Complexity of C_k-coloring in hereditary classes of graphs*

Maria Chudnovsky† Shenwei Huang‡ Paweł Rzążewski§ Sophie Spirkl¶
Mingxian Zhong‖

May 6, 2020

Abstract

For a graph F, a graph G is F*-free* if it does not contain an induced subgraph isomorphic to F. For two graphs G and H, an H*-coloring* of G is a mapping $f : V(G) \to V(H)$ such that for every edge $uv \in E(G)$ it holds that $f(u)f(v) \in E(H)$. We are interested in the complexity of the problem H-Coloring, which asks for the existence of an H-coloring of an input graph G. In particular, we consider H-Coloring of F-free graphs, where F is a fixed graph and H is an odd cycle of length at least 5. This problem is closely related to the well known open problem of determining the complexity of 3-Coloring of P_t-free graphs.

We show that for every odd $k \geq 5$ the C_k-Coloring problem, even in the list variant, can be solved in polynomial time in P_9-free graphs. The algorithm extends for the case of list version of C_k-Coloring, where k is an even number of length at least 10.

On the other hand, we prove that if some component of F is not a subgraph of a subdividecd claw, then the following problems are NP-complete in F-free graphs:

a) extension version of C_k-Coloring for every odd $k \geq 5$,

b) list version of C_k-Coloring for every even $k \geq 6$.

---

*The extended abstract of the paper was presented on ESA 2019.

†Princeton University, Princeton, NJ 08544, USA. Supported by NSF grant DMS-1763817. This material is based upon work supported in part by the U. S. Army Research Laboratory and the U. S. Army Research Office under grant number W911NF-16-1-0404.

‡College of Computer Science, Nankai University, Tianjin 300350, China. This research is partially supported by the National Natural Science Foundation of China (11801284).

§Faculty of Mathematics and Information Science, Warsaw University of Technology, Warsaw, Poland. Supported by Polish National Science Centre grant no. 2018/31/D/ST6/00062.

¶Rutgers University, Piscataway, NJ 08854, USA. This material is based upon work supported by the National Science Foundation under Award No. DMS1802201.

‖Lehman College, CUNY, Bronx, NY 10468, USA

1 Introduction

For graphs G and H, a *homomorphism from G to H* is a mapping $f : V(G) \to V(H)$ such that $f(u)f(v) \in E(H)$ for every edge $uv \in E(G)$. It is straightforward to see that if H is a complete graph with k vertices, then every homomorphism to H is in fact a k-coloring of G (and vice versa). This shows that homomorphisms can be seen as a generalization of graph colorings. Because of that, a homomorphism to H is often called an *H-coloring*, and vertices of H are called *colors*. We also say that G is *H-colorable* if G has an H-coloring.

In what follows, the target graph H is always fixed. We are interested in the complexity of the following computational problem, called H-Coloring.

Problem: H-Coloring
Instance: A graph G.
Question: Does there exist a homomorphism from G to H?

Complexity of variants of H-Coloring

Since H-Coloring is a generalization of k-Coloring, it is natural to try to extend results for k-Coloring to target graphs H which are not complete graphs. For example, it is well-known that k-Coloring enjoys a complexity dichotomy: it is polynomial-time solvable if $k \leq 2$, and NP-complete otherwise. The complexity dichotomy for H-Coloring was described by Hell and Nešetřil in their seminal paper [27]: they proved that the problem is polynomial-time solvable if H is bipartite, and NP-complete otherwise.

Since then, there have been numerous studies on variants of H-Coloring. Let us mention two of them. In the H-Precoloring Extension problem, we are given a graph G, a subset $W \subseteq V(G)$ and a mapping $h : W \to V(H)$. The problem is to decide if h can be extended to an H-coloring of G, that is, if there is an H-coloring f of G such that $f|W = h$. In the List H-Coloring problem, the input consists of a graph G with an *H-list assignment*, which is a function $L : V(G) \to 2^{V(H)}$ that assigns a subset of $V(H)$ to each vertex of G. We ask if there is an *L-coloring*, that is, and H-coloring f of G such that $f(v) \in L(v)$ for each $v \in V(G)$. In such a case we say that (G, L) is *H-colorable*. We formulate these two problems as follows.

Problem: H-Precoloring Extension
Instance: A graph G, a subset $W \subseteq V(G)$, and a mapping $h \colon W \to V(H)$.
Question: Can h be extended to an H-coloring of G?

Problem: List H-Coloring
Instance: A pair (G, L) where G is a graph and L is an H-list assignment.
Question: Does there exist an H-coloring of (G, L)?

When $H = K_k$, we write k-Precoloring Extension and List k-Coloring for K_k-Precoloring Extension and List K_k-Coloring, respectively. Clearly H-Precoloring Extension can be seen as a restriction of List H-Coloring, in which every list is either a singleton, or contains all vertices of H. This is the reason why it is sometimes called *one-or-all list homomorphism (coloring) problem* [15].

Note that List H-Coloring is at least as hard as H-Precoloring Extension, which in turn is at least as hard as H-Coloring. Thus, any algorithmic result for List H-Coloring carries over to the other two problems, and any hardness result for H-Coloring carries over to H-Precoloring Extension and List H-Coloring.

The complexity dichotomy for List H-Coloring was proven in three steps: first, for reflexive graphs H [14], then for irreflexive graphs H [15], and finally for all graphs H [16]. In general, variants of H-Coloring can be seen in a wider context of Constraint Satisfaction Problems (CSP). A full complexity dichotomy for this family of problems has been a long-standing open question, known as the *CSP dichotomy conjecture* of Feder and Vardi [19]. After a long series of partial results, the problem was finally solved very recently, independently by Bulatov [6] and by Zhuk [37].

A natural approach in dealing with computationally hard problems is to consider restricted instances, in hope to understand the boundary between easy and hard cases. For example, it is known that H-Coloring can be solved in polynomial time for perfect graphs, because it suffices to test whether $\omega(G) > \omega(H)$, which can be done in $O(|V(G)|^{|V(H)|})$ time. If $\omega(G) > \omega(H)$, then the answer is no, as there is no way to map the largest clique of G to H. Otherwise the answer is yes, since $\omega(G)$-coloring of G can be translated to a homomorphism of G to the largest clique of H, and thus to H. The situation changes when we consider the more general setting of H-Precoloring Extension and List H-Coloring. For any fixed graph H, List H-Coloring (and thus H-Precoloring Extension and H-Coloring) can be solved in polynomial time for input graphs with bounded tree-width. Combining this with an observation that any graph with a clique larger than $\omega(H)$ has no H-coloring, we obtain polynomial-time algorithms for chordal graphs [18]. For permutations graphs, List H-Coloring can also be solved in polynomial time via a recursive branching algorithm [13]. For bipartite input graphs, however, 3-Precoloring Extension (i.e., K_3-Precoloring Extension) is already NP-complete [33]. Other restricted inputs have been studied too, e.g. bounded-degree graphs [20, 17]. For more results on graph homomorphisms, we refer to the monograph by Hell and Nešetřil [26].

In this paper, we study the complexity of H-Coloring for minimal non-bipartite graphs H, namely when H is an odd cycle. We also extend these results to the List H-Coloring problem, also for even target cycles.

Although H-Coloring is well-studied for hereditary classes defined by infinitely many forbidden induced subgraph, not much is known for hereditary classes defined finitely many forbidden induced subgraphs. Here we initiate such a study for hereditary classes defined by a single forbidden induced subgraph.

Graphs with forbidden induced subgraphs

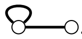
A rich family of restricted graph classes comes from forbidding some small substructures. For graphs G and F, we say that G *contains* F if F is an induced subgraph of G. By *F-free graphs* we mean the class of graphs that do not contain F. Note that this class is *hereditary*, that is, it is closed under taking induced subgraphs.

The complexity of k-Coloring for hereditary graph classes has received much attention in the past two decades and significant progress has been made. Of particular interest is the class of F-free graphs for a fixed graph F. For any fixed $k \geq 3$, the k-Coloring problem remains NP-complete for F-free graphs whenever F is not a linear forest (a collection of disjoint paths) [29, 35]. The simplest linear forests are paths, and the complexity of k-Coloring in P_t-free graphs has been studied by many researchers.

On the positive side, Hoàng, Kamiński, Lozin, Sawada, and Shu [28] gave a recursive algorithm showing that k-Coloring can be solved in polynomial time for P_5-free graphs for any fixed k. Bonomo, Chudnovsky, Maceli, Schaudt, Stein, and Zhong [5] showed that 3-Coloring can be solved in polynomial time in P_7-free graphs. Moreover, very recently, Chudnovsky, Spirkl, and Zhong proved that 4-Coloring is polynomial-time solvable in P_6-free graphs [8, 9, 10].

On the negative side, Woeginger and Sgall [36] demonstrated the NP-completeness of 5-Coloring for P_8-free graphs and 4-Coloring for P_{12}-free graphs. Later on, these NP-completeness results were improved by various researchers and the strongest result is due to Huang [30] who proved that 4-Coloring is NP-complete for P_7-free graphs and 5-Coloring is NP-complete for P_6-free graphs. These results settle the complexity of k-Coloring for P_t-free graphs for all pairs (k, t), except for the complexity of 3-Coloring for P_t-free graphs when $t \geq 8$. Interestingly, all polynomial-time results carry over to the list variant, except for the case of List 4-Coloring of P_6-free graphs, which was shown to be NP-complete by Golovach, Paulusma, and Song [22]. We refer the reader to the survey by Golovach, Johnson, Paulusma, and Song [21] for more information about coloring graphs with forbidden subgraphs.

Understanding the complexity of 3-Coloring in P_t-free graphs seems a hard problem – on the one hand, algorithms even for small values of t are difficult to construct, and on the other hand all our hardness reductions appear to introduce long induced paths. Let us mention a problem whose complexity is equally elusive: Independent Set. Alekseev [1] observed that Independent Set is NP-complete in F-free graphs whenever F is not a path or a subdivided claw. For P_t-free graphs, polynomial-time algorithms are known only for small values of t: currently, the best result is the recent polynomial-time algorithm for P_6-free graphs by Grzesik, Klimošova, Pilipczuk, and Pilipczuk [24, 25]. On the other hand, the problem is not known to be NP-hard for any fixed t.

A natural question to ask is if the similar behavior of 3-Coloring and Independent Set in P_t-free graphs is a part of a more general phenomenon. Recently, Groenland, Okrasa, Rzążewski, Scott, Seymour, and Spirkl [23] shed some light on this question by showing that if H does not contain two vertices with two common neighbors, then a very general, weighted variant of H-Coloring can be solved in time $2^{O(\sqrt{tn \log n})}$ for P_t-free graphs. Clearly K_3 does not have two vertices with two common neighbors. Moreover, Independent Set can be expressed as a weighted homomorphism to , which has the same property, and thus, for every t, both 3-Coloring and Independent Set can be solved in subexponential time in P_t-free graphs (we note that a subexponential algorithm for Independent Set in P_t-free graphs was known before [3]). This implies that if one attempts to prove NP-completeness of any of these problems in P_t-free graphs, then, assuming the Exponential Time Hypothesis [31, 32], such a reduction should be sufficiently complicated to introduce at least a quadratic blow-up of the instance.

In this paper, we study the complexity of variants of H-Coloring when H is an odd cycle of length at least five. Note that by the result of Groenland *et al.* [23], this problem can be solved in subexponential time in P_t-free graphs. We are interested in better classification of polynomial and NP-hard cases.

Our contribution

First, we show that the List C_k-Coloring can be solved in polynomial time in P_9-free graphs for every $k \geq 9$ or $k = 5$ or 7.

Theorem 1. *Let $k \geq 9$ or $k = 5$ or 7, and let (G, L) be an instance of* List C_k-Coloring *where G is a P_9-free graph of order n. Then one can determine in $O(n^{12k+3})$ time if (G, L) is C_k-colorable, and find a C_k-coloring of (G, L) if one exists.*

The algorithm is described in detail in Section 3. It builds on the recent work on 3-Coloring P_7-free graphs [5]. The high-level idea of the algorithm is the following: First, we partition the graph into a so-called *layer structure* and guess the colors of a constant number of vertices. This precoloring propagates to other vertices, reducing their lists. We keep guessing the colors of other vertices, transforming the input instance (G, L) into a set of $n^{O(k)}$ subinstances, such that both the following conditions are satisfied:

(i) (G, L) admits a C_k-coloring if and only if one of these subinstances admits a C_k-coloring;

(ii) each subinstance can be solved in polynomial time by a reduction to 2-Sat.

In Section 4, we show that the above algorithm can be improved such that it remains polynomial when k is an input (i.e., the problem is *fixed-parameter tractable* with respect to the parameter k). In particular, we prove the following theorem.

Theorem 2. *Let $k \geq 9$ or $k = 5$ or 7, and let (G, L) be an instance of* List C_k-Coloring *where G is a P_9-free graph of order n. Then one can determine in $O(n^{111})$ time if (G, L) is colorable, and find such a C_k-coloring if one exists.*

In Section 5, we study the complexity of variants of H-COLORING in F-free graphs and prove the following theorem.

Theorem 3. *Let F be a connected graph. If F is not a subgraph of a subdivided claw, then for every odd $k \geq 5$ the C_k-PRECOLORING EXTENSION problem is NP-complete for F-free graphs.*

We prove the theorem in several steps, analyzing the possible structure of F and trimming the hard cases. In most cases, we actually prove hardness for the more restricted C_k-COLORING problem.

Then we turn our attention to LIST C_k-COLORING in F-free graphs, when k is even. It is known that for general graphs this problem is polynomial time solvable for $k = 4$, and NP-complete for every $k \geq 6$ [15]. We prove the following analogue of Theorem 3.

Theorem 4. *Let F be a connected graph. If F is not a subgraph of a subdivided claw, then for every even $k \geq 6$ the* LIST C_k-COLORING *problem is NP-complete for F-free graphs.*

Observe that the statements of Theorem 3 and Theorem 4 are similar to the previously mentioned result of Alekseev for INDEPENDENT SET [1].

Finally, in Section 6, we state some open questions for future research.

2 Preliminaries

Let G be a simple graph. For $X \subseteq V(G)$, we denote by $G|X$ the subgraph induced by X, and denote by $G \setminus X$ the graph $G|(V(G) \setminus X)$. We say that X is *connected* if $G|X$ is connected. For two disjoint subsets $A, B \subset V(G)$, we say that A is *complete* to B if every vertex of A is adjacent to every vertex of B, and that A is *anticomplete* to B if every vertex of A is nonadjacent to every vertex of B. If $A = \{a\}$ we write a is complete (or anticomplete) to B to mean that $\{a\}$ is complete (or anticomplete) to B. For $X \subseteq V(G)$, we say that $e \in E(G)$ is *an edge of* X if both endpoints of e are in X. For $v \in V(G)$ we write $N_G(v)$ (or $N(v)$ when there is no danger of confusion) to mean the set of vertices of G that are adjacent to v. Observe that since G is simple, $v \notin N(v)$. For $X \subseteq V(G)$ we define $N(X) = (\bigcup_{v \in X} N(v)) \setminus X$. We say that the set S *dominates* X, or S is a *dominating set* of X if $X \subseteq S \cup N(S)$. We write that S dominates G when we mean that it dominates $V(G)$. A component of G is *trivial* if it has only one vertex and *nontrivial* otherwise.

We use $[k]$ to denote the set $\{1, 2, \ldots, k\}$. We denote by P_t the path with t vertices. A *path in a graph* G is a sequence $v_1 - \cdots - v_t$ of pairwise distinct vertices such that for any $i, j \in [t]$, $v_i v_j \in E(G)$ if and only if $|i - j| = 1$. The *length* of this path is t. We denote by $V(P)$ the set $\{v_1, \ldots, v_t\}$. If $a, b \in V(P)$, say $a = v_i$ and $b = v_j$ with $i < j$, then $a - P - b$ is the path $v_i - v_{i+1} - \cdots - v_j$, and $b - P - a$ is the path $v_j - v_{j-1} - \cdots - v_i$.

Let $k \geq 3$ be an odd integer. We use $[k]$ to denote the set $\{1, 2, \ldots, k\}$, and we denote by C_k a cycle with k vertices $1, 2, \ldots, k$ that appear along the cycle in this order. The calculations on vertices of C_k will be preformed modulo k, with 0 unified with vertex k.

We say that (G, L') is a *subinstance* of (G, L) if $L'(v) \subseteq L(v)$ for every $v \in V(G)$. Two C_k-list assignments L and L' of G are *equivalent* if (G, L) is C_k-colorable if and only if (G, L') is C_k-colorable. A C_k-list assignment L is *equivalent* to a set $\mathcal{L}$ of C_k-list assignments of a graph G if there is $L' \in \mathcal{L}$ such that (G, L) is equivalent to (G, L').

Let (G, L) be an instance of LIST C_k-COLORING. We say that the list $L(x)$ of a vertex x is *good* if $|L(x)| \in \{1, 2, 3, k\}$ and in addition

- if $|L(x)| = 2$, then $L(x) = \{i - 1, i + 1\}$ for some $i \in [k]$, and
- if $|L(x)| = 3$, then $L(x) = \{i, i - 2, i + 2\}$ for some $i \in [k]$.

We say that L is *good* if $L(v)$ is good for all $v \in V(G)$.

For an edge $vw \in E(G)$, we *update v from w* if one of the following is performed.

- If $L(w) = \{i\}$ for some $i \in [k]$, then replace the list of v by $\{i-1, i+1\} \cap L(v)$.
- If $L(w) = \{i-1, i+1\}$ for some $i \in [k]$, then replace the list of v by $\{i, i+2, i-2\} \cap L(v)$.
- If $L(w) = \{i, i-2, i+2\}$, $L(v) = \{j, j+2, j-2\}$ for some $i, j \in [k]$, then replace the list of v by $\{i-1, i+1, i-3, i+3\} \cap L(v)$.

Clearly, any update creates an equivalent subinstance of (G, L). Note that in the graph homomorphism literature this operation is usually referred to as *edge (or arc) consistency* and it is performed in the beginning of most algorithms solving variants of H-Coloring [26, 34]. However, we keep the name "update" to emphasize that we will only perform it at certain points in our algorithm. We say that an update of v from w is *effective* if the size of the list of v decreases by at least 1, and *ineffective* otherwise. Note that an update is effective if and only if there exists an element $c \in L(v)$ which is not an element of $\{i-1, i+1\}$, $\{i, i+2, i-2\}$ or $\{i-1, i+1, i-3, i+3\}$ depending on the case in the definition of an update. We observe that the update does not change the goodness of the list.

Lemma 5. *If the lists of v and w are good before updating v from w, then the list of v is good or empty after the update.*

Proof. Let $L(w)$ and $L(v)$ be the list of w and v before the update, and $L'(v)$ be the list of v after the update. We may assume that $L'(v) \neq \emptyset$ for otherwise we are done. Similarly, we may assume that the update is effective, for otherwise $L'(v) = L(v)$ is good.

If $L(w) = \{i\}$ for some $i \in [k]$, then either $L'(v) = \{i-1, i+1\}$ or $|L'(v)| = 1$. In both cases, $L'(v)$ is good.

If $L(w) = \{i-1, i+1\}$ for some $i \in [k]$, then $L'(v)$ is good unless $L'(v) = \{i-2, i+2\}$. This implies that $i-2, i+2 \in L(v)$ and $i \notin L(v)$. If $|L(v)| = 2$, then $L(v) = \{i-2, i+2\}$, which contradicts that $L(v)$ is good. So $|L(v)| \geq 3$. Since $|L(v)| < k$, it follows that $|L(v)| = 3$. But since $i \notin L(v)$ and $i-2, i+2 \in L(v)$, it follows that $L(v)$ is not good, a contradiction.

If $L(w) = \{i, i-2, i+2\}$, $L(v) = \{j, j+2, j-2\}$ for some $i, j \in [k]$ and that $|i-j| \notin \{0, 1, k-1\}$, then $L'(v)$ is good unless $L'(v) = 2$ and either $L'(v) = \{i-1, i+3\}$ (by symmetry $L'(v) = \{i+1, i-3\}$) or $L'(v) = \{i-3, i+3\}$. Since $L'(v) \subseteq L(v) = \{j, j-2, j+2\}$, it follows that $\{i-1, i+3\} = \{j-2, j+2\}$ or $\{i-3, i+3\} = \{j-2, j+2\}$. If $\{i-1, i+3\} = \{j-2, j+2\}$, then $i-1 \equiv j+2 \mod k$ and $i+3 \equiv j-2 \mod k$. This implies that $i-3 \equiv j \mod k$. So $L'(v) = L(v)$, a contradiction. If $\{i-3, i+3\} = \{j-2, j+2\}$, then $i-3 \equiv j+2 \mod k$ and $i+3 \equiv j-2 \mod k$. This implies that $k = 10$, which contradicts the assumption that k is an odd integer. $\square$

A C_k-list assignment L is said to be *reduced* if no effective update can be performed. It is well-known that one can obtain a reduced list assignment in polynomial time. We include a proof for the sake of completeness.

Lemma 6. *Let G be a graph of order n, and L be a C_k-list assignment. There exists an $O(n^3)$-time algorithm to obtain an equivalent reduced subinstance (G, L') of (G, L) or determine that (G, L) has no C_k-coloring.*

Proof. We obtain L' by performing updates exhaustively. For each edge $vw \in E(G)$, we check if the update of v from w is effective. If so, we perform the update. If the list of v becomes empty after the update, then we stop and claim that (G, L) is not C_k-colorable. Otherwise we repeat until no effective update can be found.

Since any single update results in an equivalent subinstance, we obtain a reduced subinstance that is equivalent to (G, L) at the end. It takes $O(n^2)$ time to find an effective update and $O(1)$

time to perform such a update. Moreover, since each effective update decreases the list of a vertex by at least 1 and $\sum_{v \in V(G)} |L(v)| \leq kn$, effective updates can be performed at most kn times. Thus, the total running time is $O(n^3)$. $\square$

We now introduce two more tools that are important for our purpose. The first one is purely graph-theoretic and describes the structure of P_t-free graphs.

Theorem 7 ([7])**.** *Let G be a connected P_t-free graph with $t \geq 4$. Then G has a connected dominating set D such that $G|D$ is either P_{t-2}-free or isomorphic to P_{t-2}.*

The next observation generalizes the observation by Edwards [12] that List k-Coloring can be solved in polynomial time, whenever the size of each list is at most two. This was already noted by e.g. Feder and Hell [14] and we include the proof for the sake of completeness.

Theorem 8. *Let (G, L) be an instance of* List H-Coloring *where G is of order n and $|L(v)| \leq 2$ for every $v \in V(G)$. Then one can determine in $O(n^2)$ time if (G, L) is H-colorable and find an H-coloring if one exists.*

Proof. If $L(v) = \emptyset$ for some $v \in V(G)$, then we claim that (G, L) is not C_k-colorable. Otherwise we construct a 2-Sat instance as follows.

- For every $v \in V(G)$ and $x \in L(v)$, we introduce a variable v_x. The meaning of v_x is that v_x is true if and only if v is colored with color x.
- For every $v \in V(G)$, we add a clause $\{v_x\}$ if $L(v) = \{x\}$, and we add two clauses $\{v_x, v_y\}$ and $\{\neg v_x, \neg v_y\}$ if $L(v) = \{x, y\}$. This ensures that v gets exactly one color from $L(v)$.
- For every edge $uv \in E(G)$ and every $x \in L(u)$, $y \in L(v)$, such that $xy \notin E(H)$, we add a clause $\{\neg u_x, \neg v_y\}$. This ensures that the edge uv is mapped onto an edge of H.

Obviously, (G, L) has an H-coloring if and only if the 2-Sat instance is satisfiable. In case the 2-Sat instance is satisfiable, we can obtain an H-coloring of G by setting $c(v) = x$ if v_x is true for every $v \in V(G)$ and every $x \in L(v)$. The 2-Sat instance has $O(n)$ variables and $O(n^2)$ clauses and so it can be solved in $O(n^2)$ time by [2]. $\square$

3 Polynomial algorithm for P_9-free graphs

Let $k = 5, 7$ or an integer greater than 8. In this section, we show that List C_k-Coloring can be solved in polynomial time for P_9-free graphs.

Theorem 1. *Let $k \geq 9$ or $k = 5$ or 7, and let (G, L) be an instance of* List C_k-Coloring *where G is a P_9-free graph of order n. Then one can determine in $O(n^{12k+3})$ time if (G, L) is C_k-colorable, and find a C_k-coloring of (G, L) if one exists.*

Outline of the proof. The overall strategy is to reduce the instance (G, L), in polynomial time, to polynomially many instances of 2-Sat in such a way that (G, L) is C_k-colorable if and only if at least one of the 2-Sat instances is a yes-instance. We then apply [2] to solve each 2-Sat instance in polynomial time.

More specifically, our algorithm, at a high level, has the following five phases. First, we apply Theorem 7 to show that G has a four-layer structure $\mathcal{P} = (S, X, Y, Z)$ such that the sets S, X, Y and Z form a partition of $V(G)$, and S is connected and of bounded size. The set S is called the *seed*. Second, we branch on every possible coloring of $G|S$ that respects the lists L. For each of these colorings of $G|S$, we propagate the coloring on S to the vertices of $G \setminus S$ via updates. After updating, the vertices in $S \cup X$ will have lists of size at most 2, but the vertices in $Y \cup Z$ may still have lists of size size more than 2. In the third step, we reduce the instance to polynomially many subinstances via branching in such a way that each of the subinstances avoids certain configurations, which we call *bad paths*. Moreover, we ensure that the original instance is a yes-instance if and only if at least one of the new subinstances is a yes-instance. Finally, using the fact that the subinstance has no bad paths, we may reduce the list size of vertices in $Y \cup Z$ and thus obtain an equivalent instance of 2-SAT using Theorem 8.

We now give a formal proof of the theorem.

Proof of Theorem 1. We may assume that G is connected, for otherwise we can solve the problem for each connected component of G independently. Moreover, one can determine in $O(n^3)$ time if G has a triangle. If so, we stop and claim that (G, L) is not C_k-colorable since there is no homomorphism from a triangle to C_k when $k \geq 4$. Therefore, we assume that G is triangle-free from now on.

Phase I. Obtaining a layer structure.

Claim 1. *There exists $S \subseteq V(G)$ such that $|S| \leq 7$, $G|S$ is connected and $S \cup N(S) \cup N(N(S))$ dominates G.*

Proof of Claim. We apply Theorem 7 to G: G has a connected dominating set D that induces a subgraph that is either P_7-free or isomorphic to a P_7. If $G|D$ is isomorphic to a P_7, then D is the desired set. Otherwise we apply Theorem 7 on $G|D$ to conclude that $G|D$ has a connected dominating set D' that induces a subgraph that is either P_5-free or isomorphic to a P_5. If $G|D'$ is isomorphic to a P_5, then D' is the desired set. Otherwise $G|D'$ is P_5-free. We again apply Theorem 7 on $G|D'$: $G|D'$ has a connected dominating set D'' that induces a subgraph that is either P_3-free or isomorphic to a P_3. Then $D'' \cup N(D'') \cup N(N(D''))$ dominates G. Since G is triangle-free, if $G|D''$ is P_3-free, then D'' is a clique of size at most 2. It follows that $|D''| \leq 3$ and so D'' is the desired set. ■

Let S be the set guaranteed by Claim 1. Define $X = N(S)$, $Y = N(N(S)) \setminus S$ and $Z = V(G) \setminus (X \cup Y \cup Z)$. Then (S, X, Y, Z) is a partition of $V(G)$, and S dominates X, X dominates Y and Y dominates Z, and there is no edge between S and $Y \cup Z$ or between X and Z. Moreover, S is connected. Such a quadruple (S, X, Y, Z) is called a *layer structure* of G. We write $\mathcal{P} = (S, X, Y, Z)$. The set S is called the *seed* for $\mathcal{P}$.

Phase II. Obtaining a canonical C_k-list assignment via updates.

We now branch on every possible coloring of S, respecting the lists L. Since $|S| \leq 7$, there are at most 7^k such colorings of $G|S$. Note that 7^k is a constant since k is a fixed number. To prove the theorem, therefore, it suffices to determine whether a given coloring $f : S \to [k]$ can be extended to a C_k-coloring of (G, L) in polynomial time. In the following, we fix such a coloring $f : S \to [k]$, and therefore, we are dealing with an instance (G, L') where

$$L'(v) = \begin{cases} L(v) & \text{if } v \notin S, \\ \{f(v)\} & \text{if } v \in S. \end{cases}$$

We further partition the sets S, X and Y as follows. For $1 \leq i \leq k$, let

$$S_i = \{s \in S : L(s) = \{i\}\},$$

$$X_i = \{x \in X \setminus (\bigcup_{j=1}^{i-1} X_j) : N(x) \cap S_i \neq \emptyset.\},$$

$$Y_i = \{y \in Y \setminus (\bigcup_{j=1}^{i-1} Y_j) : N(y) \cap X_i \neq \emptyset.\}.$$

Clearly, $(X_1, X_2, \ldots, X_k)$ is a partition of X and $(Y_1, Y_2, \ldots, Y_k)$ is a partition of Y.

We now perform the following updates for all $1 \leq i \leq k$ in the following order.

- For every edge sx with $s \in S_i$ and $x \in X_i$, we update x from s.
- For every edge xy with $x \in X_i$ and $y \in Y_i$, we update y from x.

We continue to denote the resulting C_k-list assignment by L'. Then $|L'(s)| = 1$ for every $s \in S$, $L'(x) \subseteq \{i-1, i+1\}$ for every $x \in X_i$ and $L'(y) \subseteq \{i, i-2, i+2\}$ for every $y \in Y_i$. We call such a C_k-list assignment L' *canonical* for $\mathcal{P} = (S, \bigcup_{i=1}^{k} X_i, \bigcup_{i=1}^{k} Y_i, Z)$.

Claim 2. *If X_i is not stable, then (G, L') is not C_k-colorable.*

Proof of Claim. Suppose that X_i contains an edge uv. In every C_k-coloring g of (G, L'), there is a $j \in [k]$ such that $\{g(u), g(v)\} = \{j, j+1\}$. Since $L'(u), L'(v) \subseteq \{i-1, i+1\}$, it follows that no such C_k-coloring exists. ■

Note that one can determine in $O(n^2)$ time if there exists an X_i that is not stable. If so, we stop and correctly determine that (G, L') is not C_k-colorable by Claim 2. Otherwise, we may assume that X_i is stable for all $1 \leq i \leq k$ from now on.

Phase III. Eliminating bad paths via branching ($O(n^{12k})$ branches).

In this phase, we shall reduce the instance (G, L') to an equivalent set of polynomially many subinstances so that every subinstance has no bad paths, which we define now.

Definition (Bad path). An induced path $a - b - c$ is a *bad path* in $\mathcal{P} = (S, X, Y, Z) = (S, \bigcup_{i=1}^{k} X_i, \bigcup_{i=1}^{k} Y_i, Z)$ if for some $i \in [k]$, $a \in Y_i$, $b, c \in (Y \cup Z) \setminus Y_i$ and $\{b, c\}$ is anticomplete to X_i. We call a the *starter* of $a - b - c$. Let $\mathcal{P}_i$ be the set of all bad paths with starters in Y_i. Note that $|\mathcal{P}_i| = O(n^3)$.

Definition (Depth). A vertex $v \in Y_i$ is of *depth ℓ* to the seed S if for every $x \in N(v) \cap X_i$, there exists an induced path $v - x - P$ of length ℓ such that $V(P) \subseteq S$.

Observe that every vertex in Y is of depth at least 3 to S (because we may assume that $|S| \geq 2$ and so no vertex in X is complete to S since G is triangle-free), and that the starter of a bad path is of depth at most 7 to S since G is P_9-free.

Note that for any C_k-coloring of (G, L') (if one exists), either there exists a bad path in $\mathcal{P}_i$ whose starter is colored with a color in $\{i-2, i+2\}$ or the starters of all bad paths in $\mathcal{P}_i$ are colored with color i. This leads to the following branching scheme.

Branching. (List change only.)

- ($2^k = O(1)$ branches.)

  For every subset $I \subseteq [k]$, we have a branch B_I intended to find possible colorings such that there exists a bad path in $\mathcal{P}_i$ whose starter is colored with a color in $\{i-2, i+2\}$ if $i \in I$, and all starters of bad paths in $\mathcal{P}_i$ are colored with color i if $i \notin I$. Clearly, (G, L') is C_k-colorable if and only if at least one of the B_I is a yes-instance. In the following, we fix a branch B_I.

- ($O(2^k n^{3k}) = O(n^{3k})$ branches.)

  We further branch to obtain a set of size $O(n^{3k})$ of subinstances within B_I by guessing, for each $i \in I$, a bad path in $\mathcal{P}_i$, and guessing the color of its starter from $\{i-2, i+2\}$. The union over all branches B_I of these subinstances is equivalent to (G, L').

  Specifically, for each element $(a_i, b_i, c_i)_{i \in I}$ in $\Pi_{i \in I} \mathcal{P}_i$, we have one branch where we set $L''(a_i) := L'(a_i) \cap \{i-2, i+2\}$ for every $i \in I$, and we set $L''(a) := L'(a) \cap \{i\}$ for every starter a of a bad path in $\mathcal{P}_i$ for every $i \notin I$. We denote the resulting C_k-list assignment by L''. For each such branch and for every element $(q_i)_{i \in I}$ in $\Pi_{i \in I} L''(a_i)$, we have one branch where $L''(a_i) := \{q_i\}$ for all $i \in I$. It follows that for all $i \in I$ and $x \in X_i \cap N(a_i)$, the only possible color for x is $(q_i + i)/2$, and so we set $L''(x) = \{(q_i + i)/2\}$.

ss: I added this sentence since we later use that these lists have size one.

Since $L''(a_i) \subseteq \{i-2, i+2\}$ for all $i \in I$, it follows that there are $2^{|I|} \leq 2^k$ branches. Let us fix one such branch and denote the resulting instance by (G, L'').

- ($O(k^{3k}) = O(1)$ branches.)

  We let I^* be the subset of $[k] \setminus I$ of indices i such that $\mathcal{P}_i$ contains a bad path. For each $i \in I^*$, we choose a bad path $a_i - b_i - c_i$ in $\mathcal{P}_i$ such that $|N(a_i) \cap X_i|$ is minimum, where the minimum is taken over all bad paths in $\mathcal{P}_i$. Choose a vertex $x_i \in N(a_i) \cap X_i$ for each $i \in I^*$. Let
  $$Q = \bigcup_{i \in I} \{b_i, c_i\} \cup \bigcup_{i \in I^*} \{b_i, c_i, x_i\},$$
  where for $i \in I$, b_i, c_i are two vertices on the bad path we guessed in the previous bullet. We branch on every possible coloring of Q, respecting the lists L. Since $|Q| \leq 3k$, the number of branches is at most k^{3k}. In the following, we fix a coloring g of Q and denote the resulting subinstance by (G, L'''), where
  $$L'''(v) = \begin{cases} L''(v) & \text{if } v \notin Q, \\ \{g(v)\} & \text{if } v \in Q. \end{cases}$$

Obtaining a new layer structure with a canonical C_k-list assignment.

We now deal with (G, L'''). Define
$$A = \bigcup_{i \in I} ((N(a_i) \cap X_i) \cup \{a_i, b_i, c_i\}) \cup \bigcup_{i \in I^*} \{x_i, a_i, b_i, c_i\},$$
and note that in L''', every vertex in A has a list of size at most 1. We update all vertices of G from all vertices in A and continue to denote the resulting C_k-list assignment by L'''. We now obtain a new partition $\mathcal{P}' = (S', X', Y', Z')$ of G as follows.

- Let $S' = S \cup A$.
- For each $1 \leq j \leq k$, let $K_j := \emptyset$. For each vertex $v \in Y \cup Z$, if v has a neighbor in S', let j be the smallest integer in $[k]$ such that there exists a vertex $s \in N(v) \cap S'$ with $L(s) = \{j\}$, and add v to K_j. For each $1 \leq j \leq k$, let $X'_j = (X_j \cup K_j) \setminus A$. Let $X' = \bigcup_{i=1}^k X'_i$.
- For $1 \leq i \leq k$, let Y'_i be the set of vertices in $V(G) \setminus (S' \cup X' \cup (\bigcup_{j<i} Y'_j))$ that have a neighbor in X'_i. Let $Y' = \bigcup_{i=1}^k Y'_i$.
- Let $Z' = V(G) \setminus (S' \cup X' \cup Y')$.

Claim 3. *The new partition $\mathcal{P}' = (S', X', Y', Z')$ is a layer structure of G and L''' is a canonical C_k-list assignment for $\mathcal{P}'$.*

Proof of Claim. From the definition of (S', X', Y', Z'), it follows that S' dominates X', X' dominates Y' and Y' dominates Z'. There are no edges between S' and $Y' \cup Z'$ and no edges between X' and Z'. Moreover, S' is connected by the definition of A and the fact that S is connected. So $\mathcal{P}' = (S', X', Y', Z')$ is a layer structure. Note that $|L'''(s)| = 1$ for every $s \in S'$, $L'''(x) \subseteq \{i-1, i+1\}$ for every $x \in X'_i$ and $L'''(y) \subseteq \{i, i-2, i+2\}$ for every $y \in Y'_i$. So L''' is canonical for $\mathcal{P}' = \left(S', \bigcup_{i=1}^k X'_i, \bigcup_{i=1}^k Y'_i, Z'\right)$. ■

Claim 4. *The following hold for $\mathcal{P}' = (S', X', Y', Z')$.*

(1) $X'_i \setminus X_i \subseteq Y \cup Z$.

(2) If a vertex in $Y' \cup Z'$ is anticomplete to X'_i, then it is anticomplete to X_i.

(3) $Y'_i \setminus Y_i$ is anticomplete to X_i.

Proof of Claim. By construction, $X'_i \setminus X_i \subseteq K_i \subseteq Y \cup Z$ and so (1) follows.

Let $v \in Y' \cup Z'$ be anticomplete to X'_i. Since $v \notin S' \cup X'$, it follows that v is anticomplete to A. Note that $X_i \setminus X'_i \subseteq A$. So (2) follows from the assumption that v is anticomplete to X'_i.

Suppose for a contradiction that there exists $y \in Y'_i \setminus Y_i$ that has a neighbor in X_i. Since $y \in Y'_i$ and Z is anticomplete to X, it follows that $y \in Y_j$ for some $j \neq i$. From the definition of the sets $Y_1, \dots, Y_k$, it follows that $y \in Y_j$ for some $j < i$. Let $x \in X_j$ be a neighbor of y. It follows that $x \notin X'_j$, for otherwise y would be in Y'_k for some $k \leq j$, which contradicts the assumption that $y \in Y'_i$. So $x \in A$, but then $y \in X'$, a contradiction. So (3) follows. ■

The following claim is the key to our branching algorithm.

Claim 5. *Let a be a starter of a bad path in $\mathcal{P}'$. If the depth of the starter of any bad path in $\mathcal{P}$ is at least ℓ, then the depth of a in $\mathcal{P}'$ is at least $\ell + 1$.*

Proof of Claim. Let $a' - b' - c'$ be a bad path in $\mathcal{P}'$ with $a' \in Y'_i$. We consider the following two cases.

Case 1: $a' \in Y_i \cap Y'_i$.

Then $\emptyset \neq N(a') \cap X_i \subseteq X'_i$. By (2), $\{b', c'\}$ is anticomplete to X_i and so $a' - b' - c'$ is also a bad path in $\mathcal{P} = (S, X, Y, Z)$. This implies that $\mathcal{P}_i \neq \emptyset$. Therefore, there exist $a, b, c, x \in S'$ such that $a - b - c$ is a bad path in $\mathcal{P}$ with $a \in Y_i$ and $x \in N(a) \cap X_i$.

We first claim that it is possible to pick a vertex $x' \in N(a') \cap X_i$ that is not adjacent to a. Recall that the branch we consider corresponds to a set $I \subseteq [k]$. If $i \in I$, then all vertices in $N(a) \cap X_i$ are in A and hence are now in S'. So every vertex in $N(a') \cap X_i$ is not adjacent to a, and our claim holds. If $i \notin I$, then $i \in I^*$, and so $a = a_i$. By the choice of a_i, it follows that $|N(a) \cap X_i| \leq |N(a') \cap X_i|$. Since $a' \in Y'_i$, it follows that a' is not adjacent to x. Therefore, there exists a vertex $x' \in N(a') \cap X_i$ such that x' is not adjacent to a.

Note that x and x' are not adjacent by Claim 2. Moreover, x' is anticomplete to $\{b', c', b, c\}$ by the definition of bad path. Let P' be the shortest path from x to x' with internal vertices contained in S. Note that P' exists since S is connected. Then P' is an induced path. Since $V(P') \setminus \{x, x'\} \subseteq S$, it follows that $V(P') \setminus \{x, x'\}$ is anticomplete to $\{a, b, c, a', b', c'\}$. Therefore, $c - b - a - x - P' - x' - a' - b' - c'$ is an induced path of order at least 9, a contradiction.

Case 2: $a' \in Y'_i \setminus Y_i$.

It follows from (3) that $N(a') \cap X'_i \subseteq X'_i \setminus X_i$. Pick a vertex $x' \in N(a') \cap X'_i$. Since $x \in X'_i \setminus X_i$, x' has a neighbor $s' \in S'$ by the definition of X'_i. By (1), $x' \in Y \cup Z$ and so $s' \in S' \setminus S = A$. This implies that there exists an index $j \in I$ such that x' is not anticomplete to $Q = \{x_j, a_j, b_j, c_j\}$ where $x_j \in N(a_j) \cap X_j$. Let $a_j - x_j - P$ be an induced path of length ℓ with $V(P) \subseteq S$. Note that $x' \in Y \cup Z$ is anticomplete $V(P) \subseteq S$. Let $x' - P'' - x_j$ be the shortest path from x' to x_j such that $V(P'') \subseteq Q$. Since a' is anticomplete to $\{x\} \cup V(P) \cup V(P'') \subseteq S'$, it follows that $a' - x' - P'' - x_j - P$ is an induced path of length at least $\ell + 1$. This proves the claim. ■

Therefore, we have obtained an equivalent set of subinstances of size $O(n^{3k})$. For each such subinstance, the minimum depth of the starter of a bad path has increased by at least 1 compared to $\mathcal{P}$ due to Claim 5. Note that the depth of any starter of a bad path in $\mathcal{P}$ is at least 3. Moreover, since G is P_9-free, the depth of any starter of a bad path is at most 7. By branching 4 times, therefore, we obtain an equivalent set of $O(n^{12k})$ subinstances such that each subinstance has no bad paths.

Phase IV. Reducing the list size of vertices in Z.

Let us now fix an instance (G, L) where $\mathcal{P} = (S, X, Y, Z)$ is a layer structure with no bad paths and L is canonical for $\mathcal{P}$. We first reduce the list size of vertices in Z.

Claim 6. *The set Z is stable and each $z \in Z$ has neighbors in at most one of $\{Y_1, Y_2, \ldots, Y_k\}$.*

Proof of Claim. Suppose by contradiction that Z has an edge z_1z_2. Let $y \in Y$ be a neighbor of z_1. Since G is triangle-free, z_2 is not adjacent to y. Then $y - z_1 - z_2$ is a bad path, a contradiction. So Z is stable. Suppose that $z \in Z$ has a neighbor $y_i \in Y_i$ and $y_j \in Y_j$ for $i \neq j$. We may assume that $i < j$. Then by the definition of $Y_1, Y_2, \ldots, Y_k$, it follows that y_j is anticomplete to X_i. So $y_i - z - y_j$ is a bad path, a contradiction. ■

For $i \in [k]$, we let $Z_i = N(Y_i) \cap Z$. By Claim 6, it follows that $Z_1, \ldots, Z_k$ is a partition of Z.

Claim 7. *There is an equivalent instance (G, L') for (G, L) such that $L'(z)$ is a subset of $\{i+1, i-1\}, \{i+1, i-3\}, \{i-1, i+3\}$ or $\{i+3, i-3\}$ for all $z \in Z_i$.*

Proof of Claim. For $z \in Z_i$, we define $c_1(z) = \{i-1\}$ if $i-1 \in L(z)$, and $c_1(z) = \{i-3\} \cap L(z)$, otherwise; we define $c_2(z) = \{i+1\}$ if $i+1 \in L(z)$, and $c_2(z) = \{i+3\} \cap L(z)$, otherwise. Now let $L'(z) = c_1(z) \cup c_2(z)$ for all $z \in Z$. It follows that $L'(z)$ is a subset of $\{i+1, i-1\}, \{i+1, i-3\}, \{i-1, i+3\}$ or $\{i+3, i-3\}$ for all $z \in Z_i$.

Since (G, L') is a subinstance of (G, L), it follows that if (G, L') has a C_k-coloring, then so does (G, L). Now suppose that (G, L) has a C_k-coloring c, and choose c such that $c(z) \in L'(z)$ for as many $z \in Z$ as possible. If $c(z) \in L'(z)$ for all $z \in Z$, then c is a C_k-coloring of (G, L'), and so (G, L') is equivalent to (G, L) and the claim follows.

Now suppose for a contradiction that there is a vertex $z \in Z$ such that $c(z) \notin L'(z)$. Since every vertex $y \in Y_i$ satisfies $L(y) \subseteq \{i, i+2, i-2\}$, $c(z) \in \{i+1, i-1, i+3, i-3\}$. If $c(z) \in \{i+1, i-1\}$, since $c(z) \in L(z)$, by the definition of $L'(z)$ it follows that $c(z) \in L'(z)$, a contradiction. So $c(z) \in \{i-3, i+3\}$. By symmetry, we may assume that $c(z) = i+3$. Since $c(z) \notin L'(z)$, it follows that $c_2(z) = i+1$, and therefore $i+1 \in L(z)$. Let $y \in N(z)$. By Claim 6, it follows that $y \in Y_i$, and consequently, $L(y) \subseteq \{i, i-2, i+2\}$. Since $c(z) = i+3$, and c is a C_k-coloring, it follows that $c(y) = i+2$ for all $y \in N(z)$. Now define c' by letting $c'(z) = i+1$, and $c'(v) = c(v)$ for all $v \neq z$. It follows that c' is a C_k-coloring of (G, L), contrary to the choice of c. This is a contradiction, and the claim follows. ■

We now modify the lists of vertices in Z as in the C_k-list assignment L' of Claim 7, and we continue to denote by the resulting list L.

Phase V. Reducing the list size of vertices in Y.

We now apply Lemma 6 on $G|S \cup X \cup Y$ to obtain a C_k-list assignment L' which is reduced on $G|S \cup X \cup Y$. Then (G, L') is an equivalent subinstance of (G, L).

If $L'(v) = \emptyset$ for some $v \in V(G)$, we stop and claim that (G, L) is not C_k-colorable. Otherwise define

$$
\begin{aligned}
S' &= \{v \in S \cup X \cup Y : |L'(v)| = 1\}, \\
X'_i &= \{v \in (X \cup Y) \setminus S' : L'(v) \subseteq \{i-1, i+1\}\}, 1 \le i \le k, \\
Y'_i &= \{v \in Y \setminus (S' \cup X' \cup \bigcup_{j<i} Y'_j) : L'(v) \subseteq \{i, i-2, i+2\}\}, 1 \le i \le k, \\
Z' &= Z \\
X' &= \bigcup_{i=1}^{k} X'_i, \\
Y' &= \bigcup_{i=1}^{k} Y'_i.
\end{aligned}
$$

Recall that we have modified the list of $z \in Z$ according to Claim 7. It follows that $|L'(v)| \le 3$ for every $v \in Y'$ and $|L'(v)| \le 2$ for every $v \in G \setminus Y'$. We prove a few properties for S', X', Y' and Z'.

Claim 8. *The following hold for S', X', Y' and Z'.*

(1) $V(G) = S' \cup X' \cup Y' \cup Z'$.

(2) For every $i \in [k]$ and $y \in Y'_i$, we have $N(y) \cap (S' \cup X') \subseteq X'_i$.

(3) For every $i \in [k]$, we have $X_i \subseteq X'_i \cup S'$, $Y'_i \subseteq Y_i$ and $Y_i \subseteq Y'_i \cup S' \cup X'$.

(4) There does not exist an induced path $a - b - c$ such that $a \in Y'_i$, $b, c \in Y' \cup Z' \setminus Y'_i$.

(5) Let C be a component in $Y' \cup Z'$ such that $V(C) \cap Y' \neq \emptyset$, then one of the followings holds:

(a) $V(C) \subseteq Y'_i$ for some $i \in [k]$;

(b) $V(C) \subseteq Y'_i \cup Z'$ for some $i \in [k]$; or

(c) $V(C) \subseteq Y'_i \cup Y'_j$ for some $i, j \in [k]$ and every edge in C has one end in Y_i and the other in Y_j;

Proof of Claim. Recall that $\mathcal{P} = (S, X, Y, Z)$ is a layer structure of G that is given at the beginning of Phase IV. Let $v \in V(G) = S \cup X \cup Y \cup Z$. If $v \in S$, then $v \in S$. If $v \in X$, then $s \in S' \cup X'$. If $v \in Y$, then $v \in S' \cup X' \cup Y'$. This proves (1).

Let $y \in Y'_i$. Then $L'(y) \subseteq \{i-2, i, i+2\}$. Since $y \notin S' \cup X'$, it follows that $L'(y) = \{i-2, i+2\}$ or $L'(y) = \{i-2, i, i+2\}$. Since L' is reduced on $G|S' \cup X' \cup Y'$, $N(y) \cap S' = \emptyset$. Pick an arbitrary vertex $x \in N(y) \cap X'_j$ for some $j \in [k]$. Then $L'(x) = \{j+1, j-1\}$. We show that $i = j$. Since L' is reduced on $G|S' \cup X' \cup Y'$, we cannot effectively update y from x. It follows that $L'(y) \subseteq \{j-2, j, j+2\}$. If $L'(y) = \{i-2, i, i+2\}$, then either $i = j$ or $i = j \pm 2$ and $k = 6$. If $L'(y) = \{i-2, i+2\}$, then either $i = j$, $i = j \pm 2$ and $k = 6$ or $i = j \pm 4$ and $k = 8$. Since $k \neq 6, 8$, it follows that $i = j$. This proves (2).

Let $x \in X_i$. Then $L(x) \subseteq \{i-1, i+1\}$. Since update cannot make the list larger, it follows that $L'(x) \subseteq L(x)$ and so $x \in X'_i \cup S'$. Let $y \in Y'_i$. Then $L'(y) \subseteq \{i, i-2, i+2\}$. Since $y \notin S' \cup X'$, it follows that $L'(y) = \{i-2, i+2\}$ or $L'(y) = \{i-2, i, i+2\}$. If $y \notin Y_i$, then $y \in Y_j$ for some $j \neq i$. So $L(y) \subseteq \{j, j-2, j+2\}$. Then $L'(y) \subseteq \{j-2, j, j+2\}$. If $L'(y) = \{i-2, i, i+2\}$, then either $i = j$ or $i - j = \pm 2$ and $k = 6$. If $L'(y) = \{i-2, i+2\}$, then either $i = j$ or $i - j = \pm 2$ and $k = 6$ or $i - j = \pm 4$ and $k = 8$. All cases lead to a contradiction, so $y \in Y_i$ and $Y'_i \subseteq Y_i$. Let $y \in Y_i$, then by construction, $y \in S' \cup X' \cup Y'$. If $y \in Y'_j$ for some $j \neq i$, then $y \in Y_j$, contrary to $y \in Y_i$. So $y \notin Y'_j$ with $j \neq i$ and it follows that $Y_i \subseteq Y'_i \cup S' \cup X'$. This proves (3).

To prove (4), it is sufficient to show that such a path $a-b-c$ is a bad path in $\mathcal{P}$. By (3) and the construction, $a \in Y_i$, $b,c \in Y \cup Z \setminus Y_i$. If $w \in \{b,c\}$ is adjacent to $x \in X_i$, then $w \in Y \setminus Y_i$. We may assume $w \in Y_j$ for $j \neq i$, then by (3) $w \in Y'_j$ and by (2) $x \in X'_j$, a contradiction to $x \in X_i \subseteq X'_i \cup S'$. It follows that $\{b,c\}$ is anticomplete to X_i and $a-b-c$ is a bad path in $\mathcal{P}$. This proves (4).

Let C be a component in $Y' \cup Z'$ such that $V(C) \cap Y' \neq \emptyset$. By Claim 6, $Z' = Z$ is stable and for every $z \in Z$, $N(z) \subseteq Y_i$ for some i. Hence for every $z \in C$, $N(z) \cap V(C) \subseteq Y'_i$ for some i. Assume that (5).(a) and (5).(b) do not hold, then there exists an edge $uv \in C$ such that $u \in Y'_i$, $v \in Y'_j$, $i,j \in [k]$ with $i \neq j$. Suppose for a contradiction there exists $w \in C$ with $w \in Y'_k \cup Z$ for some $k \neq i,j$. Let $u - v - p_1 - p_2 - \cdots - p_n = w$ be the shortest path from $\{u,v\}$ to w in C. Since $u \in Y'_i$ and $v \in Y'_j$, it follows from (4) that $p_1 \in Y'_i$, and then $p_2 \in Y'_j$, $p_3 \in Y'_i$, and so on. Inductively, it follows that $w = p_n \in Y'_i \cup Y'_j$, a contradiction. So $V(C) \subseteq Y'_i \cup Y'_j$. By (4), every edge in C has one end in Y'_i and the other in Y'_j. This proves (5). ∎

We construct a C_k-list assignment L'' as follows: for every component C in $Y' \cup Z'$ such that $V(C) \cap Y' \neq \emptyset$,

- If $V(C) \subseteq Y'_i$ for some $i \in [k]$ and $|V(C)| \geq 2$, for every $v \in V(C)$, set $L''(v) = \{i+2, i-2\} \cap L'(v)$ if $k = 5$ and set $L''(v) = \emptyset$ otherwise;
- If $V(C) \subseteq Y'_i$ for some $i \in [k]$ and $|V(C)| = 1$, for $v \in V(C)$, set $L''(v) = L'(v)$ if $|L'(v)| < 3$ and $L''(v) = \{i\}$ if $|L'(v)| = 3$;
- If $V(C) \subseteq Y'_i \cup Z'$ for some $i \in [k]$ and there exists an edge in $G|(V(C) \cap Y'_i)$, for every $v \in V(C) \cap Y'_i$, set $L''(v) = \{i+2, i-2\} \cap L'(v)$ if $k = 5$ and set $L''(v) = \emptyset$ otherwise;
- If $V(C) \subseteq Y'_i \cup Y'_{i+1}$ for some $i \in [k]$, for every $v \in V(C) \cap Y'_i$ with $|L'(v)| = 3$ set $L''(v) = L'(v) \setminus \{i-2\}$ and for every $v \in V(C) \cap Y'_{i+1}$ with $|L'(v)| = 3$ set $L''(v) = L'(v) \setminus \{i+3\}$;

And set $L''(v) = L'(v)$ for every other vertex in G. It is clear that $|L''(v)| \leq 3$ for every $v \in G$ and $|L''(v)| \leq 2$ for every $v \in G \setminus Y'$. Next we prove that (G, L'') is an equivalent subinstance of (G, L) and we can apply Theorem 8 on (G, L'').

Claim 9. *The follow holds for (G, L''):*

(1) (G, L'') is an equivalent subinstance of (G, L).

(2) Let $A = \{v \in V(G) | |L''(v)| = 3\}$, then A is a stable set and

- *For every $v \in V(G) \setminus A$, $|L(v)| \leq 2$.*
- *For every $v \in A$, $L(v) = \{i, i-2, i+2\}$ for some $i \in [k]$ and for every $u \in N(v)$, $L(u)$ is a subset of $\{i+1, i-1\}, \{i+1, i-3\}, \{i-1, i+3\}$ or $\{i+3, i-3\}$.*

Proof of Claim. Recall that for $v \in Y'_i$, if $|L'(v)| = 3$, then $L'(v) = \{i, i-2, i+2\}$. It follows from the construction of L'' that (G, L'') is an subinstance of (G, L') and therefore if (G, L'') has a C_k-coloring, then so does (G, L'). Now suppose that (G, L') has a C_k-coloring c, and choose c such that $c(v) \in L''(v)$ for as many $v \in G$ as possible. Suppose for a contradiction that there exists $v \in G$ such that $c(v) \notin L''(v)$, then $v \in C$, where C is a component C in $Y' \cup Z'$ such that $V(C) \cap Y' \neq \emptyset$ and one of the following cases hold:

- Case 1: $V(C) \subseteq Y'_i$ for some $i \in [k]$ and $|V(C)| \geq 2$. Since there exists an edge in $G|(V(C) \cap Y'_i)$ and $L'(u) \subseteq \{i, i-2, i+2\}$ for every $u \in Y'_i$, it follows that $k = 5$, $L''(v) = \{i+2, i-2\} \cap L'(v)$ and $c(v) \in \{i-2, i+2\}$, a contradiction to $c(v) \notin L''(v)$.
- Case 2: $V(C) \subseteq Y'_i$ for some $i \in [k]$ and $|V(C)| = 1$. Then $L'(v) = \{i, i+2, i-2\}$ and $L''(v) = \{i\}$. By Claim 8, $N(v) \subseteq X'_i$. It follows that for every $u \in N(v)$, $L(u) \subseteq \{i-1, i+1\}$. Now define c' by letting $c'(v) = i$ and $c'(u) = c(u)$ for every $u \neq v$. It follows that c' is a C_k-coloring of (G, L'), contrary to the choice of c.

- Case 3: $V(C) \subseteq Y_i' \cup Z'$ for some $i \in [k]$ and there exists an edge in $G|(V(C) \cap Y_i')$. Similarly to Case 1, it follows that $k = 5$, $L''(v) = \{i+2, i-2\} \cap L'(v)$ and $c(v) \in \{i-2, i+2\}$, a contradiction to $c(v) \notin L''(v)$.

- Case 4: $V(C) \subseteq Y_i' \cup Y_{i+1}'$ for some $i \in [k]$. We may assume $v \in Y_i'$, then $L'(v) = \{i, i+2, i-2\}$, $L''(v) = \{i, i+2\}$ and $c(v) = i-2$. It follows from Claim 8 that $N(v) \subseteq Y_{i+1} \cup X_i$. Let $N = \bigcup_{u \in N(v)} \{c(u)\}$, then $N \subseteq \bigcup_{u \in Y_{i+1} \cup X_i} L(u) \subseteq \{i-1, i+1, i+3\}$. Since $c(v) = i-2$, $N = \{i-1\}$. Now define c' by letting $c'(v) = i$ and $c'(u) = c(u)$ for every $u \neq v$. It follows that c' is a C_k-coloring of (G, L'), contrary to the choice of c.

This proves (1).

Let $A = \{v \in V(G) | |L''(v)| = 3\}$. Then for $|L''(v)| \leq 2$ for $v \in G \setminus A$. Pick $v \in A$, then $|L'(v)| = 3$ and $v \in Y'$. Let C be the component in $Y' \cup Z'$ contains v. By Claim 8.(5) and the construction of L'', either $V(C) \subseteq Y_i' \cup Z'$ for some $i \in [k]$ and $G|(V(C) \cap Y_i')$ is a stable set, or $V(C) \subseteq Y_i' \cup Y_j'$ for some $i, j \in [k]$, $V(C) \cap Y_i' \neq \emptyset$, and $\{i, j\} \not\subseteq \{\ell, \ell+1\}$ for any $\ell \in [k]$. First assume the latter holds and $v \in Y_i'$, then v is adjacent to $u \in Y_j'$ where $j \notin \{i, i+1, i-1\}$. Recall that L' is reduced on $G|S' \cup X' \cup Y'$. Since $L'(v) = \{i, i+2, i-2\}$ and $L'(u) \subseteq \{j, j+2, j-2\}$, it follows that $\{i, i+2, i-2\} \subseteq \{j+3, j+1, j-1, j-3\}$. Then since $j \notin \{i-1, i, i+1\}$, $\{i-2, i, i+2\} = \{j-3, j+1, j+3\}$ or $\{i-2, i, i+2\} = \{j-3, j-1, j+3\}$. In either case, it implies that $k = 6$ which is a contradiction. So $V(C) \subseteq Y_i' \cup Z'$ for some $i \in [k]$ and $G|(V(C) \cap Y_i')$ is a stable set. By Claim 8.(2), it follows that $N(v) \subseteq X_i' \cup Z$. Let $z \in Z$ adjacent to v. Then by Claim 7, $L(z)$ is a subset of $\{i+1, i-1\}, \{i+1, i-3\}, \{i-1, i+3\}$ or $\{i+3, i-3\}$. This proves (2).

■

Now we can apply Theorem 8 and this completes the proof of correctness of our algorithm. Clearly, the most expensive part of our algorithm is Phase III where we branch into $O(n^{12k})$ subinstances. Since each subinstance can be constructed in $O(n^3)$ time by Lemma 6 and each 2-SAT instance can be solved in $O(n^3)$ time by Theorem 8, the total running time is $O(n^{12k+3})$. □

4 Improving the Polynomial Result

Let $k \geq 3$, and let $h : V(G) \to V(C_k)$ be a homomorphism from a graph G to C_k. Let $c_1, \ldots, c_k$ denote the vertices of C_k in order. We define a directed graph G_h by assigning a direction to every edge $uv \in E(G)$ as follows: If $h(u) = c_i$ and $h(v) = c_{i+1}$, or if $h(u) = c_k$ and $h(v) = c_1$, we let $uv \in E(G_h)$; otherwise, we let $vu \in E(G_h)$.

Let D be a directed graph with no digons, that is, for all $u, v \in V(D)$, not both $uv \in E(D)$ and $vu \in E(D)$. A *walk* in D is a sequence $v_1, \ldots, v_j$ of vertices such that for all $i \in \{1, \ldots, j-1\}$, $v_i v_{i+1} \in E(D)$ or $v_{i+1} v_i \in E(D)$. It is a *closed walk* if in addition, $v_1 = v_j$. The *slope* $s(W)$ of a walk $W = v_1, \ldots, v_j$ is defined as

$$s(W) = |\{i \in \{1, \ldots, j-1\} : v_i v_{i+1} \in E(D)\}| - |\{i \in \{1, \ldots, j-1\} : v_{i+1} v_i \in E(D)\}|.$$

Lemma 9. *Let $t, k \in \mathbb{N}$ with $k \geq t+1$. Let G be a connected P_t-free graph, and let $h : V(G) \to V(C_k)$ be a homomorphism. Then $h(V(G))$ is contained in a $(t-1)$-vertex subpath of $V(C_k)$.*

Proof. Let G_h be as defined above. Let h' be the homomorphism given by the identity map of C_k, and let $H = (C_k)_{h'}$. Let $c_1, \ldots, c_k$ denote the vertices of C_k in order. We first prove:

Claim 10. *Let $W = v_1, \ldots, v_j$ be a walk in H with $v_1 = c_a$ and $v_j = c_b$. Then $s(W) + a - b$ is divisible by k.*

Suppose for a contradiction that W is a counterexample to Claim 10 with j minimum. If for some $i \in \{1, \ldots, j-2\}$, we have $v_i = v_{i+2}$, then $W' = v_1, \ldots, v_i, v_{i+3}, \ldots, v_j$ is a walk. Since the walk $W'' = v_i, v_{i+1}, v_{i+2}$ has slope zero, it follows that $s(W') = s(W) - s(W'') = s(W)$, and so W is not a minimum counterexample. It follows that for all $i \in \{1, \ldots, j_2\}$, $v_i \neq v_{i+2}$. By symmetry, we may assume that $v_1 = c_a$ and $v_2 = c_{a+1}$. It follows that for all $i \in \{1, \ldots, j-1\}$, $v_i v_{i+1} \in E(H)$, and for all $i \in \{1, \ldots, j\}$, $v_i = c_{i*}$ where $i - i^* + a - 1$ is divisible by k. This implies that $s(W) = j - 1$, and since $v_j = c_b$, it follows that $j - b + a - 1 = s(W) + a - b$ is divisible by k, a contradiction. This proves Claim 10.

Claim 11. *Let $W = v_1, \ldots, v_j$ be a walk in G_h with $h(v_1) = h(v_j)$. Then the slope of W is divisible by k.*

Let $W' = h(v_1), \ldots, h(v_j)$. From the definition of a homomorphism, it follows that W' is a closed walk in H. Moreover, from the definition of H, it follows that for every edge $uv \in E(G_h)$, we have $h(u)h(v) \in E(H)$; and therefore, $s(W) = s(W')$. Now Claim 11 follows from Claim 10.

Claim 12. *Let $W = v_1, \ldots, v_j$ be a walk in G_h with $h(v_1) = h(v_j)$. Then the slope of W is 0.*

Suppose for a contradiction that $W = v_1, \ldots, v_j$ is a walk of non-zero slope with $h(v_1) = h(v_j)$, and let W be chosen with j minimum. If there exist $i, i' \in \{1, \ldots, j\}$ with $i < i'$, $\{i, i'\} \neq \{1, j\}$ and $v_i = v_{i'}$, then we let $W_1 = v_1, \ldots, v_i, v_{i'+1}, \ldots, v_j$ and $W_2 = v_i, v_{i+1}, \ldots, v_{i'}$. It follows that W_1 has the same first and last vertex as W, and W_2 is a closed walk; and $s(W) = s(W_1) + s(W_2)$. This implies that at least one of $s(W_1), s(W_2)$ is non-zero, contrary to the minimality of j. It follows that $v_1, \ldots, v_{j-1}$ are distinct, and hence W is the vertex set of a (not necessarily induced) path or cycle C in G.

Since $s(W) \neq 0$ and k divides $s(W)$, it follows that C has at least k vertices. Since $k \geq t+1$, and G is P_t-free, it follows that C is not an induced path and not an induced cycle. Let $uw \in E(G)$ such that $u, w \in V(C)$, but $\{u, w\} \neq \{v_1, v_j\}$ and there is no $i \in \{1, \ldots, j-1\}$ such that $\{u, w\} = \{v_i, v_{i+1}\}$. By symmetry, we may assume that $u = v_i, w = v_{i'}$ and $i < i'$. Now let $W_3 = v_1, \ldots, v_i, v_{i'}, \ldots, v_j$ and $W_4 = v_i, v_{i+1}, \ldots, v_{i'}, v_i$. It follows that $s(W) = s(W_3) + s(W_4)$, since each consecutive pair of vertices of W occurs in exactly one of W_3, W_4; and the pair $v_i, v_{i'}$ occurs in opposite orders in W_3 and W_4. This implies that at least one of $s(W_3), s(W_4)$ is non-zero. From the choice of u and w, it follows that both W_3 and W_4 have fewer vertices than W, a contradiction. This implies Claim 12.

Let us say that u *precedes* w if $u, w \in V(G_h)$ and there is a walk $W = v_1, \ldots, v_j$ with $v_1 = u$ and $v_j = w$ such that $s(W) > 0$ in G_h. From Claim 12, it follows that no vertex precedes itself. Moreover, the definition immediately implies that this property is transitive, that is, if u precedes w and w precedes y, then u precedes y. This defines a partial order; and hence there is a vertex $u \in V(G_h)$ such that no vertex precedes u. By symmetry, we may assume that $h(u) = c_1$. Now suppose that there is a vertex $w \in V(G_h)$ with $h(w) = c_l$ for some $l \in \{t, \ldots, k\}$. Since G is connected, it follows that there is an induced u-w-path P in G, for example by taking P to be a shortest u-w-path. Let $W = v_1, \ldots, v_j$ denote the vertices of P in reverse order; it follows that W is a walk with $h(v_1) = c_l$ and $h(v_j) = c_1$. From Claim 10, we deduce that k divides $s(W) + l - 1$, and since $0 < |l - 1| < k$, it follows that $s(W) \neq 0$. Since P has length at most $t - 2$, it follows that $|s(W)| \leq t - 2 \leq k - 3$. This implies that $s(W) + l - 1 \in \{0, k\}$. Since $l - 1 \geq t - 1 > |s(W)|$, it follows that $s(W) = k - l + 1 > 0$. This implies that w precedes u, contradicting the choice of u. It follows that $h(V(G)) \subseteq \{c_1, \ldots, c_{t-1}\}$, as claimed. □

This implies the following:

Theorem 10. *Let G be connected P_t-free graph. Then, for all $k, k' > t$, G has a C_k-coloring if and only if G has a $C_{k'}$-coloring.*

It also leads to the following improvement of Theorem 1.

Theorem 2. *Let $k \geq 9$ or $k = 5$ or 7, and let (G, L) be an instance of* List C_k-Coloring *where G is a P_9-free graph of order n. Then one can determine in $O(n^{111})$ time if (G, L) is colorable, and find such a C_k-coloring if one exists.*

Proof. If $k < 10$, then this follows from Theorem 1. If $k \geq 10$, then Lemma 9 implies that every C_k-coloring of G is contained in an 8-vertex subpath of C_k. Since there are k such subpaths and List P_t-Coloring is polynomial-time solvable for every t even in general graphs [15], the result follows. $\square$

5 Hardness results

In this section we study the complexity of variants of C_k-Coloring in F-free graphs, if F is not a path. First we consider the case of odd k, and then the case of even k.

5.1 Complexity of variants of C_k-Coloring for odd $k \geq 5$

Recall that C_k-Coloring is NP-complete for every odd $k \geq 3$ [27]. In this section we prove the following theorem.

Theorem 3. *Let F be a connected graph. If F is not a subgraph of a subdivided claw, then for every odd $k \geq 5$ the C_k-*Precoloring Extension *problem is NP-complete for F-free graphs.*

We will prove Theorem 3 in several steps in which we analyze possible structure of F. We start with the following simple observation that will be repeatedly used. For the rest of this section, let $k = 2s + 1$ for $s \geq 2$.

Observation 11. *Let $s \geq 2$ be an integer and P be a $2s$-vertex path with endvertices a and b. Then the following holds.*

- *In any C_{2s+1}-coloring h of P we have $h(a) \neq h(b)$.*
- *For any distinct $i, j \in \{1, 2, \ldots, 2s + 1\}$, there exists a C_{2s+1}-coloring h of P such that $h(a) = i$ and $h(b) = j$.*

Eliminate cycles

The *girth* of a graph G, denoted by $\mathrm{girth}(G)$, is the length of a shortest cycle in G. A vertex in a graph is called a *branch vertex* if its degree is at least 3. By Γ_p we denote the class of graphs, in which the number of edges in any path joining two branch vertices is divisible by p.

We first show that the problem is NP-hard in F-free graphs, unless F is a tree in Γ_{2s-1}.

Theorem 12. *For each fixed integer $s \geq 2$ and each connected graph F, C_{2s+1}-*Coloring *is NP-complete for F-free graphs whenever F contains a cycle or is not in Γ_{2s-1}.*

Proof. It is known (see e.g. [35]) that the $(2s+1)$-Coloring problem is NP-complete for graphs of girth at least g for each fixed $g \geq 3$. We reduce this problem to C_{2s+1}-Coloring. Given a graph G, we obtain a graph G' by replacing each edge of G by a $(2s-1)$-edge path. Then it follows from Observation 11 that G is $(2s+1)$-colorable if and only if G' is C_{2s+1}-colorable. Clearly, $\mathrm{girth}(G') = \mathrm{girth}(G) \cdot (2s - 1) \geq g(2s - 1)$. Thus, if we choose $g \geq 3$ such that $g(2s - 1) > \mathrm{girth}(F)$, e.g., $g = |V(F)| + 1$, it follows that all graphs of girth at least $g(2s - 1)$ are F-free. Moreover, it is easy to see that the number of edges in any path joining two branch vertices of G' is divisible by $2s - 1$, so if $F \notin \Gamma_{2s-1}$, then G' does not contain F. $\square$

Eliminate vertices of degree at least 4

From now on it suffices to consider trees with branch vertices at distance divisible by $2s-1$. We now show that C_k-Coloring is NP-complete for F-free graphs if F contains a vertex of degree at least 4. Note that in this case every subcubic graph is F-free.

Theorem 13. *For each fixed $s \geq 2$, C_{2s+1}-Coloring is NP-complete for subcubic graphs.*

Proof. We reduce from C_{2s+1}-Coloring for general graphs. Let G be a graph. We construct an equivalent instance of C_{2s+1}-Coloring with maximum degree 3 as follows. If $\Delta(G) \leq 3$, we are done. Otherwise, let v be a vertex of degree $d \geq 4$, and let $u_1, \ldots, u_d$ be its neighbors. we replace v with a copy of R^d (see Figure 1), denoted by R_v. Each output vertex of R_v is adjacent to a distinct vertex from $u_1, \ldots, u_d$.

Note that in any C_{2s+1}-coloring of R_v, each output vertex must be mapped to the same vertex of C_{2s+1}. Therefore, the obtained graph is C_{2s+1}-colorable if and only if the original one is. Moreover, the number of vertices of degree greater than 3 in the new graph is one less than that in G. Therefore, by repeating this procedure exhaustively, we finally obtain a subcubic graph, which is an equivalent instance of C_{2s+1}-Coloring. □

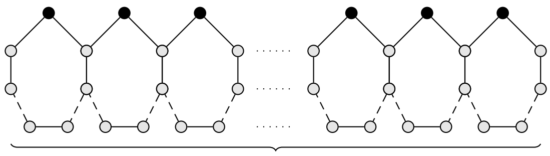


Figure 1: The graph R^d that consists of a chain of d copies of C_{2s+1}. The vertices marked black are called *output vertices.*

Eliminate multiple branch vertices

Before we prove the main theorem we need one more intermediate step that allows us to eliminate those F in which there are two branch vertices that are at distance not divisible by s. The proof is a reduction from the problem called Non-Rainbow Coloring Extension, whose instance is a 3-uniform hypergraph H and a partial coloring f of some of its vertices with colors $\{1, 2, 3\}$. We ask whether f can be extended to a 3-coloring of $V(H)$ such that no hyperedge is *rainbow* (i.e., contains three distinct colors). This problem is known to be NP-complete [4].

Theorem 14. *For each fixed integer $s \geq 2$, C_{2s+1}-Precoloring Extension is NP-complete for bipartite graphs in Γ_s.*

Proof. We reduce from Non-Rainbow Coloring Extension. Let $H = (V, E)$ be a 3-uniform hypergraph and let f be a partial 3-coloring of H. We construct an instance of C_{2s+1}-Precoloring Extension as follows.

- For each vertex $v \in V$, we introduce a variable vertex, denoted by v'. If v is precolored by f, we precolor v' with the color $f(v)$.
- For each v that is not precolored by f, we introduce $2s-2$ new vertices and precolor them with $4, 5, \ldots, 2s+1$, respectively. Then each of these new vertices is joined by a $(2s-1)$-edge path to v'. It follows from Observation 11 that each vertex v' can only be mapped to one of $1, 2, 3$, and any of these three choices is possible.
- For each hyperedge $e = \{x, y, z\} \in E$, we add a new vertex v_e and three s-edge paths connecting v_e to x',y', and z', respectively. This whole subgraph is called an *edge gadget.*

Observe that if x' is mapped to $i \in \{1,2,3\}$, then the possible colors for v_e are $\{s+i, s+i-2, \ldots, s+i-2\lfloor s/2 \rfloor\} \cup \{s+i+1, s+i+3, \ldots, s+i+1+2\lfloor s/2 \rfloor\}$. Thus, if each of x', y', z' is mapped to a different vertex from $\{1,2,3\}$, then there is no way to extend this mapping to the whole edge gadget. On the other hand, such an extension is possible whenever x', y', z' receive at most two distinct colors.

We denote by G the resulting graph. By the properties of variable vertices and edge gadgets, (H, f) is an yes-instance of Non-Rainbow Coloring Extension if and only if the precoloring of G can be extended to a C_{2s+1}-coloring of G. Clearly, G is bipartite and belongs to Γ_s. □

By Theorems 12, 13, and 14, the C_{2s+1}-Precoloring Extension problem is NP-complete for F-free graphs unless F is a tree in $\Gamma_{s(2s-1)}$ (observe that s and $2s-1$ are relatively prime). We are now ready to show that the problem is NP-hard if F has more than one branch vertex.

Theorem 15. *Let $s \geq 2$ be an integer and let F be a tree. If F contains two branch vertices, then C_{2s+1}-Coloring is NP-complete for F-free graphs.*

Proof. Let d be the distance between two closest branch vertices in F. We reduce from Positive Not-All-Equal Sat with all clauses containing exactly three literals. Consider an instance with variables $x_1, x_2, \ldots, x_n$ and clauses $D_1, D_2, \ldots, D_m$.

- We start our construction by introducing one special vertex z.
- For each variable x_i, we introduce a vertex v_i, adjacent to z.
- For each clause $D_\ell = \{x_i, x_j, x_k\}$, we introduce three new vertices $y_{\ell,i}$, $y_{\ell,j}$, and $y_{\ell,k}$, and join each pair of them with a $(2s-1)$-edge path. This guarantees that in every C_{2s+1}-coloring, they get three distinct colors. These three paths constitute the *clause gadget.*
- For each variable x_i belonging to a clause D_ℓ, we join each $y_{\ell,i}$ to v_i by a path $P_{\ell,i}$ with $2d(2s-1)+1$ edges. Let $v_i = p_1, p_2, \ldots, p_{2d(2s-1)+2} = y_{\ell,i}$ be the consecutive vertices of $P_{\ell,i}$. We add edges joining z and $p_{1+j(2s-1)}$ for every $1 \leq j \leq 2d$.

This completes the construction of a graph G. We claim that G is C_{2s+1}-colorable if and only if the initial formula is satisfiable, and that G belongs to our class.

Claim 13. *G is C_{2s+1}-colorable if and only if the initial formula is satisfiable.*

Proof of Claim. Suppose first that the formula has a satisfying assignment σ. We color the vertex z with color 2. If a variable x_i is set true by σ, we color v_i with color 1, otherwise we color v_i with color 3. Let us consider an arbitrary clause $D_\ell = \{x_i, x_j, x_k\}$. We extend this coloring to all paths $P_{\ell,i}$ ($P_{\ell,j}$, $P_{\ell,k}$, accordingly), so that the color of $p_{1+j(2s-1)}$ (for every even $2 \leq j \leq 2d$) is the same as the color of p_1. Therefore, if v_i is colored 1, then the possible colors for $y_{\ell,i}$ are 2 and $2s+1$, and if v_i is colored 3, then the possible colors for $y_{\ell,i}$ are 2 and 4. Since D_ℓ contains at least one true variable and at least one false variable, we can choose three distinct colors for $y_{\ell,i}, y_{\ell,j}$, and $y_{\ell,k}$, and extend this mapping to the remaining vertices.

Suppose now that there exists a C_{2s+1}-coloring f of G. By symmetry, we may assume that $f(z) = 2$. This implies that every v_i is colored by 1 or 3. We define the assignment: $\sigma(x_i)$ is true if $f(v_i) = 1$ and false otherwise. Suppose that σ is not satisfying, i.e., there is a clause D_ℓ with literals x_i, x_j, x_k that all have the same value. It follows that $f(v_i) = f(v_j) = f(v_k)$ and without loss of generality, we may assume that $f(v_i) = 1$. Observe that for every even $2 \leq j \leq 2d$ we have $f(p_{1+j(2s-1)}) = f(v_i) = 1$, where p_t's are consecutive vertices of $P_{\ell,i}$. This implies that $f(y_{\ell,i}) \in \{2, 2s+1\}$. Similarly, $f(y_{\ell,j}), f(y_{\ell,k}) \in \{2, 2s+1\}$. It follows that there are two of i, j, k, say i and j, such that $f(y_{\ell,i}) = f(y_{\ell,j})$. But these two vertices are connected by a path with $2s-1$ edges, which contradicts Observation 11. ■

Moreover, the constructed graph belongs to our class.

Claim 14. *G is F-free.*

Proof of Claim. Let a and b two branch vertices in F, they are at distance d. Assume by contradiction that G contains an induced copy of F. Let $g : V(F) \to V(G)$ map every vertex of F to its corresponding vertex in an induced copy of F in G.

Suppose first that $z \in g(V(F))$. Since d is divisible by $s(2s-1)$, it follows that a and b have distance at least 6 in F. Therefore, for every vertex $u \in V(F)$, there is a branch vertex in $F \setminus (N(u) \cup \{u\})$. Now let $u \in V(F)$ such that $g(u) = z$. Then $F \setminus (N(u) \cup \{u\})$ is an induced subgraph of $G \setminus (N(z) \cup \{z\})$, but the latter graph has maximum degree two, a contradiction. It follows that $z \notin g(V(F))$, and so F is an induced subgraph of $G' = G \setminus \{z\}$.

Let us now consider the possible values of $g(a)$ and $g(b)$. Since a and b have degree at least 3 in F, it follows that $g(a)$ and $g(b)$ have degree at least 3 in G'. Moreover, since a and b are at distance d in F, it follows that $g(a)$ and $g(b)$ are at distance at most d in G'.

Case 1. $g(a) = v_i$ or $g(b) = v_i$ for some i. Every vertex $u \neq v_i$ of degree at least three in G' has distance at least $2d(2s-1)+1 > d$ from v_i, a contradiction.

Case 2. $g(a) = y_{\ell,i}$ for some i and ℓ. By the first case, it follows that $g(b) = y_{\ell',j}$ for some j and ℓ'. Let Q be the a-b-path in F. Since Q has d edges, and since the number of edges of every path in G' between $y_{\ell,i}$ and $y_{\ell',j}$ for $\ell \neq \ell'$ is more than d, it follows that $\ell = \ell'$ and $g(V(Q))$ is contained in the clause gadget for D_ℓ.

Suppose first that $s \geq 3$. Since d is divisible by $s(2s-1)$, it follows that Q has $d+1 \geq 3(2s-1)+1$ vertices. However, the clause gadget for D_ℓ has $3(2s-1)$ vertices, a contradiction.

Therefore, $s = 2$. Then the clause gadget for D_ℓ is isomorphic to a nine-cycle in G' with vertices $c_1, \ldots, c_9$ in this order, say. By symmetry, we may assume that $g(a) = c_1$ and $g(b) = c_4$. Since d is divisible by $s(2s-1)$ and $s = 2$, it follows that $d = 6$, and so $g(V(Q)) = \{c_1, c_4, c_5, c_6, c_7, c_8, c_9\}$. Since F is a tree, it follows that either $c_2 \notin g(V(F))$ or $c_3 \notin g(V(F))$. By symmetry, we may assume that $c_2 \notin g(V(F))$. But c_1 has degree two in $G' \setminus \{c_2\}$, a contradiction. This concludes the proof of the claim. ■

This completes the proof of Theorem 15. □

Now Theorem 3 comes from combining the Theorems 12, 13, 14, and 15.

5.2 Complexity of variants of C_k-Coloring for even k

Recall that that in case of even k, the C_k-Coloring problem is polynomial-time solvable for general graphs [27]. However, this is no longer true in the case of List C_k-Coloring: the problem is polynomial-time solvable for $k = 4$ and NP-complete for all even $k \geq 6$ [15]. For the rest of this section $k = 2s$, where $s \geq 3$. The consecutive vertices of C_{2s} are denoted by $\{1, 2, \ldots, 2s\}$ (with $2s$ adjacent to 1).

In this section we prove Theorem 4.

Theorem 4. *Let F be a connected graph. If F is not a subgraph of a subdivided claw, then for every even $k \geq 6$ the* List C_k-Coloring *problem is NP-complete for F-free graphs.*

To be more specific, we will prove the following.

Theorem 16. *Let g be a fixed integer. For every even $k \geq 6$ the* List C_k-Coloring *problem is NP-complete for subcubic graphs with girth at least g, in which no pair of branch vertices is at distance at least g.*

Proof. We will reduce from Monotone 3-Sat, a variant of 3-Sat, in which every clause contains only positive or only negative literals. It is known that this problem is NP-complete, even if each variable appears at most three times [11].

For every variable v_i we introduce a *variable vertex* x_i with list $\{1, 3\}$. Color 1 will correspond to true assignment, while 3 will denote false. For every clause D_ℓ, we introduce a *clause vertex* d_i with list $\{1, 3, 5\}$.

Now we need to connect variable vertices with clause vertices. For $i = \{1, 3, 5\}$ we will construct a path $Q^{(i)}$, starting in a vertex a^i and ending in a vertex b^i, with appropriately chosen lists, so that the following are satisfied:

1. for $i \in \{1, 3, 5\}$, if a^i is colored 1, then for every $c \in \{1, 3, 5\}$ there is a list C_{2s}-coloring of $Q^{(i)}$ in which the color of b^i is c,
2. for $i \in \{1, 3, 5\}$ and $c \in \{1, 3, 5\} \setminus \{i\}$ there is a list C_{2s}-coloring of $Q^{(i)}$ in which the color of a^i is 3 and the color of b^i is c,
3. for $i \in \{1, 3, 5\}$ and $c \in \{1, 3, 5\} \setminus \{i\}$ there is no list C_{2s}-coloring of $Q^{(i)}$ in which a^i is colored 3 and b^i is colored i.

Additionally, we will make sure that each path has more than g vertices.

Suppose for now that we have constructed such paths. Consider a clause with three positive literals, $D_\ell = (v_p, v_q, v_r)$ and let the ordering of variables be fixed. We introduce a copy of each $Q^{(i)}$ for $i \in \{1, 3, 5\}$. We identify the vertex a^1 with x_p, the vertex a^3 with x_q, and the vertex a^5 with x_r. Finally, we identify the vertices b^1, b^3, b^5, and d_ℓ. In case of a clause D_ℓ with two positive literals, we use only paths $Q^{(1)}$ and $Q^{(3)}$, and remove the color 5 from the list of d_ℓ.

It is clear that the paths ensure that in order to find a color for d_ℓ, at least one of corresponding variable vertices must be colored 1, meaning that one of the variables in D_ℓ is true.

In order to deal with clauses with negative literals, for $i = \{1, 3, 5\}$ we will construct a path $\overline{Q}^{(i)}$, starting in a vertex $\overline{a}^i$ and ending in a vertex $\overline{b}^i$, with appropriately chosen lists, so that the following are satisfied:

1. for $i \in \{1, 3, 5\}$, if $\overline{a}^i$ is colored 3, then for every $c \in \{1, 3, 5\}$ there is a list C_{2s}-coloring of $\overline{Q}^{(i)}$ in which the color of $\overline{b}^i$ is c,
2. for $i \in \{1, 3, 5\}$ and $c \in \{1, 3, 5\} \setminus \{i\}$ there is a list C_{2s}-coloring of $\overline{Q}^{(i)}$ in which the color of $\overline{a}^i$ is 1 and the color of $\overline{b}^i$ is c,
3. for $i \in \{1, 3, 5\}$ and $c \in \{1, 3, 5\} \setminus \{i\}$ there is no list C_{2s}-coloring of $\overline{Q}^{(i)}$ in which $\overline{a}^i$ is colored 1 and $\overline{b}^i$ is colored i.

Now, for a clause $D_\ell = (\neg v_p, \neg v_q, \neg v_r)$, we introduce a copy of each $\overline{Q}^{(i)}$ for $i \in \{1, 3, 5\}$, and identify the vertex $\overline{a}^1$ with x_p, the vertex $\overline{a}^3$ with x_q, the vertex $\overline{a}^5$ with x_r, and finally all vertices $\overline{b}^1, \overline{b}^3, \overline{b}^5$, and d_ℓ. Similarly, for a clause $D_\ell = (\neg x_p, \neg x_q)$ we use paths $\overline{Q}^{(1)}$ and $\overline{Q}^{(2)}$ and remove the color 5 from the list of d_ℓ.

It is straightforward to observe that the constructed graph G admits a list C_{2s}-coloring if and only if the initial Monotone 3-Sat formula is satisfiable.

Now let us argue that is belongs to the considered class. Note that the only branch vertices in G are variable vertices and clause vertices. Since each clause contains at most three variables and each variable appears is at most 3 clauses, we conclude that G is subcubic. Finally, since all paths joining variable vertices with clause vertices have more than g vertices, we conclude G has no cycle of length at most g.

Thus in order to complete the proof, we need to show how to construct $Q^{(i)}$'s and $\overline{Q}^{(i)}$'s. Let us assume that g is even (we can do it safely, as we can always replace g with $g + 1$ and the claim will still hold). The lists of consecutive vertices on $Q^{(1)}$ are as follows (starting from a^1):

$$\underbrace{\{1, 3\}, \{2s, 4\}, \{1, 3\}, \{2s, 4\}, \ldots, \{1, 3\}, \{2s, 4\}}_{g}, \{1, 3\}, \{2s, 2\}, \{2s - 1, 3\}, \{2s, 4\}, \{1, 3, 5\}.$$

The lists of consecutive vertices on $Q^{(3)}$ are as follows (starting from a^3):

$$\underbrace{\{1,3\},\{2s,4\},\{1,3\},\{2s,4\},\ldots,\{1,3\},\{2s,4\}}_{g},\{1,3\},\{2s,2\},\{2s-1,1\},\{2s-2,2\},\ldots,$$
$$\{4,2\},\{3,1\},\{4,2s\},\{1,3,2s-1\},\{4,2s-2,2s\},\{1,3,2s-3\},\ldots,\{4,6,2s\},\{1,3,5\}$$

The lists of consecutive vertices on $Q^{(5)}$ are as follows (starting from a^5):

$$\underbrace{\{1,3\},\{2s,4\},\{1,3\},\{2s,4\},\ldots,\{1,3\},\{2s,4\}}_{g},\{1,3\},\{2s,2\},\{2s-1,3\},\ldots,\{6,2\},\{1,3,5\}.$$

It is straightforward to verify that they satisfied the required conditions. The construction of the paths $\overline{Q}^{(i)}$ for $i=\{1,3,5\}$ is analogous, with the roles of 1 and 3 switched. This completes the proof. $\square$

Now let F be a connected graph that is not a subdivided claw. Theorem 4 follows by applying Theorem 16 with $g=|F|+1$.

5.3 Subexponential algorithms and ETH-based lower bounds

Recall that by the result of Groenland *et al.* [23], for every fixed t and k, the List C_k-Coloring problem can be solved in time $2^{O(\sqrt{n \log n})}$ for P_t-free graphs.

It turns out that such subexponential algorithms for variants of C_k-Coloring are unlikely to exist for F-free graphs, if F is not a subgraph of a subdivided claw. All reductions in our hardness proofs are linear in the number of vertices (recall that the target graph is assumed to be fixed). Moreover, all problems we are reducing from can be shown to be NP-complete by a linear reduction from 3-Sat. Thus we get the following results, conditioned on the Exponential Time Hypothesis (ETH), which, along with the *sparsification lemma*, implies that 3-Sat with n variables and n clauses cannot be solved in time $2^{o(n+m)}$ [31, 32].

Corollary 17. *Unless the ETH fails, the following holds. If F is a connected graph that is not a subgraph of a subdivided claw, then for every $s \geq 2$, the problems*

a) C_{2s+1}-Precoloring Extension *and*

b) List C_{2s+2}-Coloring

cannot be solved in time $2^{o(n)}$ in F-free graphs with n vertices.

6 Conclusion

In this paper, we initiate a study of variants C_k-Coloring for F-free graphs for a fixed graph F. We show that List C_k-Coloring is polynomial-time solvable for P_9-free graphs, whenever $k=5$ or $k=7$ or $k\geq 9$. Moreover, we prove that for every $s \geq 2$ the C_{2s+1}-Precoloring Extension and List C_{2s+2}-Coloring problems are NP-complete for F-free graphs if some component of F is not a subdivided claw. Note that for the case of odd cycles, all our hardness results work for C_{2s+1}-Coloring, except for Theorem 14. Thus it is natural to ask whether analogous hardness result holds for C_{2s+1}-Coloring too.

Moreover, the following questions seem natural to explore.

- Are there values of s and t such that C_{2s+1}-Coloring is NP-complete for P_t-free graphs?
- Are there values of k and t such that List C_k-Coloring is NP-complete for P_t-free graphs?

- Is C_{2s+1}-Coloring polynomial for F-free graphs when F is a subdivided claw?

We also believe that it would be interesting study the complexity of C_k-Precoloring Extension for even k. It is known that this problem is NP-complete for every $k \geq 6$ [15].

Let us point out that, quite surprisingly, C_6-Precoloring Extension is polynomial-time solvable for graphs with maximum degree 3 [17]. Since every graph that contains a triangle is clearly a no-instance of C_6-Precoloring Extension, we conclude that the problem is polynomial-time solvable for $K_{1,4}$-free graphs.

On the other hand, it is known that C_6-Precoloring Extension is NP-complete for graphs with maximum degree 4, and for every even $k \geq 8$, C_k-Precoloring Extension is NP-complete for graphs with maximum degree 3 [17].

Finally, note that the list of each vertex v in the construction in the proof of Theorem 4 can be simulated by introducing some number of precolored vertices and joining them to v with paths of certain length. Since the newly introduced vertices are private for every original vertex, we do not introduce any new cycles, thus the constructed graph still has large girth. Thus we obtain the following.

Corollary 18. *Let F be a connected graph and $k \geq 8$ be an even integer. The following problems are NP-complete for F-free graphs:*

*a) the C_6-*Precoloring Extension *problem, if F is not a tree with $\Delta(F) \leq 4$,*

*b) the C_k-*Precoloring Extension *problem, if F is not a tree with $\Delta(F) \leq 3$.*

Let us also point out that the reduction described above introduces many further constraints on F. Moreover, if we modify the construction so that the precolored vertices used for simulating lists are not private, but shared by original vertices, further constrains can be introduced. However, the description of forbidden subgraphs is not elegant and we believe that it can be further improved.

References

[1] Vladimir E Alekseev. The effect of local constraints on the complexity of determination of the graph independence number. *Combinatorial-algebraic methods in applied mathematics*, pages 3–13, 1982.

[2] Bengt Aspvall, Michael F. Plass, and Robert Endre Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Inf. Process. Lett.*, 8(3):121–123, 1979.

[3] Gábor Bacsó, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Zsolt Tuza, and Erik Jan van Leeuwen. Subexponential-time algorithms for maximum independent set in P_t-free and broom-free graphs. *Algorithmica*, 81(2):421–438, 2019.

[4] Manuel Bodirsky, Jan Kára, and Barnaby Martin. The complexity of surjective homomorphism problems – a survey. *Discrete Applied Mathematics*, 160(12):1680 – 1690, 2012.

[5] Flavia Bonomo, Maria Chudnovsky, Peter Maceli, Oliver Schaudt, Maya Stein, and Mingxian Zhong. Three-coloring and list three-coloring of graphs without induced paths on seven vertices. *Combinatorica*, 38(4):779–801, 2018.

[6] Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 319–330. IEEE Computer Society, 2017.

[7] Eglantine Camby and Oliver Schaudt. A new characterization of P_k-free graphs. *Algorithmica*, 75(1):205–217, 2016.

[8] Maria Chudnovsky, Sophie Spirkl, and Mingxian Zhong. Four-coloring P_6-free graphs. I. extending an excellent precoloring. *CoRR*, abs/1802.02282, 2018.

[9] Maria Chudnovsky, Sophie Spirkl, and Mingxian Zhong. Four-coloring P_6-free graphs. II. finding an excellent precoloring. *CoRR*, abs/1802.02283, 2018.

[10] Maria Chudnovsky, Sophie Spirkl, and Mingxian Zhong. Four-coloring P_6-free graphs. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1239–1256. SIAM, 2019.

[11] Andreas Darmann, Janosch Döcker, and Britta Dorn. The monotone satisfiability problem with bounded variable appearances. *International Journal of Foundations of Computer Science*, 29(06):979–993, 2018.

[12] Keith Edwards. The complexity of colouring problems on dense graphs. *Theor. Comput. Sci.*, 43:337–343, 1986.

[13] Jessica Enright, Lorna Stewart, and Gábor Tardos. On list coloring and list homomorphism of permutation and interval graphs. *SIAM J. Discrete Math.*, 28(4):1675–1685, 2014.

[14] Tomas Feder and Pavol Hell. List homomorphisms to reflexive graphs. *J. Comb. Theory, Ser. B*, 72(2):236 – 250, 1998.

[15] Tomás Feder, Pavol Hell, and Jing Huang. List homomorphisms and circular arc graphs. *Combinatorica*, 19(4):487–505, 1999.

[16] Tomás Feder, Pavol Hell, and Jing Huang. Bi-arc graphs and the complexity of list homomorphisms. *Journal of Graph Theory*, 42(1):61–80, 2003.

[17] Tomás Feder, Pavol Hell, and Jing Huang. Extension problems with degree bounds. *Discrete Applied Mathematics*, 157(7):1592–1599, 2009.

[18] Tomás Feder, Pavol Hell, Sulamita Klein, Loana Tito Nogueira, and Fábio Protti. List matrix partitions of chordal graphs. *Theor. Comput. Sci.*, 349(1):52–66, 2005.

[19] Tomás Feder and Moshe Y. Vardi. Monotone monadic SNP and constraint satisfaction. In S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal, editors, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA*, pages 612–622. ACM, 1993.

[20] Anna Galluccio, Pavol Hell, and Jaroslav Nesetril. The complexity of H-colouring of bounded degree graphs. *Discrete Mathematics*, 222(1-3):101–109, 2000.

[21] Petr A. Golovach, Matthew Johnson, Daniël Paulusma, and Jian Song. A survey on the computational complexity of coloring graphs with forbidden subgraphs. *Journal of Graph Theory*, 84(4):331–363, 2017.

[22] Petr A. Golovach, Daniël Paulusma, and Jian Song. Closing complexity gaps for coloring problems on H-free graphs. *Inf. Comput.*, 237:204–214, 2014.

[23] Carla Groenland, Karolina Okrasa, Paweł Rzążewski, Alex Scott, Paul Seymour, and Sophie Spirkl. H-colouring pt-free graphs in subexponential time. *Discrete Applied Mathematics*, 267:184 – 189, 2019.

[24] Andrzej Grzesik, Tereza Klimošová, Marcin Pilipczuk, and Michal Pilipczuk. Polynomial-time algorithm for maximum weight independent set on P_6-free graphs. *CoRR*, abs/1707.05491, 2017.

[25] Andrzej Grzesik, Tereza Klimošova, Marcin Pilipczuk, and Michal Pilipczuk. Polynomial-time algorithm for maximum weight independent set on P_6-free graphs. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1257–1271. SIAM, 2019.

[26] Pavol Hell and Jaroslav Nesetril. *Graphs and Homomorphisms*. Oxford University Press, jul 2004.

[27] Pavol Hell and Jaroslav Nešetřil. On the complexity of H-coloring. *J. Comb. Theory, Ser. B*, 48(1):92–110, 1990.

[28] Chính T. Hoàng, Marcin Kamiński, Vadim V. Lozin, Joe Sawada, and Xiao Shu. Deciding k-colorability of P_5-free graphs in polynomial time. *Algorithmica*, 57(1):74–81, 2010.

[29] Ian Holyer. The NP-completeness of edge-coloring. *SIAM J. Comput.*, 10(4):718–720, 1981.

[30] Shenwei Huang. Improved complexity results on k-coloring P_t-free graphs. *Eur. J. Comb.*, 51:336–346, 2016.

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

[32] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

[33] Jan Kratochvíl. Precoloring extension with fixed color bound. *Acta Mathematica Universitatis Comenianae. New Series*, 62, 01 1993.

[34] Benoit Larose and Adrien Lemaître. List-homomorphism problems on graphs and arc consistency. *Discrete Mathematics*, 313(22):2525–2537, 2013.

[35] Vadim V. Lozin and Marcin Kamiński. Coloring edges and vertices of graphs without short or long cycles. *Contributions to Discrete Mathematics*, 2(1), 2007.

[36] Gerhard J. Woeginger and Jirí Sgall. The complexity of coloring graphs without long induced paths. *Acta Cybern.*, 15(1):107–117, 2001.

[37] Dmitriy Zhuk. A proof of CSP dichotomy conjecture. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 331–342. IEEE Computer Society, 2017.