On the parameterized complexity of coloring graphs in the absence of a linear forest *

Jean-François Couturier[1], Petr A. Golovach[2], Dieter Kratsch[1], and Daniël Paulusma[2] **

[1]Laboratoire d'Informatique Théorique et Appliquée,
Université Paul Verlaine - Metz, 57045 Metz Cedex 01, France
{couturier,kratsch}@univ-metz.fr
[2]School of Engineering and Computing Sciences, Durham University,
Science Laboratories, South Road, Durham DH1 3LE, United Kingdom
{petr.golovach,daniel.paulusma}@durham.ac.uk

Abstract. The k-Coloring problem is to decide whether a graph can be colored with at most k colors such that no two adjacent vertices receive the same color. The List k-Coloring problem requires in addition that every vertex u must receive a color from some given set $L(u) \subseteq \{1, \ldots, k\}$. Let P_n denote the path on n vertices, and $G + H$ and rH the disjoint union of two graphs G and H and r copies of H, respectively. We show that List k-Coloring is fixed-parameter tractable for graphs with no induced $rP_1 + P_2$ when parameterized by $k + r$, and that k-Coloring restricted to such graphs allows a polynomial kernel when parameterized by k. Finally, we show that List k-Coloring is fixed-parameter tractable for graphs with no induced $P_1 + P_3$ when parameterized by k.

1 Introduction

Graph coloring involves the labeling of the vertices of some given graph by integers called colors such that no two adjacent vertices receive the same color. The corresponding k-Coloring problem is to decide whether a graph can be colored with at most k colors. Because k-Coloring is NP-complete for any fixed $k \geq 3$, there has been considerable interest in studying its complexity when restricted to certain graph classes. One of the most well-known results in this respect is due to Grötschel, Lovász, and Schrijver [11] who show that k-Coloring is polynomial-time solvable for perfect graphs. More information on this classic result and on the general motivation, background and related work on coloring problems restricted to special graph classes can be found in several surveys [24, 26] on this topic. In this paper we consider graph classes defined by a forbidden induced

* The results of this paper can also be found in an extended abstract that was presented at the 37th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2011).

** This work has been supported by ANR Blanc AGAPE (ANR-09-BLAN-0159-03) and EPSRC (EP/G043434/1).

subgraph, and focus on the parameterized complexity, in contrast to previous papers on this topic [3–6, 8, 10, 12, 15–19, 21, 23, 27]. Before we summarize these results and explain our new results, we first state the necessary terminology and notations.

1.1 Terminology

We only consider finite undirected graphs $G = (V, E)$ without loops and multiple edges. We sometimes denote the vertex set of G by V_G. We write $G[U]$ to denote the subgraph of G *induced by* the vertices in U, i.e., the subgraph of G with vertex set U and an edge between two vertices $u, v \in U$ if and only if $uv \in E$. A subset $D \subseteq V$ is a *dominating* set of G if every vertex in G belongs to D or is adjacent to a vertex of D. In that case we also say that $G[D]$ is *dominating.* A subset $X \subseteq V$ is *independent* if there is no edge between any two vertices of X.

We refer to the textbook by Bondy and Murty [2] for any undefined graph terminology.

The graph P_n denotes the path on n vertices. The disjoint union of two graphs G and H is denoted $G + H$, and the disjoint union of r copies of G is denoted rG. A *linear forest* is the disjoint union of a collection of paths. Let H be a graph. We say that a graph G is *H-free* if G has no induced subgraph isomorphic to H.

A *(vertex) coloring* of a graph $G = (V, E)$ is a mapping $\phi : V \to \{1, 2, \ldots\}$ such that $\phi(u) \neq \phi(v)$ whenever $uv \in E$. Here, $\phi(u)$ is referred to as the *color* of u. A *k-coloring* of G is a coloring ϕ of G with $\phi(V) \subseteq \{1, \ldots, k\}$. Here, we used the notation $\phi(U) = \{\phi(u) \mid u \in U\}$ for $U \subseteq V$. If G has a k-coloring, then G is called *k-colorable.* Recall that the problem k-Coloring is to decide whether a given graph admits a k-coloring. Here, k is *fixed*, i.e., not part of the input. If k is part of the input then we denote the problem as Coloring. The optimization version of this problem is to determine the *chromatic number* of a graph, i.e., the smallest k such that G has a k-coloring. A *list assignment* of a graph $G = (V, E)$ is a function L that assigns a list $L(u)$ of so-called *admissible* colors to each $u \in V$. If $L(u) \subseteq \{1, \ldots, k\}$ for $u \in V$, then L is also called a *k-list assignment.* Equivalently, L is a k-list assignment if $|\bigcup_{u\in V} L(u)| \leq k$. We say that a coloring $\phi : V \to \{1, 2, \ldots\}$ *respects* L if $\phi(u) \in L(u)$ for all $u \in V$. For a fixed integer k, the List k-Coloring problem has as input a graph G with a k-list assignment L and asks whether G has a coloring that respects L.

We finish this section with a short introduction into parameterized complexity; for a more in-depth discussion we refer to Downey and Fellows [9] and Niedermeier [22]. In parameterized complexity theory, we consider the problem input as a pair (I, k), where I is the main part and k the parameter. The complexity class XP consists of parameterized decision problems Π such that for each instance (I, k) it can be decided in $\mathcal{O}(f(k)|I|^{g(k)})$ time whether $(I, k) \in \Pi$, where f and g are computable functions depending only on the *parameter* k, and $|I|$ denotes the size of I. So XP consists of parameterized decision problems which can be solved in polynomial time if the parameter is a constant. A problem is *fixed-parameter tractable* if an instance (I, k) can be solved in time $O(f(k)|I|^c)$, where f denotes a computable function and c a constant independent of k. The class $\mathsf{FPT} \subseteq \mathsf{XP}$ is the class of all fixed-parameter tractable decision problems.

A well-known technique to show that a parameterized problem Π is fixed-parameter tractable is to find a *reduction to a problem kernel*. This technique replaces an instance (I, k) of Π with a reduced instance (I', k') of Π called a *(problem) kernel*, such that the following three conditions hold:

(i) $k' \leq k$ and $|I'| \leq g(k)$ for some computable function g;
(ii) the reduction from (I, k) to (I', k') is computable in polynomial time;
(iii) (I, k) is a `Yes`-instance of Π if and only if (I', k') is a `Yes`-instance of Π.

An upper bound $g(k)$ of $|I'|$ is called the *kernel size*, and a kernel is called *polynomial* if the kernel size is polynomial in k. It is well known that a parameterized problem is fixed-parameter tractable if and only if it is kernelizable (cf. [22]).

1.2 Related work

Král', Kratochvíl, Tuza and Woeginger [17] completely determined the computational complexity of Coloring for graph classes characterized by a forbidden induced subgraph and achieved the following dichotomy.

Theorem 1 ([17]). *Let H be a fixed graph. If H is a (not necessarily proper) induced subgraph of P_4 or of P_1+P_3, then* Coloring *can be solved in polynomial time for H-free graphs; otherwise it is* NP*-complete for H-free graphs.*

Theorem 1 justifies a study into the computational complexity of the k-Coloring problem for H-free graphs. Combining results of Holyer [13], Kamiński and Lozin [15] and Leven and Galil [20] imply the following theorem (cf. [10]).

Theorem 2. *For any $k \geq 3$, the k-*Coloring *problem is* NP*-complete for the class of H-free graphs whenever H is not a linear forest.*

We now consider the case when H is a linear forest. It is known that 4-Coloring is NP-complete for P_8-free graphs [4] and that 6-Coloring is NP-complete for P_7-free graphs [3]. On the positive side, combining results of the papers by Broersma et al. [4], Couturier et al. [7], Hoàng et al. [12], and Randerath and Schiermeyer [23] leads to the following theorem (cf. [10]).

Theorem 3. *The k-*Coloring *problem can be solved in polynomial time for H-free graphs if*

- $H = rP_1 + P_2 + P_4$ *for all $k \leq 3$ and all $r \geq 0$*
- $H = rP_1 + P_6$ *for all $k \leq 3$ and all $r \geq 0$*
- $H = rP_3$ *for all $k \leq 3$ and all $r \geq 0$*
- $H = P_2 + P_3$ *for all $k \leq 4$*
- $H = rP_1 + P_5$ *for all $k \geq 0$ and all $r \geq 0$*
- $H = rP_2$ *for all $k \geq 0$ and all $r \geq 0$.*

In Theorem 3 we also used the known result that k-Coloring is polynomial-time solvable on sP_2-free graphs for any two integers k and s by combining a result of Balas and Yu [1] on the maximal number of independent sets in an sP_2-free graph with a result from Tsukiyama et al. [25] on the enumeration of such sets. Moreover, we note that k-Coloring is polynomial-time solvable on H'-free graphs whenever it is so on H-free graphs for some graph H containing H' as an induced subgraph.

As a matter of fact, all cases in Theorem 3 also hold for the List k-Coloring problem except in the case when $H = P_2 + P_3$ and $k = 4$. The computational complexity of List 4-Coloring for $(P_2 + P_3)$-free graphs is still open. Also all aforementioned NP-completeness results for k-Coloring on H-free graphs carry over to List k-Coloring. In addition, it is known that List 5-Coloring is NP-complete for P_6-free graphs [3] and for $(P_2 + P_4)$-free graphs [7].

1.3 Our results

The aim of our paper is to initiate a parameterized complexity study for the k-Coloring and List k-Coloring problem restricted to H-free graphs, when H is some fixed linear forest, in order to obtain a more subtle classification for those graphs H, for which these problems are NP-complete. We prove the following three results:

(i) List k-Coloring is fixed-parameter tractable for $(rP_1 + P_2)$-free graphs when parameterized by $k + r$;
(ii) k-Coloring restricted to $(rP_1 + P_2)$-free graphs allows a polynomial kernel when parameterized by k;
(iii) List k-Coloring is fixed-parameter tractable for $(P_1 + P_3)$-free graphs when parameterized by k.

2 $(\mathbf{r}\boldsymbol{P_1} + \boldsymbol{P_2})$-free graphs

First we consider $(rP_1 + P_2)$-free graphs. Theorem 1 tells us that already Coloring is NP-complete for $(rP_1 + P_2)$-free graphs whenever $r \geq 2$. For a graph $G = (V, E)$, we let $N(u) = \{v \in V \mid uv \in E\}$ denote the set of neighbors of a vertex $u \in V$, $N(S) = \{v \in V \setminus S \mid uv \in E \text{ for some } u \in S\}$ denotes the set of neighbors of a set $S \subseteq V$, and $N[S] = N(S) \cup S$.

We use the following lemma.

Lemma 1 ([7]). *Let G be an $(rP_1 + P_\ell)$-free graph for integers $r \geq 1$ and $\ell \geq 1$. If G contains an induced P_ℓ, then G contains a dominating induced $sP_1 + P_\ell$ for some $s < r$.*

We also need the following lemma.

Lemma 2. *Let $G = (V, E)$ be an $(rP_1 + P_2)$-free graph for some $r \geq 0$. If S is an independent set with $|S| \geq r$, then $X = V \setminus N(S)$ is a maximal independent set. Moreover, X is the unique maximal independent set containing S.*

Proof. Suppose that S is an independent set with at least r vertices. Let $X = V \setminus N(S)$. Because G is $(rP_1 + P_2)$-free and S is independent, $V \setminus N[S]$ is independent as well. Because a neighbor of a vertex of S does not belong to any independent set that contains S, we find that X is the unique maximal independent set containing S. This completes the proof of Lemma 2. □

Let G be a graph with a k-list assignment L. Let $\mathcal{G} = \{G_1, \ldots, G_p\}$ be a set of graphs, where each G_i has a $(k-1)$-list assignment L_i. Then we say that G and $\mathcal{G}$ are $(k-1)$-*compatible* if the following holds: G has a coloring respecting L if and only if there exists a graph $G_i \in \mathcal{G}$ that has a coloring respecting L_i.

We now prove the following two lemmas.

Lemma 3. *Let $k \geq 2$ and $r \geq 1$. Let $G = (V, E)$ be an $(rP_1 + P_2)$-free graph on n vertices with a k-list assignment L. If G has a maximal independent set X with at least $(r-1)k + 1$ vertices, then it is possible to find in $O(k^2 n)$ time a $(k-1)$-compatible set $\mathcal{G}$ that consists of at most k induced subgraphs of G.*

Proof. Let X be a maximal independent set with at least $(r-1)k + 1$ vertices. We perform the following procedure for every color $1 \leq i \leq k$.

1 For each vertex $v \in X$, we check whether $i \in L(v)$. If so, then we color v by i and delete v afterwards. If not, we set $L_i(v) = L(v)$.
2 For each vertex $v \in V \setminus X$, we set $L_i(v) = L(v) \setminus \{i\}$.

In this way we compute a set $\mathcal{G} = \{G_1, \ldots, G_k\}$ of at most k induced subgraphs of G, where each G_i has a $(k-1)$-list assignment L_i. The running time of this procedure is $O(k^2 n)$. We are left to show that G and $\mathcal{G}$ are $(k-1)$-compatible.

First suppose that G has a coloring respecting L. Then at least r vertices in X must have the same color. Suppose that this color is i, and let S be the set of all vertices of X colored by i. Because S is an independent set with at least r vertices and X is a maximal independent set containing S, we find that $X = V \setminus N(S)$ due to Lemma 2. Because every vertex in S has color i, every vertex in $N(S)$ cannot be colored with color i. This means that we can remove color i from the list of every vertex in $N(S)$. Then every vertex $v \in X \setminus N[S]$ with $i \in L(v)$ can safely be recolored by i if it was not colored by i already. So, in the end, every vertex in X with color i in its list gets color i, and we have removed i from the list of every vertex not in X. This means that we obtain the subgraph G_i after deleting all vertices with color i from G.

To prove the reverse implication, suppose that $\mathcal{G}$ contains a graph G_i that has a coloring respecting L_i. By construction of G_i, there is no vertex of G_i that has color i in its list. Hence, color i is not used on G_i. Because we only deleted vertices from G that were independent and that had color i in their list, we can safely color these deleted vertices by color i. In this way we obtain a coloring of G that respects L. □

Lemma 4. *Let $k \geq 2$ and $r \geq 1$. Let G be an $(rP_1 + P_2)$-free graph with $n \geq (r+1)^{k-1}((r-1)k + 1) + (r+1)\frac{(r+1)^{k-1}-1}{r}$ vertices and m edges. Then* *either G has a clique of size $k+1$ or a maximal independent set X of size at least $(r-1)k+1$. Moreover, it is possible to find such a clique or independent set in $O(k(n+m))$ time.*

Proof. The case $m = 0$ is trivial. Suppose that $m \geq 1$. We apply Lemma 1 for $\ell = 2$. This yields a dominating $sP_1 + P_2$ of G for some $s < r$. The vertex set of this subgraph is a dominating set of G that has size $s+2 \leq r+1$. Hence, it contains a vertex v of degree at least $\frac{n-(r+1)}{r+1} \geq (r+1)^{k-2}((r-1)k+1)+(r+1)\frac{(r+1)^{k-2}-1}{r}$. We can apply the same arguments inductively for the subgraph of G induced by $N(v)$. Then after at most $k-1$ steps we either obtain a clique of size $k+1$, or else we find that we cannot apply Lemma 1 any longer. The latter case means that the graph under consideration has no edges. Then its vertices form an independent set Y of size at least $(r-1)k+1 \geq r$, as $k \geq 2$. Hence, $X = V \setminus N(Y)$ is a maximal independent set due to Lemma 2. By the same lemma, X contains Y, and as such X has size at least $(r-1)k+1$.

Because a vertex of maximum degree can be found in $O(n+m)$ time, the total running time of our procedure is $O(k(n+m))$. This completes the proof of Lemma 4. □

Now we are ready to prove the following result.

Theorem 4. *The* List k-Coloring *problem is in* FPT *for* $(rP_1 + P_2)$*-free graphs when parameterized by* k *and* r.

Proof. Let G be an $(rP_1 + P_2)$-free graph on n vertices that has a k-list assignment L. If $k \leq 2$, then we can solve the problem in polynomial time. If $n < f(k,r) = (r+1)^{k-1}((r-1)k+1) + (r+1)\frac{(r+1)^{k-1}-1}{r}$, then we can solve it in $O(f(k,r)^k)$ time by brute force. Otherwise, by Lemma 4, we either find a clique of size $k+1$ or a maximal independent set of size at least $(r-1)k+1$ in $O(k(n+m))$ time. In the first case, G has no coloring respecting L. In the second case, we construct in $O(k^2 n)$ time a $(k-1)$-compatible set $\mathcal{G}$ of at most k subgraphs of G by using Lemma 3. We branch on each of them and repeat the same steps. Since the depth of the search tree is bounded by k, the desired result follows. □

If we only choose k as the parameter, then we can improve our result for the k-Coloring problem as follows. Here, we assume that $r \geq 2$ because Coloring can be solved in polynomial time for $(rP_1 + P_2)$-free graphs with $r \leq 1$, due to Theorem 1.

Theorem 5. *For any fixed integer* $r \geq 2$*, the* k-Coloring *problem restricted to* (rP_1+P_2)*-free graphs has a kernel of size* $k^2(r-1)$ *when parameterized by* k.

Proof. Let k be a positive integer, and let $G = (V,E)$ be an $(rP_1 + P_2)$-free graph for some fixed integer $r \geq 2$. If $k \leq 2$ then we can solve k-Coloring in polynomial time. Suppose that $k \geq 3$. If G has at most $k^2(r-1)$ vertices, then we are done. Suppose that G has at least $k^2(r-1)+1$ vertices. We check if G has an independent set S of r vertices such that $V \setminus N(S)$ contains at least $k(r-1)+1$ vertices. If not then we output `No`. Otherwise we give every vertex in $V \setminus N(S)$ color k. We then delete $V \setminus N(S)$ from G and check if the resulting graph G' has a $(k-1)$-coloring recursively. In this way, we either solve the problem or get an instance (G', k') of k'-Coloring where k' is an integer that is at most k and G' is a graph that has at most $k'^2(r-1)$ vertices, as desired.

We now prove that the above approach is correct. Suppose that G has at least $k^2(r-1)+1$ vertices. For every k-coloring of G, there must exist an independent set X with at least $k(r-1)+1$ vertices in G that all get the same color. We may without loss of generality assume that X is a maximal independent set. Because $k \geq 3$, we find that $k(r-1)+1 \geq r$. Hence, X contains an independent set S of size r. By Lemma 2, we find that $V \setminus N(S)$ is the unique maximal independent set containing S. Because X is maximal as well and $S \subseteq X$, we deduce that $X = V \setminus N(S)$. Hence, G has no k-coloring if G has no independent set S with $|S| = r$ and $|V \setminus N(S)| \geq k(r-1)+1$. Suppose that we find such a set S. Let $X = V \setminus N(S)$. Then, for any k-coloring of G, there exists a set $S' \subseteq X$ of vertices that get the same color, because X contains at least $k(r-1)+1$ vertices. We may assume without loss of generality that this color is k. By Lemma 2, we find that $V \setminus N(S')$ is the unique maximal independent set containing S'. Because X contains S' as well, we find that $X = V \setminus N(S')$. Because every vertex in S' received color k, no vertex in $N(S')$ will receive color k. This means that we can safely color every vertex in $X \setminus S'$ with color k as well. Consequently the graph G' obtained after deleting X must have a $(k-1)$-coloring, should G have a k-coloring.

We are left to show that the running time of our kernelization algorithm is polynomial. This follows from the following observations. First, there are at most $|V|^r$ sets of size r and r is fixed. Hence, we can check in polynomial time if G contains an independent set S of size r. Second, checking if $V \setminus N(S)$ contains at least $k(r-1)+1$ vertices and removing $V \setminus N(S)$ if this is the case can also been done in polynomial time. This completes the proof of Theorem 5. $\square$

3 $(\mathbf{P_1} + \boldsymbol{P_3})$-free graphs

In this section we consider (P_1+P_3)-free graphs. Recall that Coloring is polynomial-time solvable for (P_1+P_3)-free graphs due to Theorem 1. However, Jansen and Scheffler [14] showed that List k-Coloring is NP-complete when k is part of the input, already for complete bipartite graphs which form a subclass of the class of (P_1+P_3)-free graphs. We will show that List k-Coloring is fixed-parameter tractable for (P_1+P_3)-free graphs when parameterized by k. We must first introduce some extra terminology.

Let $G=(V,E)$ be a graph with a dominating set D. Suppose that we have ordered the vertices of D as $d_1, \ldots, d_p$. Then we can define (possibly empty) sets F_i for $i = 1, \ldots, p$ as follows. Let F_1 be the set of vertices in $V \setminus D$ adjacent to d_1, and for $i = 2, \ldots, p$, let F_i be the set of vertices in $V \setminus D$ adjacent to d_i but not to any d_h with $h \leq i-1$. The sets $F_1, \ldots, F_p$ are called *fixed* sets for D. By this definition and because D is dominating, every vertex in $V \setminus D$ belongs to exactly one fixed set F_i. We note, however, that D can have several collections of fixed sets, depending on the ordering of the vertices of D. Fixed sets have been introduced by Hoàng et al. [12] to prove that k-Coloring can be solved in polynomial time for P_5-free graphs for all fixed $k \geq 1$. We use them in a different way to prove the following result.

Theorem 6. *The* List k-Coloring *problem is in* FPT *for* (P_1+P_3)*-free graphs when parameterized by* k*.*

Proof. Let $G = (V, E)$ be a $(P_1 + P_3)$-free graph with a k-list assignment L. If G is disconnected then we consider each connected component separately. Hence, we may assume without loss of generality that G is connected.

First suppose that G has no induced P_3. Then V is a clique. If V has at least $k + 1$ vertices, then G has no coloring that satisfies L. If V has at most k vertices, then we try to color V in every possible way by brute force.

Now suppose that G contains an induced $P_3 = d_1d_2d_3$. Because G is (P_1+P_3)-free, $D = \{d_1, d_2, d_3\}$ is a dominating set. We construct fixed sets F_1, F_2, F_3 for D and prove a number of properties of these sets by a sequence of four claims.

Claim 1. If G has a coloring that respects L, then F_3 is a clique on at most $k-1$ vertices.

We prove Claim 1 as follows. We assume that G has a coloring that respects L. First suppose that F_3 is not a clique. Then there exist two vertices x and y in F_3 that are not adjacent. Consequently, d_1 and the path xd_3y form an induced $P_1 + P_3$, which is not possible. Hence F_3 is a clique. Now suppose that $|F_3| \geq k$. Then $F_3 \cup \{d_3\}$ is a clique on at least $k + 1$ vertices implying that G has no coloring respecting L. This is not possible either. Hence, $|F_3| \leq k - 1$. This completes the proof of Claim 1.

Claim 2. If G has a coloring that respects L, then $G[F_2]$ is a disjoint union of complete graphs, each of which has as most $k - 1$ vertices.

We prove Claim 2 as follows. We assume that G has a coloring that respects L. First suppose that $G[F_2]$ contains a connected component with two non-adjacent vertices. Then this component contains an induced P_3. However, d_1 is not adjacent to any vertex of F_2. Consequently, this induced P_3 and d_1 form an induced $P_1 + P_3$, which is not possible. Hence, $G[F_2]$ is a disjoint union of complete graphs. Now suppose that $G[F_2]$ contains a connected component F of size at least k. Then $F \cup \{d_2\}$ is a clique on at least $k + 1$ vertices implying that G has no coloring respecting L. This is not possible either. Hence, every connected component of $G[F_2]$ has as most $k - 1$ vertices. This completes the proof of Claim 2.

Let $X \subseteq F_1$ be the set of all vertices of F_1 that are not adjacent to any vertex of F_2.

Claim 3. If G has a coloring that respects L and $F_2 \neq \emptyset$, then X is a clique on at most $k - 1$ vertices.

We prove Claim 3 as follows. We assume that G has a coloring that respects L and that $F_2 \neq \emptyset$. First suppose that X contains two non-adjacent vertices x and y. Because $F_2 \neq \emptyset$, there exists a vertex $z \in F_2$. Then z and the path xd_1y form an induced $P_1 + P_3$, which is not possible. Hence, X is a clique. Now suppose that $|X| \geq k$. Then $X \cup \{d_1\}$ is a clique on at least $k+1$ vertices implying that G has no coloring respecting L. This is not possible either. Hence, $|X| \leq k-1$. This completes the proof of Claim 3.

Claim 4. If $G[F_2]$ has at least two connected components, then every vertex of F_2 is adjacent to every vertex of $F_1 \setminus X$.

We prove Claim 4 as follows. We assume that $G[F_2]$ has at least two connected components. Suppose that there is a vertex $x \in F_2$ that is not adjacent to a vertex $y \in F_1 \setminus X$. Because $y \notin X$, we find that y is adjacent to a vertex $z \in F_2$. If $xz \notin E$, then the path zyd_1 and the vertex x would form an induced $P_1 + P_3$. This is not possible. Hence, $xz \in E$. So, y can only be adjacent to the vertices of the connected component of $G[F_2]$ that contains x. Recall that $G[F_2]$ has at least two connected components. Let v be a vertex of another connected component of $G[F_2]$. Then v and the path xzy form an induced $P_1 + P_3$, which is not possible. Hence, every vertex of F_2 is adjacent to every vertex of $F_1 \setminus X$. This completes the proof of Claim 4.

We are now ready to describe our algorithm. We first branch by coloring the vertices of F_3 and the vertices d_1, d_2, d_3. We then consider the following three cases.

Case 1. $F_2 = \emptyset$.
For each vertex $v \in F_1$, we remove those colors from its list $L(v)$ that are a color of a neighbor of v in $F_3 \cup \{d_1, d_2, d_3\}$. We remove all vertices not in F_1 from G.

Case 2. $F_2 \neq \emptyset$ and $G[F_2]$ is connected.
We branch by coloring the vertices of F_2. Then we do the same as in Case 1.

Case 3. $G[F_2]$ has at least two components.
We first find the set X and then branch by coloring the vertices of X. Then we branch by choosing a set C of colors that will be used for the coloring of the vertices of F_2. For each vertex $u \in F_2$, we remove those colors from its list $L(u)$ that are not a color of C or that are a color of a neighbor of u in $F_3 \cup \{d_2, d_3\}$. For each vertex $v \in F_1 \setminus X$ we remove those colors from its list $L(v)$ that are a color in C or that are a color of a neighbor of v in $F_3 \cup X \cup \{d_1, d_2, d_3\}$. We remove all vertices not in $(F_1 \setminus X) \cup F_2$ from G.

Afterwards, we consider $G[F_2]$ and $G[F_1 \setminus X]$ independently and repeat the procedure above; note that $G[F_2]$ or X may be empty, depending on which case we are in.

We now prove that our algorithm is correct. By construction, we consider all possible colorings of the vertices of $F_3 \cup \{d_1, d_2, d_3\}$. Moreover, we consider all possible colorings of the vertices of F_2 if Case 2 applies. We also consider all possible colorings of the vertices of X if Case 3 applies. In the latter case we must also show that we consider all possible colorings of F_2 and $F_1 \setminus X$ by choosing the color set C to be used on the vertices of F_2. This follows from Claim 4, which tells us that every vertex of F_2 is adjacent to every vertex of $F_1 \setminus X$. Hence, a vertex in $F_1 \setminus X$ does not have the same color as a vertex in F_2 in any coloring of G. By construction of our algorithm, we find that F_2 and $F_1 \setminus X$ are separated, i.e., $L(u) \cap L(v) = \emptyset$ for all $u \in F_2$ and $v \in F_1 \setminus X$. Hence we may indeed consider $G[F_2]$ and $G[F_1 \setminus X]$ independently.

In order to show that our algorithm runs in FPT time, we consider Case 3, which is the worst case. In this case we have chosen a coloring of d_1, d_2, d_3, F_3, X and a set of colors C that are to be used on the colors of F_2. Note that there are at most 2^k different sets C. By Claims 1 and 3 we find that $F_3 \cup \{d_1, d_2, d_3\} \cup X$ consists of at most $2k+1$ vertices. Hence, there are at most $k^{2k+1} \cdot 2^k$ choices to branch on. For each branch we solve the problem for $G[F_2]$ in $O(k^{k-1}|F_2|)$ time, due to Claim 2. Because d_1 is adjacent to all vertices of F_1, at least one color cannot be used on F_1. Hence, we must solve the List $(k-1)$-Coloring problem with input $G[F_1 \setminus X]$ in every branch. Then, because the depth of the search tree is bounded by k, the desired result follows. This completes the proof of Theorem 6. □

4 Future Work

Jansen and Scheffler [14] showed that List k-Coloring is in FPT for P_4-free graphs when parameterized by k. This result together with Theorems 4 and 6 implies that the two smallest open cases parameterized by k are the cases $H = 2P_2$ and $H = 2P_1 + P_3$.

1. Is List k-Coloring parameterized by k in FPT for $2P_2$-free graphs?
2. Is List k-Coloring parameterized by k in FPT for $(2P_1 + P_3)$-free graphs?

References

1. E. Balas and C. S. Yu, On graphs with polynomially solvable maximum-weight clique problem, Networks 19, 247–253 (1989).
2. J.A. Bondy and U.S.R. Murty, Graph Theory, Springer Graduate Texts in Mathematics 244 (2008).
3. H.J. Broersma, F.V. Fomin, P.A. Golovach and D. Paulusma, Three complexity results on coloring P_k-free graphs, Proceedings of IWOCA 2009, LNCS 5874, 95–104 (2009).
4. H.J. Broersma, P.A. Golovach, D. Paulusma and J. Song, Updating the complexity status of coloring graphs without a fixed induced linear forest, Theoretical Computer Science, to appear.
5. H.J. Broersma, P.A. Golovach, D. Paulusma and J. Song, Determining the chromatic number of triangle-free $2P_3$-free graphs in polynomial time, manuscript.
6. D. Bruce, C.T. Hoàng, and J. Sawada, A certifying algorithm for 3-colorability of P_5-free graphs, Proceedings of ISAAC 2009, LNCS 5878, 594-604 (2009).

7. J.F. Couturier, P.A. Golovach, D. Kratsch and D. Paulusma, List coloring in the absence of a linear forest, Proceedings of WG 2011, LNCS, to appear. http://www.dur.ac.uk/daniel.paulusma/Papers/Submitted/rp1p5.pdf
8. K. Dabrowski, V. Lozin, R. Raman and B. Ries, Colouring vertices of triangle-free graphs, Proceedings of WG 2010, LNCS 6410, 184-195 (2010).
9. R.G. Downey and M.R. Fellows, Parameterized Complexity, Springer, 1999.
10. P.A. Golovach, D. Paulusma, and J. Song, 4-Coloring H-free graphs when H is small, Proceedings of SOFSEM 2012, LNCS, to appear.
11. M. Grötschel, L. Lovász, and A. Schrijver, Polynomial algorithms for perfect graphs, Ann. Discrete Math., Topics on Perfect Graphs 21, 325–356 (1984).
12. C.T. Hoàng, M. Kamiński, V. Lozin, J. Sawada, and X. Shu, Deciding k-colorability of P_5-free graphs in polynomial time, Algorithmica 57, 74–81 (2010).
13. I. Holyer, The NP-completeness of edge-coloring, SIAM J. Comput. 10, 718–720 (1981).
14. K. Jansen and P. Scheffler, Generalized coloring for tree-like graphs, Discrete Appl. Math. 75 (1997), 135–155.
15. M. Kamiński and V.V. Lozin, Coloring edges and vertices of graphs without short or long cycles, Contributions to Discrete Math. 2, 61–66 (2007).
16. M. Kamiński and V.V. Lozin, Vertex 3-colorability of Claw-free Graphs. Algorithmic Operations Research 21, (2007).
17. D. Král', J. Kratochvíl, Zs. Tuza, and G.J. Woeginger, Complexity of coloring graphs without forbidden induced subgraphs, Proceedings of WG 2001, LNCS 2204, 254–262 (2001).
18. J. Kratochvíl, Precoloring extension with fixed color bound, Acta Math. Univ. Comen. 62, 139–153 (1993).
19. V.B. Le, B. Randerath and I. Schiermeyer, On the complexity of 4-coloring graphs without long induced paths, Theoret. Comput. Sci. 389, 330–335 (2007).
20. D. Leven and Z. Galil, NP completeness of finding the chromatic index of regular graphs, Journal of Algorithms 4, 35—44 (1983).
21. F. Maffray and M. Preissmann, On the NP-completeness of the k-colorability problem for triangle-free graphs, Discrete Math. 162, 313–317 (1996).
22. R. Niedermeier, Invitation to Fixed-Parameter Algorithms, Oxford Lecture Series in Mathematics and its Applications. Oxford University Press, 2006.
23. B. Randerath and I. Schiermeyer, 3-Colorability $\in$ P for P_6-free graphs, Discrete Appl. Math. 136, 299–313 (2004).
24. B. Randerath and I. Schiermeyer, Vertex colouring and forbidden subgraphs - a survey, Graphs Combin. 20, 1–40 (2004).
25. S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa, A new algorithm for generating all the maximal independent sets, SIAM J. Comput. 6, 505–517 (1977).
26. Zs. Tuza, Graph colorings with local restrictions - a survey, Discuss. Math. Graph Theory 17, 161–228 (1997).
27. G.J. Woeginger and J. Sgall, The complexity of coloring graphs without long induced paths, Acta Cybernet. 15, 107–117 (2001).