{L})$ and let $t \in V(T) \setminus \mathrm{L}(T)$ be an internal node with children r and s. Let sig_r be an r-signature, sig_s be an s-signature, and sig_t be a t-signature such that:*
- *For all $p \in \{r, s\}$, there is a partial b-coloring $(\mathcal{C}_p, B_p)$ in G_p represented by sig_p, and*
- *$(\mathsf{sig}_t, \mathsf{sig}_r, \mathsf{sig}_s)$ is compatible.*

Then, there is a partial b-coloring $(\mathcal{C}_t, B_t)$ of G_t that is represented by sig_t.

▶ **Lemma 19** (♣). *Let G be a graph with rooted branch decomposition $(T, \mathcal{L})$ and let $t \in V(T) \setminus \mathrm{L}(T)$ be an internal node with children r and s. Let sig_t be a t-signature, and suppose there is a partial b-coloring $(\mathcal{C}_t, B_t)$ of G_t which is represented by sig_t. Then, there exists an r-signature sig_r and an s-signature sig_s such that*
- *for all $p \in \{r, s\}$ there is a partial b-coloring $(\mathcal{C}_p, B_p)$ represented by sig_p, and*
- *$(\mathsf{sig}_t, \mathsf{sig}_r, \mathsf{sig}_s)$ is compatible.*

3.5 The Algorithm

▶ **Definition of the table entries.** *For a node $t \in V(T)$ and a t-signature sig_t, we let $\mathsf{tab}[t, \mathsf{sig}_t] = 1$ if and only if there exists a partial b-coloring of G_t that is represented by sig_t.*

We now show that if all table entries have been computed correctly, then the solution can be read off the table entries stored at the root $\mathfrak{r}$ of the given rooted branch decomposition. Observe that since $V_{\mathfrak{r}} = V(G)$ and therefore $\overline{V_t} = \emptyset$, the equivalence relation $\sim_{\mathfrak{r}}$ has one equivalence class, namely $V(G)$.

▶ **Lemma 20** (♣). *Let G be a graph with rooted branch decomposition $(T, \mathcal{L})$ and let $\mathfrak{r} \in V(T)$ be the root of T. Let ρ be the $\mathfrak{r}$-type $(\phi_{\mathfrak{r}}, \xi_{\mathfrak{r}})$ with $\xi_{\mathfrak{r}} = 1$ and $\phi_{\mathfrak{r}}(V(G)) = \mathsf{cont}$. Let $\mathsf{sig}_{\mathfrak{r}}$ be the $\mathfrak{r}$-signature letting $\mathsf{sig}_{\mathfrak{r}}(\rho) = k$. Then, G has a b-coloring with k colors if and only if $\mathsf{tab}[\mathfrak{r}, \mathsf{sig}_{\mathfrak{r}}] = 1$.*

The table entries at the leaves are computed by brute force, and we defer the details to the full version. We compute the table entries at the internal nodes as follows.

▶ **Internal nodes of T.** *Now let $t \in V(T) \setminus \mathrm{L}(T)$ with children r and s. For each t-signature sig_t, we let $\mathsf{tab}[t, \mathsf{sig}_t] = 1$ if and only if there exists a pair $(\mathsf{sig}_r, \mathsf{sig}_s)$ of an r-signature sig_r and an s-signature sig_s such that*

(J1) $\mathsf{tab}[r, \mathsf{sig}_r] = 1$ *and* $\mathsf{tab}[s, \mathsf{sig}_s] = 1$, *and*

(J2) $(\mathsf{sig}_t, \mathsf{sig}_r, \mathsf{sig}_s)$ *is compatible.*

▶ **Lemma 21** (♣). *For each $t \in V(T)$ and t-signature sig_t, the above algorithm computes the table entry $\mathsf{tab}[t, \mathsf{sig}_t]$ correctly.*

We wrap up. By Lemma 21, the algorithm computes all table entries correctly, and by Lemma 20, the solution to the instance can be determined upon inspecting the table entries associated with the root of the given branch decomposition. Correctness of the algorithm follows. The runtime follows essentially from Observation 11 and Lemma 17. We give the details in the full version.

▶ **Theorem 22.** *There is an algorithm that solves b-*Coloring *in time* $n^{2^{\mathcal{O}(w)}}$*, where* n *denotes the number of vertices of the input graph, and* w *denotes the module-width of a given rooted branch decomposition of the input graph.*

3.6 Fall Coloring

Recall that a *fall coloring* is a special type of b-coloring where *every* vertex is a b-vertex for its color. We adapt our algorithm for b-Coloring on graphs of bounded clique-width to solve Fall Coloring, and therefore obtain the following theorem.

▶ **Theorem 23 (♣).** *There is an algorithm that solves* Fall Coloring *in time* $n^{2^{\mathcal{O}(w)}}$*, where* n *denotes the number of vertices of the input graph, and* w *denotes the module-width of a given rooted branch decomposition of the input graph.*

▶ **Proposition 24 (♣).** *The* Fall Coloring *problem on graphs on* n *vertices parameterized by the module-width* w *of the input graph is* W[1]-hard and cannot be solved in time $n^{2^{o(w)}}$, unless ETH fails. Moreover, the hardness holds even when a linear branch decomposition of width w is provided.

4 Parameterized by Vertex Cover

We conclude by stating that b-Coloring parameterized by the size of a vertex cover of the input graph is FPT.

▶ **Theorem 25 (♣).** *There is an algorithm that solves b-*Coloring *in time* $2^{\mathcal{O}(\ell^2 \log \ell)} \cdot n^{\mathcal{O}(1)}$*, where* ℓ *denotes the vertex cover number of the input graph.*

References

1 Pierre Aboulker, Édouard Bonnet, Eun Jung Kim, and Florian Sikora. Grundy coloring & friends, half-graphs, bicliques. In *STACS 2020*, volume 154 of *LIPIcs*, pages 58:1–58:18, 2020.

2 Hans-Jürgen Bandelt and Henry Martin Mulder. Distance-hereditary graphs. *Journal of Combinatorial Theory Series B*, 41:182–208, 1986.

3 Rémy Belmonte, Eun Jung Kim, Michael Lampis, Valia Mitsou, and Yota Otachi. Grundy distinguishes treewidth from pathwidth. In *ESA 2020*, volume 173 of *LIPIcs*, pages 14:1–14:19, 2020.

4 J. Adrian Bondy and Uppaluri S. R. Murty. *Graph Theory*, volume 244 of *Graduate Texts in Mathematics*. Springer, 2008.

5 Flavia Bonomo, Guillermo Durán, Frederic Maffray, Javier Marenco, and Mario Valencia-Pabon. On the b-coloring of cographs and P_4-sparse graphs. *Graphs and Combinatorics*, 25(2):153–167, 2009.

6 Flavia Bonomo, Oliver Schaudt, Maya Stein, and Mario Valencia-Pabon. b-Coloring is NP-hard on co-bipartite graphs and polytime solvable on tree-cographs. *Algorithmica*, 73(2):289–305, 2015.

7 Victor A. Campos, Carlos V. Lima, Nicolas A. Martins, Leonardo Sampaio, Marcio C. Santos, and Ana Silva. The b-chromatic index of graphs. *Discrete Mathematics*, 338(11):2072–2079, 2015.

8 Victor A. Campos, Carlos Vinicius G. C. Lima, and Ana Silva. Graphs of girth at least 7 have high b-chromatic number. *European Journal of Combinatorics*, 48:154–164, 2015.

9 Victor A. Campos, Cláudia Linhares-Sales, Rudini Sampaio, and Ana Karolinna Maia. Maximization coloring problems on graphs with few P_4. *Discrete Applied Mathematics*, 164:539–546, 2014.

10 Victor A. Campos and Ana Silva. Edge-b-coloring trees. *Algorithmica*, 80(1):104–115, 2018.

11 Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.

12 Bruno Courcelle, Joost Engelfriet, and Grzegorz Rozenberg. Handle-rewriting hypergraph grammars. *Journal of Computer and System Sciences*, 46(2):218–270, 1993.

13 Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1-3):77–114, 2000.

14 Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

15 Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, Heidelberg, fifth edition, 2016.

16 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013.

17 Jean E. Dunbar, Sandra M. Hedetniemi, S.T. Hedetniemi, David P. Jacobs, J. Knisely, R.C. Laskar, and Douglas F. Rall. Fall colorings of graphs. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 33:257–274, 2000.

18 Brice Effantin, Nicolas Gastineau, and Olivier Togni. A characterization of b-chromatic and partial grundy numbers by induced subgraphs. *Discrete Mathematics*, 339(8):2157–2167, 2016.

19 Wolfgang Espelage, Frank Gurski, and Egon Wanke. How to solve NP-hard graph problems on clique-width bounded graphs in polynomial time. In *WG 2001*, pages 117–128, 2001.

20 Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, and Saket Saurabh. Intractability of clique-width parameterizations. *SIAM Journal on Computing*, 39(5):1941–1956, 2010.

21 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 Transactions on Algorithms*, 15(1):9:1–9:27, 2019.

22 Wayne Goddard and Michael A. Henning. Independent domination in graphs: A survey and recent results. *Discrete Mathematics*, 313(7):839–854, 2013.

23 Martin Charles Golumbic and Udi Rotics. On the clique-width of some perfect graph classes. *International Journal of Foundations of Computer Science*, 11(03):423–443, 2000.

24 Frédéric Havet, Claudia Linhares Sales, and Leonardo Sampaio. b-coloring of tight graphs. *Discrete Applied Mathematics*, 160(18):2709–2715, 2012.

25 Frédéric Havet and Leonardo Sampaio. On the Grundy and *b*-chromatic numbers of a graph. *Algorithmica*, 65:885–899, 2013.

26 Pinar Heggernes and Jan Arne Telle. Partitioning graphs into generalized dominating sets. *Nordic Journal on Computing*, 5(2):128–142, 1998.

27 Russell Impagliazzo and Ramamohan Paturi. On the complexity of *k*-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

28 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

29 Robert W. Irving and David F. Manlove. The b-chromatic number of a graph. *Discrete Applied Mathematics*, 91(1-3):127–141, 1999.

30 Lars Jaffke and Paloma T. Lima. A complexity dichotomy for critical values of the b-chromatic number of graphs. *Theoretical Computer Science*, 815:182–196, 2020.

31 Klaus Jansen. Complexity results for the optimum cost chromatic partition problem, 1997.

32 Jan Kratochvíl, Zsolt Tuza, and Margit Voigt. On the b-chromatic number of graphs. In *WG 2002*, pages 310–320, 2002.

33 Renu Laskar and Jeremy Lyle. Fall colouring of bipartite graphs and cartesian products of graphs. *Discrete Applied Mathematics*, 157(2):330–338, 2009.

34 Juho Lauri and Christodoulos Mitillos. Complexity of fall coloring for restricted graph classes. In *30th IWOCA*, pages 352–364. Springer, 2019.

35 Jeremy Lyle, Nate Drake, and Renu Laskar. Independent domatic partitioning or fall coloring of strongly chordal graphs. *Congressus Numerantium*, 172:149–159, 2005.

36 Johann A. Makowsky and Udi Rotics. On the clique-width of graphs with few P_4's. *International Journal of Foundations of Computer Science*, 10(03):329–348, 1999.

37 Christodoulos Mitillos. *Topics in Graph Fall-Coloring*. PhD thesis, Illinois Institute of Technology, 2016.

38 Fahad Panolan, Geevarghese Philip, and Saket Saurabh. On the parameterized complexity of b-chromatic number. *Journal of Computer and System Sciences*, 84:120–131, 2017.

39 Michaël Rao. *Décompositions de graphes et algorithmes efficaces*. PhD thesis, University of Metz, 2006.

40 Michaël Rao. Clique-width of graphs defined by one-vertex extensions. *Discrete Mathematics*, 308(24):6157–6165, 2008.

41 Leonardo Sampaio. *Algorithmic aspects of graph colourings heuristics*. PhD thesis, Université Nice Sophia Antipolis, 2012.

42 Ana Silva. Graphs with small fall-spectrum. *Discrete Applied Mathematics*, 254:183–188, 2019.

43 Ana Shirley Ferreira da Silva. *The b-chromatic number of some tree-like graphs*. PhD thesis, Université Joseph-Fourier - Grenoble I, 2010.

44 Jan Arne Telle and Andrzej Proskurowski. Algorithms for vertex partitioning problems on partial k-trees. *SIAM Journal on Discrete Mathematics*, 10(4):529–550, 1997.

45 Clara Inés Betancur Velasquez, Flavia Bonomo, and Ivo Koch. On the b-coloring of P_4-tidy graphs. *Discrete Applied Mathematics*, 159(1):60–68, 2011.

46 Egon Wanke. k-NLC graphs and polynomial algorithms. *Discrete Applied Mathematics*, 54:251–266, 1994.