Closing Complexity Gaps for Coloring Problems on H-Free Graphs ⋆

Petr A. Golovach[1], Daniël Paulusma[2], and Jian Song[2]

[1] Department of Informatics, Bergen University,
PB 7803, 5020 Bergen, Norway
petr.golovach@ii.uib.no
[2] School of Engineering and Computing Sciences, Durham University,
Science Laboratories, South Road, Durham DH1 3LE, United Kingdom
{daniel.paulusma,jian.song}@durham.ac.uk

Abstract. If a graph G contains no subgraph isomorphic to some graph H, then G is called H-free. A coloring of a graph $G = (V, E)$ is a mapping $c : V \rightarrow \{1, 2, \ldots\}$ such that no two adjacent vertices have the same color, i.e., $c(u) \neq c(v)$ if $uv \in E$; if $|c(V)| \leq k$ then c is a k-coloring. The Coloring problem is to test whether a graph has a coloring with at most k colors for some integer k. The Precoloring Extension problem is to decide whether a partial k-coloring of a graph can be extended to a k-coloring of the whole graph for some integer k. The List Coloring problem is to decide whether a graph allows a coloring, such that every vertex u receives a color from some given set $L(u)$. By imposing an upper bound ℓ on the size of each $L(u)$ we obtain the ℓ-List Coloring problem. We first classify the Precoloring Extension problem and the ℓ-List Coloring problem for H-free graphs. We then show that 3-List Coloring is NP-complete for n-vertex graphs of minimum degree $n-2$, i.e., for complete graphs minus a matching, whereas List Coloring is fixed-parameter tractable for this graph class when parameterized by the number of vertices of degree $n-2$. Finally, for a fixed integer $k > 0$, the List k-Coloring problem is to decide whether a graph allows a coloring, such that every vertex u receives a color from some given set $L(u)$ that must be a subset of $\{1, \ldots, k\}$. We show that List 4-Coloring is NP-complete for P_6-free graphs, where P_6 is the path on six vertices. This completes the classification of List k-Coloring for P_6-free graphs.

1 Introduction

Graph coloring involves the labeling of the vertices of some given graph by integers called colors such that no two adjacent vertices receive the same color. The corresponding decision problem is called Coloring and is to decide whether a graph can be colored with at most k colors for some given integer k. Because Coloring is NP-complete for any fixed $k \geq 3$, its computational complexity has been widely studied for special graph classes, see e.g. the surveys of Randerath

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

and Schiermeyer [17] and Tuza [20]. In this paper, we consider the Coloring problem together with two natural and well-studied variants, namely Precoloring Extension and List Coloring for graphs characterized by some forbidden induced subgraph. Before we summarize related results and explain our new results, we first state the necessary terminology.

Terminology. We only consider finite undirected graphs $G = (V, E)$ without loops and multiple edges. The graph P_r denotes the path on r vertices. The disjoint union of two graphs G and H is denoted $G + H$, and the disjoint union of r copies of G is denoted rG. Let G be a graph and $\{H_1, \ldots, H_p\}$ be a set of graphs. We say that 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. The *complement* of a graph G denoted by $\overline{G}$ has vertex set $V(G)$ and an edge between two distinct vertices if and only if these vertices are not adjacent in G.

A *coloring* of a graph $G = (V, E)$ is a mapping $c : V \to \{1, 2, \ldots\}$ such that $c(u) \neq c(v)$ whenever $uv \in E$. We call $c(u)$ the *color* of u. A *k-coloring* of G is a coloring c of G with $1 \leq c(u) \leq k$ for all $u \in V$. The problem k-Coloring is to decide whether a given graph admits a k-coloring. Here, k is *fixed*, i.e., not part of the input. If k is part of the input, then we denote the problem as Coloring. A *list assignment* of a graph $G = (V, E)$ is a function L that assigns a list $L(u)$ of so-called *admissible* colors to each $u \in V$. If $L(u) \subseteq \{1, \ldots, k\}$ for each $u \in V$, then L is also called a *k-list assignment*. The *size* of a list assignment L is the maximum list size $|L(u)|$ over all vertices $u \in V$. We say that a coloring $c : V \to \{1, 2, \ldots\}$ *respects* L if $c(u) \in L(u)$ for all $u \in V$. The List Coloring problem is to test whether a given graph has a coloring that respects some given list assignment. For a fixed integer k, the List k-Coloring problem has as input a graph G with a k-list assignment L and asks whether G has a coloring that respects L. For a fixed integer ℓ, the ℓ-List Coloring problem has as input a graph G with a list assignment L of size at most ℓ and asks whether G has a coloring that respects L. In *precoloring extension* we assume that a (possibly empty) subset $W \subseteq V$ of G is precolored by a *precoloring* $c_W : W \to \{1, 2, \ldots k\}$ for some integer k, and the question is whether we can extend c_W to a k-coloring of G. For a fixed integer k, we denote this problem as k-Precoloring Extension. If k is part of the input, then we denote this problem as Precoloring Extension.

Note that k-Coloring can be viewed as a special case of k-Precoloring Extension by choosing $W = \emptyset$, and that k-Precoloring Extension can be viewed as a special case of List k-Coloring by choosing $L(u) = \{c_W(u)\}$ if $u \in W$ and $L(u) = \{1, \ldots, k\}$ if $u \in W \setminus V$. Moreover, List k-Coloring can be readily seen as a special case of k-List Coloring. Hence, we can make the following two observations for a graph class $\mathcal{G}$. If k-Coloring is NP-complete for $\mathcal{G}$, then k-Precoloring Extension is NP-complete for $\mathcal{G}$, and consequently, List k-Coloring and hence k-List Coloring are NP-complete for $\mathcal{G}$. Conversely, if k-List Coloring is polynomial-time solvable on $\mathcal{G}$, then List k-Coloring is polynomial-time solvable on $\mathcal{G}$, and consequently, k-Precoloring Extension is polynomial-time solvable on $\mathcal{G}$, and then also k-Coloring is polynomial-time solvable on $\mathcal{G}$.

Related and New Results. Král', Kratochvíl, Tuza and Woeginger [11] showed the following dichotomy for Coloring for H-free graphs.

Theorem 1 ([11]). *Let H be a fixed graph. If H is a (not necessarily proper) induced subgraph of P_4 or of P_1+P_3, then* Coloring *can be solved in polynomial time for H-free graphs; otherwise it is* NP*-complete for H-free graphs.*

In Section 2 we use Theorem 1 and a number of other results from the literature to obtain the following two dichotomies, which complement Theorem 1. Theorem 3 shows amongst others that Precoloring Extension is polynomial-time solvable on (P_1+P_3)-free graphs, which contain the class of $3P_1$-free graphs, i.e., complements of triangle-free graphs. As such, this theorem also generalizes a result of Hujter and Tuza [8] who showed that Precoloring Extension is polynomial-time solvable on complements of bipartite graphs.

Theorem 2. *Let ℓ be a fixed integer, and let H be a fixed graph. If $\ell \leq 2$ or H is a (not necessarily proper) induced subgraph of P_3, then ℓ-*List Coloring *is polynomial-time solvable on H-free graphs; otherwise ℓ-*List Coloring *is* NP*-complete for H-free graphs.*

Theorem 3. *Let H be a fixed graph. If H is a (not necessarily proper) induced subgraph of P_4 or of $P_1 + P_3$, then* Precoloring Extension *can be solved in polynomial time for H-free graphs; otherwise it is* NP*-complete for H-free graphs.*

In Section 3 we consider the List Coloring problem for $(3P_1, P_1 + P_2)$-free graphs, i.e., graphs that are obtained from a complete graph after removing the edges of some matching. We also call such a graph a *complete graph minus a matching*. Our motivation to study this graph class comes from the fact that List Coloring is NP-complete on almost all non-trivial graph classes, such as can be deduced from Theorem 2 and from other results known in the literature. For example, List Coloring is NP-complete for complete bipartite graphs [10], complete split graphs [10], line graphs of complete graphs [14], and more over, even for (not necessarily vertex-disjoint) unions of two complete graphs [9]; we refer to Table 1 in the paper by Bonomo, Durán and Marenco [1] for an overview. It is known that List Coloring can be solved in polynomial time for block graphs [9], which contain the class of complete graphs and trees. Our aim was to extend this positive result. However, as we show, already 3-List Coloring is NP-complete for complete graphs minus a matching. As a positive result, we show that List Coloring is fixed-parameter tractable for complete graphs minus a matching when parameterized by the number of matching edges removed.

In Section 4, we consider the List k-Coloring problem. As we explained, this problem is closely related to the problems k-Coloring and k-Precoloring Extension. In contrast to Coloring and Precoloring Extension (cf. Theorems 1 and 3), the complexity classifications of k-Coloring and k-Precoloring

	k-Coloring				k-Precoloring Extension				List k-Coloring			
r	$k=3$	$k=4$	$k=5$	$k \geq 6$	$k=3$	$k=4$	$k=5$	$k \geq 6$	$k=3$	$k=4$	$k=5$	$k \geq 6$
$r \leq 5$	P	P	P	P	P	P	P	P	P	P	P	P
$r=6$	P	?	?	?	P	?	NP-c	NP-c	P	**NP-c**	NP-c	NP-c
$r=7$	?	?	?	NP-c	?	NP-c	NP-c	NP-c	?	NP-c	NP-c	NP-c
$r \geq 8$	?	NP-c	NP-c	NP-c	?	NP-c	NP-c	NP-c	?	NP-c	NP-c	NP-c

Table 1. The complexity of k-Coloring, k-Precoloring Extension and List k-Coloring on P_r-free graphs for fixed k and r. The bold entry is our new result.

Extension for H-free graphs are yet to be completed, even when H is a path. Hoàng et al. [6] showed that for any $k \geq 1$, the k-Coloring problem can be solved in polynomial time for P_5-free graphs. Randerath and Schiermeyer [16] showed that 3-Coloring can be solved in polynomial time for P_6-free graphs. These results are complemented by the following hardness results: 4-Coloring is NP-complete for P_8-free graphs [3] and 6-Coloring is NP-complete for P_7-free graphs [2]. Also the computational complexity of the List k-Coloring problem is still open for P_r-free graphs. Hoàng et al. [6] showed that their polynomial-time result on k-Coloring for P_5-free graphs is in fact valid for List k-Coloring for any fixed $k \geq 1$. Broersma et al. [2] generalized the polynomial-time result of Randerath and Schiermeyer [16] for 3-Coloring on P_6-free graphs to List 3-Coloring on P_6-free graphs. In addition, they showed that 5-Precoloring Extension is NP-complete for P_6-free graphs [2], whereas 4-Precoloring Extension is known to be NP-complete for P_7-free graphs [3]. Table 1 summarizes all existing results for these three problems restricted to P_r-free graphs. We prove that List 4-Coloring is NP-complete for P_6-free graphs. Because List 3-Coloring is polynomial-time solvable on P_6-free graphs [2], we completely characterized the computational complexity of List k-Coloring for P_6-free graphs. In Table 1 we indicate this result in bold. All cases marked by "?" in Table 1 are still open.

2 Classifying Precoloring Extension and 3-List Coloring

The following well-known lemma (cf. [1]) is obtained by modeling the List Coloring problem on n-vertex complete graphs with a k-list assignment as a maximum matching problem for an $(n+k)$-vertex bipartite graph; as such we may apply the Hopcroft-Karp algorithm [7] to obtain the bound on the running time.

Lemma 1. List Coloring *can be solved in* $O((n+k)^{\frac{5}{2}})$ *time on* n*-vertex complete graphs with a* k*-list assignment.*

We are now ready to state the proofs of Theorems 2 and 3.

The proof of Theorem 2. Early papers by Erdös, Rubin and Taylor [4] and Vizing [21] already observed that 2-List Coloring is polynomial-time solvable on general graphs. Hence, we can focus on the case $\ell \geq 3$. Because the ℓ-COLORING problem is a special case of the ℓ-LIST COLORING problem, the following results are useful. Kamiński and Lozin [13] showed that for any $k \geq 3$, the k-COLORING problem is NP-complete for the class of graphs of girth (the length of a shortest induced cycle) at least p for any fixed $p \geq 3$. Their result implies that for any $\ell \geq 3$, the ℓ-COLORING problem, and consequently, the ℓ-LIST COLORING problem is NP-complete for the class of H-free graphs whenever H contains a cycle. The proof of Theorem 4.5 in the paper by Jansen and Scheffler [10] is to show that 3-LIST COLORING is NP-complete on P_4-free graphs but as a matter of fact shows that 3-LIST COLORING is NP-complete on complete bipartite graphs, which are (P_1+P_2)-free. The proof of Theorem 11 in the paper by Jansen [9] is to show that LIST COLORING is NP-complete for (not necessarily vertex-disjoint) unions of two complete graphs but as a matter of fact shows that 3-LIST COLORING is NP-complete for these graphs. As the union of two complete graphs is $3P_1$-free, this means that 3-LIST COLORING is NP-complete for $3P_1$-free graphs. This leaves us with the case when H is a (not necessarily proper) induced subgraph of P_3. By Lemma 1 we can solve LIST COLORING in polynomial time on complete graphs. This means that we can solve ℓ-LIST COLORING in polynomial time on P_3-free graphs for any $\ell \geq 1$. Hence we have proven Theorem 2. □

The proof of Theorem 3. Let H be a fixed graph. If H is not an induced subgraph of P_4 or of $P_1 + P_3$, then Theorem 1 tells us that COLORING, and consequently, PRECOLORING EXTENSION is NP-complete for H-free graphs. Jansen and Scheffler [10] showed that PRECOLORING EXTENSION is polynomial-time solvable for P_4-free graphs. Hence, we are left with the case $H = P_1 + P_3$.

Let (G, k, c_W) be an instance of PRECOLORING EXTENSION, where G is a $(P_1 + P_3)$-free graph, k is an integer and $c_W : W \to \{1, \ldots, k\}$ for some $W \subseteq V(G)$ is a precoloring. We first prove how to transform (G, k, c_W) in polynomial time into a new instance $(G', k', c_{W'})$ with the following properties:

(i) G' is a $3P_1$-free subgraph of G, $k' \leq k$ and $c_{W'} : W' \to \{1, \ldots, k\}$ for some $W \subseteq W' \subseteq V(G)$ is a precoloring;

(ii) $(G', k', c_{W'})$ is a yes-instance if and only if (G, k, c_W) is a yes-instance.

Suppose that G is not $3P_1$-free already. Then G contains at least one triple T of three independent vertices. Let $u \in T$. Here we make the following choice if possible: if there exists a triple of three independent vertices that intersects with W, then we choose T to be such a triple and pick $u \in T \cap W$.

Let $S = V(G) \setminus (\{u\} \cup N(u))$. Because G is $(P_1 + P_3)$-free, $G[S]$ is the disjoint union of a set of complete graphs $D_1, \ldots, D_p$ for some $p \geq 2$; note that $p \geq 2$ holds, because the other two vertices of T must be in different graphs D_i and D_j. We will use the following claim.

Claim 1. Every vertex in $V(D_1) \cup \cdots \cup V(D_p)$ is adjacent to exactly the same vertices in $N(u)$.

We prove Claim 1 as follows. First suppose that w and w' are two vertices in two different graphs D_i and D_j, such that w is adjacent to some vertex $v \in N(u)$.

Then w' is adjacent to v, as otherwise w' and u, v, w form an induced $P_1 + P_3$ in G, which is not possible. Now suppose that w and w' are two vertices in the same graph D_i, say D_1, such that w is adjacent to some vertex $v \in N(u)$. Because $p \geq 2$, the graph D_2 is nonempty. Let w^* be in D_2. As we just showed, the fact that w is adjacent to v implies that w^* is adjacent to v as well. By repeating this argument with respect to w^* and w', we then find that w' is adjacent to v. Hence, we have proven Claim 1.

We now proceed as follows. First suppose that $u \in W$. By symmetry we may assume that $c_W(u) = k$. Then we assign color k to an arbitrary vertex of every D_i that does not contain a vertex colored with k already and that contains at least one vertex outside W. If $u \notin W$, then by our choice of u no vertex from $V(D_1) \cup \cdots \cup V(D_p)$ belongs to W. Either $c_W(W) = \{1, \ldots, k\}$, and we find (in polynomial time) that (G, k, c_W) is a no-instance, or $c_W(W) \subset \{1, \ldots, k\}$, and then we may assume that $c_W(W) \subseteq \{1, \ldots, k-1\}$ by symmetry. In that case we assign color k to u and also to an arbitrary vertex of every D_i. Afterward, in both cases, we remove all vertices colored k from G. In both cases this leads to a new instance $(G', k-1, c_{W'})$ that satisfies condition (i) except that G' may not be $3P_1$-free, and that satisfies condition (ii) due to Claim 1. We repeat this step until the graph is $3P_1$-free as claimed. Note that this takes polynomial time in total, because every step takes polynomial time and in every step the number of vertices of the graph reduces by at least 1.

Due to the above, we may assume without loss of generality that G is $3P_1$-free. We now apply the same algorithm as Hujter and Tuza [8] used for solving Precoloring Extension on complements of bipartite graphs. Because G is $3P_1$-free, G has no three mutually nonadjacent vertices. Suppose that u and v are two nonadjacent vertices in W. Then every vertex of $V(G) \setminus \{u, v\}$ is adjacent to at least one of $\{u, v\}$. This means that we can remove u, v if they are both colored alike by c_W in order to obtain a new instance $(G - \{u, v\}, k-1, c_{W \setminus \{u,v\}})$ that is a yes-instance of Precoloring Extension if and only if (G, k, c_W) is a yes-instance. If u and v are colored differently by c_W, then we add an edge between them. We perform this step for any pair of non-adjacent vertices in W. Afterward, we have found in polynomial time a new instance (G^*, k^*, c_{W^*}) with the following properties. First, $|V(G^*)| \leq |V(G)|$, $k^* \leq k$ and $c_{W^*} : W^* \to \{1, \ldots, k\}$ is a precoloring defined on some clique W^* of G^*. Second, (G^*, k^*, c_{W^*}) is a yes-instance if and only if (G, k, c_W) is a yes-instance. Hence, we may consider (G^*, k^*, c_{W^*}) instead. Because W^* is a clique, we find that (G^*, k^*, c_{W^*}) is a yes-instance if and only if G^* is k^*-colorable. Because G^* is $3P_1$-free, we can solve the later problem by using Theorem 1 (which in this case comes down to computing the size of a maximum matching in the complement of G^*). This completes the proof for the case $H = P_1 + P_3$. Consequently, we have proven Theorem 3. □

3 List Coloring for Complete Graphs Minus a Matching

We prove that 3-List Coloring is NP-complete for complete graphs minus a matching. In order to this we use a reduction from a variant of Not-All-Equal

3-Satisfiability with positive literals only, which we denote as Not-All-Equal ($\leq 3, 2/3$)-Satisfiability with positive literals. The Not-All-Equal 3-Satisfiability problem is NP-complete [18] and is defined as follows. Given a set $X = \{x_1, x_2, ..., x_n\}$ of logical variables, and a set $C = \{C_1, C_2, ..., C_m\}$ 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? The variant Not-All-Equal ($\leq 3, 2/3$)-Satisfiability with positive literals asks the same question but takes as input an instance I that has a set of variables $\{x_1, \ldots, x_n\}$ and a set of literal clauses $\{C_1, \ldots, C_m\}$ over X with the following properties. Each C_i contains either 2 or 3 literals, and these literals are all positive. Moreover, each literal occurs in at most three different clauses. One can prove that Not-All-Equal ($\leq 3, 2/3$)-Satisfiability is NP-complete by a reduction from Not-All-Equal-3-Satisfiability via a well-known folklore trick.

Let I be an arbitrary instance of Not-All-Equal ($\leq 3, 2/3$)-Satisfiability with positive literals. We let $x_1, x_2, \ldots, x_n$ be the variables of I, and we let $C_1, C_2, \ldots, C_m$ be the clauses of I. We define a graph G_I with a list assignment L of size three in the following way. We represent every variable x_i by a vertex with $L(x_i) = \{1_i, 2_i\}$ in G_I. We say that these vertices are of x-type and these colors are of 1-type and 2-type, respectively. For every clause C_p with two variables we fix an arbitrary order of its variables x_h, x_i and we introduce a set of vertices $C_p, a_{p,h}, a_{p,i}, b_{p,h}, b_{p,i}$ that have lists of admissible colors $\{3_p, 4_p\}$, $\{1_h, 3_p\}$, $\{1_i, 4_p\}$, $\{2_h, 4_p\}$, $\{2_i, 3_p\}$, respectively, and we add edges $C_pa_{p,h}$, $C_pb_{p,h}$, $C_pa_{p,i}$, $C_pb_{p,i}$, $a_{p,h}x_h$, $b_{p,h}x_h$, $a_{p,i}x_i$, $b_{p,i}x_i$. For every clause C_p with three variables we fix an arbitrary order of its variables x_h, x_i, x_j and we introduce a set of vertices $C_p, a_{p,h}, a_{p,i}, a_{p,j}$, $b_{p,h}$, $b_{p,i}$, $b_{p,j}$ that have lists of admissible colors $\{3_p, 4_p, 5_p\}$, $\{1_h, 3_p\}$, $\{1_i, 4_p\}$, $\{1_j, 5_p\}$, $\{2_h, 5_p\}$, $\{2_i, 3_p\}$, $\{2_j, 4_p\}$, respectively, and we add edges $C_pa_{p,h}$, $C_pb_{p,h}$, $C_pa_{p,i}$, $C_pb_{p,i}$, $C_pa_{p,j}$, $C_pb_{p,j}$, $a_{p,h}x_h$, $b_{p,h}x_h$, $a_{p,i}x_i$, $b_{p,i}x_i$, $a_{p,j}x_j$, $b_{p,j}x_j$. We say that the new vertices are of C-type, a-type and b-type, respectively. We say that the new colors are of 3-type, 4-type and 5-type, respectively. For each variable x_j that occurs in three clauses we fix an arbitrary order of the clauses C_p, C_q, C_r, in which it occurs. Then we do as follows. First, we modify the lists of $a_{p,j}$, $a_{q,j}$, $b_{p,j}$ and $b_{q,j}$. In $L(a_{p,j})$ we replace color 1_j with a new color $1'_j$. In $L(a_{q,j})$ we replace color 1_j with a new color $1''_j$. In $L(b_{p,j})$ we replace color 2_j with a new color $2'_j$. In $L(b_{q,j})$ we replace color 2_j with a new color $2''_j$. Next we introduce four vertices $a'_{p,j}, a'_{q,j}, b'_{p,j}, b'_{q,j}$ with lists of admissible colors $\{1_j, 1'_j\}, \{1'_j, 1''_j\}, \{2_j, 2'_j\}, \{2'_j, 2''_j\}$, respectively. We say that these vertices are of a'-type or b'-type, respectively. We say that the new colors are also of 1-type or 2-type, respectively. We add edges $a_{p,j}a'_{p,j}$, $a'_{p,j}a'_{q,j}$, $a'_{p,j}x_j$, $a_{q,j}a'_{q,j}$, $b_{p,j}b'_{p,j}$, $b'_{p,j}b'_{q,j}$, $b'_{p,j}x_j$, $b_{q,j}b'_{q,j}$. We add an edge between any two not yet adjacent vertices of G_I whenever they have no common color in their lists. In Figure 1 we give an example, where in order to increase the visibility we display the complement graph $\overline{G_I}$ of G_I instead of G_I itself.

As can be seen from Figure 1, the graph $\overline{G_I}$ is isomorphic to the disjoint union of a number of P_1s and P_2s. This means that G_I is a complete graph

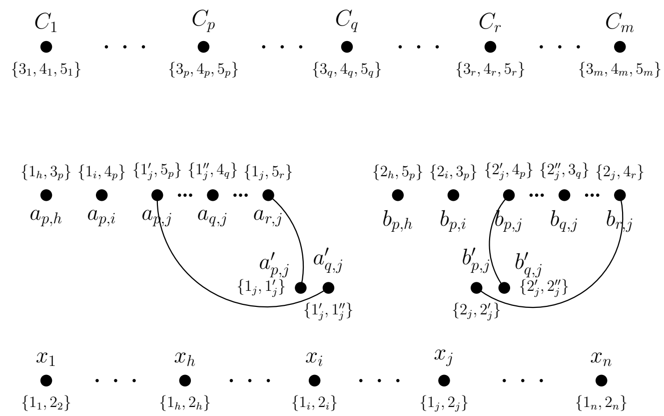

Fig. 1. An example of a graph $\overline{G_I}$ in which a clause C_p and a variable x_j are highlighted. Note that in this example C_p is a clause with ordered variables x_h, x_i, x_j, and that x_j is a variable contained in ordered clauses C_p, C_q and C_r.

minus a matching. This leads us to Lemma 2, whereas the hardness reduction is stated in Lemma 3. The proofs of both lemmas have been omitted.

Lemma 2. *The graph G_I is a complete graph minus a matching.*

Lemma 3. *The graph G_I has a coloring that respects L if and only if I has a satisfying truth assignment in which each clause contains at least one true and at least one false literal.*

Recall that complete graphs minus a matching are exactly those graphs that are $(3P_1, P_1 + P_2)$-free, or equivalently, graphs of minimum degree at least $n-2$, where n is the number of vertices. By observing that 3-List Coloring belongs to NP and using Lemmas 2 and 3, we have proven Theorem 4.

Theorem 4. *The* 3-List Coloring *problem is* NP*-complete for complete graphs minus a matching.*

To complement Theorem 4 we finish this section with the next result, which has as a consequence that List Coloring problem is fixed-parameter tractable on complete graphs minus a matching when parameterized by the number of removed matching edges, or equivalently, for n-vertex graphs G of minimum degree at least $n - 2$ when parameterized by the number of vertices of degree $n - 2$. The proof of Theorem 5 uses Lemma 1; we omit the details.

Theorem 5. *The* List Coloring *problem can be solved in $O(2^p(n+k)^{\frac{5}{2}})$ time on pairs (G, L) where G is an n-vertex graph with p pairs of non-adjacent vertices and L is a k-list assignment.*

4 List 4-Coloring for P_6-Free Graphs

To prove that List 4-Coloring is NP-complete for P_6-free graphs we reduce from Not-All-Equal 3-Satisfiability with positive literals. From an arbitrary instance I of Not-All-Equal 3-Satisfiability with variables x_1, $x_2, \ldots, x_n$ and clauses $C_1, C_2, \ldots, C_m$ that contain positive literals only, we build a graph G_I with a 4-list assignment L. Next we show that G_I is P_6-free and that G_I has a coloring that respects L if and only if I has a satisfying truth assignment in which each clause contains at least one true and at least one false literal. To obtain the graph G_I with its 4-list assignment L we modify the construction of the (P_7-free but not P_6-free) graph used to prove that 4-Precoloring Extension is NP-complete for P_7-free graphs [3]; proof details are omitted.

5 Concluding Remarks

The main tasks are to determine the computational complexity of Coloring for AT-free graphs and to solve the open cases marked "?" in Table 1. This table shows that so far all three problems k-Coloring, k-Precoloring Extension and List k-Coloring behave similarly on P_r-free graphs. Hence, our new NP-completeness result on List 4-Coloring for P_6-free graphs may be an indication that 4-Coloring for P_6-free graphs is NP-complete, or otherwise at least this result makes clear that new proof techniques not based on subroutines that solve List 4-Coloring are required for proving polynomial-time solvability.

Another open problem, which is long-standing, is to determine the computational complexity of the Coloring problem for the class of asteroidal triple-free graphs, also known as *AT-free* graphs. An *asteroidal triple* is a set of three mutually non-adjacent vertices such that each two of them are joined by a path that avoids the neighborhood of the third, and AT-free graphs are exactly those graphs that contain no such triple. We note that unions of two complete graphs are AT-free. Hence NP-completeness of 3-List Coloring for this graph class [9] immediately carries over to AT-free graphs. Stacho [19] showed that 3-Coloring is polynomial-time solvable on AT-free graphs. Recently, Kratsch and Müller [12] extended this result by proving that List k-Coloring is polynomial-time solvable on AT-free graphs for any fixed positive integer k. Marx [15] showed that Precoloring Extension is NP-complete for proper interval graphs, which form a subclass of AT-free graphs. An *asteroidal set* in a graph G is an independent set $S \subseteq V(G)$, such that every triple of vertices of S forms an asteroidal triple. The *asteroidal number* is the size of a largest asteroidal set in G. Note that complete graphs are exactly those graphs that have asteroidal number at most one, and that AT-free graphs are exactly those graphs that have asteroidal number at most two. We observe that Coloring is NP-complete for the class of graphs with asteroidal number at most three, as this class contains the class of $4P_1$-free graphs and for the latter graph class one may apply Theorem 1.

References

1. F. Bonomo, G. Durán and J. Marenco, Exploring the complexity boundary between coloring and list-coloring, Ann. Oper. Res. 169 (2009) 3–16.
2. H.J. Broersma, F.V. Fomin, P.A. Golovach and D. Paulusma, Three complexity results on coloring P_k-free graphs, Proc. IWOCA 2009, LNCS 5874, 95–104 (2009).
3. H.J. Broersma, P.A. Golovach, D. Paulusma and J. Song, Updating the complexity status of coloring graphs without a fixed induced linear forest, Theoretical Computer Science 414, 9–19 (2012) .
4. P. Erdös, A.L. Rubin and H. Taylor, Choosabilty in graphs, Proc. West Coast Conference on Combinatorics, Graph Theory and Computing, (1979) 125–157.
5. P.A. Golovach, D. Paulusma, and J. Song, 4-Coloring H-free graphs when H is small, Proc. SOFSEM 2012, LNCS 7147, 289–300 (2012).
6. 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).
7. J.E. Hopcroft and R.M. Karp, An $n^{\frac{5}{2}}$ algorithm for maximum matchings in bipartite graphs, SIAM J. Comput. 2, 225–231 (1973).
8. M. Hujter and Zs. Tuza, Precoloring extension. II. Graph classes related to bipartite graphs, Acta Math. Univ. Comenianae Vol. LXII, 1–11 (1993).
9. K. Jansen, Complexity Results for the Optimum Cost Chromatic Partition Problem, Universität Trier, Mathematik/Informatik, Forschungsbericht 96–41 (1996).
10. K. Jansen and P. Scheffler, Generalized coloring for tree-like graphs, Discrete Appl. Math. 75, 135–155 (1997).
11. D. Král', J. Kratochvíl, Zs. Tuza, and G.J. Woeginger, Complexity of coloring graphs without forbidden induced subgraphs, Proc. WG 2001, LNCS 2204, 254–262 (2001).
12. D. Kratsch and H. Müller, Colouring AT-free graphs, Proc. ESA 2012, LNCS, to appear.
13. 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).
14. M. Kubale, Some results concerning the complexity of restricted colorings of graphs. Discrete Applied Mathematics 36, 35-46 (1992).
15. D. Marx, Precoloring extension on unit interval graphs, Discrete Applied Mathematics 154, 995–1002 (2006).
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, Proc. STOC 1978, 216–226 (1978).
19. J. Stacho. 3-Colouring AT-free graphs in polynomial time, Algorithmica 64, 384–399 (2012).
20. Zs. Tuza, Graph colorings with local restrictions - a survey, Discuss. Math. Graph Theory 17, 161–228 (1997).
21. V.G. Vizing, Coloring the vertices of a graph in prescribed colors, in Diskret. Analiz., no. 29, Metody Diskret. Anal. v. Teorii Kodov i Shem 101, 3–10 (1976).