Fine-grained parameterized complexity analysis of graph coloring problems

Fine-Grained Parameterized Complexity Analysis of Graph Coloring Problems*

Lars Jaffke†1 and Bart M. P. Jansen[2]

[1]Department of Informatics, University of Bergen, Postboks 7803, N-5020 Bergen, Norway, l.jaffke@uib.no

[2]Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands, b.m.p.jansen@tue.nl

January 25, 2017

Abstract

The q-Coloring problem asks whether the vertices of a graph can be properly colored with q colors. Lokshtanov et al. [SODA 2011] showed that q-Coloring on graphs with a feedback vertex set of size k cannot be solved in time $\mathcal{O}^*((q-\varepsilon)^k)$, for any $\varepsilon > 0$, unless the Strong Exponential-Time Hypothesis (SETH) fails. In this paper we perform a fine-grained analysis of the complexity of q-Coloring with respect to a hierarchy of parameters. We show that even when parameterized by the vertex cover number, q must appear in the base of the exponent: Unless ETH fails, there is no universal constant θ such that q-Coloring parameterized by vertex cover can be solved in time $\mathcal{O}^*(\theta^k)$ for all fixed q. We apply a method due to Jansen and Kratsch [Inform. & Comput. 2013] to prove that there are $\mathcal{O}^*((q-\varepsilon)^k)$ time algorithms where k is the vertex deletion distance to several graph classes $\mathcal{F}$ for which q-Coloring is known to be solvable in polynomial time. We generalize earlier ad-hoc results by showing that if $\mathcal{F}$ is a class of graphs whose $(q+1)$-colorable members have bounded treedepth, then there exists some $\varepsilon > 0$ such that q-Coloring can be solved in time $\mathcal{O}^*((q-\varepsilon)^k)$ when parameterized by the size of a given modulator to $\mathcal{F}$. In contrast, we prove that if $\mathcal{F}$ is the class of paths – some of the simplest graphs of unbounded treedepth – then no such algorithm can exist unless SETH fails.

1 Introduction

In an influential paper from 2011, Lokshtanov et al. showed that for several problems, straightforward dynamic programming algorithms for graphs of bounded treewidth are essentially optimal unless the Strong Exponential Time Hypothesis (SETH) fails [13]. (Section 2.2 gives the definitions of the two Exponential Time Hypotheses, see [4, Chapter 14] or the survey [16] for further details.) Some of the lower bounds, as the one for q-Coloring, even hold for parameters such as the feedback vertex number, which form an upper bound on the treewidth but may be arbitrarily much larger. For other problems such as Dominating Set, the tight lower bound of $\Omega((3-\varepsilon)^k)$ holds for the parameterization pathwidth, but is not known for the parameterization feedback vertex set. In general, moving to a parameterization that takes larger values might enable running times with a smaller base of

*This research was partially funded by the Networks programme via the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research. The second author was supported by NWO Veni grant "Frontiers in Parameterized Preprocessing".

†The research was done while this author was at CWI, Amsterdam.

the exponent. In this paper, we therefore investigate the parameterized complexity of the q-Coloring and q-List-Coloring problems from a more fine-grained perspective.

In particular, we consider a hierarchy of graph parameters — ordered by their expressive strength — which is a common method in parameterized complexity, see e.g. [7] for an introduction. One of the strongest parameters for a graph problem is the number of vertices in a graph, in the following denoted by n. Björklund et al. showed that the chromatic number $\chi(G)$ (the smallest number of colors q such that G is q-colorable) of a graph G can be computed in time $\mathcal{O}^*(2^n)$ [2], so the base of the exponent in the runtime of the algorithm is independent of the value of $\chi(G)$. We show that if you consider a slightly weaker parameter, the size k of a vertex cover of G, it is very unlikely that there is a constant θ, such that q-Coloring can be solved in time $\mathcal{O}^*(\theta^k)$ for all fixed $q \in \mathcal{O}(1)$: It would imply that ETH is false.

However, we show that there is a simple algorithm that solves q-Coloring parameterized by vertex cover, and for which the base of the exponential in its runtime is strictly smaller than the base q that is potentially optimal for the treewidth parameterization. (A proof of the following proposition is deferred to the beginning of Section 3.)

Proposition 1. *There is an algorithm which decides whether a graph G is q-colorable and runs in time $\mathcal{O}^*((q-1.11)^k)$, where k denotes the size of a* given *vertex cover of G.*

On the other hand, the above algorithm does not obviously generalize to other parameterizations. To derive more general results about obtaining non-trivial runtime bounds for parameterized q-Coloring, we study graph classes with small *vertex modulators* to several graph classes $\mathcal{F}$: Given a graph $G = (V, E)$, a vertex modulator $X \subseteq V$ to $\mathcal{F}$ is a subset of its vertices such that if we remove X from G the resulting graph is a member of $\mathcal{F}$, i.e. $G - X \in \mathcal{F}$. If $|X| \leq k$, we say that $G \in \mathcal{F} + kv$. (For example, graphs that have a vertex cover of size at most k are Independent $+ kv$ graphs.) Hence, we study the following problems which were first investigated in this parameterized setting by Cai [3].

> q-(List-)Coloring on $\mathcal{F} + kv$ Graphs
> **Input:** An undirected graph G and a modulator $X \subseteq V(G)$ such that $G - X \in \mathcal{F}$ (and lists $\Lambda\colon V \to 2^{[q]}$).
> **Parameter:** $|X| = k$, the size of the modulator.
> **Question:** Can we assign each vertex v a color (from its list $\Lambda(v)$) such that adjacent vertices have different colors?

Given a No-instance (G, Λ) of q-List-Coloring we call (G', Λ') a No-subinstance of (G, Λ), if G' is an induced subgraph of G and for all vertices $v \in V(G')$: $\Lambda(v) = \Lambda'(v)$ such that (G', Λ') is also No. We show that if a graph class $\mathcal{F}$ has small No-certificates for q-List-Coloring then q-(List-)Coloring on $\mathcal{F} + kv$ graphs can be solved in time $\mathcal{O}^*((q-\varepsilon)^k)$, for some $\varepsilon > 0$. This notion was introduced by Jansen and Kratsch to prove the existence of polynomial kernels for said parameterizations [11].

In addition to that, we give some further structural insight into hereditary graph classes $\mathcal{F}$, for which $\mathcal{F}+kv$ graphs have non-trivial algorithms: We show that if the $(q+1)$-colorable members of $\mathcal{F}$ have bounded treedepth, then $\mathcal{F} + kv$ has $\mathcal{O}^*((q-\varepsilon)^k)$ time algorithms for q-Coloring when parameterized by the size k of a given modulator, for some $\varepsilon > 0$. We prove that this *treedepth-boundary* is in some sense tight: Arguably the most simple graphs of unbounded treedepth are paths. We show that q-Coloring cannot be solved in time $\mathcal{O}^*((q-\varepsilon)^k)$ for any $\varepsilon > 0$ on Path $+ kv$ graphs, unless SETH fails — strengthening the lower bound for Forest $+ kv$ graphs [13] via a somewhat simpler construction. Using this strengthened lower bound, we prove that if a hereditary graph class $\mathcal{F}$ excludes a complete bipartite graph $K_{t,t}$ for some constant t, then $\mathcal{F} + kv$ has $\mathcal{O}^*((q-\varepsilon)^k)$ time algorithms for q-(List-)Coloring *if and only if* the $(q+1)$-colorable members of $\mathcal{F}$ have bounded treedepth.

The rest of the paper is organized as follows: In Section 2 we give some fundamental definitions used throughout the paper. We present some upper bounds in the hierarchy in Section 3 and lower bounds in Section 4. In Section 5 we present the aforementioned tight relationship between the parameter treedepth and the existence of algorithms for q-Coloring with nontrivial runtime and we give concluding remarks in Section 6.

2 Preliminaries

We assume the reader to be familiar with the basic notions in graph theory and parameterized complexity and refer to [4, 5, 6, 8] for an introduction. We now give the most important definitions which are used throughout the paper.

We use the following notation: For $a, b \in \mathbb{N}$ with $a < b$, $[a] = \{1, \ldots, a\}$ and $[a..b] = \{a, a+1, \ldots, b\}$. The $\mathcal{O}^*$-notation suppresses polynomial factors in the input size n, i.e. $\mathcal{O}^*(f(n, \cdot)) = \mathcal{O}(f(n, \cdot) \cdot n^{\mathcal{O}(1)})$. For a function $f\colon X \to Y$, we denote by $f_{|X'}$ the restriction of f to $X' \subseteq X$.

2.1 Graphs and Parameters

Throughout the paper a graph G with vertex set $V(G)$ and edge set $E(G)$ is finite and simple. We sometimes shorthand $'V(G)'$ ($'E(G)'$) to $'V'$ ($'E'$) if it is clear from the context. For graphs G, G' we denote by $G' \subseteq G$ that G' is a subgraph of G, i.e. $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$. We often use the notation $n = |V|$ and $m = |E|$. For a vertex $v \in V(G)$, we denote by $N_G(v)$ (or simply $N(v)$, if G is clear from the context) the set of *neighbors* of v in G, i.e. $N_G(v) = \{w \in V(G) \mid \{v, w\} \in E(G)\}$.

For a vertex set $V' \subseteq V(G)$, we denote by $G[V']$ the subgraph *induced* by V', i.e. $G[V'] = (V', E(G) \cap V' \times V')$. A graph class $\mathcal{F}$ is called *hereditary*, if it is closed under taking induced subgraphs.

We now list a number of graph classes which will be important for the rest of the paper. A graph G is *independent*, if $E(G) = \emptyset$. A *cycle* is a connected graph all of whose vertices have degree two. A graph is a *forest*, if it does not contain a cycle as an induced subgraph and a *linear forest* if additionally its maximum degree is at most two. A connected forest is a *tree* and a tree of maximum degree at most two is a *path*. A graph G is a *split graph*, if its vertex set $V(G)$ can be partitioned into sets $W, Z \subseteq V(G)$ such that $G[W]$ is a clique and $G[Z]$ is independent. We define the class $\bigcup$Split containing all graphs that are disjoint unions of split graphs. A graph G is a *cograph* if it does not contain P_4, a path on four vertices, as an induced subgraph. A graph is *chordal*, if it does not have a cycle of length at least four as an induced subgraph. A cochordal graph is the edge complement of a chordal graph and the class $\bigcup$Cochordal contains all graphs that are disjoint unions of cochordal graphs.

Definition 2 (Parameterized Problem)**.** Let Σ be an alphabet. A *parameterized problem* is a set $\Pi \subseteq \Sigma^* \times \mathbb{N}$, the second component being the *parameter* which usually expresses a structural measure of the input. A parameterized problem is (strongly uniform) *fixed-parameter tractable* (fpt) if there exists an algorithm to decide whether $\langle x, k\rangle \in \Pi$ in time $f(k) \cdot |x|^{\mathcal{O}(1)}$ where f is a computable function.

The main focus of our research is how the function $f(k)$ behaves for q-Coloring w.r.t. different structural graph parameters, such as the size of a vertex cover.

In this paper we study a *hierarchy of parameters*, a term which we will now discuss. For a detailed introduction we refer to [7, Section 3]. For notational convenience, we denote by Π_p a parameterized problem with parameterization p. Suppose we have a graph problem and two parameterizations $p(G)$ and $p'(G)$ regarding some structural graph measure. We call parameterization $p'(G)$ *larger* than $p(G)$ if there is a function f, such that $f(p'(G)) \geq p(G)$ for all graphs G. Modulo some technicalities, we can then observe that if a problem Π_p is fpt, then $\Pi_{p'}$ is also fpt. This induces a partial ordering on all parameterizations based on which a hierarchy can be defined.

2.2 Exponential-Time Hypotheses

In 2001, Impagliazzo et al. made two conjectures about the complexity of q-SAT — the problem of finding a satisfying assignment for a Boolean formula in conjunctive normal form with clauses of size at most q [9, 10]. These conjectures are known as the Exponential-Time Hypothesis (ETH) and Strong Exponential-Time Hypothesis (SETH), formally defined below. For a survey of conditional lower bounds based on such conjectures, see [16].

Conjecture 3 (ETH [9])**.** *There is an $\varepsilon > 0$, such that* 3-SAT *on n variables cannot be solved in time $\mathcal{O}^*(2^{\varepsilon n})$.*

Conjecture 4 (SETH [9, 10])**.** *For every $\varepsilon > 0$, there is a $q \in \mathcal{O}(1)$ such that* q-SAT *on n variables cannot be solved in time $\mathcal{O}^*((2-\varepsilon)^n)$.*

3 Upper Bounds

In this section we present upper bounds for parameterized q-Coloring. In particular, in Section 3.1 we show that if a graph class $\mathcal{F}$ has No-certificates of constant size, then there exist $\mathcal{O}^*((q-\varepsilon)^k)$ time algorithms for q-Coloring on $\mathcal{F}+kv$ graphs for some $\varepsilon > 0$ depending on $\mathcal{F}$. In Section 3.2 we show that if the $(q+1)$-colorable members of a hereditary graph class $\mathcal{F}$ have bounded treedepth, then $\mathcal{F}$ has No-certificates of small size.

We begin by proving Proposition 1 and repeat its statement.

Proposition. *There is an algorithm which decides whether a graph G is q-colorable and runs in time $\mathcal{O}^*((q-1.11)^k)$, where k denotes the size of a* given *vertex cover of G.*

Proof. Let $X \subseteq V(G)$ be the given vertex cover of G of size k. We observe that if G is q-colorable, then any valid q-coloring of G can be extended from a valid q-coloring of $G[X]$. We know that in any q-coloring $\gamma\colon V \to [q]$ there is a color class that contains at most $\lfloor k/q \rfloor$ vertices in X. The algorithm now works as follows. We enumerate all sets $S \subseteq X$ of size at most $\lfloor k/q \rfloor$ and check whether they are independent. If so, let S' denote the set consisting of S together with all vertices in $V \setminus X$ that do not have a neighbor in S. Note that $G[S']$ is independent. We then recurse on the instance $G - S'$ with q decreased by one (and the size of the modulator decreased by $|S|$). Once $q = 2$, we check whether the remaining graph is 2-colorable (or equivalently, bipartite) in linear time.

We now compute the exponential dependence of the runtime by induction on q. As base cases we consider $q \in \{1,2,3\}$. The cases $q = 1$ and $q = 2$ are trivial, since the problem can be solved in polynomial time. For $q = 3$, the number of generated subproblems is bounded by $\sum_{\ell=0}^{\lfloor k/3 \rfloor} \binom{k}{\ell}$, which is at most $2^{H(1/3)k}$, where $H(x) = -x\log_2(x) - (1-x)\log_2(1-x)$ is the binary entropy [8, page 427]. Since $H(1/3) \leq 0.9183$, the algorithm generates at most $2^{0.9183k} \leq 1.89^k$ subproblems, all of which can be solved in polynomial time. For the induction step, let $q > 3$ and assume for the induction hypothesis that for $(q-1)$, the exponential dependence of the running time is upper bounded by $(q-1-1.11)^k$. Since the algorithm enumerates all subsets of X of size ℓ for each $\ell \in [\lfloor k/q \rfloor]$, and the size of the parameter decreases by ℓ in each call, using the induction hypothesis we find that the exponential term in the running time is upper bounded by

$$\sum_{\ell=0}^{\lfloor k/q \rfloor} \binom{k}{\ell} (q-2.11)^{k-\ell} \leq \sum_{\ell=0}^{k} \binom{k}{\ell} (q-2.11)^{k-\ell} \cdot 1^{\ell} = (q-2.11+1)^k = (q-1.11)^k,$$

since $\sum_{i=0}^{n} \binom{n}{i} \cdot a^i \cdot b^{n-i} = (a+b)^n$ by the Binomial Theorem.

We now argue the correctness of the algorithm, again by induction on q. The base cases, $q = 1$ and $q = 2$ are again trivially correct. For the induction step, consider $q > 2$ and assume for the induction hypothesis that the recursive calls to solve $(q-1)$-Coloring are correct. Suppose G has a q-coloring γ and let $T \subseteq V(G)$ denote the color class with the fewest vertices from X. Then, $|T \cap X| \leq k/q$, so the algorithm guesses the set $S = T \cap X$. Since the corresponding set S' contains all vertices in $G - X$ that do not have a neighbor in S and γ is a proper coloring, we can conclude that $S' \supseteq T$. Hence, $G - S'$ is a subgraph of the $(q-1)$-colorable graph induced by the other color classes of γ which the algorithm detects correctly by the induction hypothesis. Conversely, any $(q-1)$-coloring for $G - S'$ can be lifted to a q-coloring of G by giving all vertices in the independent set S' the same, new, color. □

3.1 Small No-Certificates

In earlier work [11], Jansen and Kratsch studied the kernelizability of q-Coloring and established a generic method to prove the existence of polynomial kernels for several parameterizations of q-Coloring. We now show that we can use their method to prove the existence of $\mathcal{O}^*((q-\varepsilon)^k)$ time algorithms, for some $\varepsilon > 0$, for several graph classes $\mathcal{F} + kv$ as well.

We first introduce the necessary terminology. Let (G, Λ) be an instance of q-List-Coloring. We call (G', Λ') a *subinstance* of (G, Λ), if G' is an induced subgraph of G and $\Lambda(v) = \Lambda'(v)$ for all $v \in V(G')$.

Definition 5 ($g(q)$-size No-certificates)**.** Let $g\colon \mathbb{N} \to \mathbb{N}$ be a function. A graph class $\mathcal{F}$ is said to have *$g(q)$-size* No*-certificates* for q-List-Coloring if for all No-instances (G, Λ) of q-List-Coloring with $G \in \mathcal{F}$ there is a No-subinstance (G', Λ') on at most $g(q)$ vertices.

Theorem 6. *Let $\mathcal{F}$ be a graph class with $g(q)$-size* No*-certificates for q-*List-Coloring*. Then, there is an $\varepsilon > 0$, such that q-*List-Coloring *(and hence, q-*Coloring*) on $\mathcal{F} + kv$ graphs can be solved in time $\mathcal{O}^*((q-\varepsilon)^k)$ given a modulator to $\mathcal{F}$ of size at most k. In particular, the algorithm runs in time $\mathcal{O}^*\left(\sqrt[g(q)\cdot q]{q^{g(q)\cdot q} - 1}^{\,k}\right)$, where the degree of the hidden polynomial depends on $g(q)$.*

Proof. Let $G \in \mathcal{F}+kv$ with vertex modulator X, such that $\mathcal{F}$ has $g(q)$-size No-certificates for q-List-Coloring. The idea of the algorithm is to enumerate partial colorings of X, except some colorings for which it is clear that they cannot be extended to a proper coloring of the entire instance. The latter can occur as follows: After choosing a coloring for some vertices of X and removing the chosen colors from the lists of their neighbors, a No-subinstance appears in the graph $G - X$. If the minimal No-subinstances have constant size, then for any given instance, either *all* proper colorings on X can be extended onto $G - X$, or there is a way to find a constant-size set $X' \subseteq X$ of vertices for which at least one of the $q^{|X'|}$ colorings would trigger a No-subinstance and can therefore be discarded. Branching on the remaining relevant colorings for X' then gives a nontrivial running time. An outline is given in Algorithm 1.

The main condition (line 2) checks whether the input graph G contains the graph of a minimal No-instance as an induced subgraph. If so, we look for a neighborhood of $V(G')$ in X (the sets $X_1, \ldots, X_q$), which can block the colors that are on the lists Λ but not on the lists of the minimal No-instance. If these conditions are satisfied, then we know that we can exclude the coloring on $X_1, \ldots, X_q$ which assigns each vertex $v \in X_c$ the color c (for all $c \in [q]$): This coloring induces a No-subinstance on (G, Λ). It suffices to use sets X_c of at most $g(q)$ vertices each. To induce the No-instance, in the worst case we need a different vertex in X_c for each of the $g(q)$ vertices in H that do not have c on their list. Hence, as described from line 4 on, we enumerate all colorings $\gamma\colon \mathcal{X} \to [q]$ (where $\mathcal{X} = \bigcup_i X_i$) except the one we just identified as not being extendible to $G - X$. For each such γ, we make a copy of the current instance and 'assign' each vertex v corresponding to a vertex in $\mathcal{X}$ the color $\gamma(v)$: We remove $\gamma(v)$ from the lists of its neighbors and then remove v from the copy instance. In the worst case we therefore recurse on $q^{q \cdot g(q)} - 1$ instances with the size of the vertex modulator decreased by $q \cdot g(q)$. If during a branch in the computation, the condition in line 2 is not satisfied, then we know that there is no coloring on the modulator that cannot be extended to the vertices outside the modulator and hence it is sufficient to compute whether $G[X]$ is q-list-colorable using the standard $\mathcal{O}^*(2^n)$ algorithm for computing the chromatic number [2]. As soon as one branch returns Yes, we can terminate the algorithm, since we found a valid list coloring.

Input : A graph $G \in \mathcal{F} + kv$ with vertex modulator X and $\Lambda\colon V \to 2^{[q]}$.
Output: YES, if G is q-list-colorable, NO otherwise.

1 Let ζ be the set of NO-instances of q-LIST-COLORING for $\mathcal{F}$ of size at most $g(q)$, which is computed once by complete enumeration;
2 **if** *there exist* $(H, \Lambda_H) \in \zeta$, $G' \subseteq G - X$ *and* $X_1, \ldots, X_q \subseteq X$ *of size at most* $g(q)$ *each such that:*
  1. $\exists$ *isomorphism* $\varphi\colon V(G') \to V(H)$
  2. *For all* $c \in [q]$ *and* $v \in X_c$ *we have* $c \in \Lambda(v)$
  3. $(\forall v \in V(G'))(\forall c \in \Lambda(v) \setminus \Lambda_H(\varphi(v)))\ \exists w \in X_c$ *with* $\{v, w\} \in E(G)$
3 **then**
4   **foreach** *proper coloring* $\gamma\colon \mathcal{X} \to [q]$ *where* $\mathcal{X} = \bigcup_i X_i$ *and* $\forall v \in \mathcal{X}$*:* $\gamma(v) \in \Lambda(v)$ **do**
5   **if** $(\forall c \in [q])(\forall v \in X_c):\ \gamma(v) = c$ **then**
6   Skip this coloring, it is not extendible to $G - X$;
7   **else**
8   Create a copy (G'', Λ'') of (G, Λ) and denote by $\mathcal{X}''$ the vertex set in G'' corresponding to $\mathcal{X}$ in G;
9   For each vertex $v \in \mathcal{X}''$ and each neighbor w of v: Remove $\gamma(v)$ from $\Lambda''(w)$;
10   Recurse on $(G'' - \mathcal{X}'', \Lambda'')$;
11   **if** *the recursive call returns* YES **then**
12   **Return** YES and terminate the algorithm;
13   **Return** NO;
14 **else**
15   Decide whether $(G[X], \Lambda)$ is q-list-colorable and if so, return YES;

Algorithm 1: q-LIST-COLORING for $\mathcal{F} + kv$ graphs where $\mathcal{F}$ has $g(q)$-size NO-certificates.

Claim 7. *If the condition of line 2 does not hold, then G is q-list-colorable if and only if $G[X]$ is q-list-colorable.*

Proof. The forward direction is trivial since any proper coloring of G yields a proper coloring of its induced subgraph $G[X]$. To prove the reverse direction, we show that if the condition of line 2 fails, any proper q-list-coloring of $G[X]$ can be extended to a proper q-list-coloring of the entire graph.

Suppose that $\gamma\colon X \to [q]$ is a proper q-list-coloring of $G[X]$. Define a q-list-coloring instance $(G - X, \Lambda')$ on the graph $G - X$, where for each vertex $v \in V(G - X)$ the list of allowed colors is $\Lambda'(v) \coloneqq \Lambda(v) \setminus \{\gamma(u) \mid u \in N_G(v) \cap X\}$. If $(G - X, \Lambda')$ has a proper q-list-coloring γ', then we can obtain a proper q-list-coloring for G by following γ on the vertices in X and γ' on the vertices outside X. The fact that the colors for vertices in X are removed from the Λ'-lists of their neighbors ensures that the resulting coloring is proper, and since each list of Λ' is a subset of the corresponding list in Λ, the coloring satisfies the list requirements. We therefore complete the proof by showing that $(G - X, \Lambda')$ must be a YES-instance. Assume for a contradiction that $(G-X, \Lambda')$ has answer NO. Since $G-X \in \mathcal{F}$, which has $g(q)$-size NO-certificates, there is an induced subinstance (G', Λ'') of $(G - X, \Lambda')$ on at most $g(q)$ vertices, where G' is an induced subgraph of $G - X$ and therefore of G. Since (G', Λ'') is a NO-instance on at most $g(q)$ vertices, the instance $(H \coloneqq G', \Lambda_H \coloneqq \Lambda'')$ is contained in the set of enumerated small NO-instances. For each $v \in V(G')$, for each color c that belongs to $\Lambda(v)$ but not to $\Lambda'(v) = \Lambda''(v)$ we have $\gamma(u) = c$ for some $u \in N_G(v) \cap X$, by definition of Λ'. Initialize $X_1, \ldots, X_q$ as empty vertex sets. For each $v \in V(G')$ and color $c \in \Lambda(v) \setminus \Lambda''(v)$, add such a vertex u to X_c. Since γ satisfies the list constraints, for each vertex $v \in X_c$ with $c \in [q]$ we have $c \in \Lambda(v)$. Hence these structures satisfy the conditions of line 2; a contradiction. ⌟

Claim 8. *If the condition of line 5 holds, then the coloring γ cannot be extended to a proper q-list-coloring of G.*

Proof. To extend the coloring γ to the entire graph G, each vertex v of $G - X$ has to receive a color of $\Lambda(v) \setminus \{\gamma(u) \mid u \in N_G(v) \cap X\}$, since the color of v must differ from that of its neighbors. For each vertex v in the subgraph G', for each color c in $\Lambda(v) \setminus \Lambda_H(\phi(v))$ there is a neighbor of v in X_c (by condition 3 of line 2) that is colored c (by line 5). Hence the colors available for v in an extension form a subset of $\Lambda_H(\phi(v))$. But since G' is isomorphic to H, and (H, Λ_H) is a NO-instance, no such extension is possible as it would yield a proper q-list-coloring of (H, Λ_H). ⌟

Using these claims we prove correctness by induction on the nesting depth of recursive calls in which the condition of line 2 is satisfied. If line 2 is not satisfied (which includes the base case of the induction), then the algorithm is correct by Claim 7 and the fact that we invoke a correct algorithm in line 15 as a subroutine [2]. Now, suppose that the condition of line 2 is satisfied, and assume by the induction hypothesis that the recursive calls (line 10) are correct. Let (G, Λ) with modulator X be the current instance. We recurse on each possible proper q-list-coloring of the set $\mathcal{X}$, except the one described in the condition in line 5 for which Claim 8 shows it cannot be extended to a proper q-list-coloring. If (G, Λ) has a proper q-list-coloring γ, then in the branch where we correctly guess the restriction of γ onto the vertices in $\mathcal{X}$ we find a YES-answer: the restriction of γ on $G'' - \mathcal{X}''$ is a proper q-list-coloring of $(G'' - \mathcal{X}, \Lambda'')$ since the colors we removed from the lists were not used on $G'' - \mathcal{X}''$ (they were used on their neighbors in $\mathcal{X}''$). Conversely, if some recursive call yields a YES-answer, then since we restricted the lists before going into recursion, we can extend a proper q-list-coloring on the smaller instance with the coloring γ on $\mathcal{X}$ to obtain a proper q-list-coloring of (G, Λ).

We now analyze the runtime. Since q is a constant, $g(q)$ is constant as well and computing the set ζ in line 1 can be done in constant time. Using the same argument we observe that the condition in line 2 checks a polynomial number of options: The size of ζ and the size of its elements are constant and hence there is a polynomial number (at most $|\zeta| \cdot n^{g(q)}$) of subgraphs of G to consider. Since $t \leq g(q)$, we can enumerate all isomorphisms and all sets $X_1, \ldots, X_q$ with an additional polynomial overhead. Hence the work in each iteration, excluding the recursive calls and line 15, is polynomial.

Line 15 can be done in time $\mathcal{O}^*(2^{k+q})$, which is $\mathcal{O}^*(2^k)$ for constant q, using the $\mathcal{O}^*(2^n)$ algorithm for CHROMATIC NUMBER [2] and the following classic reduction from q-list-coloring to q-coloring. Instance $(G[X], \Lambda)$ has a proper q-list-coloring if and only if the following graph is q-colorable: starting from $G[X]$, add a q-clique whose vertices represent the q colors, and edges between every $v \in X$ and the clique-vertices whose colors do not appear on $\Lambda(v)$.

Using these facts we bound the total runtime. In the worst case we branch on $q^{q \cdot g(q)} - 1$ instances in which the size of the modulator decreased by $q \cdot g(q)$. By standard techniques [15, Proposition 8.1], this branching vector can be shown to generate a search tree with $\mathcal{O}(\sqrt[g(q) \cdot q]{q^{g(q) \cdot q} - 1}^{\,k})$ nodes. If the work at each node of the tree is polynomial, we therefore get a total runtime bound matching the theorem statement. If we do not execute line 15, then indeed a single iteration takes polynomial time. If line 15 is executed, then we spend $\mathcal{O}^*(2^k)$ time on the iteration. However, in that case we do not recurse further, so the time spent solving the problem on $G[X]$ can be discounted against the fact that we do not explore a search tree of size $\sqrt[g(q) \cdot q]{q^{g(q) \cdot q} - 1}^{\,k} > 2^k$ for $q \geq 3$. The time bound follows.

This concludes the proof of Theorem 6, noting that we can apply any algorithm for q-LIST-COLORING to solve an instance of q-COLORING by giving each vertex in a given instance of q-COLORING a full list. □

In the light of [11, Lemmas 2-4] we can apply Theorem 6 to a number of graph classes.

Corollary 9 (of Thm. 6 and Cor. 1 and 2 and Lemmas 2, 3 and 4 in [11])**.** *There is an $\varepsilon > 0$, such that the q-*COLORING *and q-*LIST-COLORING *problems on $\mathcal{F} + kv$ graphs can be solved in time $\mathcal{O}^*((q - \varepsilon)^k)$* given *a modulator to $\mathcal{F}$ of size k, where $\mathcal{F}$ is one of the following classes:* INDEPENDENT, $\bigcup$SPLIT, $\bigcup$COCHORDAL *and* COGRAPH.

Remark. Rather than on the maximum *size* of any minimal No-instance of a graph class $\mathcal{F}$, the runtime of the algorithm described in Theorem 6 depends on their maximum *deficiency*, defined as $d(G, \Lambda) = \sum_{v \in V(G)} q - |\Lambda(v)|$ (as we need one vertex in the modulator for each color we want to block from the list of a vertex in the No-instance). We would like to note that the runtime of Algorithm 1 can be improved when analyzing the deficiency of the No-instances constructed in the proofs of [11, Lemmas 2-4].

3.2 Bounded Treedepth

We now show that if the $(q+1)$-colorable members of a hereditary graph class $\mathcal{F}$ have treedepth at most t, then $\mathcal{F}$ has q^t-size No-certificates. For a detailed introduction to the parameter treedepth and its applications, we refer to [14, Chapter 6].

Definition 10 (Treedepth)**.** Let G be a connected graph. A *treedepth decomposition* $\mathcal{T} = (V(G), F)$ is a rooted tree on the vertex set of G such that the following holds. For $v \in V(G)$, let $\mathcal{A}_v$ denote the set of ancestors of v in $\mathcal{T}$. Then, for each edge $\{v, w\} \in E(G)$, either $v \in \mathcal{A}_w$ or $w \in \mathcal{A}_v$.

The *depth* of $\mathcal{T}$ is the number of vertices on a longest path from the root to a leaf. The *treedepth* of a connected graph is the minimum depth of all its treedepth decompositions. The treedepth of a disconnected graph is the maximum treedepth of its connected components.

The main result of this section is the following.

Lemma 11. *Let $\mathcal{F}$ be a hereditary graph class whose $(q+1)$-colorable members have treedepth at most t. Then, $\mathcal{F}$ has q^t-size* No*-certificates for q-*List-Coloring*.*

Proof. Consider an arbitrary No-instance (G, Λ) of q-List-Coloring for a graph $G \in \mathcal{F}$. If G is not $(q+1)$-colorable (ignoring the lists Λ), then remove an arbitrary vertex from G. Since this lowers the chromatic number by at most one, the resulting graph will still be a No-instance of q-Coloring and therefore of q-List-Coloring. Repeat this step until arriving at a subinstance (G', Λ') that is $(q+1)$-colorable. By assumption, G' has treedepth at most t. Fix an arbitrary treedepth decomposition for G' of depth at most t. We use the decomposition to find a No-subinstance by a recursive algorithm. Given a No-instance (G, Λ) and a treedepth decomposition $\mathcal{T}$ of G of depth at most t, it marks a set $M \subseteq V(G)$ such that the subinstance induced by M is still a No-instance and $|M| \leq q^t$.

If the treedepth decomposition has depth one, then mark a vertex with an empty list (which must exist if the answer is No). When the decomposition has depth > 1, then do the following. Let $\mathcal{T}$ be a tree of the decomposition that represents a connected component C that cannot be list colored. Let r be its root. For each color $c \in \Lambda(r)$, create a list coloring instance $(C - \{r\}, \Lambda_c)$ on a graph of treedepth $t-1$ as follows. The graph is $C - \{r\}$ and its decomposition consists of $\mathcal{T}$ minus its root (which therefore splits into a forest), and the lists equal the old lists except that we remove c from the lists of all of r's neighbors. Observe that the subinstance has answer No, since otherwise the component C has a proper coloring. Recursively call the algorithm on this smaller instance to get a set M_c that preserves the fact that $(C - \{r\}, \Lambda_c)$ has answer No. After getting the answers from all the recursive calls, mark the vertices in the set M containing the root r together with the union of the sets M_c for all $c \in \Lambda(r)$.

To bound the size of the set M, let $h(t)$ denote the maximum number of marked vertices in a treedepth decomposition of depth t. Clearly, $h(1) = 1$. If $t > 1$, we recurse in at most q ways on instances of treedepth $t-1$, hence the number of marked vertices is described by the recurrence $h(t) \leq q \cdot h(t-1) + 1$ which resolves to $h(t) \leq \frac{q^t - 1}{q-1}$ and hence $h(t) \leq q^t$, as claimed.

We now prove that the above described marking procedure preserves the No-answer of an instance of q-List-Coloring. We use induction on t, the depth of a treedepth decomposition $\mathcal{T}$ (with root r) of the graph G of a q-List-Coloring No-instance (G, Λ). The base case $t = 1$ is trivially correct: A graph has treedepth one if and only if it is independent and since a graph is q-list-colorable if and only if its connected components are

q-list-colorable, the only minimal No-instance of treedepth one is a single vertex with an empty list, which we marked in the procedure. Now suppose for the induction hypothesis that $t > 1$ and for all $t' < t$, the marking procedure is correct. Consider a treedepth decomposition $\mathcal{T}$ of a connected component C of (a subgraph of) G and the set M of currently marked vertices. Suppose for the sake of a contradiction that $(G[M], \Lambda_{|M})$ is a Yes-instance with proper list-coloring $\gamma \colon M \to [q]$. Let $C_{\gamma(r)}$ denote the connected component of $C - \{r\}$ we branched on for color $\gamma(r)$ and $M_{\gamma(r)}$ the set of marked vertices in $C_{\gamma(r)}$. By the induction hypothesis (which applies since $C_{\gamma(r)}$ has treedepth at most $t-1$), we know that $(G[M_{\gamma(r)}], \Lambda_{\gamma(r)})$ is a No-instance of q-List-Coloring. But $\gamma_{|M_{\gamma(r)}}$ is a valid solution for that instance if γ is a proper coloring: the color of r cannot appear on its neighbors in $M_{\gamma(r)}$, and therefore $\gamma_{|M_{\gamma(r)}}$ satisfies the list constraints of $\Lambda_{\gamma(r)}$. This contradicts the fact that $(g[M_{\gamma(r)}], \Lambda_{\gamma(r)})$ is a No-instance. □

To see the versatility of Lemma 11, observe that the vertices of a $(q+1)$-colorable split graph can be partitioned into a clique of size at most $(q+1)$ and an independent set, which makes it easy to see that they have treedepth at most $q+2$. Since the treedepth of a disconnected graph equals the maximum of the treedepth of its connected components, we then get a finite (q^{q+2}) bound on the size of minimal No-instances for q-List-Coloring on $\bigcup$ Split graphs. An ad-hoc argument was needed for this in earlier work [11, Lemma 2], albeit resulting in a better bound $(q + 4^q)$.

4 Lower Bounds

In this section we prove lower bounds for q-Coloring in the parameter hierarchy. Since in the following, the '$\mathcal{F} + kv$'-notation is more convenient for the presentation of our results, we will mostly refer to graphs which have a vertex cover of size k as Independent $+ kv$ graphs and graphs that have a feedback vertex set of size k as Forest $+ kv$ graphs.

In Section 4.1 we show that there is no universal constant θ, such that q-Coloring on Independent $+ kv$ graphs can be solved in time $\mathcal{O}^*(\theta^k)$ for all fixed $q \in \mathcal{O}(1)$, unless ETH fails. We generalize the lower bound modulo SETH for Forest $+ kv$ graphs [13] to Linear Forest$+kv$ (and Path$+kv$) graphs in Section 4.2. Note that by the constructions we give in their proofs, the lower bounds also hold in case a modulator of size k to the respective graph class is given.

4.1 No Universal Constant for Independent+kv graphs

The following theorem shows that, unless ETH fails, the runtime of any algorithm for q-Coloring parameterized by vertex cover (equivalently, on Independent $+ kv$ graphs), always has a term depending on q in the base of the exponent.

Theorem 12. *There is no (universal) constant θ, such that for all fixed $q \in \mathcal{O}(1)$, q-Coloring on Independent $+ kv$ graphs can be solved in time $\mathcal{O}^*(\theta^k)$, unless ETH fails.*

Proof. Assume we can solve q-Coloring on Independent $+ kv$ graphs in time $\mathcal{O}^*(\theta^k)$. We will use this hypothetical algorithm to solve 3-SAT in $\mathcal{O}^*(2^{\varepsilon n})$ time for arbitrarily small $\varepsilon > 0$, contradicting ETH. We present a way to reduce an instance φ of 3-SAT to an instance of $3q$-List-Coloring for q an arbitrary power of 2. The larger q is, the smaller the vertex cover of the constructed graph will be. It will be useful to think of a color $c \in [q]$ ($q = 2^t$ for some $t \in \mathbb{N}$) as a bitstring of length t, which naturally encodes a truth assignment to t variables. The entire color range $[3q]$ partitions into three consecutive blocks of q colors, so that the same truth assignment to t variables can be encoded by three distinct colors $c, c+q$, and $c+2q$ for some $c \in [q]$. The reason for the threefold redundancy is that clauses in φ have size three and will become clear in the course of the proof.

Given an instance φ of 3-SAT, we create a graph G_{3q} and lists $\Lambda \colon V(G_{3q}) \to [3q]$ as follows. First, we add $\lceil n/\log q \rceil$ vertices $v_{1,i}$ (where $i \in [\lceil n/\log q \rceil]$) to $V(G_{3q})$, whose colorings will correspond to the truth assignments of the variables $x_1, \ldots, x_n$ in φ. We let

$\Lambda(v_{1,i}) = [q]$ for all these vertices. In particular, the variable x_i will be encoded by vertex $v_{1,\lceil i/\log q\rceil}$. We add two more layers of vertices $v_{2,i}, v_{3,i}$ (where $i \in [\lceil n/\log q\rceil]$) to G_{3q} whose lists will be $\Lambda(v_{2,i}) = [(q+1)..2q]$ and $\Lambda(v_{3,i}) = [(2q+1)..3q]$, respectively (for all i). Throughout the proof, we denote the set of all these *variable vertices* by $\mathcal{V} = \bigcup_{i,j} v_{i,j}$, where $i \in [3]$ and $j \in [\lceil n/\log q\rceil]$.

For each $i \in [2]$ and $j \in [\lceil n/\log q\rceil]$ we do the following. For each pair of colors $c \in [((i-1)q+1)..(i\cdot q)]$ and $c' \in [(i\cdot q+1)..((i+1)q)]$ such that $c+q \neq c'$, we add a vertex $u^{i,j}_{c,c'}$ with list $\Lambda(u^{i,j}_{c,c'}) = \{c, c'\}$ and make it adjacent to both $v_{i,j}$ and $v_{i+1,j}$. Note that this way, we add $\mathcal{O}(q^2)$ and hence a constant number of vertices for each such i and j. We denote the set of all vertices $u^{\cdot,\cdot}_{\cdot,\cdot}$ for all i and j by $\mathcal{U}$.

Claim 13. *Let $i \in [2]$ and $j \in [\lceil n/\log q\rceil]$. In any proper list-coloring of G_{3q}, the color $c \in [((i-1)q+1)..(i\cdot q)]$ appears on $v_{i,j}$ if and only if the color $c+q$ appears on $v_{i+1,j}$. If color $c \in [((i-1)q+1)..(i\cdot q)]$ appears on $v_{i,j}$ and $c' = q+c$ appears on $v_{i+1,j}$, then all vertices $u^{i,j}_{\cdot,\cdot}$ can be assigned a color from their list that does not appear on a neighbor.*

Proof. We first observe that the lists of $v_{i,j}$ and $v_{i+1,j}$ are $\Lambda(v_{i,j}) = [((i-1)q+1)..(i\cdot q)]$ and $\Lambda(v_{i+1,j}) = [(i\cdot q+1)..(i+1)q]$, respectively. Suppose that c appears on $v_{i,j}$. Then, for every color $c' \in [(i\cdot q+1)..((i+1)q)]$ with $c' \neq c+q$ there is a neighbor $u^{i,j}_{c,c'}$ of $v_{i,j}$ with list $\Lambda(u^{i,j}_{c,c'}) = \{c, c'\}$. Since c already appears on a neighbor of $u^{i,j}_{c,c'}$, we know that in each proper coloring, $u^{i,j}_{c,c'}$ must be colored c', blocking this color for its neighbor $v_{i+1,j}$. As this prevents any color $c' \neq c+q$ from appearing on $v_{i+1,j}$, in any proper list-coloring that vertex is colored $c+q$. (A proof of the converse works the same way.)

Now suppose c appears on $v_{i,j}$ and $c+q$ appears on $v_{i,j}$. Then any vertex $u^{i,j}_{c',c''}$ created by the process above has $\{c', c''\} \neq \{c, c+q\}$ by construction. Hence $u^{i,j}_{c',c''}$ can safely be assigned a color of $\{c', c''\} \setminus \{c, c+q\}$, which does not appear on any of its neighbors. ⌟

Claim 13 shows that in any proper list-coloring of $\mathcal{V}$, there is a threefold redundancy: If color c appears on $v_{1,i}$, then color $c+q$ appears on $v_{2,i}$ and $c+2q$ appears on $v_{3,i}$. We associate a proper list-coloring of $\mathcal{V}$ with the truth assignment whose True/False assignment to the i-th block of $\log q$ consecutive variables follows the 1/0-bit pattern in the least significant $\log q$ bits of the binary expansion of the color of vertex $v_{1,i}$. Conversely, given a truth assignment to $x_1, \ldots, x_n$ we associate it to the coloring of $\mathcal{V}$ where the color of vertex $v_{1,i}$ is given by the number whose least significant $\log q$ bits match the truth assignment to the i-th block of $\log q$ variables, and any remaining bits are set to 0. The colors of $v_{2,i}$ and $v_{3,i}$ are q and $2q$ higher than the color of $v_{1,i}$.

For each clause $C_j \in \varphi$ we will now add a number of *clause vertices* to ensure that if C_j is not satisfied by a given truth assignment of its variables, then the corresponding coloring of the vertices $\mathcal{V}$ cannot be extended to (at least) one of these clause vertices.

Let $C_j \in \varphi$ be a clause with variables x_{j_1}, x_{j_2}, and x_{j_3}. Then, $v_{1,\lceil j_1/\log q\rceil}, v_{1,\lceil j_2/\log q\rceil}$, and $v_{1,\lceil j_3/\log q\rceil}$ denote the vertices whose colorings encode the truth assignments of the respective variables. In the following, let $j'_i = \lceil j_i/\log q\rceil$ for $i \in [3]$. Note that there is precisely one truth assignment of the variables x_{j_1}, x_{j_2}, and x_{j_3} that does not satisfy C_j. Choose $\ell_1, \ell_2, \ell_3 \in \{0,1\}$ such that $\ell_i = 0$ if and only if the i-th variable in C_j appears negated. For $i \in [3]$ let $F_i \subseteq [q]$ be those colors whose binary expansion differs from ℓ_i at the $(j_i \bmod (\log q))$-th least significant bit, and define $F_i^{+q} := \{q+c \mid c \in F_i\}$ and $F_i^{+2q} := \{2q+c \mid c \in F_i\}$. This implies that the truth assignment encoded by a proper list-coloring of $\mathcal{V}$ falsifies the i-th literal of C_j if and only if it uses a color from F_i on vertex v_{1,j'_i}. By Claim 13, this happens if and only if it uses a color from F_i^{+q} on vertex v_{2,j'_i}, which happens if and only if it uses a color of F_i^{+2q} on vertex v_{3,j'_i}. Hence the assignment encoded by a proper list-coloring satisfies clause C_j if and only if the colors appearing on $(v_{1,j'_1}, v_{2,j'_2}, v_{3,j'_3})$ do not belong to the set $F_1 \times F_2^{+q} \times F_3^{+2q}$. To encode the requirement that C_j be satisfied into the graph G_{3q}, for each $(\gamma_1, \gamma_2, \gamma_3) \in F_1 \times F_2^{+q} \times F_3^{+2q}$ we add a vertex $w_{\gamma_1,\gamma_2,\gamma_3}$ to G_{3q} that is adjacent to v_{1,j'_1}, v_{2,j'_2}, and v_{3,j'_3} and whose list is $\{\gamma_1, \gamma_2, \gamma_3\}$. The threefold redundancy we incorporated ensures that the three colors in each forbidden triple are all

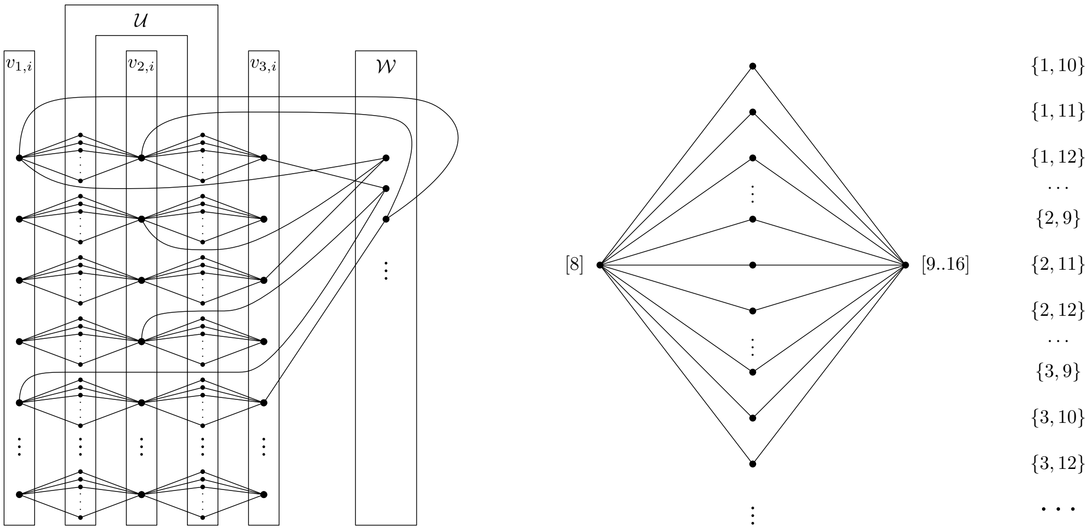

Figure 1: An illustration of the reduction given in the proof of Theorem 12. On the left there is a schematic overview and on the right an example of a subgraph induced by two vertices $v_{1,j}$ and $v_{2,j}$ together with the corresponding vertices in $\mathcal{U}$ for 24-List-Coloring (where the lists of the vertices in the middle are displayed to their right).

distinct. Therefore, if one of the three neighbors of $w_{\gamma_1,\gamma_2,\gamma_3}$ does not receive its forbidden color, then $w_{\gamma_1,\gamma_2,\gamma_3}$ can properly receive that color. This would not hold if there could be duplicates among the forbidden colors. The reduction is finished by adding these vertices for each clause $C_j \in \varphi$. We denote the set of clause vertices by $\mathcal{W}$. For an illustration see Figure 1.

Claim 14. *The formula φ has a satisfying assignment, if and only if the graph G_{3q} obtained via the above reduction is $3q$-list-colorable.*

Proof. Suppose φ has a satisfying assignment $\psi\colon [n] \to \{0,1\}$. Let γ_ψ be the corresponding proper coloring of $\mathcal{V}$, as described above. We argue that γ_ψ can be extended to the vertices $\mathcal{W}$ as well. Let $C_j \in \varphi$ be a clause on variables x_{j_1}, x_{j_2}, and x_{j_3} and let $w_{\gamma_1,\gamma_2,\gamma_3} \in \mathcal{W}$ be a vertex we introduced in the construction above for C_j. For $i \in [3]$, let $\gamma_\psi^i = \gamma_\psi(v_{i,\lceil j_i/\log q\rceil})$.

Since γ_ψ encodes a satisfying assignment, we know that there exists an $i^* \in [3]$, such that $\gamma_\psi^{i^*} \neq \gamma_{i^*}$ (since otherwise, ψ is not a satisfying assignment to φ). Hence, the color γ_{i^*} is not blocked from the list of vertex $w_{\gamma_1,\gamma_2,\gamma_3}$ which can then be properly colored. By Claim 13 we know that the remaining vertices $\mathcal{U}$ can be properly list-colored as well.

Conversely, suppose that G_{3q} is properly list-colored. We show that each proper coloring must correspond to a truth assignment that satisfies φ. For the sake of a contradiction, suppose that there is a proper list-coloring $\gamma_\psi\colon V(G) \to [3q]$ which encodes a truth assignment ψ that does not satisfy φ. Let $C_j \in \varphi$ denote a clause which is not satisfied by ψ on variables x_{j_1}, x_{j_2}, and x_{j_3}. For $i \in [3]$, we denote by $\gamma_\psi^i = \gamma_\psi(v_{i,\lceil j_i/\log q\rceil})$ the colors of the variable vertices encoding the truth assignment of the variables in C_j. Since ψ does not satisfy C_j we know that we added a vertex $w_{\gamma_\psi^1,\gamma_\psi^2,\gamma_\psi^3}$ to $\mathcal{W}$, which is adjacent to $v_{1,\lceil j_1/\log q\rceil}$, $v_{2,\lceil j_2/\log q\rceil}$, and $v_{3,\lceil j_3/\log q\rceil}$. This means that the colors $\gamma_\psi^1, \gamma_\psi^2$, and γ_ψ^3 appear on a vertex which is adjacent to $w_{\gamma_\psi^1,\gamma_\psi^2,\gamma_\psi^3}$ and hence the coloring γ_ψ is improper, a contradiction. ⌟

We have shown how to reduce an instance of 3-SAT to an instance of $3q$-List-Coloring. We modify the graph G_{3q} to obtain an instance of q-Coloring which preserves the correctness of the reduction. We add a clique K_{3q} of $3q$ vertices to G_{3q}, each of whose vertices represents one color. We make each vertex in $v \in \mathcal{V} \cup \mathcal{W} \cup \mathcal{U}$ adjacent to each vertex in K_{3q} that represents a color which does not appear on v's list in the list-coloring instance.

(The same trick was used in the proof of Theorem 6.1 in [13].) It follows that the graph without K_{3q} has a proper list-coloring if and only if the new graph has a proper $3q$-coloring.

We now compute the size of G_{3q} in terms of n and q and give a bound on the size of a vertex cover of G_{3q}. We observe that $|\mathcal{V}| = 3\lceil n/\log q\rceil$, $|\mathcal{U}| = \mathcal{O}(q^2 \cdot \lceil n/\log q\rceil)$, and clearly, $|V(K_{3q})| = 3q$. To bound the size of $\mathcal{W}$, we observe that for each clause C_j, we added $(2^{\log q-1})^3$ vertices (since we considered all triples of bitstrings of length $\log q$ where one character is fixed in each string) and hence $|\mathcal{W}| = \mathcal{O}(q^3 \cdot m)$ with m the number of clauses in ϕ. It is easy to see that $\mathcal{V} \cup V(K_{3q})$ is a vertex cover of G_{3q} and hence G_{3q} has a vertex cover of size $3\lceil n/\log q\rceil + 3q$.

Assuming there is an algorithm that solves q-Coloring on Independent+kv graphs in time $\mathcal{O}^*(\theta^k)$ together with an application of the above reduction (whose correctness follows from Claim 14) would yield an algorithm for 3-SAT that runs in time

$$\begin{aligned} &\theta^{3\lceil n/\log q\rceil+3q} \cdot ((q^2+3)\lceil n/\log q\rceil + 3q + q^3 \cdot m)^{\mathcal{O}(1)} = \theta^{3\lceil n/\log q\rceil+3q} \cdot (n+m)^{\mathcal{O}(1)} \\ = \; &\theta^{3\lceil n/\log q\rceil+3q} \cdot n^{\mathcal{O}(1)} = \mathcal{O}^*\left(\theta^{3\lceil n/\log q\rceil+3q}\right) = \mathcal{O}^*\left(2^{\frac{3\log\theta}{\log q}n}\right). \end{aligned}$$

Hence, for any $\varepsilon > 0$ we can choose a constant q large enough such that $(3\log\theta)/(\log q) < \varepsilon$ and Theorem 12 follows. □

4.2 No Nontrivial Runtime Bound for Path+kv Graphs

We now strengthen the lower bound for Forest+kv graphs due to [13] to the more restrictive class of Linear Forest + kv graphs. The key idea in our reduction is that we treat the clause size in a satisfiability instance as a constant, which allows for constructing a graph of polynomial size. The following lemma describes the clause gadget that will be used in the reduction.

Lemma 15. *For each $q \geq 3$ there is a polynomial-time algorithm that, given $(c_1, \ldots, c_m) \in [q]^m$, outputs a q-list-coloring instance (P, Λ) where P is a path of size $\mathcal{O}(m)$ containing distinguished vertices $(\pi_1, \ldots, \pi_m)$, such that the following holds. For each $(d_1, \ldots, d_m) \in [q]^m$ there is a proper list-coloring γ of P in which $\gamma(\pi_i) \neq d_i$ for all i, if and only if $(c_1, \ldots, c_m) \neq (d_1, \ldots, d_m)$.*

Proof. The path P consists of consecutive vertices $v_0, v_1, \ldots, v_{6m}, v_{6m+1}$. Vertex v_0 is the source and v_{6m+1} is the sink. The remaining $6m$ vertices are split into m groups $D_1, \ldots, D_m$ consisting of six consecutive vertices $v_{6(i-1)+1}, \ldots, v_{6i}$ ($i \in [m]$) each. We first add some colors to the lists of these vertices which are allowed regardless of $(c_1, \ldots c_m)$. Later we will add some more colors to the lists of selected vertices to obtain the desired behavior.

Initialize the 'default' list of vertex v_i for $i \in [6m]$ to contain the two colors $\{(i \bmod 3) + 1, (i + 1 \mod 3) + 1\}$, so that the first few lists are $\{2, 3\}$, $\{3, 1\}$, $\{1, 2\}$, and so on. Initialize $\Lambda(v_0) := \Lambda(v_{6m+1}) := \{2\}$. With these lists, there is no proper list-coloring of P. The color for the source vertex is fixed to 2, forcing the color of v_1 to 3, which forces v_2 to 1, and generally forces v_i to color $(i+1) \bmod 3+1$. Hence v_{6m} is forced to $(6m+1) \bmod 3+1 = 2$, creating a conflict with the sink v_{6m+1} which is also forced to color 2.

We now introduce additional colors on some lists, and identify the distinguished vertices $\pi_1, \ldots, \pi_m$ among the vertices $v_{i'}$ (where $i' \in [6m]$), to allow proper list-colorings under the stated conditions. (Note that in the rest of the proof, we will make use of two symbols for any distinguished vertex, depending on which is more convenient at the time: π_i where $i \in [m]$ and $v_{i'}$ where $i' \in [6m]$.) For a group D_i of six consecutive vertices, the *interior* of the group consists of the middle four vertices. For each index $i \in [m]$, choose π_i as a vertex from the interior of group D_i such that c_i is not on the default list of colors for π_i. Since there is no color that appears on all of the default lists of the four interior vertices, this is always possible. Add c_i to the list of allowed colors for π_i. This completes the construction of the list-coloring instance (P, Λ). For an illustration see Figure 2.

It is easy to see that the construction can be performed in polynomial time. To conclude the proof, we argue that (P, Λ) has the desired properties. Observe that if $(d_1, \ldots, d_m) =$

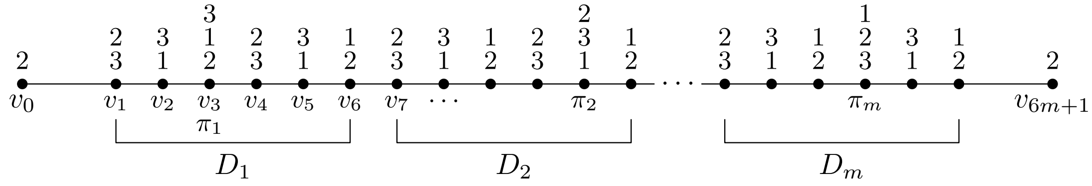

Figure 2: An example 3-List-Coloring instance created as in the proof of Lemma 15, where $(c_1, \ldots, c_m) = (3, 2, \ldots, 1)$.

$(c_1, \ldots, c_m)$, then a proper list-coloring γ of P in which $\gamma(\pi_i) \neq d_i = c_i$ for all $i \in [m]$ would in fact be a proper list-coloring of P under the default lists before augmentation, which is impossible as we argued earlier. It remains to argue that when $(d_1, \ldots, d_m)$ differs from $(c_1, \ldots, c_m)$ in at least one position, then P has a proper list-coloring γ with $\gamma(\pi_i) \neq d_i$ for all $i \in [m]$. To construct such a list-coloring, for each index $i \in [m]$ with $c_i \neq d_i$, assign vertex π_i the color c_i. Since the vertices π_i are interior vertices of their groups, the distinguished vertices are pairwise nonadjacent and this does not result in any conflicts. For distinguished vertices π_i with $c_i = d_i$, we will assign π_i a color from the default list of vertex π_i; since c_i is not on the default list this results in the desired color-avoidance. We therefore conclude by verifying that the remaining vertices can be assigned a proper color from their default list.

To do so, assign the source vertex its forced color and propagate the coloring as described above, until we reach the first distinguished vertex π_i with $c_i \neq d_i$ (where $i \in [m]$). Let $i' \in [6m]$ denote the index of π_i among all vertices of P, i.e. $\pi_i = v_{i'}$. In the current partial coloring, $v_{i'-1}$ received color $((i'-1)+1) \bmod 3 + 1 = i' \bmod 3 + 1$ which is a color on the default list of $v_{i'}$. Hence, we do not create a conflict between vertices $v_{i'-1}$ and $v_{i'}$ as we gave $v_{i'}$ the color c_i which was not on $v_{i'}$'s default list by construction. The other color on the default list of $v_{i'}$ is $(i'+1) \bmod 3 + 1$, which is also on the list of $v_{i'+1}$, as $\Lambda(v_{i'+1}) = \{(i'+1) \bmod 3 + 1, (i'+2) \bmod 3 + 1\}$. Hence, assigning $v_{i'+1}$ color $(i'+1) \bmod 3 + 1$ does not create a conflict between $v_{i'}$ and $v_{i'+1}$, again since we assigned $v_{i'}$ a color which was not on its default list.

- If i was the last index for which $c_i \neq d_i$, then, for all $i'' \in [(i'+2)..6m]$ we continue giving vertex $v_{i''}$ color $i'' \bmod 3 + 1$. This way the sink can be properly list-colored.
- If not, we give $v_{i'+2}$ color $i' \bmod 3 + 1$ $(= (i'+3) \bmod 3 + 1)$. Note that since all distinguished vertices are interior vertices of the groups, $v_{i'+2}$ cannot be a distinguished vertex and hence has not been previously assigned a color. We now propagate this coloring along the path as before until we reach the next distinguished vertex which has already been assigned a color.

We repeat the construction until all vertices are properly list-colored. □

Theorem 16. *For any $\varepsilon > 0$ and constant $q \geq 3$, q-Coloring on Linear Forest + kv graphs cannot be solved in time $\mathcal{O}^*((q-\varepsilon)^k)$, unless SETH fails.*

Proof. To prove the theorem, we will first show that if q-List-Coloring on Linear Forest+ kv graphs can be solved in time $\mathcal{O}^*((q-\varepsilon)^k)$ for $q \geq 3$ and some $\varepsilon > 0$, then s-SAT can be solved in time $\mathcal{O}^*((2-\delta)^n)$ for some $\delta > 0$ and any $s \in \mathcal{O}(1)$, contradicting SETH. By the same argument as in the proof of Theorem 12, we then extend the lower bound to q-Coloring.

Suppose we have an instance φ of s-SAT on variables $x_1, \ldots, x_n$. We construct a graph G and lists $\Lambda\colon V(G) \to [q]$, such that G is properly list-colorable if and only if φ is satisfiable. The first part of the reduction is inspired by the reduction of Lokshtanov et al. [13, Theorem 6.1], which we repeat here for completeness. We choose an integer constant p depending on q and ε and group the variables of φ into t groups $F_1, \ldots, F_t$ of size $\lfloor \log q^p \rfloor$ each. We call a truth assignment for the variables in F_i a *group assignment*. We say that a group assignment satisfies clause $C_j \in \varphi$ if C_j contains at least one literal which is set to True by the group assignment. For each group F_i, we add a set of p vertices $v_i^1, \dots, v_i^p$ to G, in the following denoted by $\mathcal{V}_i$ with $\Lambda(v_i^j) = [q]$ for all i and j. Each coloring of the vertices $\mathcal{V}_i$ will encode one group assignment of F_i. We fix some efficiently computable injection $f_i \colon \{0,1\}^{|F_i|} \to [q]^p$ that assigns to each group assignment for F_i a distinct p-tuple of colors. This is possible since there are $q^p \geq 2^{|F_i|}$ possible colorings of p vertices. For a variable $x_i \in \varphi$ we can identify the set of vertices whose colorings encode the assignment of the group containing x_i. Since each group has size $\lfloor \log q^p \rfloor$, the truth assignments of a variable $x_i \in \varphi$ are encoded by (some) colorings of the vertices in $\mathcal{V}_{i'}$, where $i' = \lceil i / \lfloor \log q^p \rfloor \rceil$.

We now construct the main part of the graph G. Let $C_j \in \varphi$ be a clause on variables $x_{j_1}, \dots, x_{j_{s'}}$, where $s' \in [s]$. The truth assignments of these variables are encoded by the colorings of the vertices in $\mathcal{V}_{C_j} = \bigcup_{i \in [s']} \mathcal{V}_{\lceil j_i / \lfloor \log q^p \rfloor \rceil}$. We say that a coloring $\mu \colon \mathcal{V}_{C_j} \to [q]$ is a *bad* coloring for C_j if there is a group for which the coloring does not represent a group assignment, or if the group assignments encoded by μ do not satisfy clause C_j.

For each bad coloring μ we construct a path using Lemma 15 which ensures that G is not properly list-colorable if μ appears on $\mathcal{V}_{C_j}$. Let $j'_i = \lceil j_i / \lfloor \log q^p \rfloor \rceil$ and consider the following vector of colors induced by μ:

$$c_\mu = \left(\mu\left(v^1_{j'_1}\right), \dots, \mu\left(v^p_{j'_1}\right), \dots, \mu\left(v^1_{j'_{s'}}\right), \dots, \mu\left(v^p_{j'_{s'}}\right) \right) \tag{1}$$

We add to G a path P_{c_μ} constructed according to Lemma 15 with c_μ as the input vector of colors. Let $(\pi_1, \dots, \pi_{p \cdot s'})$ denote the distinguished vertices of P_{c_μ}. We make each variable vertex $v^\ell_{j'_i} \in \mathcal{V}_{C_j}$ (where $i \in [s']$ and $\ell \in [p]$) adjacent to the distinguished vertex $\pi_{p \cdot (i-1) + \ell}$ in P_{c_μ}, intending to ensure that if all vertices in $\mathcal{V}_{C_j}$ are colored according to μ, then this partial list-coloring on G cannot be extended to P_{c_μ}. Adding such a path for each clause in φ and each bad coloring finishes the construction of (G, Λ).

We first count the number of vertices in G and then prove the correctness of the reduction. There are $\mathcal{O}(n)$ variable vertices and for each of the m clauses, there are at most $q^{p \cdot s}$ bad colorings, each of which adds a path on at most $\mathcal{O}(p \cdot s)$ vertices to G, by Lemma 15. Hence, the number of vertices in G is at most

$$\mathcal{O}\left(n + m \cdot q^{p \cdot s}(p \cdot s)\right) = \mathcal{O}(n + m) = n^{\mathcal{O}(1)}, \tag{2}$$

as $p, q, s \in \mathcal{O}(1)$ and $m = \mathcal{O}(n^s)$.

Claim 17. *(G, Λ) is properly q-list-colorable if and only if φ has a satisfying assignment.*

Proof. Suppose φ has a satisfying assignment ψ. For each group $\mathcal{V}_i$ the assignment φ dictates a group assignment, which corresponds to a coloring on $\mathcal{V}$ by the chosen injection f_i. Let $\gamma_\psi \colon \bigcup_i \mathcal{V}_i \to [q]$ denote the coloring of the variable vertices that encodes ψ. We argue that γ_ψ can be extended to the rest of G, respecting the lists Λ. For every $C_j \in \varphi$ on variables $x_{j_1}, \dots, x_{j_{s'}}$ and every bad coloring $\mu \colon \bigcup_{i=1}^{s'} \mathcal{V}_{j'_i} \to [q]$ w.r.t. C_j (where $j'_i = \lceil j_i / \lfloor \log q^p \rfloor \rceil$), we added a path P_{c_μ} to G, constructed according to Lemma 15, whose distinguished vertices we denote by $(\pi_1, \dots, \pi_{p \cdot s'})$. Note that c_μ denotes the vector representation of the coloring μ as in (1). Let c_γ denote the vector representation of γ restricted to the variable vertices $\cup_{i=1}^{s'} \mathcal{V}_{j'_i}$, appearing in the same order as in c_μ. Since γ_ψ encodes a satisfying assignment of φ, $c_\mu \neq c_\gamma$. Hence, by Lemma 15, we can extend γ_ψ to P_{c_μ} without creating a conflict; it asserts that there is a proper list-coloring γ' on P_{c_μ} such that $\gamma(v^\ell_{j'_i}) = c_\gamma(p \cdot (i-1) + \ell) \neq \gamma'(\pi_{p \cdot (i-1) + \ell})$ for all $i \in [s']$ and $\ell \in [p]$. Hence, every pair of adjacent vertices between the vertices of P_{c_μ} and the vertices encoding the truth assignments of the variables in C_j can be list-colored properly and we can conclude that γ_ψ can be extended to P_{c_μ} and subsequently, to all of G.

Now suppose (G, Λ) has a proper list-coloring γ and assume for the sake of a contradiction that φ does not have a satisfying assignment. Then, the restriction of any list-coloring of G to (some of) the variable vertices $\bigcup_i \mathcal{V}_i$ must be a bad coloring for some clause in φ. Let C_j denote such a clause for γ and let c_γ denote the corresponding vector of colors, restricted to the variable vertex groups that encode the truth assignments to the variables in C_j. We added a path P_{c_γ} to G which by Lemma 15 cannot be properly list-colored such that each distinguished vertex gets a color which is different from the color of the variable vertex it is adjacent to. Hence, one of the distinguished vertices of P_{c_γ} creates a conflict and we have a contradiction. ⌟

Since G consists of the variable vertices attached to a set of disjoint paths, we observe the following.

Observation 18. *$\bigcup_i \mathcal{V}_i$ is a modulator to* Linear Forest.

The previous observation can easily be verified, since G consists of the variable vertices attached to a set of disjoint paths. By Claim 17 and Observation 18 we can now finish the proof in the same way as the proof of [13, Theorem 6.1], in particular Lemma 6.4 yields the claim.

Claim 19 (Cf. Lemma 6.4 in [13])**.** *If q-*List-Coloring *on* Linear Forest $+ kv$ *graphs can be solved in time $\mathcal{O}^*((q-\varepsilon)^k)$ for some $\varepsilon < 1$, then s-*SAT *can be solved in $\mathcal{O}^*((2-\delta)^n)$ time, for some $\delta < 1$ and any $s \in \mathcal{O}(1)$.*

Proof. Let $\lambda := \log_q (q-\varepsilon)^k < 1$, such that $(q-\varepsilon)^k = q^{\lambda k}$. Note that by (2), the size of G is polynomial in n, the number of variables of φ. We choose a sufficiently large p such that $\delta' = \lambda \frac{p}{p-1} < 1$. Given an instance φ of s-SAT, we use the above reduction to obtain (G, Λ), an instance of q-List-Coloring. Correctness follows from Claim 17. By Observation 18 we know that G has a modulator to Linear Forest of size $p\lceil \frac{n}{\lfloor p \log q \rfloor} \rceil$. By the choice of p we have $\lambda p \lceil \frac{n}{\lfloor p \log q \rfloor} \rceil \leq \lambda p \frac{n}{(p-1)\log q} + \lambda p \leq \delta' \frac{n}{\log q} + \lambda p$. Hence, s-SAT can be solved in $\mathcal{O}^*(2^{\delta' n + \lambda p}) = \mathcal{O}^*(2^{\delta' n}) = \mathcal{O}^*((2-\delta)^n)$ time for some $\delta > 0$ which does not depend on s. ⌟

We have given a reduction from s-SAT to q-List-Coloring on Linear Forest $+ kv$ graphs. As in the proof of Theorem 12, we can make the reduction work for q-Coloring as well by adding a clique K_q of q vertices to the graph, each of which represents one color and then making each vertex in G adjacent to each vertex in K_q which represents a color that is not on its list. Since this increases the size of the modulator by q, which is a constant, this does not affect asymptotic runtime bounds and completes the proof of Theorem 16. □

Note that we can modify the reduction in the proof of Theorem 16 to give a lower bound for Path $+ kv$ graphs as well: We simply connect all paths that we added to the graph to one long path, adding a vertex with a full list between each pair of adjacent paths.

Corollary 20. *For any $\varepsilon > 0$ and constant $q \geq 3$, q-*Coloring *on* Path$+kv$ *graphs cannot be solved in time $\mathcal{O}^*((q-\varepsilon)^k)$, unless* SETH *fails.*

5 A Tighter Treedepth Boundary

In Lemma 11 we showed that if the $(q+1)$-colorable members of a hereditary graph class $\mathcal{F}$ have bounded treedepth, then $\mathcal{F}$ has constant-size No-certificates for q-List-Coloring and hence $\mathcal{F} + kv$ has nontrivial algorithms for q-(List-)Coloring parameterized by the size of a given modulator to $\mathcal{F}$. One might wonder whether a graph class $\mathcal{F} + kv$ has nontrivial algorithms for q-Coloring parameterized by a given modulator to $\mathcal{F}$ *if and only if* all $(q+1)$-colorable members in $\mathcal{F}$ have bounded treedepth. However, this is not the case. In [11, Lemma 4] the authors showed that q-Coloring parameterized by the size of a modulator to the class Cograph has nontrivial algorithms. Clearly, complete bipartite graphs are cographs and it is easy to see that (the 2-colorable balanced biclique) $K_{n,n}$ has treedepth $n+1$. In this section we show that, unless SETH fails, bicliques are in some sense the only obstruction to this treedepth boundary.

We use a combinatorial theorem which in combination with Corollary 20 will yield the result.

Theorem 21 (Corollary 3.6 in [12], Theorem 1 in [1])**.** *For any $s, k \in \mathbb{N}$ there is a $P(s,k) \in \mathbb{N}$ such that any graph with a path of length $P(s,k)$ either contains an induced path of length s, or a K_k subgraph, or an induced $K_{k,k}$ subgraph.*

Theorem 22. *Let $\mathcal{F}$ be a hereditary class of graphs for which there exists a $t \in \mathbb{N}$ such that $K_{t,t}$ is not contained in $\mathcal{F}$, let $q \geq 3$, and suppose* SETH *is true. Then, q-*Coloring *parameterized by a given vertex modulator to $\mathcal{F}$ of size k has $\mathcal{O}^*((q-\varepsilon)^k)$ time algorithms for some $\varepsilon > 0$, if and only if all $(q+1)$-colorable graphs in $\mathcal{F}$ have bounded treedepth.*

Proof. Assume the stated conditions hold for $\mathcal{F}$ and t. In one direction, if all the $(q+1)$-colorable graphs in $\mathcal{F}$ have their treedepth bounded by a constant, then there are constant-size No-certificates for q-List-Coloring on $\mathcal{F}$ by Lemma 11, implying the existence of nontrivial algorithms by Theorem 6.

For the other direction, suppose that there is no finite bound on the treedepth of $(q+1)$-colorable graphs in $\mathcal{F}$. We claim that $\mathcal{F}$ contains all paths, which will prove this direction using Corollary 20. If the longest (simple) path in a graph G has length k, then G has treedepth at most k since any depth-first search tree forms a valid treedepth decomposition, and has depth at most k since all its root-to-leaf paths are paths in G. Hence a graph of treedepth more than n contains a path of length more than n. Since the $(q+1)$-colorable graphs in $\mathcal{F}$ have arbitrarily large treedepth, the preceding argument shows that for any n there is a $(q+1)$-colorable graph in $\mathcal{F}$ containing a path of length more than n. In particular, for any n there is a $(q+1)$-colorable graph G_n in $\mathcal{F}$ containing a (not necessarily induced) path of length $P(n, \max(t, q+2))$, the Ramsey number of Theorem 21. Hence graph G_n contains an induced path of length n, a clique of size $\max(t, q+2)$, or an induced biclique with sets of size $\max(t, q+2)$. Since a $(q+2)$-clique is not $(q+1)$-colorable, G_n contains no such clique. If G_n contains an induced biclique subgraph with sets of size $\max(t, q+2)$, then since $\mathcal{F}$ is hereditary it would contain $K_{t,t}$, which contradicts our assumption on $\mathcal{F}$. Hence G_n contains an induced path of length n, implying that $\mathcal{F}$ contains the induced path of length n since it is hereditary. As this holds for all n, class $\mathcal{F}$ contains all paths, implying by Corollary 20 and SETH that there are no nontrivial algorithms for q-List-Coloring parameterized by the size of a given vertex modulator to $\mathcal{F}$. □

6 Conclusion

In this paper we have presented a fine-grained parameterized complexity analysis of the q-Coloring and q-List-Coloring problems. We showed that if a graph class $\mathcal{F}$ has No-certificates for q-List-Coloring of bounded size or if the $(q+1)$-colorable members of $\mathcal{F}$ (where $\mathcal{F}$ is hereditary) have bounded treedepth, then there is an algorithm that solves q-Coloring on graphs in $\mathcal{F} + kv$ (graphs with vertex modulators of size k to $\mathcal{F}$) in time $\mathcal{O}^*((q-\varepsilon)^k)$ for some $\varepsilon > 0$ (depending on $\mathcal{F}$). The parameter treedepth revealed itself as a boundary in some sense: We showed that Path $+ kv$ graphs do not have $\mathcal{O}^*((q-\varepsilon)^k)$ time algorithms for any $\varepsilon > 0$ unless SETH is false — and paths are arguably the simplest graphs of unbounded treedepth. Furthermore we proved that if a graph class $\mathcal{F}$ does not have large bicliques, then $\mathcal{F} + kv$ graphs have $\mathcal{O}^*((q-\varepsilon)^k)$ time algorithms, for some $\varepsilon > 0$, if and only if $\mathcal{F}$ has bounded treedepth.

Treedepth is an interesting graph parameter which in many cases also allows for polynomial space algorithms where e.g. for treewidth this is typically exponential. It would be interesting to see how the problems studied by Lokshtanov et al. [13] behave when parameterized by treedepth. Naturally, a fine-grained parameterized complexity analysis as we did might be interesting for other problems as well.

Open Problem. Consider a different problem than q-Coloring, for example another problem studied in [13]. For which parameters in the hierarchy can we improve upon the base of the exponent of the SETH-based lower bound? Does the parameter treedepth establish a diving line in this sense as well?

References

[1] Aistis Atminas, Vadim V. Lozin, and Igor Razgon. Linear time algorithm for computing a small biclique in graphs without long induced paths. In *SWAT*, pages 142–152. Springer, 2012.

[2] Andreas Björklund, Thore Husfeldt, and Mikko Koivisto. Set partitioning via inclusion-exclusion. *SIAM Journal on Computing*, 39(2):546–563, 2009. Based on two extended abstracts appearing in FOCS '06.

[3] Leizhen Cai. Parameterized complexity of vertex colouring. *Discrete Applied Mathematics*, 127(3):415–429, 2003.

[4] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 1st edition, 2015.

[5] Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, 4th edition, 2010. Corrected reprint 2012.

[6] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

[7] Michael R. Fellows, Bart M. P. Jansen, and Frances Rosamond. Towards fully multivariate algorithmics: Parameter ecology and the deconstruction of computational complexity. *European Journal of Combinatorics*, 34(3):541–566, 2013. Previously app. in IWOCA '09.

[8] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*, volume XIV of *Texts in Theoretical Computer Science*. Springer, 2006.

[9] Russel Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

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

[11] Bart M. P. Jansen and Stefan Kratsch. Data reduction for graph coloring problems. *Information & Computation*, 231:70–88, 2013. Previously app. in FCT '11.

[12] Bart M. P. Jansen and Dániel Marx. Characterizing the easy-to-find subgraphs from the viewpoint of polynomial-time algorithms, kernels and turing kernels. *ArXiv e-prints*, 2015. arxiv:1410.0855, also app. in SODA '15 pp. 616-629.

[13] Daniel Lokshstanov, Dániel Marx, and Saket Saurabh. Known algorithms for graphs of bounded treewidth are probably optimal. In *SODA*, pages 777–789. SIAM, 2011.

[14] Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity. Graphs, Structures and Algorithms*, volume 28 of *Algorithms and Combinatorics*. Springer, 2012.

[15] Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[16] Virginia Vassilevska Williams. Hardness of easy problems: Basing hardness on popular conjectures such as the strong exponential time hypothesis. In *IPEC*, volume 43 of *LIPIcs*, pages 16–28, 2015.