



ELSEVIER

Contents lists available at SciVerse ScienceDirect

European Journal of Combinatorics

journal homepage: www.elsevier.com/locate/ejc

Three complexity results on coloring P_k -free graphs[☆]

Hajo Broersma^{a,1}, Fedor V. Fomin^b, Petr A. Golovach^b, Daniël Paulusma^a

^a School of Engineering and Computing Sciences, Durham University, DH1 3LE Durham, United Kingdom

^b Department of Informatics, University of Bergen, PB 7803, 5020 Bergen, Norway

ARTICLE INFO

Article history:

Available online 24 August 2012

ABSTRACT

We prove three complexity results on vertex coloring problems restricted to P_k -free graphs, i.e., graphs that do not contain a path on k vertices as an induced subgraph. First of all, we show that the pre-coloring extension version of 5-coloring remains NP-complete when restricted to P_6 -free graphs. Recent results of Hoàng et al. imply that this problem is polynomially solvable on P_5 -free graphs. Secondly, we show that the pre-coloring extension version of 3-coloring is polynomially solvable for P_6 -free graphs. This implies a simpler algorithm for checking the 3-colorability of P_6 -free graphs than the algorithm given by Randerath and Schiermeyer. Finally, we prove that 6-coloring is NP-complete for P_7 -free graphs. This problem was known to be polynomially solvable for P_5 -free graphs and NP-complete for P_8 -free graphs, so there remains one open case.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

In this paper we consider computational complexity issues related to vertex coloring problems restricted to P_k -free graphs. Due to the fact that the usual ℓ -COLORING problem is NP-complete for any fixed $\ell \geq 3$, there has been considerable interest in studying its complexity when restricted to certain graph classes. Without doubt one of the most well-known results in this respect is that ℓ -COLORING is polynomially solvable for perfect graphs. More information on this classic result and

[☆] An extended abstract of this paper appeared in the proceedings of IWOCA 2009.

E-mail addresses: hajo.broersma@durham.ac.uk, h.j.broersma@utwente.nl (H. Broersma), fedor.fomin@ii.uib.no (F.V. Fomin), petr.golovach@ii.uib.no (P.A. Golovach), daniel.paulusma@durham.ac.uk (D. Paulusma).

¹ Current address: Faculty of EEMCS, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands. Tel.: +44 1913341712.

related work on coloring problems restricted to graph classes can be found in, e.g., [11,13]. Instead of repeating what has been written in so many papers over the years, we also refer to these surveys for motivation and background. Here we continue the study of ℓ -COLORING and its variants for P_k -free graphs, a problem that has been studied in several earlier papers by different groups of researchers (see, e.g., [2,3,6,8–10,15]). We summarize all these results in the table in Section 5.

1.1. Terminology

We refer to [1] for standard graph theory terminology and to [5] for terminology on computational complexity.

Let $G = (V, E)$ be a graph and k a positive integer. We say that G is P_k -free if G does not have a path on k vertices as an induced subgraph.

A (vertex) coloring of a graph $G = (V, E)$ is a mapping $\phi : V \rightarrow \{1, 2, \dots\}$ such that $\phi(u) \neq \phi(v)$ whenever $uv \in E$. Here $\phi(u)$ is usually referred to as the color of u in the coloring ϕ of G . An ℓ -coloring of G is a mapping $\phi : V \rightarrow \{1, 2, \dots, \ell\}$ such that $\phi(u) \neq \phi(v)$ whenever $uv \in E$. The problem ℓ -COLORING asks if a given graph has an ℓ -coloring.

In list-coloring we assume that $V = \{v_1, v_2, \dots, v_n\}$ and that for every vertex v_i of G there is a list L_i of admissible colors (a subset of the natural numbers). Given these lists, a list-coloring of G is a coloring $\phi : V \rightarrow \{1, 2, \dots\}$ such that $\phi(v_i) \in L_i$ for all $i \in \{1, 2, \dots, n\}$; we say that ϕ respects the lists L_i .

In pre-coloring extension we assume that a (possibly empty) subset $W \subseteq V$ of G is pre-colored with $\phi_W : W \rightarrow \{1, 2, \dots\}$ and the question is whether we can extend ϕ_W to a coloring of G . If ϕ_W is restricted to $\{1, 2, \dots, \ell\}$ 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. In fact, we consider a slight variation on the latter problem which can be considered as list coloring, but which has the flavor of pre-coloring: lists have varying sizes including some of size 1. We will slightly abuse terminology and call these problems pre-coloring extension problems too.

1.2. Results of this paper

We prove the following three complexity results on vertex coloring problems restricted to P_k -free graphs.

First of all, in Section 2 we show that the pre-coloring extension version of 5-COLORING remains NP-complete when restricted to P_6 -free graphs. Recent results of Hoàng et al. [6] imply that this problem is polynomially solvable on P_5 -free graphs. Their algorithm for ℓ -COLORING for any fixed ℓ is in fact a list-coloring algorithm where the lists are from the set $\{1, 2, \dots, \ell\}$.

Secondly, in Section 3 we show that the pre-coloring extension version of 3-COLORING is polynomially solvable for P_6 -free graphs. The 3-COLORING problem was known to be polynomially solvable for P_6 -free graphs from a paper by Randerath and Schiermeyer [10]. Their approach is as follows. First they note that the input graph G may be assumed to be K_4 -free, i.e., does not contain a complete graph on four vertices as a subgraph, as otherwise it is not 3-colorable. Their algorithm then determines if G contains a C_5 . If so, it exploits the existence of this C_5 in G in a clever way. If not, the authors use the Strong Perfect Graph Theorem to deduce that G is perfect. This allows them to use the polynomial time algorithm of Tucker [12] for finding a χ -coloring of a K_4 -free perfect graph. Here χ denotes the *chromatic number* of a graph, i.e., the smallest ℓ such that the graph is ℓ -colorable. We follow a different approach. First, our algorithm is independent of the Strong Perfect Graph Theorem, and second it uses a recent characterization of P_6 -free graphs in terms of dominating subgraphs [14]. This way we can indeed show that the pre-coloring extension version of 3-COLORING is polynomially solvable for P_6 -free graphs, whereas the approach of Randerath and Schiermeyer [10] does not immediately lead to this result. The reason for this lies in the second part of their algorithm that focuses on K_4 -free perfect graphs. Already for a subclass of this class, namely the class of bipartite graphs, Kratochvíl [8] showed that the pre-coloring extension version of 3-COLORING is an NP-complete problem.

Finally, in Section 4 we show that 6-COLORING is NP-complete for P_7 -free graphs. This problem was known to be polynomially solvable for P_5 -free graphs [6] and NP-complete for P_8 -free graphs [15], so there remains one open case.

2. Pre-coloring extension of 5-coloring for P_6 -free graphs

In this section we show that the pre-coloring extension version of 5-COLORING remains NP-complete when restricted to P_6 -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 [5] is also known as HYPERGRAPH 2-COLORABILITY and is defined as follows. Given a set $X = \{x_1, x_2, \dots, x_n\}$ of logical variables, and a set $\mathcal{C} = \{C_1, C_2, \dots, C_m\}$ 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 and define a graph G_I and a pre-coloring on some vertices of G_I , and next we show that G_I is P_6 -free and that the pre-coloring on G_I can be extended to a 5-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.

Theorem 1. *The pre-coloring extension version of 5-COLORING is NP-complete for P_6 -free graphs.*

Proof. Let I be an instance of NAE 3SATPL with variables $\{x_1, x_2, \dots, x_n\}$ and clauses $\{C_1, C_2, \dots, C_m\}$. We define a graph G_I corresponding to I and lists of admissible colors for its vertices based on the following construction. We note here that the lists we introduce below are only there for convenience to the reader; it will be clear later that all lists other than $\{1, 2, \dots, 5\}$ are in fact forced by the pre-colored vertices.

1. We introduce one new vertex for each of the clauses, and use the same labels C_1, C_2, \dots, C_m for these m vertices; we assume that for each of these vertices there is a list $\{1, 2, 3\}$ of admissible colors. We say that these vertices are of C -type and use \mathcal{C} to denote the set of C -type vertices.
2. We introduce one new vertex for each of the variables, and use the same labels x_1, x_2, \dots, x_n for these n vertices; we assume that for each of these vertices there is a list $\{4, 5\}$ of admissible colors. We say that these vertices are of x -type and use \mathcal{X} to denote the set of x -type vertices.
3. We join all C -type vertices to all x -type vertices to form a complete bipartite graph with $|\mathcal{C}| |\mathcal{X}|$ edges.
4. For each clause C_j we fix an arbitrary order of its variables x_i, x_k , and x_r , and we introduce three pairs of new vertices $\{a_{i,j}, b_{i,j}\}, \{a_{k,j}, b_{k,j}\}, \{a_{r,j}, b_{r,j}\}$; we assume the following lists of admissible colors for these three pairs, respectively: $\{\{1, 4\}, \{2, 5\}\}, \{\{2, 4\}, \{3, 5\}\}, \{\{3, 4\}, \{1, 5\}\}$. We say that these vertices are of a -type and b -type, and use \mathcal{A} and \mathcal{B} to denote the set of a -type and b -type vertices, respectively. We add edges between x -type and a -type vertices whenever the first index of the a -type vertex is the same as of the x -type vertex, and similarly for the b -type vertices. We add edges between C -type and a -type vertices whenever the second index of the a -type vertex is the same as the index of the C -type vertex, and similarly for the b -type vertices. Hence each clause with three variables is represented by three 4-cycles that have one C -type vertex in common.
5. For each a -type vertex we introduce a copy of a $K_{2,3}$, as follows: for $a_{i,j}$ we add five vertices $\{p_{i,j,1}, \dots, p_{i,j,5}\}$, and we add all edges between $\{p_{i,j,1}, p_{i,j,2}, p_{i,j,3}\}$ and $\{p_{i,j,4}, p_{i,j,5}\}$. We say that these vertices are of p -type and use \mathcal{P} to denote the set of p -type vertices. We add edges between each a -vertex and the p -vertices of its corresponding $K_{2,3}$ depending on its list of admissible colors. In particular, we join the a -vertex to the three p -vertices of its $K_{2,3}$ that have a third index which is not in its list of admissible colors. So, if $a_{i,j}$ has list $\{1, 4\}$, we join it to $p_{i,j,2}, p_{i,j,3}, p_{i,j,5}$. We use \mathcal{P}_1 to denote the set of all p -type vertices with the third index in $\{1, 2, 3\}$ and $\overline{\mathcal{P}}_1$ to denote all other p -type vertices.
6. For each b -type vertex we introduce a new copy of a $K_{2,3}$ on five vertices of q -type, in the same way as we introduced the p -type vertices for the a -type vertices. Edges are added in a similar way, depending on the indices and the lists. We use \mathcal{Q} to denote the set of q -type vertices, \mathcal{Q}_1 to denote the set of all q -type vertices with third index in $\{1, 2, 3\}$ and $\overline{\mathcal{Q}}_1$ to denote all other q -type vertices.

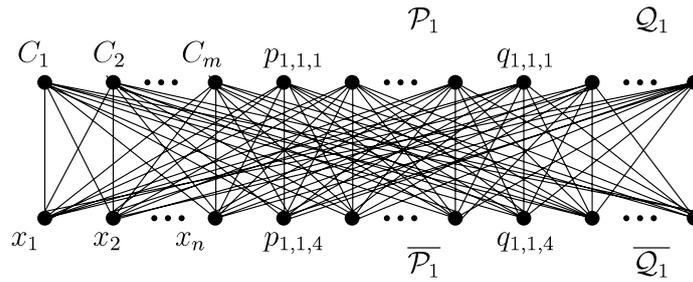


Fig. 1. The (complete bipartite) subgraph of G_I induced by vertices of type C, p, q, x .

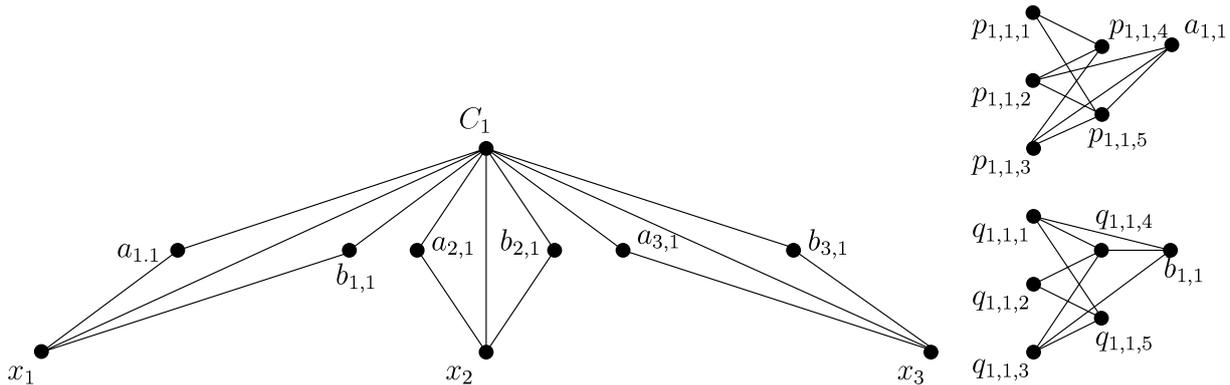


Fig. 2. (i) The subgraph of G_I for clause C_1 with ordered variables x_1, x_2, x_3 . (ii) How $a_{1,1}$ and $b_{1,1}$ are connected to \mathcal{P} and \mathcal{Q} , respectively.

7. We join all the p -type and q -type vertices with third indices 1, 2, 3 to all the p -type and q -type vertices with third indices 4, 5 to form a complete bipartite graph with $|\mathcal{P}_1 \cup \mathcal{Q}_1| |\overline{\mathcal{P}_1} \cup \overline{\mathcal{Q}_1}|$ edges.
8. We join all x -type vertices to all p -type and q -type vertices with third indices 1, 2, 3.
9. We join all C -type vertices to all p -type and q -type vertices with third indices 4, 5.
10. We pre-color all the p -type and q -type vertices according to their third index, so $p_{i,j,\ell}$ will be pre-colored with color $\ell \in \{1, 2, \dots, 5\}$. Note that we can now in fact replace all lists introduced earlier by $\{1, 2, \dots, 5\}$, since the shorter lists will be forced by the given pre-coloring.

See Figs. 1 and 2 for sketches of the ingredients in the construction of the graph G_I ; in Fig. 2 we illustrate an example in which C_1 is a clause with ordered variables x_1, x_2, x_3 .

We now prove that G_I is P_6 -free. In order to obtain a contradiction, suppose that the graph G_I contains an induced subgraph H that is isomorphic to P_6 . We first consider the complete bipartite subgraph with bipartition classes $V_1 = \mathcal{C} \cup \mathcal{P}_1 \cup \mathcal{Q}_1$ and $V_2 = \mathcal{X} \cup \overline{\mathcal{P}_1} \cup \overline{\mathcal{Q}_1}$.

Suppose that H contains at least four vertices from $V_1 \cup V_2$. Since P_6 contains no independent set of cardinality four, H then contains at least one vertex from each of V_1 and V_2 . This either yields a vertex with degree at least three in H or a cycle on four vertices in H , a contradiction. Hence $|V(H) \cap (V_1 \cup V_2)| \leq 3$. Since $\mathcal{A} \cup \mathcal{B}$ is an independent set, we also have $|V(H) \cap (\mathcal{A} \cup \mathcal{B})| \leq 3$. Since $|V(H)| = 6$, this implies that both inequalities are in fact equalities.

Let $V(H) = \{v_1, v_2, \dots, v_6\}$ and $E(H) = \{v_1v_2, v_2v_3, v_3v_4, v_4v_5, v_5v_6\}$. By symmetry, we may assume that either $\{v_1, v_3, v_5\} \subset V(H) \cap (\mathcal{A} \cup \mathcal{B})$ or $\{v_1, v_3, v_6\} \subset V(H) \cap (\mathcal{A} \cup \mathcal{B})$. Noting that every vertex of $\mathcal{P} \cup \mathcal{Q}$ has at most one neighbor in $\mathcal{A} \cup \mathcal{B}$, in both cases $v_2 \in \mathcal{C} \cup \mathcal{X}$. We next observe that every vertex of $\mathcal{A} \cup \mathcal{B}$ has precisely one neighbor in \mathcal{C} and precisely one neighbor in \mathcal{X} . This implies that we can neither have $\{v_2, v_4\} \subset \mathcal{X}$ nor $\{v_2, v_4\} \subset \mathcal{C}$. Since $v_2v_4 \notin E(G_I)$, we cannot have $v_4 \in \mathcal{C} \cup \mathcal{X}$. This rules out the first case, and in the remaining case we may assume $\{v_1, v_3, v_6\} \subset V(H) \cap (\mathcal{A} \cup \mathcal{B})$, with $v_2 \in \mathcal{C} \cup \mathcal{X}$ and $v_4 \in \mathcal{P} \cup \mathcal{Q}$. Since v_5 is a neighbor of v_4 while v_2 is not a neighbor of v_4 , we find that $v_5 \notin \mathcal{C} \cup \mathcal{X}$. Hence $v_5 \in \mathcal{P} \cup \mathcal{Q}$. Because v_4v_5 is an edge and v_4, v_5 both belong to $\mathcal{P} \cup \mathcal{Q}$, one of them belongs to V_1 and the other one to V_2 . However, then either v_2v_4 or v_2v_5 is an edge of G_I , because $v_2 \in \mathcal{C} \cup \mathcal{X}$ is either adjacent to all vertices in V_1 or else to all vertices in V_2 . This is not possible, and we conclude that G_I is P_6 -free.

We claim that I has a truth assignment in which each clause contains at least one true and at least one false literal if and only if the pre-coloring of G_I can be extended to a 5-coloring of G_I .

First suppose that I has a satisfying truth assignment in which each clause contains at least one true and at least one false literal. We use color 4 to color the x -type vertices representing the true literals and color 5 for the false literals. Now consider the lists assigned to the a -type and b -type vertices that come in pairs chosen from $\{\{1, 4\}, \{2, 5\}\}, \{\{2, 4\}, \{3, 5\}\}, \{\{3, 4\}, \{1, 5\}\}$. If the adjacent x -type vertex has color 4, color 1, 2 or 3 is forced on one of the adjacent a -type or b -type vertices, respectively, while on the other one we can use color 5; similarly, if the adjacent x -type vertex has color 5, color 2, 3 or 1 is forced on one of the adjacent a -type or b -type vertices, respectively, while on the other one we can use color 4. Since precisely two of the three x -type vertices of one clause gadget have the same color, this leaves at least one of the colors 1, 2 and 3 admissible for the C -type vertex representing the clause. By coloring the vertices associated with each clause and variable as described above, a 5-coloring of the pre-colored graph G_I is obtained.

Now suppose that we have a 5-coloring of the graph G_I that respects the pre-coloring. Then each of the x -type vertices has color 4 or 5, and each of the C -type vertices has color 1, 2 or 3. We define a truth assignment that sets a variable to TRUE if the corresponding x -type vertex has color 4, and to FALSE otherwise. Suppose that one of the clauses contains only true literals. Then the three x -type vertices in the corresponding clause gadget of G_I all have color 4. Now consider the lists assigned to the a -type and b -type vertices of this gadget that come in pairs chosen from $\{\{1, 4\}, \{2, 5\}\}, \{\{2, 4\}, \{3, 5\}\}, \{\{3, 4\}, \{1, 5\}\}$. Since the adjacent x -type vertices all have color 4, colors 1, 2 and 3 are forced on three of the a -type and b -type vertices adjacent to the C -type vertex of this gadget, a contradiction, since the C -type vertex has color 1, 2 or 3. This proves that every clause contains at least one false literal. Analogously, every clause contains at least one true literal. This completes the proof of [Theorem 1](#). \square

3. Pre-coloring extension of 3-coloring for P_6 -free graphs

In this section we show that the pre-coloring extension version of 3-COLORING is polynomially solvable for P_6 -free graphs. A key ingredient in our approach is the following characterization of P_6 -free graphs [[14](#)]. Here a subgraph H of a graph G is said to be a dominating subgraph of G if every vertex of $V(G) \setminus V(H)$ has a neighbor in H .

Lemma 2 ([[14](#)]). *A graph G is P_6 -free if and only if each connected induced subgraph of G on more than one vertex contains a dominating induced cycle on six vertices or a dominating (not necessarily induced) complete bipartite subgraph. Moreover, these dominating subgraphs can be obtained in polynomial time.*

Another key ingredient in our approach is the following lemma. Its proof 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 [[4](#)] and is folklore now, see also [[6,10](#)].

Lemma 3 ([[4](#)]). *Let G be a graph in which every vertex has a list of admissible colors of size at most 2. Then checking if G has a list-coloring is solvable in polynomial time.*

An important subroutine in our algorithm works as follows. Let G be a graph in which every vertex has a list of admissible colors. 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 the unique color in its list from the lists of its neighbors. Next we remove u from G . We 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. We note the following.

Lemma 4. *A graph G with lists of admissible colors on its vertices can be updated in polynomial time. If this results in a vertex with an empty list, then G does not have a list-coloring respecting the original lists.*

We are now ready to state the main result of this section. We prove a slightly stronger statement, namely that we can decide in polynomial time whether a P_6 -free graph, in which each vertex has a list of admissible colors from the set $\{1, 2, 3\}$, has a coloring respecting these lists; note that a pre-coloring corresponds to lists of size 1 on the pre-colored vertices.

Theorem 5. *The pre-coloring extension version of 3-COLORING can be solved in polynomial time for P_6 -free graphs.*

Proof. Suppose that our instance graph $G = (V, E)$ is connected (otherwise we treat the components of G separately) and that we have lists of admissible colors from the set $\{1, 2, 3\}$ on each vertex of G . We show how to check in polynomial time whether G allows a 3-coloring respecting these lists.

We first check if G has a dominating C_6 . We can do this in $O(|V|^6)$ time by brute force. If so, we can solve our problem as follows. We assume a coloring on the C_6 (respecting the lists) and apply Lemma 4. Since all original lists are subsets of $\{1, 2, 3\}$ and the vertices not in the C_6 are dominated by the C_6 , their new lists have size at most 2. This means that we can apply Lemma 3. Because the number of possible 3-colorings of the C_6 is at most 3^6 , we can check all of them if necessary.

Now suppose that G does not have a dominating C_6 . Then, by Lemma 2, we can construct in polynomial time a dominating complete (not necessarily induced) bipartite graph H of G with bipartition classes A and B . As we cannot assume that H has a bounded size, we must use the special structure of P_6 -free graphs in a more advanced way. Below we show how.

Claim 1. *In any eligible 3-coloring of G at least one of the sets A, B is monochromatic.*

We prove Claim 1 as follows. Suppose that both A and B contain two vertices with different colors. Then either 4 colors must be used on $A \cup B$ or two vertices with the same color are adjacent. Both cases are not possible.

Due to Claim 1 we can proceed as follows. We first assume that A is monochromatic. If this does not result in a 3-coloring of G we repeat the procedure assuming that B is monochromatic.

So, from now on, we assume that all vertices of A are colored with color 1 (possibly after renaming the colors). We apply Lemma 4. Let G' denote the resulting graph after restoring one vertex $a \in A$ and its incident edges back into the graph; we need such a vertex later, in order to make use of the P_6 -freeness. So, in G' , the list of every vertex except a has size 2 or 3. Let R denote the subset of all vertices of G' with lists of size 3. If $R = \emptyset$, then we are done by Lemma 3.

Suppose that $R \neq \emptyset$. Note that the vertices in R are not adjacent to any vertex of A in the original graph G . Then they must be adjacent to at least one vertex of B , because H is a dominating subgraph of G . Since H is complete bipartite, all vertices of $B \cap V(G')$ are in $N_{G'}(a)$, and we redefine $B := N_{G'}(a)$ for convenience. We observe that every vertex of B has list $\{2, 3\}$, and consequently, R must be a subset of $Q = V(G') \setminus (\{a\} \cup B)$. We observe that B dominates R but not necessarily all vertices of Q . We analyze pairs of adjacent vertices of Q and distinguish a number of cases.

Case 1. Q contains an edge pq such that p is adjacent to a vertex $b \in B \setminus N_{G'}(q)$ and q is adjacent to a vertex $c \in B \setminus N_{G'}(p)$.

First note that the set $S = \{a, b, c, p, q\}$ induces a C_5 with possibly an additional edge bc in G' . Let R' be the subset of R consisting of vertices not dominated by S . If $R' = \emptyset$, we check all $O(3^5)$ eligible 3-colorings of S and apply Lemma 3 for every such coloring. Suppose the contrary, i.e., $R' \neq \emptyset$. Let R'_1 consist of all vertices x of R' so that b or c has a neighbor in $B \cap N_{G'}(x)$. Let R'_2 consist of all vertices x of $R' \setminus R'_1$ so that both p and q have a neighbor in $B \cap N_{G'}(x)$. Let $R'_3 = R' \setminus (R'_1 \cup R'_2)$.

Claim 2. *Any eligible 3-coloring of S will reduce the list size of every vertex in $R'_1 \cup R'_2$ by at least one color.*

We prove Claim 2 as follows. A 3-coloring on S would color b, c , and at least one of p, q with color 2 or 3. Consequently, it will fix the color of every vertex $y \in B$ that is adjacent to b, c or to both p and q , because vertices in B have list $\{2, 3\}$. This has as a further consequence that the list of every neighbor of such y will be reduced by at least one color. By definition, $R'_1 \cup R'_2$ only contains such neighbors. This proves Claim 2.

Suppose that $R'_3 = \emptyset$. Then, by Claim 2, we can apply Lemma 3 every time we guess a 3-coloring of S . Suppose that $R'_3 \neq \emptyset$. Because R is dominated by B , every vertex $x \in R'_3$ has a neighbor in B . By definition, there is no edge between $B \cap N_{G'}(x)$ and $\{b, c\}$, and only one of $\{p, q\}$ may have a neighbor in $B \cap N_{G'}(x)$. However, every $y \in B \cap N_{G'}(x)$ must be adjacent to one of p, q ; otherwise $xyabpq$ is an induced P_6 . This means that we can partition R'_3 into two sets T_1, T_2 , where T_1 consists of all vertices of R'_3 , whose neighbors in B are adjacent to p and not to q , and T_2 consists of all vertices of R'_3 , whose

neighbors in B are adjacent to q and not to p . Because $R'_3 \neq \emptyset$, at least one of T_1, T_2 is nonempty, and we analyze two subcases.

Case 1a. $T_1 \neq \emptyset$ and $T_2 \neq \emptyset$.

Let D_i be the set of vertices in B that have a neighbor in T_i for $i = 1, 2$.

Claim 3. Every vertex in D_1 is adjacent to every vertex in D_2 .

We prove Claim 3 as follows. Let $b' \in D_1$ and $c' \in D_2$. Suppose that $b'c' \notin E(G')$. By definition, b' has a neighbor $p' \in T_1$, and c' has a neighbor $q' \in T_2$. Then $p'q' \in E(G')$; otherwise $p'b'pqc'q'$ is an induced P_6 . However, then $qcab'p'q'$ is an induced P_6 . This is not possible and completes the proof of Claim 3.

We now proceed as follows. Every eligible 3-coloring of S colors at least one of p, q with color 2 or 3. As a direct consequence, one of D_1, D_2 becomes monochromatic, because all the vertices in $D_1 \cup D_2 \subseteq B$ have list $\{2, 3\}$. Due to Claim 3, also the other set in $\{D_1, D_2\}$ becomes monochromatic. This means that the list size of every vertex in $R'_3 = T_1 \cup T_2$ is reduced by at least one color. By Claim 2, the same holds for every vertex in $R'_1 \cup R'_2$. Thus we may apply Lemma 3 every time we guess a 3-coloring of S .

Case 1b. $T_1 = \emptyset$ or $T_2 = \emptyset$.

We assume without loss of generality that $T_1 = \emptyset$. If q receives color 2 or 3 in the guessed 3-coloring of S then, as before, the subset of B that consists of vertices adjacent to q becomes monochromatic, and consequently, the list size of every vertex in $R'_3 = T_2$ reduces by at least one color. Recall that the same holds for every vertex in $R'_1 \cup R'_2$ due to Claim 2. This means that we may apply Lemma 3 every time we guess a 3-coloring of S .

Suppose that using color 2 or 3 on q does not result in a 3-coloring of G' in the end. Then we assign color 1 to q and update G' without removing a . We check if we are in Case 1. If so, we repeat the (polynomial time) procedure described in Case 1. If not, then we check whether we are in Case 2 or Case 3 described below; note that these two cases together cover all remaining possibilities.

Case 2. Case 1 does not apply, and Q contains an edge pq such that p is adjacent to a vertex $b \in B \cap N(q)$ and q is adjacent to a vertex $c \in B \setminus N(p)$.

The set $S = \{a, b, c, p, q\}$ now induces a C_5 with an edge bq and possibly an additional edge bc in G' . We define R' as in Case 1. If $R' = \emptyset$, then we are done just as in Case 1. Otherwise we define R'_1, R'_2, R'_3 as in Case 1. Then, in case $R'_3 = \emptyset$, we are done just as in Case 1. Suppose that $R'_3 \neq \emptyset$. We define T_1, T_2 as in Case 1. Suppose that $T_1 \neq \emptyset$. Then there exists a vertex $p' \in T_1$ with a neighbor $b' \in B$ such that b' is adjacent to p and not to q . Then we contradict our assumptions since we are in Case 1 with b' instead of b . Hence $T_1 = \emptyset$, and we can proceed as in Case 1b.

Case 3. Every two adjacent vertices $p, q \in Q$ have the same neighbors in B .

This means that all vertices in each component of Q have the same neighbors in B . We may assign color 1 to every vertex in Q that has color 1 in its list but that does not have a neighbor with color 1 in its list. Afterwards, we update G' (hence a is removed as well). Let \mathcal{F} be the set of components of the resulting graph and consider each component $F \in \mathcal{F}$ separately.

Suppose that F only contains vertices whose lists have size at most 2. Then we can apply Lemma 3. Suppose that F contains at least one vertex x with a list of size 3. Because x is dominated by B , there must exist vertices in B that are adjacent to x and that still have list $\{2, 3\}$, so $B \cap V(F) \neq \emptyset$. Let $y \in B \cap V(F)$.

Claim 4. Assigning color 2 or 3 to y reduces the list of every vertex in $B \cap V(F)$ with at least one color.

We prove Claim 4 as follows. Let \mathcal{C} be the set of components in the subgraph of F induced by $B \cap V(F)$. Let C be the component in \mathcal{C} that contains y . Then C is a bipartite graph, every vertex of which has list $\{2, 3\}$. Hence, fixing a color of y fixes the color of all vertices in C . Let $C' \in \mathcal{C} \setminus \{C\}$ be a component that is connected to C in F by a path P that has all its internal vertices in Q .

First suppose that P has at least two internal vertices x, x' . By the assumption of Case 3, x and x' share the same neighbors in B . Hence, a neighbor in C of the internal vertices of P must receive the same color as a neighbor in C' . Now suppose that P has exactly one internal vertex x . If x has list $\{2, 3\}$, coloring C fixes the color of x and consequently the color of C' . If 1 is a color in the list of x , then by

construction x has a neighbor x^* with 1 in its list. By the assumption of Case 3, x and x^* share the same neighbors in C' . Hence, we may add x^* as an internal vertex of P and return to the previous case, in which P has two internal vertices. We repeat these arguments for components in \mathcal{C} connected to C or C' by a path that has all its internal vertices in Q , and so on. This proves Claim 4.

We now proceed as follows. We first consider the case in which y gets color 2. Then, by Claim 4, all colors on B are fixed and we may apply Lemma 3. If this does not lead to a 3-coloring of F , then we give y color 3 and apply Lemma 3 as well.

After checking every $F \in \mathcal{F}$ separately, we have either found (in polynomial time) an eligible 3-coloring of every component of \mathcal{F} , or a component in \mathcal{F} that does not allow an eligible 3-coloring. In the first case we have found an eligible 3-coloring of G . In the second case we conclude that there does not exist an eligible 3-coloring of G with monochromatic A (and we need to verify if such a coloring exists with monochromatic B). This completes the proof of Theorem 5. \square

4. 6-Coloring for P_7 -free graphs

In this section we prove that 6-COLORING is NP-complete for P_7 -free graphs. We use a reduction from 3-SATISFIABILITY (3SAT). We consider an arbitrary instance I of 3SAT and define a graph G_I , and next we show that G_I is P_7 -free and that G_I is 6-colorable if and only if I has a satisfying truth assignment.

Theorem 6. *The 6-COLORING problem is NP-complete for P_7 -free graphs.*

Proof. Let I be an arbitrary instance of 3SAT with variables $\{x_1, x_2, \dots, x_n\}$ and clauses $\{C_1, C_2, \dots, C_m\}$. We define a graph G_I corresponding to I based on the following construction.

1. We introduce a gadget on 8 new vertices for each of the clauses, as follows. For each clause C_j we introduce a gadget with vertex set:

$$\{a_{j,1}, a_{j,2}, a_{j,3}, b_{j,1}, b_{j,2}, b_{j,3}, c_{j,1}, c_{j,2}\}$$
 and edge set:

$$\{a_{j,1}a_{j,2}, a_{j,1}a_{j,3}, a_{j,2}a_{j,3}, a_{j,1}b_{j,1}, a_{j,2}b_{j,2}, a_{j,3}b_{j,3}, b_{j,1}c_{j,1}, b_{j,1}c_{j,2}, b_{j,2}c_{j,1}, b_{j,2}c_{j,2}, b_{j,3}c_{j,1}, b_{j,3}c_{j,2}, c_{j,1}c_{j,2}\}.$$
 We say that these vertices are of a -type, b -type and c -type. These vertices induce disjoint components in G_I which we will call clause-components.
2. We introduce a gadget on 3 new vertices for each of the variables, as follows. For each variable x_i we introduce a complete graph with vertex set $\{x_i, \bar{x}_i, y_i\}$. We say that these vertices are of x -type (both the x_i and the \bar{x}_i vertices) and of y -type. These vertices induce disjoint triangles in G_I which we will call variable-components.
3. 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 add the edges $b_{j,h}x_{i_h}$ or $b_{j,h}\bar{x}_{i_h}$ depending on whether x_{i_h} or \bar{x}_{i_h} is a literal in C , respectively. We also add the edge $b_{j,h}y_{i_h}$ for $h = 1, 2, 3$.
4. We introduce three additional vertices d_1, d_2 and z , and join d_1 and d_2 by an edge. We join all x_i to d_1 by edges, and all \bar{x}_i to d_2 .
5. We join z to all vertices of y -type, a -type, and c -type, and to d_1 and d_2 .
6. We join all the x -type vertices and y -type vertices to all the a -type and c -type vertices.
7. Finally, we join d_1 and d_2 to all the a -type, b -type and c -type vertices.

See Figs. 3–5 for an example of a graph G_I . In this example, C_1 is a clause with literals x_1, \bar{x}_2 , and x_3 .

We now prove that G_I is P_7 -free. In order to obtain a contradiction, suppose that the graph G_I contains an induced subgraph H that is isomorphic to P_7 . We observe that two distinct variable-components do not share a b -type vertex as a common neighbor.

First suppose that H contains both d_1 and d_2 . Then, since $d_1d_2 \in E(H)$ and H has no cycles and no vertices with degree more than 2, H does neither contain z nor any vertices of a -type, b -type or c -type, and at most two x -type vertices (with one positive and one negative literal in the case where it contains two). The longest path we can obtain is a P_6 , a contradiction. We conclude that H contains at most one of the vertices d_1 and d_2 .

Next suppose that H contains both d_1 and z . Then, since $d_1z \in E(H)$ and H has no cycles and no vertices with degree more than 2, H neither contains d_2 nor any vertices of a -type or c -type, and at

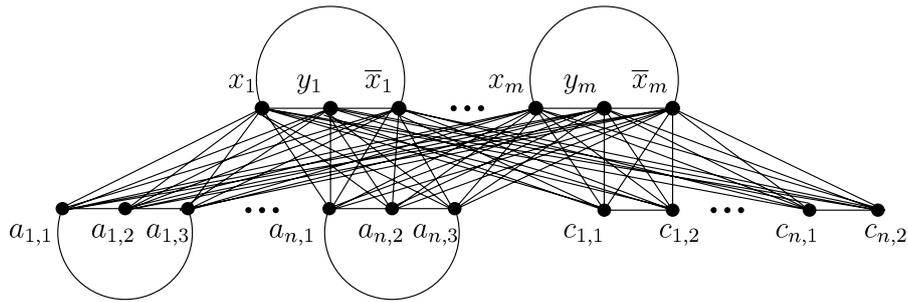


Fig. 3. The subgraph of G_I induced by vertices of type a, c, x, y .

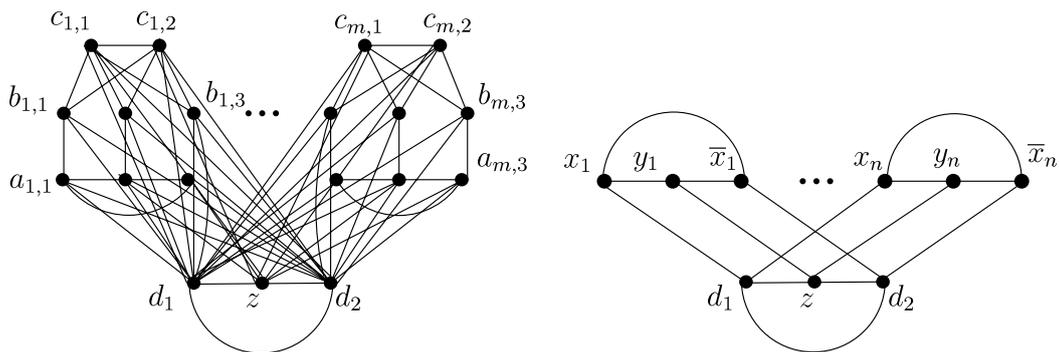


Fig. 4. (i) The subgraph of G_I induced by d_1, d_2, z and vertices of type a, b, c . (ii) The subgraph of G_I induced by d_1, d_2, z and vertices of type x, y .

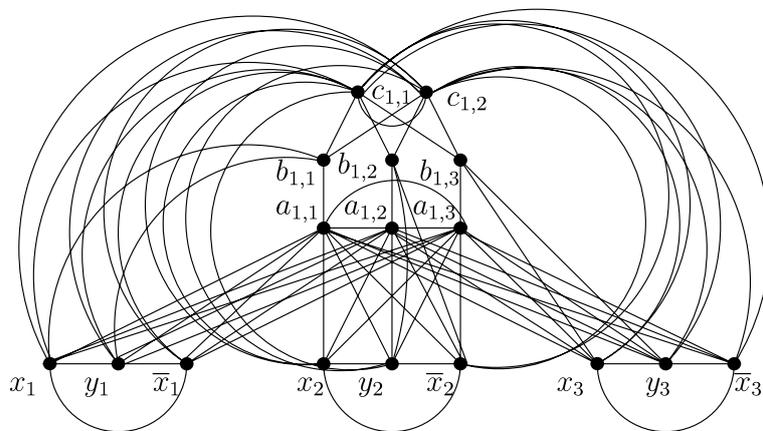


Fig. 5. The subgraph of G_I for clause C_1 with ordered literals x_1, \bar{x}_2, x_3 .

most one b -type and at most one y -type vertex. Since $|V(H)| = 7$, this implies that H contains at least three vertices of x -type. Since d_1 is adjacent to all x_i and to z , H contains at most one x_i . So H contains at least two vertices \bar{x}_j and \bar{x}_k . In H , \bar{x}_j can only have neighbors in $\{x_j, y_j, b\}$ where b is the only possible b -type vertex in H . Recall that b can be adjacent to at most one of the variable-components. Since H contains at most one y -type vertex, at most one b -type vertex, and at most one x_i , this means that there cannot be three distinct vertices \bar{x}_j, \bar{x}_k and \bar{x}_r in H . So we conclude that H contains precisely one y -type vertex, one b -type vertex, one x_i and two distinct \bar{x}_j and \bar{x}_k (where possibly $i = j$ or $i = k$). But now d_1 has degree 3 in H , a contradiction. We conclude that H contains at most one of the vertices d_1 and z . By symmetry, H contains at most one of the vertices d_2 and z , and hence at most one of d_1, d_2 and z .

Next we are going to show that H contains at most two b -type vertices. To the contrary, first suppose that H contains at least four b -type vertices. Because the b -type vertices form an independent set, H contains exactly four of them, and the other three vertices of H also form an independent set. This implies that the other three are either of a -type and c -type or of x -type and y -type. The latter

cannot occur, because two variable-components do not share a b -type vertex as a common neighbor. Hence, all vertices of H are of a -type, b -type and c -type. This implies that H is a subgraph of one clause-component, a contradiction. Next suppose that H contains precisely three b -type vertices. If $z \notin V(H)$, then, since all the x -type and y -type vertices are joined to all the a -type and c -type vertices, the other four vertices are either of a -type and c -type or of x -type and y -type. Again the former cannot occur since H is not a subgraph of one clause-component and the latter cannot occur, because two variable-components do not share a b -type vertex as a common neighbor. So we conclude that $z \in V(H)$. Then z and the three b -type vertices form an independent set in H , and the other three vertices also form an independent set in H . Just as in the case when H contains four b -type vertices, the only possibility is that these three other vertices are of a -type and c -type. But since z is adjacent to all a -type and c -type vertices, we obtain a contradiction. We conclude that H contains at most two b -type vertices.

Together with the earlier conclusion that H contains at most one of d_1 , d_2 and z , this implies that H contains at least four vertices from the set of all a -type, c -type, x -type and y -type vertices. Due to the adjacencies between these vertices and the fact that H has neither cycles nor vertices with degree more than 2, we find that all four are either of a -type and c -type or of x -type and y -type. In the former case z , d_1 and d_2 are no vertices of H . But then all vertices of H are of a -type, b -type and c -type, so H is contained in one clause-component, a contradiction. In the latter case we know that H contains vertices from at least two variable-components. Since these components have no b -type vertex as a common neighbor, they are connected through one of d_1 , d_2 and z . Hence H contains vertices of precisely two of these components, implying that H contains precisely two b -type vertices. It is not difficult to check that the b -type vertices have degree 1 in H . This in turn implies that d_1 and d_2 are no vertices of H . Hence $z \in V(H)$ and z has two y -type neighbors in H . The other two vertices of H are of x -type and each of these x_i or \bar{x}_i is adjacent to a b -type vertex and to y_i in H . But then this y_i and this b -type vertex are adjacent, our final contradiction. We conclude that G_I is P_7 -free.

We claim that I has a satisfying truth assignment if and only if G_I is 6-colorable.

First suppose that I has a satisfying truth assignment. We use color 4 or 5 to color the x -type vertices representing the true literals and color 6 for the false literals. In particular, if x_i is true, we use color 5 to color the corresponding vertex; if \bar{x}_i is true, we use color 4 to color the corresponding vertex. We use color 4 or color 5 to color the y -type vertices, depending on the colors we used for the x -type vertices. This yields a proper 3-coloring of all the variable-components with colors 4, 5 and 6. We extend this 3-coloring by using color 6 for z and colors 4 and 5 for d_1 and d_2 , respectively. For the true literals of C_j , we can use color 6 for the corresponding b -type vertex, and color 1 for the other b -type vertices of the corresponding clause-component. Since each clause contains at least one true literal, we note that we do not use color 1 for all three b -type vertices of the clause-components. We can now use colors 2 and 3 for the c -type vertices and colors 1, 2 and 3 for the a -type vertices to extend the coloring to a 6-coloring of G_I .

Now suppose that we have a 6-coloring of G_I with colors $\{1, 2, \dots, 6\}$. We assume that vertex z has color 6, that d_2 has color 5, and that d_1 has color 4. This implies that all a -type and c -type vertices have colors from $\{1, 2, 3\}$, and all three colors are used on the a -type vertices, and two of the three on the c -type vertices. This implies that all x -type vertices have colors from $\{4, 5, 6\}$ and all y -type vertices from $\{4, 5\}$. Without loss of generality, suppose that in one of the clause-components, the c -type vertices have colors 2 and 3. Then the b -type vertices in this clause-component can only have colors from $\{1, 6\}$. If all of them have color 1, we obtain a contradiction with the coloring of the three a -type vertices in this component. So at least one of the b -type vertices has color 6. The same holds if we had assumed another choice for the two colors used on the c -type vertices. This implies that the corresponding x -type vertex has color 4 or 5. We define a truth assignment that sets a literal to FALSE if the corresponding x -type vertex has color 6, and to TRUE otherwise. In this way we obtain a satisfying truth assignment for I . This completes the proof of [Theorem 6](#). \square

5. Conclusions and open problems

We proved that the pre-coloring extension version of 5-COLORING remains NP-complete for P_6 -free graphs. Hoàng et al. [6] showed that the pre-coloring extension version of ℓ -COLORING is polynomially

Table 1

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

P_k -free	ℓ							
	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

solvable on P_5 -free graphs for any fixed ℓ . In contrast, determining the chromatic number is NP-hard on P_5 -free graphs [7]. We showed that the pre-coloring extension version of 3-COLORING is polynomially solvable for P_6 -free graphs. Finally, we proved that 6-COLORING is NP-complete for P_7 -free graphs. Recently, Broersma et al. [2] showed that 4-COLORING is NP-complete for P_8 -free graphs and that the pre-coloring extension version of 4-COLORING is NP-complete for P_7 -free graphs. All these results together lead to the following Table 1 that shows the current status of ℓ -COLORING and its extension version for P_k -free graphs. This table also shows which cases are still open. We finish this paper with two other open problems on 3-COLORING that have intrigued many researchers: the complexity of 3-COLORING is open for graphs with diameter 2, and for graphs with diameter 3.

Acknowledgments

The first author was supported by EPSRC Grant EP/G043434/1. The second and third authors were supported by the Norwegian Research Council. The fourth author was supported by EPSRC Grant EP/D053633/1.

References

[1] J.A. Bondy, U.S.R. Murty, Graph Theory, in: Springer Graduate Texts in Mathematics, vol. 244, Springer, Berlin, 2008.
 [2] H.J. Broersma, P.A. Golovach, D. Paulusma, J. Song, Updating the complexity status of coloring graphs without a fixed induced linear forest, Theoretical Computer Science (in press).
 [3] D. Bruce, C.T. Hoàng, J. Sawada, A certifying algorithm for 3-colorability of P_5 -free graphs, in: Y. Dong, D.-Z. Du, O.H. Ibarra (Eds.), Proceedings of the 20th International Symposium on Algorithms and Computation, ISAAC 2009, in: Lecture Notes in Computer Science, vol. 5878, Springer, Berlin, 2009, pp. 594–604.
 [4] K. Edwards, The complexity of coloring problems on dense graphs, Theoretical Computer Science 43 (1986) 337–343.
 [5] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.
 [6] C.T. Hoàng, M. Kamiński, V. Lozin, J. Sawada, X. Shu, Deciding k -colorability of P_5 -free graphs in polynomial time, Algorithmica 57 (2010) 74–81.
 [7] D. Král, J. Kratochvíl, Zs. Tuza, G.J. Woeginger, Complexity of coloring graphs without forbidden induced subgraphs, in: A. Brandstädt, V.B. Le (Eds.), Proceedings of the 27th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2001, in: Lecture Notes in Computer Science, vol. 2204, Springer, Berlin, 2001, pp. 254–262.
 [8] J. Kratochvíl, Precoloring extension with fixed color bound, Acta Mathematica Universitatis Comenianae 62 (1993) 139–153.
 [9] V.B. Le, B. Randerath, I. Schiermeyer, On the complexity of 4-coloring graphs without long induced paths, Theoretical Computer Science 389 (2007) 330–335.
 [10] B. Randerath, I. Schiermeyer, 3-Colorability $\in P$ for P_6 -free graphs, Discrete Applied Mathematics 136 (2004) 299–313.
 [11] B. Randerath, I. Schiermeyer, Vertex colouring and forbidden subgraphs—a survey, Graphs and Combinatorics 20 (2004) 1–40.
 [12] A. Tucker, A reduction procedure for coloring perfect K_4 -free graphs, Journal of Combinatorial Theory, Series B 43 (1987) 151–172.
 [13] Zs. Tuza, Graph colorings with local restrictions—a survey, Discussiones Mathematicae Graph Theory 17 (1997) 161–228.
 [14] P. van 't Hof, D. Paulusma, A new characterization of P_6 -free graphs, Discrete Applied Mathematics 158 (2010) 731–740.
 [15] G.J. Woeginger, J. Sgall, The complexity of coloring graphs without long induced paths, Acta Cybernetica 15 (2001) 107–117.