Updating the complexity status of coloring graphs without a fixed induced linear forest*

Hajo Broersma, Petr A. Golovach, Daniël Paulusma, and Jian Song **

School of Engineering and Computing Sciences, Durham University,
Science Laboratories, South Road, Durham DH1 3LE, UK
{hajo.broersma,petr.golovach,daniel.paulusma,jian.song}@durham.ac.uk

Abstract. A graph is H-free if it does not contain an induced subgraph isomorphic to the graph H. The graph P_k denotes a path on k vertices. The ℓ-Coloring problem is the problem to decide whether a graph can be colored with at most ℓ colors such that adjacent vertices receive different colors. We show that 4-Coloring is NP-complete for P_8-free graphs. This improves a result of Le, Randerath, and Schiermeyer, who showed that 4-Coloring is NP-complete for P_9-free graphs, and a result of Woeginger and Sgall, who showed that 5-Coloring is NP-complete for P_8-free graphs. Additionally, we prove that the precoloring extension version of 4-Coloring is NP-complete for P_7-free graphs, but that the precoloring extension version of 3-Coloring can be solved in polynomial time for $(P_2 + P_4)$-free graphs, a subclass of P_7-free graphs. Here $P_2 + P_4$ denotes the disjoint union of a P_2 and a P_4. We denote the disjoint union of s copies of a P_3 by sP_3 and involve Ramsey numbers to prove that the precoloring extension version of 3-Coloring can be solved in polynomial time for sP_3-free graphs for any fixed s. Combining our last two results with known results yields a complete complexity classification of (precoloring extension of) 3-Coloring for H-free graphs when H is a fixed graph on at most 6 vertices: the problem is polynomial-time solvable if H is a linear forest; otherwise it is NP-complete.

1 Introduction

Graph coloring involves the labeling of the vertices of some given graph by integers called colors such that adjacent vertices receive different colors. The corresponding ℓ-Coloring problem is the problem to decide whether a graph can be colored with at most ℓ colors. Due to the fact that the ℓ-Coloring problem is NP-complete for any fixed $\ell \geq 3$, there has been a considerable interest in studying its complexity when restricted to certain graph classes, in particular graph classes that can be characterized by forbidden induced subgraphs. We refer to two survey papers [17, 20] for more details. Instead of repeating what

* Preliminary and shortened versions containing some of the results of this paper will appear in the proceedings of WG 2010 and ISAAC 2010

** This work has been supported by EPSRC (EP/G043434/1).

has been written in so many papers over the years, we also refer to these surveys for motivation and background.

In the first part of this paper we continue the study of ℓ-Coloring for P_k-free graphs and narrow down the complexity gap on 4-Coloring for P_k-free graphs by proving two NP-completeness results. This setting has been studied in several earlier papers by different groups of researchers [3, 5, 10, 13–16, 21]. In the second part of the paper we concentrate on (precoloring extension of) 3-Coloring for H-free graphs, where H is a linear forest. We prove two polynomial-time results that together with existing results settle the complete complexity classification of 3-Coloring for H-free graphs, where H is a fixed graph on at most 6 vertices. Before we discuss the existing results and our new results in more detail we must first introduce some terminology.

1.1 Terminology

We only consider finite undirected graphs without loops and multiple edges. We refer to the textbook by Bondy and Murty [2] for any undefined graph terminology. The graph P_k denotes the path on k vertices, and the graph K_k denotes the complete graph on k vertices. The disjoint union of two graphs G and H is denoted $G + H$, and the disjoint union of k copies of G is denoted kG. A *linear forest* is the disjoint union of a collection of paths.

Let $\{H_1, \ldots, H_p\}$ be a set of graphs. We say that a graph G is $(H_1, \ldots, H_p)$-*free* if G has no induced subgraph isomorphic to a graph in $\{H_1, \ldots, H_p\}$; if $p = 1$, we sometimes write H_1-free instead of (H_1)-free. A K_3-free graph is also called *triangle-free*.

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. An *ℓ-coloring* of G is a coloring ϕ of G with $\phi(V) \subseteq \{1, \ldots, \ell\}$. Here we use the notation $\phi(U) = \{\phi(u) \mid u \in U\}$ for $U \subseteq V$. If G has an ℓ-coloring, then G is called *ℓ-colorable*. We let $\chi(G)$ denote the *chromatic number* of G, i.e., the smallest ℓ such that G has an ℓ-coloring. We recall that the problem ℓ-Coloring is to decide whether a given graph admits an ℓ-coloring. The related Vertex Coloring problem is the problem of determining the chromatic number of a given graph.

In *list-coloring* we assume that $V = \{v_1, v_2, \ldots, v_n\}$ and that for every vertex v_i of G there is a list L_i of *admissible* colors (a subset of the natural numbers). We say that a coloring $\phi : V \to \{1, 2, \ldots\}$ *respects* these lists if $\phi(v_i) \in L_i$ for all $i \in \{1, 2, \ldots, n\}$. We also call ϕ a *list-coloring* in this case.

In *precoloring extension* we assume that a (possibly empty) subset $W \subseteq V$ of G is precolored with $\phi_W : W \to \{1, 2, \ldots\}$ and the question is whether we can extend ϕ_W to a coloring of G. If ϕ_W is restricted to $\{1, 2, \ldots, \ell\}$ and we want to decide whether we can extend the precoloring to an ℓ-coloring of G, we say we deal with the problem ℓ-Precoloring Extension.

1.2 Related work and new results

Results of Hoàng et al. [10] imply that ℓ-Precoloring Extension is solvable in polynomial time on P_5-free graphs for any fixed ℓ. In contrast, determining the chromatic number is NP-hard for P_5-free graphs, as shown by Král' et al. [13], whereas this problem is polynomial-time solvable for P_4-free graphs (because a P_4-free graph is perfect, and the chromatic number of a perfect graph can be determined in polynomial time, a classic result due to Grötschel et al. [9]). In an earlier paper [3] we established the following three results. Firstly we proved that 6-Coloring is NP-complete for P_7-free graphs, secondly that 3-Precoloring Extension is polynomial-time solvable for P_6-free graphs, and thirdly that 5-Precoloring Extension is NP-complete for P_6-free graphs. Le, Randerath, and Schiermeyer [15] proved that 4-Coloring is NP-complete for P_9-free graphs. Woeginger and Sgall [21] showed that 5-Coloring is NP-complete for P_8-free graphs. In Section 2 we present a common improvement to the results by Le et al. [15] and Woeginger and Sgall [21] by showing that 4-Coloring is NP-complete for P_8-free graphs. In Section 3 we prove that 4-Precoloring Extension is NP-complete for P_7-free graphs. All these results together lead to Table 1, which shows the current status of ℓ-Coloring and ℓ-Precoloring Extension for P_k-free graphs. In this table the contributions of this paper are indicated in bold. Table 1 also shows which cases are still open.

P_k-free	$\ell \rightarrow$ 3	3*	4	4*	5	5*	≥ 6	$\geq 6^*$
$k \leq 5$	P	P	P	P	P	P	P	P
$k = 6$	P	P	?	?	?	NP-c	?	NP-c
$k = 7$	?	?	?	**NP-c**	?	NP-c	NP-c	NP-c
$k = 8$	?	?	**NP-c**	**NP-c**	NP-c	NP-c	NP-c	NP-c
$k \geq 9$	?	?	NP-c	NP-c	NP-c	NP-c	NP-c	NP-c

Table 1. The complexity of ℓ-Coloring and ℓ-Precoloring Extension (marked by *) on P_k-free graphs for combinations of fixed k and ℓ.

It seems hard to extend the polynomial-time result on 3-Coloring for P_6-free graphs (which follows from the corresponding result on 3-Precoloring Extension [3] but which was originally proven by Randerath and Schiermeyer [16]) to P_7-free graphs. This motivates our focus on H-free graphs, where H is a linear forest on at most 6 vertices. Such graphs are P_7-free whenever H is an induced subgraph of the P_7.

The focus on linear forests as forbidden induced subgraphs can also be justified by the following result. Kamiński and Lozin [11] showed that 3-Coloring is NP-complete for the class of graphs of girth (the length of a shortest induced cycle) at least p for any fixed $p \geq 3$. This immediately implies that 3-Coloring and hence also 3-Precoloring Extension is NP-complete for the class of H-free graphs if H contains a cycle.

The 3-Coloring problem is also NP-complete for the class of claw-free graphs (graphs with no induced 4-vertex star $K_{1,3}$), even for the subclass of claw-free graphs that are also diamond-free and 4-regular, as shown by Kochol, Lozin and Randerath [12]. This immediately implies that 3-Coloring, and hence also 3-Precoloring Extension, is NP-complete for the class of H-free graphs if H is a forest that contains a vertex with degree at least 3.

We show in Section 4 that there are exactly two open nontrivial complexity questions on 3-Precoloring Extension for H-free graphs when H is a linear forest on at most six vertices. The first case is when $H = P_2 + P_4$, and the second case is when $H = 2P_3$. In Section 4.1 we prove that 3-Precoloring Extension is polynomial-time solvable for $(P_2 + P_4)$-free graphs. In Section 4.2 we solve the second case by showing that 3-Precoloring Extension is polynomial-time solvable for sP_3-free graphs for any fixed positive integer s. The key ingredient for proving the latter result is the use of Ramsey numbers to deduce the existence of a small dominating set in sP_3-free graphs that admit a k-coloring (or more specifically, a 3-coloring).

Together with existing results the cases we solved for $(P_2 + P_4)$-free and $2P_3$-free graphs yield the following complete characterization of the complexity status of 3-Precoloring Extension for H-free graphs, where H is a fixed graph on at most 6 vertices.

Theorem 1. *Let H be a fixed graph on at most 6 vertices. Then* 3-Precoloring Extension *for H-free graphs is polynomial-time solvable if H is a linear forest; otherwise it is* NP*-complete.*

We also show in Section 4.1 that our results imply that the Vertex Coloring problem for triangle-free $(P_2 + P_4)$-free graphs can be solved in polynomial time. This has recently been proved independently by Dabrowski et al. [6].

For triangle-free graphs this result together with the existing results and the open case we recently solved in a related paper [4] yields the following characterization theorem.

Theorem 2 ([4]). *Let H be a fixed graph on at most 6 vertices. Then* Vertex Coloring *for triangle-free H-free graphs is polynomial-time solvable if H is a forest and $H \neq K_{1,5}$; otherwise it is* NP*-hard.*

2 4-Coloring for P_8-free graphs

In this section we prove that 4-Coloring is NP-complete for P_8-free graphs. We use a reduction from the 3-Satisfiability problem, which is an NP-complete problem [8]. We consider an arbitrary instance I of 3-Satisfiability that has variables $\{x_1, x_2, \ldots, x_n\}$ and clauses $\{C_1, C_2, \ldots, C_m\}$ and define a graph G_I. Next we show that G_I is P_8-free and that G_I is 4-colorable if and only if I has a satisfying truth assignment.

Here is the construction that defines G_I.

- For each clause C_j we introduce a 7-vertex cycle with vertex set

$$\{b_{j,1}, b_{j,2}, c_{j,1}, c_{j,2}, c_{j,3}, d_{j,1}, d_{j,2}\}$$

  and edge set

$$\{b_{j,1}c_{j,1}, c_{j,1}d_{j,1}, d_{j,1}c_{j,2}, c_{j,2}d_{j,2}, d_{j,2}c_{j,3}, c_{j,3}b_{j,2}, b_{j,2}b_{j,1}\}.$$

  We say that these vertices are of b-type, c-type and d-type, respectively. They induce disjoint 7-cycles (i.e., cycles on 7 vertices) in G_I which we call *clause-components* in the sequel.

- For each variable x_i we introduce a copy of a K_2, i.e., two vertices joined by an edge $x_i\overline{x}_i$. We say that both x_i and $\overline{x}_i$ are of x-type, and we call the corresponding disjoint K_2s in G_I *variable-components* in the sequel.

- For every clause C_j we fix an arbitrary order of its variables $x_{i_1}, x_{i_2}, x_{i_3}$. For $h = 1, 2, 3$ we either add the edge $c_{j,h}x_{i_h}$ or the edge $c_{j,h}\overline{x}_{i_h}$ depending on whether x_{i_h} or $\overline{x}_{i_h}$ is a literal in C_j, respectively.

- We add an edge between any x-type vertex and any b-type vertex. We also add an edge between any x-type vertex and any d-type vertex.

- We introduce one additional new vertex a which we make adjacent to all b-type, c-type and d-type vertices.

See Figure 1 for an example of a graph G_I. In this example C_1 is a clause with ordered literals $x_{i_1}, \overline{x}_{i_2}, x_{i_3}$ and C_m is a clause with ordered literals $\overline{x}_1, x_{i_3}, x_n$. The thick edges indicate the connections between the literal vertices and the c-type vertices of the clause gadgets. We omitted the indices from the labels of the clause gadget vertices to increase the visibility.

We now prove two lemmas. Lemma 1 shows that the graph G_I is P_8-free (in fact it shows a slightly stronger statement as this will be of use for us in Section 3). In Lemma 2 we prove that G_I admits a 4-coloring if and only if I has a satisfying truth assignment.

Lemma 1. *The graph G_I is P_8-free. Moreover, every induced path in G_I on seven vertices contains a.*

Proof. Let P be an induced path in G_I. We show that G_I is P_8-free by proving that P has at most seven vertices. We also show that P contains a in case P has exactly seven vertices. We distinguish a number of cases and subcases.

Case 1. $a \notin V(P)$.

Case 1a. P contains no x-type vertex.
This means that P is contained in one clause-component, which is isomorphic to an induced 7-cycle. Consequently, P has at most 6 vertices.

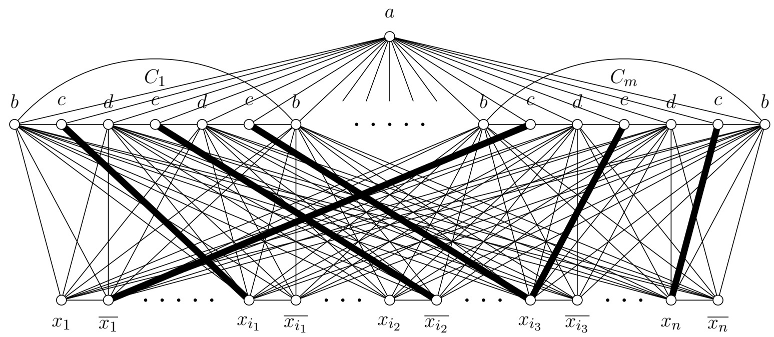

Fig. 1. The graph G_I in which clauses $C_1 = \{x_{i_1}, \overline{x}_{i_2}, x_{i_3}\}$ and $C_m = \{\overline{x}_1, x_{i_3}, x_n\}$ are illustrated.

Case 1b. P contains exactly one x-type vertex.
Let x_i be this vertex. Then P contains vertices of at most two clause-components. Since x_i is adjacent to all b-type and d-type vertices, we then find that P contains at most two vertices of each of the clause-components. Hence P has at most 5 vertices.

Case 1c. P contains exactly two x-type vertices.
First suppose that these vertices are adjacent, say P contains x_i and $\overline{x}_i$. By the same reasoning as above we find that P has at most 4 vertices.

Now suppose the two x-type vertices of P are not adjacent. By symmetry, we may assume that P contains x_h and x_i. If P contains no b-type vertex and no d-type vertex, then there is no subpath in P from x_h to x_i, a contradiction. If P contains two or more vertices of b-type and d-type, then P contains a cycle, another contradiction. Hence P contains exactly one vertex z that is of b-type or d-type. Then $x_h z x_i$ is a subpath in P. If both x_h and x_i have a neighbor in $V(P)\backslash\{z\}$, then this neighbor must be of c-type, and consequently an end vertex of P (because a c-type vertex is adjacent to only one x-type vertex). Hence P contains at most five vertices.

Case 1d. P contains at least three x-type vertices.
Then P contains no b-type vertex and no d-type vertex, because such vertices would have degree 3 in P. However, on the other hand the three x-type vertices come from at least two different variable-components. Since any c-type vertex is adjacent to exactly one x-type vertex, P must contain a b-type or d-type vertex to connect the x-type vertices of P to one another. We conclude that this subcase is not possible.

Case 2. $a \in V(P)$.
First suppose a is an end vertex of P. If $|V(P)| \geq 2$ then P contains exactly one

vertex that is of b-type, c-type or d-type. Since every x-type vertex is adjacent to only one other x-type vertex, this means that P can have at most four vertices.

Now suppose a is not an end vertex of P. Then P contains exactly two vertices that are of b-type, c-type or d-type. By the same arguments as above, we then find that P has at most 7 vertices. This completes the proof of Lemma 1. □

Lemma 2. *The graph G_I is 4-colorable if and only if I has a satisfying truth assignment.*

Proof. Suppose we have a 4-coloring of G_I with colors $\{1, 2, 3, 4\}$. We may assume without loss of generality that a has color 1, that $b_{1,1}$ has color 3 and that $b_{1,2}$ has color 4. This implies that all x-type vertices have a color from $\{1, 2\}$. Furthermore, for $i = 1, \ldots, n$, if x_i has color 1 then $\overline{x}_i$ has color 2, and vice versa. Hence we find that all b-type and d-type vertices have a color from $\{3, 4\}$. Then by symmetry we may assume that every $b_{j,1}$ has color 3 and every $b_{j,2}$ has color 4. This means that every $c_{j,1}$ has a color from $\{2, 4\}$, every $c_{j,2}$ has a color from $\{2, 3, 4\}$ and every $c_{j,3}$ has a color from $\{2, 3\}$. Now suppose there is a clause C_j with each of its three literals colored by color 2. Then $c_{j,1}$ must have color 4 and $c_{j,3}$ must have color 3. Consequently, $d_{j,1}$ has color 3 and $d_{j,2}$ has color 4. Then $c_{j,2}$ cannot have a color in a proper 4-coloring of G_I. Hence this is not possible and we find that at least one literal in every clause is colored by color 1. This means we can define a truth assignment that sets a literal to `false` if the corresponding x-type vertex has color 2, and to `true` otherwise. So a 4-coloring of G_I implies a satisfying truth assignment for I.

For the converse, suppose I has a satisfying truth assignment. We use color 1 to color the x-type vertices representing the true literals and color 2 for the false literals. Since each clause contains at least one true literal, we note that we can color $c_{j,1}$, $c_{j,2}$ and $c_{j,3}$ and also all other remaining vertices in a straightforward way. This implies a 4-coloring for G_I and completes the proof of Lemma 2. □

By Lemmas 1 and 2 we obtain the main result of this section.

Theorem 3. *The* 4-Coloring *problem is* NP*-complete for P_8-free graphs.*

3 4-Precoloring Extension for P_7-free graphs

In this section we show that 4-Precoloring Extension is NP-complete for the class of P_7-free graphs. We use a reduction from the Not-All-Equal 3-Satisfiability problem with positive literals only. This NP-complete problem [18] is also known as Hypergraph 2-Colorability and is defined as follows. Given a set $X = \{x_1, x_2, \ldots, x_n\}$ of logical variables, and a set $C = \{C_1, C_2, \ldots, C_m\}$ of three-literal clauses over X in which all literals are positive, does there exist a truth assignment for X such that each clause contains at least one true literal and at least one false literal?

We consider an arbitrary instance I of Not-All-Equal 3-Satisfiability that has variables $\{x_1, x_2, \ldots, x_n\}$ and clauses $\{C_1, C_2, \ldots, C_m\}$, and we define a graph G_I^* with a precoloring on some vertices of G_I^*. Then we show that G_I^* is P_7-free and that the precoloring on G_I^* can be extended to a 4-coloring of G_I^* if and only if I has a satisfying truth assignment in which each clause contains at least one true literal and at least one false literal.

Here is the construction that defines G_I^* with a precoloring.

- For each clause C_j we introduce a gadget with vertex set

  $$\{a_{j,1}, a_{j,2}, a_{j,3}, b_{j,1}, b_{j,2}, c_{j,1}, c_{j,2}, c_{j,3}, d_{j,1}, d_{j,2}\}$$

  and edge set

  $$\{a_{j,1}c_{j,1}, a_{j,2}c_{j,2}, a_{j,3}c_{j,3}, b_{j,1}c_{j,1}, c_{j,1}d_{j,1}, d_{j,1}c_{j,2}, c_{j,2}d_{j,2}, d_{j,2}c_{j,3}, c_{j,3}b_{j,2}, b_{j,2}b_{j,1}\},$$

  and a disjoint gadget called the *copy* with vertex set

  $$\{a'_{j,1}, a'_{j,2}, a'_{j,3}, b'_{j,1}, b'_{j,2}, c'_{j,1}, c'_{j,2}, c'_{j,3}, d'_{j,1}, d'_{j,2}\}$$

  and edge set

  $$\{a'_{j,1}c'_{j,1}, a'_{j,2}c'_{j,2}, a'_{j,3}c'_{j,3}, b'_{j,1}c'_{j,1}, c'_{j,1}d'_{j,1}, d'_{j,1}c'_{j,2}, c'_{j,2}d'_{j,2}, d'_{j,2}c'_{j,3}, c'_{j,3}b'_{j,2}, b'_{j,2}b'_{j,1}\}.$$

  We say that all these vertices (so including the vertices in the copy) are of a-type, b-type, c-type and d-type, respectively. They induce $2m$ disjoint 10-vertex components in G_I^* which we will call *clause-components* in the sequel. We precolor every $a_{j,h}$ by 1 and every $a'_{j,h}$ by 2.

- Every variable x_i will be represented by a vertex in G_I^*, and we say that these vertices are of x-type.

- For every clause C_j we fix an arbitrary order of its variables $x_{i_1}, x_{i_2}, x_{i_3}$ and add edges $c_{j,h}x_{i_h}$ and $c'_{j,h}x_{i_h}$ for $h = 1, 2, 3$.

- We add an edge between every x-type vertex and every b-type vertex. We also add an edge between every x-type vertex and every d-type vertex.

- We add an edge between every a-type vertex and every b-type vertex. We also add an edge between every a-type vertex and every d-type vertex.

In Figure 2 we illustrate an example in which C_j is a clause with ordered variables $x_{i_1}, x_{i_2}, x_{i_3}$. The thick edges indicate the connection between the variables vertices and the c-type vertices of the two copies of the clause gadget. The dashed thick edges indicate the connections between the (precolored) a-type and c-type vertices of the two copies of the clause gadget. We omitted the indices from the labels of the clause gadget vertices to increase the visibility.

We now prove two lemmas. Lemma 3 shows that the graph G_I^* is P_7-free. In Lemma 4 we prove that the precoloring of G_I^* can be extended to a 4-coloring of G_I^* if and only if I has a truth assignment in which each clause contains at least one true and at least one false literal.

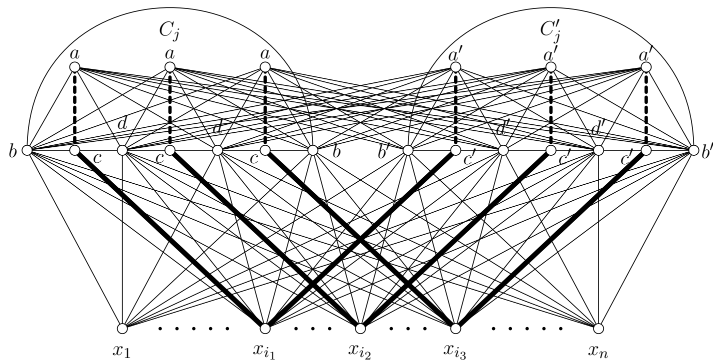

Fig. 2. The graph G_I^* for the clause $C_j = \{x_{i_1}, x_{i_2}, x_{i_3}\}$.

Lemma 3. *The graph G_I^* is P_7-free.*

Proof. Let P be an induced path in G_I^*. We show that P has at most six vertices. We distinguish the following cases.

Case 1. P contains no a-type vertex.
Let I' be the instance obtained from I after adding a copy of every clause. Then P is contained in a graph isomorphic to the graph $G_{I'}$ as defined in Section 2 after removing the negative variables. Hence, by Lemma 1, P contains at most six vertices.

Case 2. P contains exactly one a-type vertex.
We may assume without loss of generality that $a_{1,1}$ is this vertex.

Suppose $a_{1,1}$ is an end vertex of P. If the (only) neighbor of $a_{1,1}$ is a b-type or d-type vertex, then P contains no other b-type or d-type vertex. This implies that P can contain at most two other vertices, namely one x-type vertex and one c-type vertex. Suppose the neighbor of $a_{1,1}$ on P is a c-type vertex (so this neighbor must be $c_{1,1}$). Then P contains no b-type vertex and no d-type vertex. Hence P contains at most two other vertices, namely an x-type vertex and another c-type vertex. Hence P has at most four vertices.

Suppose $a_{1,1}$ is not an end vertex of P. Suppose $c_{1,1}$ is a neighbor of $a_{1,1}$ on P. Let z be the other neighbor of $a_{1,1}$ on P. Then z must be a b-type or d-type vertex. This means that P contains no other b-type or d-type vertex, and if P contains an x-type vertex this vertex must be adjacent to z. Then $c_{1,1}$ is an end vertex of P, and P contains at most two other vertices, namely an x-type vertex and another c-type vertex. Hence P has at most five vertices.

Suppose $c_{1,1}$ is not a neighbor of $a_{1,1}$ on P. Then both neighbors of $a_{1,1}$ are b-type or d-type vertices. This means that P contains no x-type vertex. Consequently, P can contain at most two other (c-type) vertices. Hence P has at most five vertices.

Case 3. P contains exactly two a-type vertices.
Suppose P contains no b-type vertex and no d-type vertex. Then the a-type vertices must be the end vertices of P that must be joined in P by two c-type vertices and one x-type vertex. Hence P has five vertices.

Suppose P contains a vertex z that is a b-type or d-type vertex. Because such a vertex is adjacent to both a-type vertices, P contains no other b-type or d-type vertex and P does not contain an x-type vertex. Then P might only contain at most two other vertices (that are of c-type). This means that P has at most five vertices.

Case 4. P contains at least three a-type vertices.
Then P contains no b-type vertex and no d-type vertex. Hence, every a-type vertex of P must be an end vertex of P. This is not possible and completes the proof of Lemma 3. □

Lemma 4. *The precoloring of G_I^* can be extended to a 4-coloring of G_I^* if and only if I has a truth assignment in which each clause contains at least one true and at least one false literal.*

Proof. Suppose the precoloring of G_I^* can be extended to a 4-coloring of G_I^*. Since $a_{1,1}$ with color 1 and $a'_{1,1}$ with color 2 are adjacent to every b-type vertex, we may assume by symmetry that every $b_{j,1}$ and every $b'_{j,1}$ has color 3, whereas every $b_{j,2}$ and every $b'_{j,2}$ has color 4. This implies the following. Firstly, it implies that all x-type vertices have a color from $\{1, 2\}$. Consequently, all d-type vertices must have a color from $\{3, 4\}$. Secondly, it implies that every $c_{j,1}$ has a color from $\{2, 4\}$, every $c_{j,2}$ has a color from $\{2, 3, 4\}$ and every $c_{j,3}$ has a color from $\{2, 3\}$. Thirdly, it implies that every $c'_{j,1}$ has a color from $\{1, 4\}$, every $c_{j,2}$ has a color from $\{1, 3, 4\}$ and every $c_{j,3}$ has a color from $\{1, 3\}$.

Now suppose there is a clause C_j with each of its three literals colored by color 2. Then $c_{j,1}$ must have color 4 and $c_{j,3}$ must have color 3. Consequently, $d_{j,1}$ has color 3 and $d_{j,2}$ has color 4. Then $c_{j,2}$ cannot have a color in a proper 4-coloring. Hence this is not possible and we find that at least one literal in every clause is colored by color 1. By considering the copies, in a similar way we find that at least one literal in every clause is colored by color 2. Hence, we can define a truth assignment that sets a literal to `false` if the corresponding x-type vertex has color 2, and to `true` otherwise. So a 4-coloring of G_I^* that extends the precoloring on G_I^* implies a truth assignment for I in which each clause contains at least one true and at least one false literal.

For the converse, suppose I has a satisfying truth assignment in which each clause contains at least one true and at least one false literal. We use color 1 to color the x-type vertices representing the true literals and color 2 for the false literals. Since each clause contains at least one true literal, we can color $c_{j,1}$, $c_{j,2}$ and $c_{j,3}$, respecting the precoloring. Similarly, since each clause contains at least one false literal, we can color $c'_{j,1}$, $c'_{j,2}$ and $c'_{j,3}$, respecting the precoloring. We color all other remaining uncolored vertices in a straightforward way. This completes the proof of Lemma 4. □

By Lemmas 3 and 4 we obtain the main result of this section.

Theorem 4. *The* 4-Precoloring Extension *problem is* NP*-complete for* P_7*-free graphs.*

4 3-Precoloring Extension for H-free graphs

Here we consider 3-Precoloring Extension for H-free graphs, where H is a subgraph of P_7 on at most 6 vertices. We can use the polynomial-time algorithm of one of our earlier papers [3] for solving this problem when H is an induced subgraph of P_6, because in these cases any H-free graph is also P_6-free. This means that the following fifteen cases remain:

$$\begin{array}{lll} H_1 = 6P_1 & H_6 \ = 3P_1 + P_3 & H_{11} = P_1 + P_2 + P_3 \\ H_2 = 5P_1 & H_7 \ = 2P_1 + 2P_2 & H_{12} = P_1 + P_5 \\ H_3 = 4P_1 & H_8 \ = 2P_1 + P_3 & H_{13} = 3P_2 \\ H_4 = 4P_1 + P_2 & H_9 \ = 2P_1 + P_4 & H_{14} = P_2 + P_4 \\ H_5 = 3P_1 + P_2 & H_{10} = P_1 + 2P_2 & H_{15} = 2P_3. \end{array}$$

We first consider H_i for $i = 1, \ldots, 12$. For these graphs we use the following observation, the proof of which follows from the fact that the decision problem in these cases can be modelled and solved as an instance of the 2-Satisfiability problem. This approach has been introduced by Edwards [7] and is folklore now.

Observation 1 ([7]) *Let G be a graph in which every vertex has a list of admissible colors of size at most 2. Then checking whether G has a coloring respecting these lists is solvable in polynomial time.*

Proposition 1. *Let H be a graph. If* 3-Precoloring Extension *is solvable in polynomial time for H-free graphs, then it is also solvable in polynomial time for $(H + P_1)$-free graphs.*

Proof. Let G be an $(H+P_1)$-free graph with precoloring $\phi_W : W \to \{1, 2, 3\}$ for some $W \subseteq V(G)$. If G is H-free, we are done. Otherwise, we use ϕ_W to construct a list of admissible colors for each vertex in G.

Suppose G contains an induced subgraph H' that is isomorphic to H. Because G is $(H+P_1)$-free, every vertex in $V(G)\backslash V(H')$ must be adjacent to a vertex in H'. We guess a coloring of $V(H')$ that respects the lists. Afterwards we apply Observation 1. Since H' has a fixed size, the number of guesses is polynomially bounded. □

Using our polynomial-time algorithm [3] that solves 3-Precoloring Extension for P_6-free graphs, and (repeatedly) applying Proposition 1 yields polynomial-time results of the same problem for H_i-free graphs for $i = 1, \ldots, 12$.

In Section 4.1 we consider $H_{14} = P_2 + P_4$. The polynomial-time result for 3-Precoloring Extension for the cases $H_{13} = 3P_2$ and $H_{15} = 2P_3$ follow from our more general result for this problem for sP_3-free graphs, which we prove in Section 4.2. We note that the case H_{13} can also be solved by modifying the known proof of the corresponding result for 3-Coloring for sP_2-free graphs that uses a result from Balas and Yu [1] on the maximal number of independent sets in an sP_2-free graph and a result from Tsukiyama et al. [19] on the enumeration of such sets.

4.1 3-Precoloring Extension for $(P_2 + P_4)$-free graphs

In this section we describe how to test in polynomial time whether a given (P_2+P_4)-free graph G with precoloring $\phi_W : W \rightarrow \{1, 2, 3\}$ for some $W \subseteq V(G)$ allows a coloring $\phi : V(G) \rightarrow \{1, 2, 3\}$ with $\phi(u) = \phi_W(u)$ for all $u \in W$.

Theorem 5. *The* 3-Precoloring Extension *problem can be solved in polynomial time for* $(P_2 + P_4)$*-free graphs.*

Proof. Let G be a $(P_2 + P_4)$-free graph. We start by making two assumptions. Firstly, we assume that G is connected as otherwise we apply our algorithm on each component of G. Secondly, we assume that G contains an induced subgraph H isomorphic to P_6. If not, then G would be P_6-free and we could use the polynomial-time algorithm for P_6-free graphs [3] to solve this problem.

We use ϕ_W to construct a list of admissible colors for each vertex in G. We guess a coloring of H respecting these lists and start our algorithm, which we run at most 3^6 times as this is an upper bound on the number of possible 3-colorings of H. From the description of the algorithm it will be immediately clear that its running time is polynomial in $|V(G)|$.

Our algorithm first applies the following subroutine. Let $U \subseteq V(G)$ contain all vertices that have a list consisting of exactly one color. For every vertex $u \in U$ we remove this single color $c(u)$ from the lists of its neighbors. If this results in an empty list at some vertex, then we output No. We remove u from G and repeat this process in the remaining graph as long as there exists a vertex with a list of size 1. This process is called *updating* the graph. Note that during this procedure we also removed all vertices of H. We restore the vertices of H back into G. We may assume that G is still connected; otherwise, due to the $(P_2 + P_4)$-freeness of G, every component not containing H is a single vertex and can be colored trivially. Let S be the set of vertices that still have a list of admissible colors of size 3. If $S = \emptyset$, then we can apply Observation 1.

Suppose $S \neq \emptyset$. Let T be the set of vertices of $V(G) \setminus V(H)$ that have at least one neighbor in H. Because we colored every vertex in H and updated G, every vertex of T has a list of exactly two admissible colors, and consequently, $S \cap (V(H) \cup T) = \emptyset$. Since G contains no induced P_2+P_4, we find that $V(G) \setminus (V(H) \cup T)$, and consequently S, is an independent set in G. Since we assume that G is still connected, each vertex in S has at least one neighbor in T (so $T \neq \emptyset$).

For convenience we order the vertices of H along the P_6 as $p_1, p_2, \ldots, p_6$, starting with vertex p_1 with degree 1 in H. Let $T^* \subseteq T$ consist of all vertices in T that have a neighbor in S. Let T_1 denote the subset of vertices of T^* adjacent to p_1, p_3, p_5 and not to p_2, p_4, p_6; let T_2 denote the subset of vertices of T^* adjacent to p_2, p_4, p_6 and not to p_1, p_3, p_5; let T_3 denote the subset of vertices of T^* adjacent to p_2, p_5 and not to p_1, p_3, p_4, p_6.

Claim 1. $T^* = T_1 \cup T_2 \cup T_3$.

We prove Claim 1 as follows. Because $T_1 \cup T_2 \cup T_3 \subseteq T^*$ by definition, we only have to prove that $T^* \subseteq T_1 \cup T_2 \cup T_3$. Let $u \in T^*$. Because u has a list of two admissible colors, u is not adjacent to two adjacent vertices of H (as these vertices have different colors). By definition, u has a neighbor $v \in S$.

Case 1. u is adjacent to p_1*.*

Then u is not adjacent to p_2. Hence, p_4p_5 and vup_1p_2 form an induced $P_2 + P_4$ in G unless u is adjacent to a vertex of $\{p_4, p_5\}$.

Suppose u is adjacent to p_4. Then u is neither adjacent to p_3 nor to p_5. Recall that u is not adjacent to p_2. Then u must be adjacent to p_6, as otherwise vup_1p_2 and p_5p_6 form an induced $P_2 + P_4$ in G. However, now we find that p_2p_3 and vup_6p_5 form an induced P_2+P_4 in G. We conclude that u cannot be adjacent to p_4. Because u is not adjacent to p_4 and u must be adjacent to a vertex of $\{p_4, p_5\}$, we find that u is adjacent to p_5. Then u is not adjacent to p_6. Recall that u is not adjacent to p_2. This means that u must be adjacent to p_3, as otherwise p_2p_3 and vup_5p_6 form an induced $P_2 + P_4$ in G. Hence, we obtain $u \in T_1$.

Case 2. u is not adjacent to p_1 *but u is adjacent to* p_2*.*

Then p_4p_5 and vup_2p_1 form an induced $P_2 + P_4$ in G unless u is adjacent to a vertex from $\{p_4, p_5\}$. Suppose u is adjacent to p_4. Then u is not adjacent to p_5. This means that u is adjacent to p_6, as otherwise p_5p_6 and vup_2p_1 form an induced $P_2 + P_4$ in G. Hence, we obtain $u \in T_2$. Suppose u is not adjacent to p_4. Then u must be adjacent to p_5. This means that u is not adjacent to p_6. Because u is adjacent to p_2, we find that u is not adjacent to p_3. Recall that u is not adjacent to p_1. Hence, we obtain $u \in T_3$.

Case 3. u is neither adjacent to p_1 *nor to* p_2 *but u is adjacent to* p_3*.*

Then p_5p_6 and vup_3p_2 form an induced $P_2 + P_4$ in G unless u is adjacent to a vertex from $\{p_5, p_6\}$. Suppose u is adjacent to p_5. Then u is not adjacent to p_6, and we find that p_1p_2 and vup_5p_6 form an induced $P_2 + P_4$ in G. Suppose u is adjacent to p_6. Then u is not adjacent to p_5, and we find that p_1p_2 and vup_6p_5 form an induced $P_2 + P_4$ in G. Both cases are not possible.

Because the remaining cases follow from symmetry, we conclude that $u \in T_1 \cup T_2 \cup T_3$. Hence $T^* \subseteq T_1 \cup T_2 \cup T_3$, and we have proven Claim 1.

Claim 2. *Either* $T_1 \cup T_2$ *or* T_3 *is empty.*

We prove Claim 2 as follows. Assume $T_1 \cup T_2 \neq \emptyset$ and $T_3 \neq \emptyset$. Without loss of generality, assume there is a vertex $u \in T_1$ and a vertex $v \in T_3$. By definition, u is adjacent to p_1, p_3 and p_5. Since u has a list of 2 admissible colors, p_1, p_3 and p_5 are colored by the same color, say color 1. Because p_2 is adjacent to p_1, vertices p_1 and p_2 have different colors. Thus the colors of p_2 and p_5 are different. Then v has only one admissible color in its list. This contradiction proves Claim 2.

Using Claim 2 we distinguish two cases.

Case 1. $T_1 \cup T_2$ is empty and T_3 is not empty.

Since every vertex in T_3 has a list of 2 admissible colors, p_2 and p_5 are colored the same. Recall that S is an independent set. Hence we can safely color all the vertices in S by the same color as p_2 and p_5. We are left to apply Observation 1.

Case 2. T_3 is empty and $T_1 \cup T_2$ is not empty.

If one of T_1 and T_2 is empty, say $T_2 = \emptyset$, we proceed as in Case 1. We now assume that none of T_1 and T_2 is empty. As before, this means that p_1, p_3, p_5 must have the same color, say color 1, whereas p_2, p_4, p_6 also have the same color, say color 2. Recall that S is an independent set. Hence, we can safely color all vertices of S that only have neighbors in T_1 by color 1, and all vertices of S that only have neighbors in T_2 by color 2. Afterwards we remove them from G. If no vertices of S remain we apply Observation 1. Suppose S did not become empty. Then each (remaining) vertex of S has a neighbor in T_1 and T_2. We first try the case that all vertices of T_1 receive color 2. For this coloring of T_1, all vertices in S get reduced lists of size at most 2, so we can again apply Observation 1.

We are left to consider the possibility that color 3 is used on at least one vertex of T_1. We try all possible $|T_1| = O(|V(G)|)$ choices in which we give one fixed vertex $x \in T_1$ color 3. Below we describe what we do for each such choice.

We first update G. If G then only contains vertices that have a list of admissible colors of size 2, we apply Observation 1. Otherwise, we restore x and all vertices of H back into G and redefine sets T_1, T_2 and S accordingly. We find that no vertex in T_2 is adjacent to x, because such vertex would have received color 1 and would have been removed when we were updating G. Furthermore, by definition of S, no vertex in S is adjacent to x, and we may again assume that each vertex in S is adjacent to a vertex in T_1 and to a vertex in T_2.

Let y be an arbitrary vertex of T_2. Suppose there exists an edge ab such that $a \in T_2$, $b \in S$ and y is not adjacent to a, b. Then G contains an induced $P_2 + P_4$ formed by xp_1 and bap_6y. This is not possible. Hence, the vertex y is adjacent to at least one of the vertices of every edge ab with $a \in T_2$ and $b \in S$. We consider both the case in which y gets color 1 and in which y gets color 3. Then, in each case, we reduce the list of admissible colors of each vertex $s \in S$ by at least one, which can be seen as follows. Suppose y gets color $i \in \{1, 3\}$. If s is adjacent to y, then s cannot receive color i. Otherwise, as we just proved, s is adjacent to a neighbor of y in T_2. Because this neighbor is in T_2, it is adjacent to p_6 as well, and consequently it gets color $j = 1$ if $i = 3$ and color $j = 3$ if $i = 1$. This means that s cannot receive color j. Hence, after updating the graph we may apply Observation 1. This finishes Case 2, and thus the description of our algorithm is completed. □

Corollary 1. *The* VERTEX COLORING *problem can be solved in polynomial time for the class of triangle-free $(P_2 + P_4)$-free graphs.*

Proof. We first note that a graph can be colored with at most one color if and only if it consists of isolated vertices only. Secondly, a graph can be colored with at most two colors if and only if it is bipartite. Both cases can clearly be checked in polynomial time. By Theorem 5, we can also check in polynomial time whether a triangle-free $(P_2 + P_4)$-free graph is 3-colorable. We complete the proof by showing that every triangle-free $(P_2 + P_4)$-free graph can always be colored with at most 4 colors. This means that such a graph has chromatic number 4 if it is not 3-colorable.

Let G be a triangle-free $(P_2 + P_4)$-free graph. We may assume that G is connected and that $|V(G)| \geq 5$. Consider an induced P_2 of G, say with vertices v_1 and v_2. Let G' be the graph of G obtained from G after removing v_1, v_2 and all their neighbors. Then G' contains no triangle and no induced P_4, because G is triangle-free and $(P_2 + P_4)$-free. This implies that G' is bipartite. Because G is triangle-free, the set of neighbors of v_1 is an independent set and the set of neighbors of v_2 is an independent set.

Due to the above we can define the following 4-coloring of G. Assign color 1 to all neighbors of v_1 (including v_2) and color 2 to all neighbors of v_2 (including v_1). Assign colors 3 and 4 to all the vertices of G' corresponding to the bipartition of G'. This completes the proof of Corollary 1. $\square$

4.2 3-Precoloring Extension for sP_3-free graphs

In this section we describe how to test in polynomial time whether a given sP_3-free graph admits a 3-coloring extending a given precoloring. Our polynomial-time algorithm heavily relies on a number of structural properties of 3-colorable sP_3-free graphs. Before we present these properties we must first introduce some additional terminology.

Let $G = (V, E)$ be a graph. For a subset $U \subseteq V$ we define $N_G(U) = \{v \in V \setminus U \mid uv \in E \text{ for some } u \in U\}$. A set $D \subseteq V$ *dominates* a set $S \subseteq V$ if $S \subseteq D \cup N_G(D)$; if $S = V$ then we say that D is a *dominating set* of G. Let I be an independent set in G, and let X be a subset of $V \setminus I$. We write $I(X) := N_G(X) \cap I$ and $I(\overline{X}) := I \setminus N_G(X)$, so $I = I(X) \cup I(\overline{X})$ and $I(X) \cap I(\overline{X}) = \emptyset$. If every vertex in $N_G(I) \setminus X$ has exactly one neighbor in $I(\overline{X})$ then we say that X *pseudo-dominates* I. This notion plays a crucial role in the design of our algorithm. An example of a set X that pseudo-dominates a set I is illustrated in Figure 3.

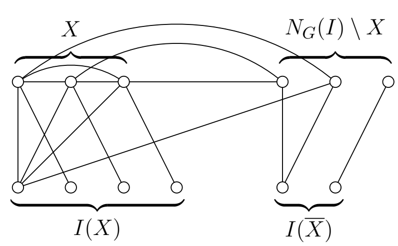

Fig. 3. A set X that pseudo-dominates a set I.

The second key ingredient in our approach is the following well-known result from Ramsey Theory (see, e.g., Bondy and Murty [2]). For positive integers p and q, the *Ramsey number* $r(p,q)$ is the smallest number of vertices n such that all graphs on n vertices contain an independent set of size p or a clique of size q. Ramsey's Theorem states that such a number exists for all positive integers p and q. In fact, by a classical result of Erdős and Szekeres, $r(p,q) \leq \binom{p+q-2}{q-1}$.

Using Ramsey numbers we can prove the following structural result on pseudo-dominating sets.

Lemma 5. *Let I be an independent set in a k-colorable sP_3-free graph $G = (V,E)$ for some integer $s \geq 2$. Then $G[V\backslash I]$ contains a set X of cardinality at most $r(s,k+1)$ that pseudo-dominates I.*

Proof. We may assume that $G[V\backslash I]$ contains more than $r(s,k+1)$ vertices; otherwise there is nothing to prove. Let X be a subset of $V \setminus I$ with $r(s,k+1)$ vertices that dominates the maximum number of vertices in I over all subsets of $V \setminus I$ of size $r(s,k+1)$.

Suppose that X does not pseudo-dominate I. Then there is a vertex $v \in N_G(I) \setminus X$ adjacent to at least two vertices in $I(\overline{X})$. This means that any vertex $x \in X$ has two neighbors u_x, w_x in I that are not adjacent to any vertex in $X\backslash\{x\}$; otherwise the set $X' = (X \cup \{v\}) \setminus \{x\}$ of size $r(s,k+1)$ dominates more vertices than X, contradicting the maximality of X. From the definition of Ramsey number $r(s,k+1)$, we find that X has an independent set $\{x_1,\ldots,x_s\}$ or a clique of size $k+1$. In the first case we have an induced sP_3 consisting of the s paths $u_{x_i}x_iw_{x_i}$, contradicting the sP_3-freeness of G. In the second case, when X has a clique of $k+1$ vertices, G is not k-colorable, which contradicts our assumptions as well. This completes the proof. $\square$

We note that the upper bound on the size of X in Lemma 5 can be slightly improved for $s = 2$, as we show in a recent paper [4].

Before we prove our main result of this section, we need two more lemmas.

Lemma 6. *Let G be an sP_3-free graph that contains a set X and an independent set I, such that X pseudo-dominates I. Let $k \geq 1$. If $I(\overline{X})$ contains more than $k(s-1)$ vertices with degree at least k in G, then G is not k-colorable.*

Proof. Let G, I and X be defined as in the statement of the lemma. Let $k \geq 1$. Let S be the subset of vertices in $I(\overline{X})$ that have degree at least k in G. Suppose $|S| > k(s-1)$. In order to derive a contradiction, assume G has a k-coloring c.

Every vertex in G with degree at least k must have at least two neighbors with the same color in c. Thus, because every vertex in S has degree at least k, every vertex in S has at least two neighbors with the same color. Because S contains more than $k(s-1)$ vertices, this means that there exist s vertices $u_1,\ldots,u_s$ in S and a color j such that each u_i has (at least) two neighbors x_i and y_i with color j. The set $\{x_1,\ldots x_s\} \cup \{y_1,\ldots,y_s\}$ is an independent set because all its vertices have the same color, namely color j. Because X pseudo-dominates I, a vertex u_i is neither adjacent to x_h nor to y_h whenever $h \neq i$. This implies that the s paths $x_iu_iy_i$ form an induced sP_3 in G, which is not possible. This contradiction yields that G is not k-colorable. ◻

Lemma 7. *Let $G = (V, E)$ be an sP_3-free graph with a set $W \subseteq V$ such that each vertex in W is precolored with a color from $\{1, \ldots, k\}$ and every vertex in $V \setminus W$ has degree at least k for some integer k. If G has a k-coloring extending the precoloring on W, then G contains a set D of size at most $k \cdot\ r(s, k+1) + (k^2 + 3) \cdot (s - 1)$ that dominates $V \setminus W$.*

Proof. Let G be an sP_3-free graph such that every vertex in $V(G) \setminus W$ has degree at least k for some integer k. Assume that G has a k-coloring extending the precoloring on W. Then G is K_{k+1}-free. If $s = 1$, then every component of G is a complete graph at most k vertices. Consequently, G has no vertex of degree at least k, and the statement of the lemma holds. Assume that $s \geq 2$ and assume without loss of generality that G is not $(s-1)P_3$-free. Let S be the vertex set of an induced subgraph of G that is isomorphic to $(s-1)P_3$, hence $|S| = 3(s - 1)$. If S is a dominating set of G, then the statement of the lemma holds.

Suppose S is not a dominating set of G. Let G' be the graph obtained from G after removing $S \cup W$ and all vertices in $N_G(S)$. Because G is (K_{k+1}, sP_3)-free and S induces an $(s - 1)P_3$, we find that G' is (K_{k+1}, P_3)-free. Hence, every component of G' is isomorphic to a graph from $\{K_1, \ldots, K_k\}$.

We partition the vertices of G' into at most k independent sets $I_1, \ldots, I_k$ as follows. First we form I_1 by taking exactly one vertex from each component of G'. We remove I_1 from G' and repeat the above step to obtain I_2 if there were any vertices of G' left. We proceed in this way until all vertices of G' have been used. This will happen after at most k steps, because every component of G' has at most k vertices at the start of this procedure.

We apply Lemma 5 to each I_h in order to find a set X_h with $|X_h| \leq r(s, k+1)$ in G that pseudo-dominates I_h for $h = 1, \ldots, k$.

We apply Lemma 6 to G and each I_h in order to find that $|I_h(\overline{X_h})| \leq k(s-1)$ for $h = 1, \ldots, k$. Then $D = S \cup X_1 \cup \cdots \cup X_k \cup I_1(\overline{X_1}) \cup \cdots \cup I_k(\overline{X_k})$ has at most $3(s-1) + k \cdot r(s, k+1) + k \cdot k \cdot (s-1) = k \cdot\ r(s, k+1) + (k^2 + 3) \cdot (s - 1)$ vertices. Because D is a dominating set in G, this completes the proof of Lemma 7. ◻

We observe that Lemma 7 involves a minimum degree condition. This is not a problem for our algorithm due to the following procedure. Let $G = (V, E)$ be a graph with a set $W \subseteq V$ such that each vertex in W is precolored with a color from $\{1, \ldots, k\}$. Remove all vertices of $V \setminus W$ with degree at most $k - 1$ from G. Propagate this until we obtain a graph $G^*_{\geq k}$ such that all vertices not in W have degree at least k. Note that $G^*_{\geq k}$ may be the empty graph (this happens when $W = \emptyset$ and G is $(k - 1)$-degenerate). We observe the following.

Observation 2 *Let $G = (V, E)$ be a graph with a set $W \subseteq V$ such that each vertex in W is precolored with a color from $\{1, \ldots, k\}$ for some fixed integer k. Then $G^*_{\geq k}$ can be obtained in polynomial time, and $G^*_{\geq k}$ has a k-coloring extending the precoloring of W if and only if G has a k-coloring extending the precoloring of W. Furthermore, for any set $\mathcal{H}$ of graphs, $G^*_{\geq k}$ is $\mathcal{H}$-free if G is $\mathcal{H}$-free.*

Proof. Because we must search for at most $|V|$ times for a vertex of degree at most $k-1$, the graph $G^*_{\geq k}$ can be obtained in polynomial time. Let $\mathcal{H}$ be a set of graphs. Because we only remove vertices in order to get $G^*_{\geq k}$, we find that $G^*_{\geq k}$ is $\mathcal{H}$-free if G is $\mathcal{H}$-free.

We now prove the remaining statement. If G has a k-coloring extending the precoloring of W then so has $G^*_{\geq k}$, because $G^*_{\geq k}$ is a subgraph of G. To prove the reverse implication, suppose that $G^*_{\geq k}$ has a k-coloring extending the precoloring of W. By considering the vertices of $V \backslash V(G^*_{\geq k})$ in the reverse order of their removal, there is always a color from $\{1, \ldots, k\}$ available to color them. $\square$

We are now ready to prove the main result of this section.

Theorem 6. *The* 3-Precoloring Extension *problem can be solved in polynomial time for the class of sP_3-free graphs for any fixed integer $s \geq 1$.*

Proof. Let $G = (V, E)$ be an sP_3-free graph with a set $W \subseteq V$ such that each vertex in W is precolored with a color from $\{1, 2, 3\}$ for some fixed integer k. By Observation 2 we may assume that every vertex in $V \backslash W$ has degree at least 3. By brute force, we search for a set D of size at most $3r(s, 4)+12(s-1)$ that dominates $V \setminus W$. This takes $O(|V|^{3r(s,4)+12(s-1)})$ time, which is polynomial because s is fixed. If such a set does not exist, then the precoloring of W cannot be extended to a 3-coloring of G due to Lemma 7. Otherwise, if we find such a dominating set D, then we guess a possible 3-coloring of D that respects the precoloring on W and we apply Observation 1. Because $|D| \leq 3r(s, 4)+12(s-1)$, the total number of these guesses is $3^{|D|} \leq 3^{3r(s,4)+12(s-1)}$. This number is constant because s is fixed. Hence, our algorithm runs in polynomial time. $\square$

5 Future research

The complexity status of the 3-Coloring and 3-Precoloring Extension problem restricted to H-free graphs is open for many linear forests H on seven or more vertices, in particular for paths. As we can see from Table 1, it is even unknown whether there exists a fixed integer $k \geq 7$ such that 3-Precoloring Extension is NP-complete for P_k-free graphs. This indicates how difficult these complexity questions are and puts the results of this paper in the right perspective. For larger values of ℓ, Table 1 shows that more is known on the complexity status of the ℓ-Coloring problem restricted to P_k-free graphs. The currently sharpest known results are that 4-Coloring is NP-complete for P_8-free graphs and that 6-Coloring is NP-complete for P_7-free graphs. It is an open problem whether there exists an integer ℓ such that ℓ-Coloring is NP-complete for P_6-free graphs.

Another question we would like to answer is whether the ℓ-Coloring problem and the ℓ-Precoloring Extension problem always have the same computational complexity for P_k-free graphs. Is there a pair (k, ℓ) such that the

ℓ-Coloring problem is polynomial-time solvable for P_k-free graphs whereas the ℓ-Precoloring Extension problem is NP-complete for P_k-free graphs?

One can explore various directions to extend the polynomial-time results in this paper. As an example, determining the complexity of the following problems is still open.

1. k-Coloring for $2P_3$-free graphs for any fixed $k \geq 4$;
2. k-Coloring for triangle-free sP_3-free graphs for any fixed $s \geq 3$ and $k \geq 4$;
3. 3-Coloring for triangle-free P_7-free graphs.

Note that we proved the polynomial-time complexity of 4-Coloring for triangle-free $2P_3$-free graphs in a recent paper [4], yielding that Vertex Coloring for triangle-free $2P_3$-free graphs is polynomial-time solvable. We also note that Dabrowski et al. [6] showed that Vertex Coloring is polynomial-time solvable for triangle-free sP_2-free graphs for any fixed $s \geq 2$.

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, In: IWOCA 2009, LNCS 5874, 95–104 (2009).
4. 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, http://www.dur.ac.uk/daniel.paulusma/Papers/Submitted/2p3.pdf.
5. D. Bruce, C.T. Hoàng, and J. Sawada, A certifying algorithm for 3-colorability of P_5-free graphs, Proceedings of the 20th International Symposium on Algorithms and Computation (ISAAC 2009), Lecture Notes in Computer Science 5878, 594-604 (2009).
6. K. Dabrowski, V. Lozin, R. Raman and B. Ries, Colouring vertices of triangle-free graphs, Proceedings of the 36th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2010), Lecture Notes in Computer Science, to appear. preprint available at http://www.warwick.ac.uk/∼mariaq/Triangle-free.pdf
7. K. Edwards, The complexity of coloring problems on dense graphs. Theoret. Comput. Sci. 43, 337–343 (1986).
8. M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco (1979).
9. M. Grötschel, L. Lovász, and A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, Combinatorica 1, 169–197 (1981).
10. 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).
11. 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).
12. M. Kochol, V.V. Lozin and B. Randerath, The 3-Colorability Problem on Graphs with Maximum Degree Four, SIAM J. Comput. 32, 1128–1139 (2003).

13. D. Král', J. Kratochvíl, Zs. Tuza, and G.J. Woeginger, Complexity of coloring graphs without forbidden induced subgraphs, In: WG 2001, LNCS 2204, 254–262 (2001).
14. J. Kratochvíl, Precoloring extension with fixed color bound, Acta Math. Univ. Comen. 62, 139–153 (1993).
15. 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).
16. B. Randerath and I. Schiermeyer, 3-Colorability $\in$ P for P_6-free graphs, Discrete Appl. Math. 136, 299–313 (2004).
17. B. Randerath and I. Schiermeyer, Vertex colouring and forbidden subgraphs - a survey, Graphs Combin. 20, 1–40 (2004).
18. T. J. Schaefer, The complexity of satisfiability problems, in Conference Record of the Tenth Annual ACM Symposium on Theory of Computing (San Diego, Calif., 1978), ACM, New York, 1978, pp. 216–226.
19. 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).
20. Zs. Tuza, Graph colorings with local restrictions - a survey, Discuss. Math. Graph Theory 17, 161–228 (1997).
21. G.J. Woeginger and J. Sgall, The complexity of coloring graphs without long induced paths, Acta Cybernet. 15, 107–117 (2001).