3-Colouring graphs without triangles or induced paths on seven vertices

Flavia Bonomo∗1, Maria Chudnovsky†2, Peter Maceli2, Oliver Schaudt3, Maya Stein‡4, and Mingxian Zhong2

1 CONICET and Departamento de Computación, FCEN, Universidad de Buenos Aires, Argentina. E-mail: fbonomo@dc.uba.ar
2 Columbia University, New York, NY 10027, USA. E-mails: mchudnov@columbia.edu, plm2109@columbia.edu, mz2325@columbia.edu
3 Institut für Informatik, Universität zu Köln, Germany. E-mail: schaudto@uni-koeln.de
4 Centro de Modelamiento Matemático, Universidad de Chile, Santiago, Chile. E-mail: mstein@dim.uchile.cl

Abstract

We present an algorithm to 3-colour a graph $G$ without triangles or induced paths on seven vertices in $O(|V(G)|^7)$ time. In fact, our algorithm solves the list 3-colouring problem, where each vertex is assigned a subset of $\{1, 2, 3\}$ as its admissible colours.

Keywords: 3-colouring, list 3-colouring, $\{P_7, \text{triangle}\}$-free graphs, polynomial time algorithm.
MSC: 05C15, 05C85, 05C38.

1 Introduction

A colouring of a graph $G = (V, E)$ is a function $f : V \to \mathbb{N}$ such that $f(v) \neq f(w)$ whenever $vw \in E$. A $k$-colouring is a colouring $f$ such that $f(v) \leq k$ for every $v \in V$. The vertex colouring problem takes as input a graph $G$ and a natural number $k$, and consists in deciding whether $G$ is $k$-colourable or not. This well-known problem is one of Karp’s 21 NP-complete problems [13]. It remains NP-complete even for triangle-free graphs [17], even if $k \geq 3$ is fixed.

In order to take into account particular constraints that arise in practical settings, more elaborate models of vertex colouring have been defined in the literature. One of such generalized models is the list-colouring problem, which considers a prespecified set of available colours for each vertex. Given a graph $G = (V, E)$ and a finite list $L(v) \subseteq \mathbb{N}$ for each vertex $v \in V$, the list-colouring problem asks for a list-colouring of $G$, i.e., a colouring $f$ such that $f(v) \in L(v)$ for
every \(v \in V\). Since list-colouring generalizes \(k\)-colouring, it is NP-complete as well. Nevertheless, if \(|L(v)| \leq 2\) for each vertex \(v\) in \(V\), the instance can be solved in \(O(|V|^2)\) time, by reducing the problem to a 2-SAT instance [5, 19].

Many classes of graphs where the vertex colouring problem is polynomially solvable are known, the most prominent being the class of perfect graphs [6]. For a set \(M\) of graphs, a graph \(G\) is \(M\)-free if no member of \(M\) is an induced subgraph of \(G\), and for a graph \(H\), a graph \(G\) is \(H\)-free if it is \(\{H\}\)-free. Let \(P_k\) and \(C_k\) denote a path and a cycle on \(k\) vertices, respectively. We will call the graph \(C_3\) a triangle.

The following is an overview of the complexity of colouring problems on \(H\)-free graphs. Kamiński and Lozin [12] and independently Kráľ, Kratochvíl, Tuza, and Woeginger [14], proved that, for any fixed \(k \geq 3\) and \(g \geq 3\), the \(k\)-colouring problem is NP-complete for the class of graphs containing no cycle of length less than \(g\). In particular, for any graph \(H\) containing a cycle, the \(k\)-colouring problem is NP-complete for \(k \geq 3\) on \(H\)-free graphs. On the other hand, if \(H\) is a forest with a vertex of degree at least 3, then \(k\)-colouring is NP-complete for \(H\)-free graphs and \(k \geq 3\) [9, 15]. Combining these results, the remaining cases are those in which \(H\) is a union of disjoint paths. There is a nice recent survey by Hell and Huang on the complexity of colouring graphs without paths and cycles of certain lengths [7].

The strongest known results related with our work are due to Huang [10] who proved that 4-colouring is NP-complete for \(P_7\)-free graphs, and that 5-colouring is NP-complete for \(P_6\)-free graphs. On the positive side, Hoang, Kamiński, Lozin, Sawada, and Shu [8] have shown that \(k\)-colouring can be solved in polynomial time on \(P_5\)-free graphs for any fixed \(k\). These results give a complete classification of the complexity of \(k\)-colouring \(P_t\)-free graphs for any fixed \(k \geq 5\), and leave only the case of \(P_6\)-free graphs open for \(k = 4\). For \(k = 3\), it is not known whether or not there exists any \(t\) such that 3-colouring is NP-complete on \(P_t\)-free graphs. Randerath and Schiermeyer [18] gave a polynomial time algorithm for 3-colouring \(P_6\)-free graphs. We solve in this paper the 3-colouring problem on \(P_7\)-free graphs containing no triangle.

This paper represents the work of two groups working on the same problem. First, the second, third and last author came up with an involved solution [4], that they have not yet published. Then the remaining authors were able to find a significant simplification [1], further extending similar ideas. This manuscript contains the simplified version. The case of 3-colouring graphs that are \(P_7\)-free but contain a triangle was solved by the second, third and last author [3].

**Theorem 1.** Given a \(\{P_7,\text{triangle}\}\)-free graph \(G\), it can be decided in \(O(|V(G)|^7)\) time whether \(G\) admits a 3-colouring. If a 3-colouring exists, it can be computed in the same time.

Our algorithm can be used even if each vertex is assigned a list of admissible colours from \(\{1, 2, 3\}\), and thus solves the so-called list 3-colouring problem. This is not trivial: in the class of \((P_6, C_5)\)-free graphs for example, 4-colouring can be solved in polynomial time [2] while the list 4-colouring problem is NP-complete [11].

After proving Theorem 1 in the subsequent sections, we sketch in Section 5 how the list 3-colouring problem can be solved.

### 1.1 Basic definitions

Let \(G\) be a finite, simple, loopless, undirected graph, with vertex set \(V(G)\) and edge set \(E(G)\). The graph \(G\) will be called trivial if \(|V(G)| = 1\).

Let \(v \in V(G)\) and \(A \subseteq V(G)\). Denote by \(N(v)\) the set of neighbours of \(v\) in \(V(G)\), and by \(N(A)\) the set of neighbors of vertices of \(A\) in \(V(G)\). Two vertices \(u\) and \(v\) of a graph \(G\) are
false twins if and only if $N(u) = N(v)$ (in particular, they are non-adjacent). Let $A, B \subseteq V(G)$. We denote by $N_B(A)$ the set $N(A) \cap B$. For any $W \subseteq V(G)$, $G[W]$ denotes the subgraph of $G$ induced by $W$. If $H$ is an induced subgraph of $G$ (resp. a subset of vertices of $G$), we denote by $G - H$ the graph $G[V(G) - V(H)]$ (resp. $G[V(G) - H]$).

A stable set is a subset of pairwise non-adjacent vertices. A graph $G$ is bipartite if $V(G)$ can be partitioned into two stable sets.

2 The structure of the graph $G$

2.1 The core structure

Let a $\{P_7, \text{triangle}\}$-free graph $G$ be given, with $n$ vertices and $m$ edges. We may assume that $G$ is connected. If $G$ is bipartite, which can be decided in $O(m)$ time, then $G$ is 2-colourable and we are done. So assume $G$ is not bipartite. Since $G$ is $\{P_7, \text{triangle}\}$-free, the shortest odd cycle of $G$ has either length 5 or length 7, and it is an induced cycle. We first discuss the case that $G$ contains no $C_5$.

Claim 1. If $G$ is $C_5$-free, then after identifying false twins in $G$, the remaining graph is $C_7$.

Proof. As argued above, since $G$ is not bipartite and contains no $P_7$, triangle, or $C_5$, we know $G$ contains an induced cycle $C = v_1, \ldots, v_7$ of length 7. Suppose some vertex $v \in V(G - C)$ has neighbours in $C$. If $v$ has only one neighbour $v_i$, then $G$ contains an induced $P_7$, namely $v, v_i, v_{i+1}, \ldots, v_{i-2}$. (As usual, index operations are done modulo 7). By the absence of triangles, $v$ has at most three neighbours in $C$, and they are pairwise non-consecutive. If $v$ has two neighbours $v_i, v_{i+3}$ at distance three in $C$, then $v_i, v_{i+1}, v_{i+2}, v_{i+3}, v$ induce a $C_5$, a contradiction. So $v$ has only two neighbours and they are at distance two in $C$.

For $i = 1, \ldots, 7$, let $V_i$ be the set formed by $v_i$ and the vertices not in $C$ whose neighbours in $C$ are $v_{i-1}$ and $v_{i+1}$. As $G$ is triangle-free, $V_i$ is a stable set. Since $G$ is $P_7$-free, every vertex in $V_i$ is adjacent to every vertex in $V_{i+1}$. Moreover, since $G$ is connected and $P_7$-free, there are no vertices outside $\bigcup_{i=1}^7 V_i$. As $G$ is $\{\text{triangle}, C_5\}$-free, there are no edges between $V_i$ and $V_j$, for $j \notin \{i+1, i-1\}$. So, for each $i$, the vertices of $V_i$ are false twins, and after identifying them we obtain $C_7$. □

If $G$ is a ‘blown-up’ $C_7$, then it is clearly 3-colourable (as false twins can use the same colour). Thus, Claim 1 enables us to assume $G$ has a cycle $C$ of length 5, say its vertices are $c_1, c_2, c_3, c_4, c_5$, in this order. From now on, all index operations will be done modulo 5. Because $G$ has no triangles, the neighbourhood $N_C$ of $V(C)$ in $G$ is comprised of 10 sets (some of these possibly empty):

- sets $T_i$, whose neighbourhood on $C$ is equal to $\{c_{i-1}, c_{i+1}\}$;

- sets $D_i$, whose only neighbour on $C$ is $c_i$;

where the indices $i$ go from 1 to 5. Note that, because of $G$ being triangle-free, the sets $T_i$ and $D_i$ are each stable. We set $S := V(C) \cup \bigcup_{i=1}^5 T_i \cup \bigcup_{i=1}^5 D_i$. Note that $V(C)$ has no neighbours in $G - S$. 
2.2 The non-trivial components of \( G - S \)

The following list of claims narrows down the structure of the non-trivial components of \( G - S \).

**Claim 2.** If \( xy \) is an edge in \( G - S \), then \( x \) and \( y \) have no neighbours in any of the sets \( D_i \).

*Proof.* Suppose that \( x \) has a neighbour \( u \) in \( D_1 \). Then as \( G \) is triangle-free, \( y \) is not adjacent to \( u \). So \( y xu c_1 c_2 c_3 c_4 \) is an induced \( P_7 \), a contradiction. The other cases are symmetric. \( \square \)

**Claim 3.** If \( xyz \) is an induced \( P_3 \) in \( G - S \), then the neighbourhoods of \( x \) and \( z \) inside each \( T_i \) are identical.

*Proof.* Suppose that \( x \) has a neighbour \( u \) in \( T_1 \) that is not a neighbour of \( z \). Then, as \( G \) is triangle-free, \( y \) is not adjacent to \( u \), either. Also, \( x \) and \( z \) are not adjacent. So \( y xu c_2 c_3 c_4 \) is an induced \( P_7 \), a contradiction. We argue similarly for all other \( T_i \)'s. \( \square \)

**Claim 4.** \( G - S \) is bipartite.

*Proof.* Assume \( G - S \) has an odd cycle \( C' \). Take a shortest path \( P = c' p_1 \ldots p_k s \) from \( V(C') \) to \( S \). Since \( G \) is triangle-free, and \( C' \) is odd, there is a induced path \( c'_1 c'_2 c'_3 p_1 \) with \( c'_1 \in V(C') \). Further, as \( P \) was chosen as a shortest path, there are no edges of the form \( c'_i p_j \) except for \( c'_3 p_1 \), and no edges of the form \( p_j s \) except for \( p_k s \). So we can complete \( c'_1 c'_2 c'_3 p_1 \ldots p_k s \) with three vertices from \( C \) to obtain an induced path of length at least 7, a contradiction. \( \square \)

**Claim 5.** Let \( M \) be a non-trivial component of \( G - S \). Then there is a partition of \( V(M) \) into stable sets \( U_1, U_2 \) such that all vertices in \( U_i \) have the same set \( N_i \) of neighbours in \( S \), at least one of \( N_1, N_2 \) is non-empty, and \( N_1 \cap N_2 = \emptyset \).

*Proof.* Directly from the two previous claims, and the fact that the graph is connected and triangle-free. \( \square \)

We need one more claim about independent edges outside \( S \).

**Claim 6.** Suppose \( uv, xy \in E(G) \) induce a \( 2K_2 \) in \( G - S \). Then, for every \( i = 1, \ldots, 5 \), \( N_{T_i}(x) \cup N_{T_i}(y) \subseteq N_{T_i}(v) \cup N_{T_i}(w) \), or \( N_{T_i}(v) \cup N_{T_i}(w) \subseteq N_{T_i}(x) \cup N_{T_i}(y) \).

*Proof.* If none of these inclusions holds, then there are vertices \( u, z \in T_i \) such that \( uv, yz \in E(G) \) and \( u \notin N_{T_i}(x) \cup N_{T_i}(y), \ z \notin N_{T_i}(v) \cup N_{T_i}(w) \) (after possibly swapping some names). Since \( G \) is triangle-free, also \( u \notin N_{T_i}(w) \) and \( z \notin N_{T_i}(x) \). So, after possibly swapping some names, \( xyzc_{i+1}uw \) is an induced \( P_7 \), a contradiction. \( \square \)

(Observe that we can not extend the last claim to the neighbourhood in all of \( S \), because then \( u \) and \( z \) might be adjacent.)

2.3 The trivial components of \( G - S \)

Let \( W \) be the set of isolated vertices in \( G - S \). We will first show some properties of \( W \) and their neighbours in \( S \).

**Claim 7.** There is no vertex in \( W \) having neighbours in both \( D_i \) and \( D_{i+1} \), \( i = 1, \ldots, 5 \).

*Proof.* Suppose \( w \) has neighbours \( d_1 \) in \( D_1 \) and \( d_2 \) in \( D_2 \). Then \( d_1 wd_2 c_2 c_3 c_4 c_5 \) is a \( P_7 \) in \( G \), a contradiction. The other cases are symmetric. \( \square \)
As both $W$ and $D_i$ are stable, $G[W \cup D_i]$ is bipartite, for every $i = 1, \ldots, 5$. We now show some properties similar to the ones we showed for the non-trivial components of $G - S$ in Section 2.2.

**Claim 8.** If $xyz$ is a $P_3$ in $G[W \cup D_i]$, then the neighbourhoods of $x$ and $z$ inside $T_i$ are identical, for $i = 1, \ldots, 5$.

**Proof.** Suppose that $x$ has a neighbour $u$ in $T_i$ that is not a neighbour of $z$. Then as $G$ is triangle-free, $y$ is not adjacent to $u$, either. So $zyxc_{i+1}c_{i+2}c_{i+3}$ is an induced $P_7$, a contradiction. \hfill \Box

**Claim 9.** Let $M$ be a non-trivial component of $G[W \cup D_i]$, $i \in \{1, \ldots, 5\}$. Then all vertices in $M \cap W$ have the same set of neighbours in $T_i$, and all vertices in $M \cap D_i$ have the same set of neighbours in $T_i$.

**Proof.** Directly from the previous claim, and the fact that $G[W \cup D_i]$ is bipartite. \hfill \Box

Finally, we extend Claim 6 to connected components of $G[W \cup D_i]$ (for $i = 1, \ldots, 5$).

**Claim 10.** Let $i \in \{1, \ldots, 5\}$. Suppose $vw, xy \in E(G)$ induce a $2K_2$ in $G[(G - S) \cup D_i]$. Then, $N_{T_i}(x) \cup N_{T_i}(y) \subseteq N_{T_i}(v) \cup N_{T_i}(w)$, or $N_{T_i}(v) \cup N_{T_i}(w) \subseteq N_{T_i}(x) \cup N_{T_i}(y)$.

**Proof.** If none of these inclusions holds, then there are vertices $u, z \in T_i$ such that $uv, yz \in E(G)$ and $u \notin N_{T_i}(x) \cup N_{T_i}(y)$, $z \notin N_{T_i}(v) \cup N_{T_i}(w)$ (after possibly swapping some names). Since $G$ is triangle-free, also $u \notin N_{T_i}(w)$ and $z \notin N_{T_i}(x)$. So, after possibly swapping names, $xyzc_{i+1}uvw$ is an induced $P_7$, a contradiction. \hfill \Box

### 3 The algorithm

Our strategy, as usual for this kind of problem, is to reduce 3-colouring to a polynomial number of instances of list-colouring where the lists have length at most two. The latter problem is known to be solvable in polynomial time via a reduction to 2-SAT [5, 19]. To be precise, we will reduce our problem to a polynomial number of instances of list-colouring where every vertex either has a list of size at most 2, or there is a colour $j \in \{1, 2, 3\}$ missing in the list of each of its neighbours, and so $v$ can safely use colour $j$.

First of all, we fix a colouring of the 5-cycle $C$ (there are 5 essentially different colourings). For each $D \in \{T_1, D_1, \ldots, T_5, D_5\}$, the colouring of $C$ either determines the colour of $D$, or the vertices of $D$ lose a colour. Notice that, once we have fixed the colouring of $C$, three of the $T_i$ have their colour determined, while two of them (consecutive sets, actually) have two possible colours left. For instance, if we colour $c_1, c_2, c_3, c_4, c_5$ with 1, 2, 1, 2, 3, respectively, then vertices in $T_1$ will be forced to have colour 1, vertices in $T_4$ colour 2, vertices in $T_5$ colour 3, vertices in $T_2$ have the options $\{2, 3\}$ and vertices in $T_3$ have the options $\{1, 3\}$. The vertices in $D_i$, $i = 1, \ldots, 5$, have lost one colour each.

We will work with this colouring of $V(C)$, since the other 4 are totally symmetric. In addition to what we observed above about possible colourings of the sets $T_i$ and $D_i$, we know that each neighbour of $T_1 \cup T_4 \cup T_5$ has already lost a colour. We have to deal now with the vertices having neighbours only in $T_2$, $T_3$, and $D_i$, for $i = 1, \ldots, 5$, and vertices having no neighbours in $S$.

Note that, by Claim 2, no connected component of $G[(G - S) \cup D_i]$ may contain at the same time edges from $G - S$ and vertices of $D_i$. So, from Claims 5, 9 and 10 we know that for each
Figure 1: Scheme of the coloured cycle and its neighbours with their possible colours.

For any partial colouring as above and for any vertex $x$ of some non-trivial component $M$ of $G - S$, we have the following. Either $x$ has only two colours left on its list, or there is a colour $j \in \{1, 2, 3\}$ missing in the list of each of the neighbours of $x$ (and so $x$ can safely use colour $j$).
Proof. By Claim 2, for each partial colouring as above, either there are two vertices of different
colour in $N(M) \cap S$, or $N(M) \cap S$ is completely coloured. In the first case, either $x$ is adjacent
to a coloured vertex in $S$ (so it loses a colour), or, by Claim 5, its neighbours in $M$ have two
neighbours in $S$ of different colour, thus fixed their colour, and again, $x$ loses a colour. In the
second case, if $x$ has neighbours in $S$, then it loses a colour. If not, then again by Claim 5, all
the neighbours of $x$ are in $M$ and have lost a common colour $j$ from their coloured neighbours
in $S$, so we are done. 

Claim 12. For any partial colouring as above and for any vertex of any non-trivial component
of $G[W \cup D_i]$, we have the following. If $x$ has a neighbour in $T_i$, then $x$ has lost a colour on its
list.

Proof. The proof is analogous to the proof of the previous claim, replacing Claim 5 with Claim 9,
and $S$ with $T_i$. 

Claim 13. In any partial colouring as above, every vertex $w \in W$ having neighbours both in $T_2$
and in $T_3$ loses a colour.

Proof. Notice that either $T_2$ is monochromatic with colour 2, or $T_3$ is monochromatic with colour
1 (in either case, $w$ loses a colour), or there are two non-adjacent vertices $x_2 \in T_2$ and $x_3 \in T_3$
having colour 3. Let $w$ in $W$ have neighbours $y_2$ in $T_2$ and $y_3$ in $T_3$. Observe that $y_2$ and $y_3$
are not adjacent because there are no triangles. Then either $w$ is adjacent to $x_2$ or to $x_3$ (and hence
loses a colour), or $y_2$ is adjacent to $x_3$, or $y_3$ is adjacent to $x_2$, since otherwise $x_3c_1y_3wy_2c_1x_2$
is an induced $P_7$ in $G$, a contradiction. If $y_2$ is adjacent to $x_3$ then it has to be coloured 2, and if
$y_3$ is adjacent to $x_2$ then it has to be coloured 1. In either case, $w$ loses a colour.

In total, we enumerate
\[
O(|N^2_1| + |N^2_2 \setminus N^2_1| + \cdots + |N^2_{r_2+1} \setminus N^2_{r_2}|) \cdot (|N^3_1| + |N^3_2 \setminus N^3_1| + \cdots + |N^3_{r_3+1} \setminus N^3_{r_3}|)) \\
= O(|T_2| \cdot |T_3|)
\]
many partial colourings, that is, $O(n^2)$. For each of these, we will solve a set of instances of
list-colouring with lists of size at most two, after dealing with the trivial components of $G - S$
as detailed in the next paragraphs.

Since $G$ is connected, each $w \in W$ has a neighbour in $S$. If $w$ has a neighbour in a set with
fixed colour, then $w$ has at most 2 colours on his list. On the other hand, if there is a colour
$j \in \{1, 2, 3\}$ missing in the list of each of its neighbours, then $w$ can safely use colour $j$. So
we have to deal only with vertices in $W$ having no neighbours in $T_1$, $T_4$ and $T_5$, and having
neighbours in at least two sets with different colour options in $\{T_2, T_3, D_1, \ldots, D_5\}$. Vertices of
$W$ having neighbours in $T_2$ and $D_2$, or in $T_3$ and $D_3$, have already lost a colour by Claim 12.
Vertices of $W$ having neighbours in $T_2$ and $T_3$, have already lost a colour by Claim 13.
So, the types of vertices in $W$ we need to consider are the following (a scheme of the situation
can be seen in Figure 1).

- **Type 1**: vertices in $W$ having neighbours in $T_2$ and in $D_5$; symmetrically, vertices in $W$
having neighbours in $T_3$ and $D_5$; vertices in $W$ having neighbours in $T_2$ and $D_4$; vertices
in $W$ having neighbours in $T_3$ and $D_4$.

- **Type 2**: vertices in $W$ having neighbours in $D_2$ and in $D_5$; symmetrically, vertices in $W$
having neighbours in $D_3$ and $D_5$; and vertices in $W$ having neighbours in $D_1$ and $D_4$. 


Pick an arbitrary vertex \( v_i \in D_i \), for \( i = 1, 4, 5 \). We now extend each partial colouring from above by further enumerating the colouring of some vertices in \( D_i \), for all \( i = 1, 4, 5 \) simultaneously and independently. For this, say that \( \{a_i, b_i\} \) are the possible colours in \( D_i \).

(e) Vertex \( v_i \) gets colour \( a_i \) and there is a vertex \( v' \) in \( D_i - \{v_i\} \) getting colour \( b_i \).

(f) Vertex \( v_i \) gets colour \( b_i \) and there is a vertex \( v' \) in \( D_i - \{v_i\} \) getting colour \( a_i \).

(g) All vertices in \( D_i \) get colour \( a_i \).

(h) All vertices in \( D_i \) get colour \( b_i \).

Again, every 3-colouring of \( G \) that agrees with the colouring of the induced \( C_5 \) is an extension of one of the enumerated partial colourings.

Notice that for each \( D_i \) these are \( O(|D_i|) \) cases, so the total number of combinations of cases is \( O(n^3) \). Again, we discard a partial colouring if there are adjacent vertices receiving the same colour, or when some vertex has no colour left.

It remains to show that for each partial colouring, all the remaining vertices in \( W \) (Type 1 and Type 2) lose a colour.

First we discuss vertices of Type 1. Suppose some \( w \) in \( W \) has neighbours in \( T_2 \) and in \( D_5 \). For each case of (e)–(h), either \( D_5 \) is monochromatic with colour 1 (so \( w \) loses a colour), or there is a vertex \( x \) in \( D_5 \) with colour 2. Then either \( w \) is adjacent to \( x \) (and loses a colour), or every neighbour \( z \) of it in \( T_2 \) is adjacent to \( x \), otherwise, by triangle freeness, \( xczvwzxc_2c_3 \), where \( v \) is some neighbour of \( w \) in \( D_5 \), is a \( P_7 \), a contradiction. Then \( z \) has to be coloured 3, and \( w \) loses a colour. The other cases are symmetric.

Now we discuss vertices of Type 2. Suppose some \( w \) in \( W \) has neighbours in \( D_2 \) and in \( D_5 \). Such a vertex can be dealt with in exactly the same way as the vertices of Type 1. The only difference is the path \( P_7 \), which here is given by \( xcvzwzxc_2c_3 \), where \( x, v \in D_5 \) and \( z \in D_2 \) are neighbours of \( w \). The other cases are symmetric.

4 The overall complexity of the algorithm

4.1 Finding the \( C_5 \) and partitioning the graph

We check if \( G \) is bipartite in \( O(m) \) time. If it is not, we will find either an induced \( C_5 \) or an induced \( C_7 \). If we find an induced \( C_7 \), let us call it \( X \), we check the neighbourhood of \( N(X) \) in \( X \): either we find a \( C_5 \) or we partition the vertices into sets that are candidates to be false twins of the vertices of \( X \) (see Claim 1). The adjacencies must hold because of the \( P_7 \)-freeness, and the sets are stable and there are no edges between vertices in sets corresponding to vertices at distance 2 in \( X \) because of the triangle-freeness. While checking for edges between vertices in sets corresponding to vertices at distance 3 in \( X \), we will either find a \( C_5 \) or conclude that the graph is \( C_5 \)-free and after identifying false twins we obtain \( C_7 \). For this, we just need to check for each edge, if their endpoints are labelled with numbers corresponding to vertices at distance 3 in \( X \). The whole step can be done in \( O(m) \) time.

Assuming we found an induced \( C_5 \), partitioning the neighbours \( S \) of this \( C_5 \) into the sets \( T_1, D_1, \ldots, T_5, D_5 \) can be also done in linear time. Moreover, finding the connected components of \( S \) and their neighbourhoods on each of the \( T_i \)'s can be done in \( O(m) \) time.
4.2 Exploring the cases to get the list-colouring instances

The number of distinct colourings of the $C_5$ to consider is 5. For each of them, using the notation of the colouring of the last section, there are $O(n^2)$ many combinations to be tested on $T_2$ and $T_3$ in order to deal with the non-trivial components of $G - S$, $G[W \cup D_2]$ and $G[W \cup D_3]$, and the vertices in $W$ having neighbours in both $T_2$ and $T_3$ (cases (a)–(d)). For each of them, we have to test all the combinations for $D_1$, $D_4$ and $D_5$ (cases (e)–(h)), that are $O(n^3)$ many, in order to deal with the remaining vertices in $W$. Finally, for each of these possibilities, we have to solve a list-colouring instance with lists of size at most 2, which can be done in $O(n^7)$ time [5, 19]. Thus, the overall complexity of the algorithm is $O(n^7)$.

5 List 3-colouring

When there is no induced $C_5$, the graph is either bipartite or, according to Claim 1, a blown-up $C_7$. In the bipartite case, we may use the algorithm of the second and third author [16]. It solves the list 3-colouring problem for bipartite $P_7$-free graphs in $O(n^6)$ time. In the case of a blown-up $C_7$, we may simply identify twins that have identical lists. Since we have at most 8 possible distinct lists, the remaining graph has at most 56 vertices and we are done.

When the input graph has an induced $C_5$, the strategy consists of reducing our problem to a number of instances of list-colouring where every vertex $v$ either has a list of size at most 2, or there is a colour $j \in \{1, 2, 3\}$ missing in the list of each of its neighbours, and so $v$ can safely use colour $j$. In order to do that, we pre-colour some vertices, in particular those of the $C_5$. To take into account the list restrictions, we have to consider all possible (feasible) colourings of the $C_5$ (at most 30 instead of 5), since the colours are not playing symmetric roles now. Also, we may simply close a branch in the enumeration whenever a vertex is assigned a colour not contained in its list. Finally, for the cases in which our argument was that for some vertex $v$ there is a colour $j \in \{1, 2, 3\}$ missing in the list of each of its neighbours so we can assign colour $j$ to $v$, it may happen now that $j$ is not in the list of $v$. But then $v$ has a list of size at most 2, which is equally useful for our purposes.

Acknowledgements

BSS thank Daniël Paulusma for his observation that the present algorithm can actually deal with list 3-colourings. CMZ are grateful to Paul Seymour and Juraj Stacho for many fruitful discussions. Finally, they thank Alex Scott for telling them about the problem that this paper solves.

References

[1] F. Bonomo, M. Stein, and O. Schaudt. 3-colouring graphs without triangles or induced paths on seven vertices. arXiv:1410.0040 [math.CO], September 2014.

[2] M. Chudnovsky, P. Maceli, J. Stacho, and M. Zhong. 4-coloring $P_6$-free graphs with no induced 5-cycles. arXiv:1407.2487v1 [cs.DM], July 2014.

[3] M. Chudnovsky, P. Maceli, and M. Zhong. Three-coloring graphs with no induced seven-vertex path II: using a triangle. In preparation.
[4] M. Chudnovsky, P. Maceli, and M. Zhong. Three-coloring graphs with no induced seven-vertex path I: the triangle-free case. arXiv:1409.5164v1 [math.CO], September 2014.

[5] P. Erdős, A. Rubin, and H. Taylor. Choosability in graphs. Congressus Numerantium, 26:125–157, 1979.

[6] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. Combinatorica, 1:169–197, 1981.

[7] P. Hell and S. Huang. Complexity of coloring graphs without paths and cycles. arXiv:1310.0340v1 [cs.DM], September 2013.

[8] C.T. Hoàng, M. Kamiński, V.V. Lozin, J Sawada, and X. Shu. Deciding $k$-colorability of $P_5$-free graphs in polynomial time. Algorithmica, 57:74–81, 2010.

[9] I. Holyer. The NP-completeness of edge-coloring. SIAM Journal on Computing, 10:718–720, 1981.

[10] S. Huang. Improved complexity results on $k$-coloring $P_t$-free graphs. In Proceedings of the Mathematical Foundations of Computer Science 2013, volume 7551 of Lecture Notes in Computer Science, pages 551–558, 2013.

[11] S. Huang, M. Johnson, and D. Paulusma. Narrowing the complexity gap for colouring $(C_s,P_t)$-free graphs. arXiv:1407.1480v1 [cs.CC], July 2014.

[12] M. Kamiński and V.V. Lozin. Coloring edges and vertices of graphs without short or long cycles. Contributions to Discrete Mathematics, 2:61–66, 2007.

[13] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, Complexity of Computer Computations, pages 85–103. Plenum Press, New York, 1972.

[14] D. Král, J. Kratochvíl, Zs. Tuza, and G.J. Woeginger. Complexity of coloring graphs without forbidden induced subgraphs. In M.C. Golumbic, M. Stern, A. Levy, and G. Morgenstern, editors, Proceedings of the International Workshop on Graph-Theoretic Concepts in Computer Science 2001, volume 2204 of Lecture Notes in Computer Science, pages 254–262, 2001.

[15] D. Leven and Z. Galil. NP-completeness of finding the chromatic index of regular graphs. Journal of Algorithms, 4:35–44, 1983.

[16] P. Maceli. Excluding induced paths: Graph structure and coloring. PhD thesis, Columbia University.

[17] F. Maffray and M. Preissmann. On the NP-completeness of the $k$-colorability problem for triangle-free graphs. Discrete Mathematics, 162:313–317, 1996.

[18] B. Randerath and I. Schiermeyer. $3$-Colorability $\in P$ for $P_6$-free graphs. Discrete Applied Mathematics, 136:299–313, 2004.

[19] V. Vizing. Coloring the vertices of a graph in prescribed colors. Metody Diskretnogo Analiza, 29:3–10, 1976.