Narrowing Down the Gap on the Complexity of Coloring P_k-Free Graphs

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 P_k-free if it does not contain an induced subgraph isomorphic to a path on k vertices. We show that deciding whether a P_8-free graph can be colored with at most four colors is an NP-complete problem. 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 pre-coloring extension version of 4-coloring is NP-complete for P_7-free graphs, but that the pre-coloring extension version of 3-coloring is polynomially solvable for $(P_2 + P_4)$-free graphs, a subclass of P_7-free graphs.

1 Introduction

Due to the fact that the usual 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 [14, 17] for surveys. Instead of repeating what has been written in so many papers over the years, and in order to save as much space as possible for relevant details related to our results, we also refer to these surveys for motivation and background. Here we continue the study of ℓ-Coloring for P_k-free graphs. This setting has been studied in several earlier papers by different groups of researchers (see, e.g., [3, 5, 9–13, 18]). Before we summarize their results we first introduce the necessary terminology.

Terminology. We only consider finite undirected graphs without loops and multiple edges. We refer to [2] for any undefined graph terminology. The graph P_k denotes the path 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. Given two graphs G and H we say that 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*

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

D.M. Thilikos (Ed.): WG 2010, LNCS 6410, pp. 63–74, 2010.

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 $U \subseteq V$. We let $\chi(G)$ denote the *chromatic number* of G, i.e., the smallest ℓ such that G has an ℓ-coloring. The problem ℓ-Coloring is the problem to decide whether a given graph admits an ℓ-coloring.

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 *pre-coloring extension* we assume that a (possibly empty) subset $W \subseteq V$ of G is pre-colored 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 extend it to an ℓ-coloring of G, we say we deal with the *pre-coloring extension version of ℓ-*Coloring.

Known results. Results of Hoàng et al. [9] imply that the pre-coloring extension version of ℓ-Coloring is polynomially solvable on P_5-free graphs for any fixed ℓ. In contrast, determining the chromatic number is NP-hard for P_5-free graphs [10], whereas this problem is polynomially 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 [8]). Le, Randerath, and Schiermeyer [12] proved that 4-Coloring is NP-complete for P_9-free graphs. Woeginger and Sgall [18] showed that 5-Coloring is NP-complete for P_8-free graphs. In [3] we established the following three results. Firstly we proved that 6-Coloring is NP-complete for P_7-free graphs, secondly that the pre-coloring extension version of 3-Coloring is polynomially solvable for P_6-free graphs, and thirdly that the pre-coloring extension version of 5-Coloring is NP-complete for P_6-free graphs. All these results together lead to the following table that shows the current status of ℓ-Coloring and its pre-coloring extension version for P_k-free graphs. This table also shows which cases are still open.

Table 1. The complexity of ℓ-Coloring and its pre-coloring extension version (marked by *) on P_k-free graphs for combinations of fixed k and ℓ

P_k-free \ $\ell \to$	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
$k = 8$	?	?	?	?	NP-c	NP-c	NP-c	NP-c
$k \geq 9$	?	?	NP-c	NP-c	NP-c	NP-c	NP-c	NP-c

Our results and paper organization. In Section 2 we present a common improvement to results in [12] and [18] by showing that 4-Coloring is NP-complete for P_8-free graphs. In Section 3 we give a closely related result showing that the pre-coloring extension version of 4-Coloring is NP-complete for P_7-free graphs. It seems hard to extend our result from [3] on the pre-coloring extension version of 3-Coloring for P_6-free graphs to P_7-free graphs. This motivates our focus on subclasses of P_7-free graphs, namely H-free graphs, where H is a linear forest on at most 6 vertices. We show in Section 4 that the first nontrivial case is $H = P_2 + P_4$ and that the pre-coloring extension version of 3-Coloring is polynomially solvable for $(P_2{+}P_4)$-free graphs. Section 5 contains the conclusions and mentions open problems.

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 3-Satisfiability (3SAT), which is an NP-complete problem [7]. We consider an arbitrary instance I of 3SAT 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 complete this section by proving 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.

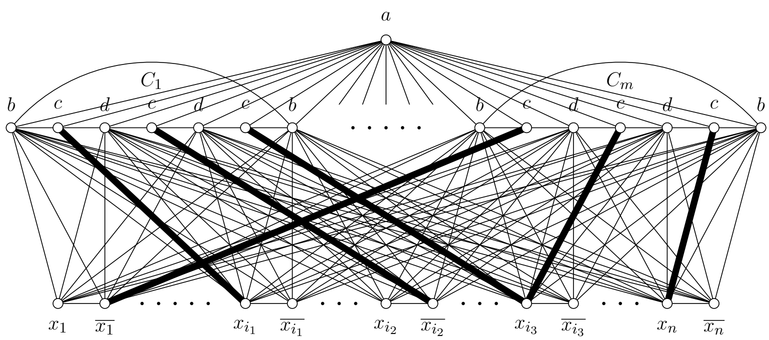

Fig. 1. The graph G_I in which clauses $C_1 = \{x_{i_1}, \overline{x}_{i_2}, x_{i_3}\}$ and $C_m = \{\overline{x}_1, x_{i_3}, x_n\}$ are illustrated

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.

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. □

3 Pre-coloring Extension of 4-Coloring for P_7-Free Graphs

In this section we show that the pre-coloring extension version of 4-Coloring is NP-complete for the class of P_7-free graphs. We use a reduction from Not-All-Equal 3-Satisfiability with positive literals only, which we denote as NAE 3SATPL. This NP-complete problem [15] 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 NAE 3SATPL 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 pre-coloring on some vertices of G_I^*. Then we show that G_I^* is P_7-free and that the pre-coloring 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 pre-coloring.

- 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 pre-color 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 (pre-colored) 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.

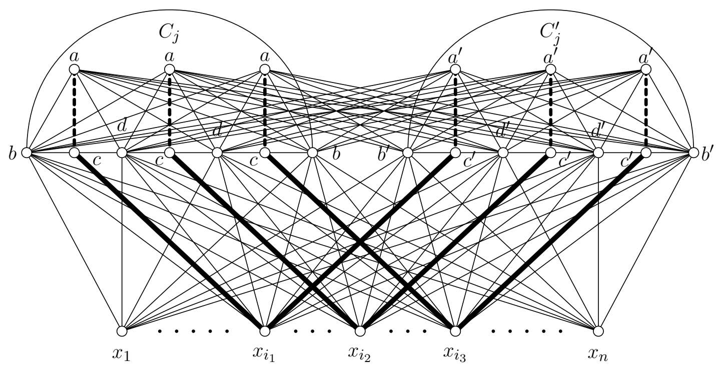

Fig. 2. The graph G_I^* for the clause $C_j = \{x_{i_1}, x_{i_2}, x_{i_3}\}$

We complete this section by proving two lemmas. Lemma 3 shows that the graph G_I^* is P_7-free. Its proof is postponed to the journal version of our paper. In Lemma 4 we prove that the pre-coloring 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.

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

Lemma 4. *The pre-coloring 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 pre-coloring 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 pre-coloring 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 pre-coloring. 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 pre-coloring. We color all other remaining uncolored vertices in a straightforward way. This completes the proof of Lemma 4. □

4 Pre-coloring Extension of 3-Coloring for Subclasses of P_7-Free Graphs

Here we consider the pre-coloring extension version of 3-Coloring 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 [3] for solving this problem when H is an induced subgraph of P_6, because then any H-free graph is also P_6-free. Then the following 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 need the following observation, the proof of which follows from the fact that the decision problem in this case can be modeled and solved as a 2SAT-problem. This approach has been introduced by Edwards [6] and is folklore now, see also [9] and [13].

Observation 1 ([6]). *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 the pre-coloring extension version of 3-*COLORING *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 pre-coloring $\phi_W : W \rightarrow \{1, 2, 3\}$ 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 the polynomial-time algorithm of [3] that solves the pre-coloring extension version of 3-COLORING for P_6-free graphs, and (repeatedly) applying Proposition 1 yields polynomial-time results of the same problem for H_i-free graphs for $i = 1, \ldots, 12$.

The case $H_{13} = 3P_2$ follows from the more general result that 3-COLORING is polynomial-time solvable for sP_2-free graphs for any $s \geq 1$. This is known already and can be seen as follows. Balas and Yu [1] showed that for any $s \geq 1$ the number of maximal independent sets in an sP_2-free graph $G = (V, E)$ is bounded by a polynomial. These maximal independent sets can then be efficiently enumerated by applying the algorithm of Tsukiyama, Ide, Ariyoshi and Shirakawa [16]. We note that G has a 3-coloring if and only if V can be partitioned into at most 3 independent sets V_1, V_2, V_3, one of which may be assumed to be maximal. Hence, for each maximal independent set I in G we check if the subgraph induced by $V\backslash I$ is bipartite. This can be done in polynomial time.

In Section 4.1 we consider H_{14}.

4.1 Pre-coloring Extension of 3-Coloring for $(P_2 + P_4)$-Free Graphs

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

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 of [3] to solve our 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 removed all vertices of H. We restore them 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 G is 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.

Because every vertex $u \in T$ has a list of two admissible colors, u is not adjacent to two adjacent vertices of H (as these vertices have different colors). By considering a vertex in T^* together with one of its neighbors in S and using the $(P_2 + P_4)$-freeness of G, we then find that $T^* = T_1 \cup T_2 \cup T_3$.

Claim 1. *Either $T_1 \cup T_2$ or T_3 is empty.*

We prove Claim 1 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 1.

Using Claim 1 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

Table 2. An update of Table 1

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

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 $O(|V(G)|)$ 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 bap_6y and xp_1. 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 all possible colorings of y. This way we reduce the list of admissible colors of each vertex in S by at least one (either directly or via one of its neighbors in T_2) and we apply Observation 1. This finishes Case 2, and thus the description of our algorithm is completed.

5 Conclusions

Due to our new results we can update Table 1. This yields Table 2. Positions in this table marked by "?" are still open. We also showed that 3-Coloring is polynomial-time solvable for H-free graphs if H is any fixed linear forest on at most 6 vertices, except when $H = 2P_3$. Recently, we showed that 3-Coloring is also polynomial-time solvable for $2P_3$-free graphs [4].

References

1. Balas, E., Yu, C.S.: On graphs with polynomially solvable maximum-weight clique problem. Networks 19, 247–253 (1989)
2. Bondy, J.A., Murty, U.S.R.: Graph Theory. In: Springer Graduate Texts in Mathematics, vol. 244 (2008)
3. Broersma, H.J., Fomin, F.V., Golovach, P.A., Paulusma, D.: Three complexity results on coloring P_k-free graphs. In: Fiala, J., Kratochvíl, J., Miller, M. (eds.) IWOCA 2009. LNCS, vol. 5874, pp. 95–104. Springer, Heidelberg (2009)
4. Broersma, H.J., Golovach, P.A., Paulusma, D., Song, J.: On coloring graphs without induced forests. In: ISAAC 2010 (to appear, 2010)
5. Bruce, D., Hoàng, C.T., Sawada, J.: A certifying algorithm for 3-colorability of P_5-free graphs. In: Dong, Y., Du, D.-Z., Ibarra, O. (eds.) ISAAC 2009. LNCS, vol. 5878, pp. 594–604. Springer, Heidelberg (2009)
6. Edwards, K.: The complexity of coloring problems on dense graphs. Theoret. Comput. Sci. 43, 337–343 (1986)
7. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)
8. Grötschel, M., Lovász, L., Schrijver, A.: The ellipsoid method and its consequences in combinatorial optimization. Combinatorica 1, 169–197 (1981)
9. Hoàng, C.T., Kamiński, M., Lozin, V., Sawada, J., Shu, X.: Deciding k-colorability of P_5-free graphs in polynomial time. Algorithmica 57, 74–81 (2010)
10. Král', D., Kratochvíl, J., Tuza, Z., Woeginger, G.J.: Complexity of coloring graphs without forbidden induced subgraphs. In: Brandstädt, A., Le, V.B. (eds.) WG 2001. LNCS, vol. 2204, pp. 254–262. Springer, Heidelberg (2001)
11. Kratochvíl, J.: Precoloring extension with fixed color bound. Acta Math. Univ. Comen. 62, 139–153 (1993)
12. Le, V.B., Randerath, B., Schiermeyer, I.: On the complexity of 4-coloring graphs without long induced paths. Theoret. Comput. Sci. 389, 330–335 (2007)
13. Randerath, B., Schiermeyer, I.: 3-Colorability $\in$ P for P_6-free graphs. Discrete Appl. Math. 136, 299–313 (2004)
14. Randerath, B., Schiermeyer, I.: Vertex colouring and forbidden subgraphs - a survey. Graphs Combin. 20, 1–40 (2004)
15. Schaefer, T.J.: The complexity of satisfiability problems. In: Conference Record of the Tenth Annual ACM Symposium on Theory of Computing, San Diego, Calif., pp. 216–226. ACM, New York (1978)
16. Tsukiyama, S., Ide, M., Ariyoshi, H., Shirakawa, I.: A new algorithm for generating all the maximal independent sets. SIAM J. Comput. 6, 505–517 (1977)
17. Tuza, Z.: Graph colorings with local restrictions - a survey. Discuss. Math. Graph Theory 17, 161–228 (1997)
18. Woeginger, G.J., Sgall, J.: The complexity of coloring graphs without long induced paths. Acta Cybernet. 15, 107–117 (2001)