Coloring graphs with forbidden induced subgraphs

Maria Chudnovsky*

Abstract. Since graph-coloring is an NP-complete problem in general, it is natural to ask how the complexity changes if the input graph is known not to contain a certain induced subgraph H. Results of Kamínski and Lozin, Holyer, and Levin and Galil imply that the problem remains NP-complete, unless H is the disjoint union of paths. Recently, the question of coloring graphs that do not contain certain induced paths has received considerable attention. Only one case of that problem remains open for k-coloring when $k \geq 4$, and that is the case of 4-coloring graphs with no induced 6-vertex path. However, little is known for 3-coloring. In this paper we survey known results on the topic, and discuss recent developments.

Mathematics Subject Classification (2010). Primary 05C15; secondary 05C85.

Keywords. Graph coloring, induced subgraphs, coloring algorithms.

1. Introduction

Let G be a graph. We denote by $V(G)$ the vertex set of G, and by $E(G)$ the edge set of G. For a positive integer k, a *k-coloring* of G is a function $c : V(G) \to \{1, \ldots, k\}$ such that $c(u) \neq c(v)$ for every adjacent pair of vertices u, v; if such a function exists, we say that G *admits a k-coloring* or *is k-colorable.* The *chromatic number* $\chi(G)$ of a graph G is the smallest number k for which G admits a k-coloring. The algorithmic problem of determining the chromatic number of a graph is notoriously difficult; in fact it was one of the initial problems Karp showed to be NP-complete [14]. The algorithmic question remains difficult if we fix the parameter $k \geq 3$ (as opposed to allowing it to be part of the input), and ask to determine whether a given graph is k-colorable. This problem (knows as *the k-coloring problem*) was shown to be NP-complete in [18]. (Determining if a graph is 2-colorable is easy). It is therefore of interest to establish classes of graphs in which the k-coloring problem can be solved efficiently (i.e. in time that is a polynomial function of the size of the input graph). In this paper we focus on classes of graphs defined by forbidding certain substructures, called induced subgraphs.

*Partially supported by NSF grants IIS-1117631 and DMS-1265803.

2. Making the problem easier

In this paper when we say that an algorithm runs "in polynomial time" or is a "polynomial-time algorithm" we always mean "time polynomial as a function of the number of vertices in the input graph"; we will not concern ourselves with the exact polynomial function that provides a bound on the complexity.

A *cycle* C_t is a graph with vertices $c_1, \dots, c_t$ such that c_i is adjacent to c_j if and only if $|i-j| =\in \{1, t-1\}$, and a path P_t is a graph with vertices $p_1, \dots, p_t$ such that p_i is adjacent to p_j if and only if $|i-j| = 1$. Let G be a graph. An *induced subgraph* of G is a graph H such that $V(H) \subseteq V(G)$, and $uv \in E(H)$ if and only if $uv \in E(G)$ for all $u, v \in V(H)$. Given graphs G and F we say that G *contains* F if F is isomorphic to an induced subgraph of G. G is F*-free* if G does not contain F. The question then becomes: for which graphs F and integers k can the k-coloring problem be solved in polynomial time for F-free graphs?

The discussion in the remainder of this section assumes that $P \neq NP$. Unfortunately, the news is not good on this front, because of the following powerful result of Kaminski and Lozin [13]:

Theorem 2.1. *For every $k, g \geq 3$, the problem of k-coloring graphs with no cycles of length at most g is NP-complete.*

Applying 2.1 with $g = |V(H)|$, we then obtain the following:

Theorem 2.2. *Let H be a graph with a cycle. For every $k \geq 3$, the problem of k-coloring H-free graphs is NP-complete.*

In other words, if the k-coloring problem is polynomial-time solvable for the class of H-free graphs (where $k \geq 3$), then H is a forest. It turns out that further restrictions need to be placed on H before excluding H as an induced subgraph may impact the complexity of k-coloring. A *k-edge-coloring* of a graph G is a function $c : E(G) \to \{1, \dots, k\}$ such that $c(e) \neq c(f)$ for every pair e, f of edges of G that share and end. The algorithmic problem of k-edge-coloring is NP-complete for all fixed $k \geq 3$ [12, 15].

Next consider a construction. The *line graph* of a graph G, denoted by $L(G)$, is the graph with vertex set $E(G)$, and e and f are adjacent in $L(G)$ if and only if they share an end in G. Clearly k-edge-coloring G and k-coloring $L(G)$ are equivalent problems. A *claw* is the graph with four vertices x, y, z, w, and three edges xy, xz, xw. It is an easy exercise to show that line graphs are claw-free. We therefore deduce:

Theorem 2.3. *For every $k \geq 3$, the problem of k-coloring claw-free graphs is NP-complete.*

Consequently,

Theorem 2.4. *Let H be a graph that contains a claw. For every $k \geq 3$, the problem of k-coloring H-free graphs is NP-complete.*

A *component* of a graph is its maximal connected subgraph. Together 2.2 and 2.4 imply the following:

Theorem 2.5. *Let $k \geq 3$ be an integer, and H be a graph. If the problem of determining whether an H-free graph is k-colorable can be solved in polynomial time, then every component of H is a path.*

The remainder of this paper deals with coloring H-free graphs, where H is a path.

3. Excluding induced paths

In this section we summarize what is currently known about the complexity of coloring graphs with certain induced paths excluded. It turns out that excluding short induced paths does in fact help with coloring. P_2-free graphs have no edges, P_3-free graphs are disjoint unions of complete graphs, and P_4-free graphs with at least two vertices are either not connected or not connected in the complement (by a theorem of Seinsche [17]); in all cases polynomial-time coloring algorithms can easily be constructed. The first non-trivial case is the class of P_5-free graphs:

Theorem 3.1. *[10] For every integer k, the k-coloring problem can be solved in polynomial time for the class of P_5-free graphs.*

On the other hand,

Theorem 3.2. *[11]*

1. *The 5-coloring problem is NP-complete for the class of P_6-free graphs.*
2. *The 4-coloring problem is NP-complete for the class of P_7-free graphs.*

This immediately implies that:

Theorem 3.3. *[11]*

1. *The k-coloring problem is NP-complete for the class of P_t-free graphs for all $k \geq 5$ and $t \geq 6$.*
2. *The 4-coloring problem is NP-complete for the class of P_t-free graphs for all $t \geq 7$.*

To deduce 3.3 from 3.2, observe that for every integer $t \geq 0$ a P_t-free graph is also P_{t+1}-free, and further note that if G is a P_t-free graph that is a difficult instance for k-coloring (where $k \geq 3$ and $t \geq 2$), then the graph obtained from G by adding a new vertex adjacent to all members of $V(G)$ is a P_t-free graph that is a difficult instance for $k+1$-coloring.

Finally, in [16] it is shown that:

Theorem 3.4. *[16] The 3-coloring problem can be solved in polynomial time for the class of P_6-free graphs.*

Our focus here is on a new result of [4, 5]:

Theorem 3.5. *[4, 5] The 3-coloring problem can be solved in polynomial time for the class of P_7-free graphs.*

Thus, at the time of the writing of this manuscript, the following cases remain open:

1. the complexity of 3-coloring P_t-free graphs where $t \geq 8$, and
2. the complexity of 4-coloring P_6-free graphs.

Recently there has been some progress on the second question:

Theorem 3.6. *[11] The 4-coloring problem can be solved in polynomial time for the class of graphs that are both P_6-free and C_4-free.*

and

Theorem 3.7. *[6] The 4-coloring problem can be solved in polynomial time for the class of graphs that are both P_6-free and C_5-free.*

There are many similarities between the proofs of 3.5 and 3.7, and we will discuss them both.

4. List coloring

Given a graph G and a function L from $V(G)$ to the set of all subsets of positive integers, a *coloring* of (G, L) is a function $c : V(G) \to \bigcup_{v \in V(G)} L(v)$ such that $c(u) \neq c(v)$ for every adjacent pair uv of vertices of G, and $c(v) \in L(v)$ for every $v \in V(G)$. In this case c is also called *list coloring*, and $L(v)$ is called *the list of* v. We say that (G, L) is *colorable* if such a coloring exists. Clearly k-coloring is an instance of list coloring, where $L(v) = \{1, \dots, k\}$ for all $v \in V(G)$.

Since list coloring is a generalization of coloring, it is an NP-complete problem in general. However, a few special cases can be solved in polynomial time, and we make use of this fact in our work. The first such case is the following:

Theorem 4.1. *[8] There is a polynomial-time algorithm with the following specifications:*

Input: *A pair (G, L) such that $|L(v)| \leq 2$ for all $v \in V(G)$.*

Output: *A coloring of (G, L), or a determination that none exists.*

The proof of 4.1 consists of a reduction to a well-known polynomial-time solvable problem called *2-SAT* that we do not describe here. In fact, a slightly more general result can be obtained using similar techniques (a set of vertices S is monochromatic in a given coloring c if $c(v) = c(u)$ for all $u, v \in S$.):

Theorem 4.2. *[1] There is a polynomial-time algorithm with the following specifications:*

Input:

1. *a pair (G, L) such that $|L(v)| \leq 2$ for all $v \in V(G)$, and*

2. *a list $S_1, \ldots, S_{|V(G)|^t}$ of subsets of $V(G)$, where t is an integer.*

Output: *A coloring of (G, L), so that each of $S_1, \ldots, S_{|V(G)|^t}$ is monochromatic, or a determination that none exists.*

Here is another, much more difficult, result from [1]

Theorem 4.3. *[1] There is a polynomial-time algorithm with the following specifications:*

Input: *A pair (G, L) such that G is P_6-free and $L(v) \subseteq \{1, 2, 3\}$ for all $v \in V(G)$.*

Output: *A coloring of (G, L), or a determination that none exists.*

Armed with these three theorems, our strategy for both 3.5 and 3.7 is to reduce the original problem of k-coloring a graph G to a polynomial number of problems $(G_1, L_1), \ldots, (G_t, L_t)$, such that

1. G is k-colorable if and only if there exists $i \in \{1, \ldots, t\}$ such that (G_i, L_i) is colorable,

2. each of $(G_1, L_1), \ldots, (G_t, L_t)$ can be solved efficiently using 4.1, 4.2 or 4.3, and

3. a k-coloring of G can be reconstructed from a coloring of (G_i, L_i) in polynomial time.

We remark that the following natural extension of 3.5 remains open:

Question 4.4. *Given a pair (G, L) where G is a P_7-free graph, and $L(v) \subseteq \{1, 2, 3\}$ for every $v \in V(G)$, what is the complexity of determining whether (G, L) is colorable?*

The corresponding extension of 3.7 is open as well. In the next section we will explain where our methods fall short for attacking these questions.

The only positive result we can report in this direction is the following:

Theorem 4.5. *[3] There is a polynomial-time algorithm with the following specifications:*

Input: *A pair (G, L) such that G is a bipartite P_7-free graph and $L(v) \subseteq \{1, 2, 3\}$ for all $v \in V(G)$.*

Output: *A coloring of (G, L) or a determination that none exists.*

5. Tools for 3-coloring

A *dominating set* in a graph G is a subset $S \subseteq V(G)$ such that every vertex of $V(G) \setminus S$ has a neighbor in S. For a subset X of $V(G)$, we denote by $G|X$ the subgraph of G induced by X. For $v \in V(G)$, we denote by $N(v)$ the set of vertices of G adjacent to v (in particular, $v \notin N(v)$).

In view of 4.1, the following seems like a natural approach to constructing the algorithm of 3.5:

1. Prove that there exists an integer K such that every connected P_7-free graph has a dominating set of size at most K.
2. Find a dominating set S of size at most K in G.
3. Let c be a coloring of $G|S$. Set $L(v) = \{c(v)\}$ for every $v \in S$, and
$$L(v) = \{1,2,3\} \setminus \bigcup_{u \in N(v)} \{c(u)\}$$
for every $v \in V(G) \setminus S$.

Clearly this is a polynomial-time procedure that reduces the original problem to at most $(3|V(G)|)^K$ instances (G, L), where each instance can be efficiently solved using 4.1.

Unfortunately, the first statement above is not true as stated, but the following questions is of interest:

Question 5.1. *Which P_7-free graphs G have a dominating set of size at most $\log |V(G)|$?*

So we cannot follow the approach outlined above directly. Instead, we identify a number of efficiently detectable "reducible configurations" in the input graph, making the graph simpler without changing its colorability properties, until a small dominating set emerges (possibly considering a number of different candidates for being a dominating set).

Next we list some examples of reducible configurations. For a graph G, a set $X \subseteq V(G)$ and a vertex $v \in V(G) \setminus X$, we say that v is *complete* to X if v is adjacent to every vertex of X, and that v is *anticomplete* to X if v has no neighbor in X.

Dominating vertex. $u, v \in V(G)$ such that $N(u) \subseteq N(v)$. Please note that it follows that u and v are non-adjacent. Clearly in this case G is 3-colorable if and only if $G \setminus u$ is 3-colorable (by making u and v be the same color). We remark that this simple reduction is the first obstacle we encounter when trying to apply our methods to 4.4, for in the list coloring setting we would additionally need to require that $L(v) \subseteq L(u)$, which is not an inherent structural property of a graph, and thus is harder to impose.

Homogeneous pair of stable sets. A *stable set* is a set of vertices all pairwise non-adjacent. Let $A, B \subseteq V(G)$ be disjoint and non-empty. We say that (A, B)

is a *homogeneous pair* in G if every vertex of $V(G) \setminus (A \cup B)$ with a neighbor in A is complete to A, and the same for B. If in addition both A and B are stable sets, then (A, B) is a *homogeneous pair of stable sets.* Let (A, B) be a homogeneous pair of stable sets such that there is at least one edge between A and B, and suppose that $|A| + |B| \geq 3$. Let $a \in A$ be adjacent to $b \in B$, and let $G' = G \setminus ((A \setminus \{a\}) \cup (B \setminus \{b\}))$. It is now easy to see that G is 3-colorable if and only if G' is.

Connected neighborhood. Let $v \in V(G)$ be such that the graph $N = G|N(v)$ is connected. If N is not 2-colorable, then clearly G is not 3-colorable. Since we can check in polynomial time if a graph is 2-colorable, we may assume that N is bipartite, and, since it is connected, it has a unique 2-coloring; let N_1, N_2 be the color classes. Now, in every 3-coloring of G, the sets N_1 and N_2 are monochromatic. Let G' be obtained from $G \setminus v$ by, for $i = 1, 2$, replacing each N_i by a new vertex n_i adjacent to $(\bigcup_{n \in N_i} N(n)) \setminus \{v\}$. Now G' is P_7-free, and G is 3-colorable if and only if G' is.

The algorithm of 3.5 consists of two main parts. The first part deals with triangle-free P_7-free graphs, and exploits the structural information that follows from these assumptions. The second part deals with $P7$-free graphs that contain a triangle. Here the structure is more complex, but, on the other hand, if such a graph does have a 3-coloring, some of it can easily be seen to be forced, which makes the analysis simpler. In the next two sections we discuss the two parts of the algorithm.

6. 3-coloring P_7-free graphs: the triangle-free case

The goal of this section is to describe the algorithm of 3.5 for the case when the input is a triangle-free graph. Let G be a graph. If P is an induced path with vertices $p_1, \ldots, p_t$ in G, where $p_i p_j \in E(G)$ if and only if $|j - i| = 1$, we write "$p_1 - \ldots - p_t$ is an induced path in G" (or "is a P_t in G"). Similarly, if C is an induced cycle with vertices $c_1, \ldots, c_t$ in G, where $c_i c_j \in E(G)$ if and only if $|j - i| \in \{1, t - 1\}$, we write "$c_1 - \ldots - c_t - c_1$ is an indcued cycle in G" (or "is a C_t in G"). The *complement* G^c of G is the graph with vertex set $V(G)$ and such that $uv \in E(G^c)$ if and only if $uv \notin E(G)$. A *clique* in G is a set of vertices all pairwise adjacent. The *clique number* of G, denoted by $\omega(G)$, is the largest size of a clique in G. G is called *perfect* if $\omega(H) = \chi(H)$ for every induced subgraph H of G. Clearly, for any fixed integer k, a perfect graph G is k-colorable if and only if G does not contain a clique of size $k + 1$, and so testing if a perfect graph is k-colorable can be done in polynomial time, simply by examining all subsets of size $k + 1$. (In fact, a much stronger statement and deeper result is true: one can find the chromatic number of a perfect graph in polynomial time [9], but we do not need this here.)

The following is a well-known structural fact about perfect graphs, the Strong Perfect Graph Theorem:

Theorem 6.1. *[7] A graph G is perfect if and only if no induced subgraph of G or G^c is isomorphic to C_{2n+1} with $n \geq 2$.*

By 6.1, if a triangle-free P_7-free graph is not perfect, then it contains either a C_5 or a C_7. Thus in this case it is easy to check if the input graph is perfect (as with coloring, a much harder theorem is that testing perfection can be done in polynomial time [2], but we do not need it), and, in view of the discussion in this first paragraph of the section, we may assume that it is not, for otherwise we are done.

Let us thus assume that the input graph G is connected, P_7-free, triangle-free, and contains a C_7, say $c_1 - c_2 - c_3 - c_4 - c_5 - c_6 - c_7 - c_1$ (the case of C_5 uses similar ideas); write $C = \{c_1, \dots, c_7\}$. For $i \geq 1$, let X_i be the set of vertices at distance i from C, thus the vertices of X_1 have neighbors in C, the vertices of X_2 are anticomplete to C but have neighbors in X_1, etc. Since G is triangle-free, it is easy to check that for every $x \in X_1$ there exist $p, q, r \in C$, such that $x - p - q - r$ is an induced path in G, and therefore, $X_i = \emptyset$ for $i \geq 4$. For every $x \in X_1$, let $P(x)$ denote some such path $x - p - q - r$. We may assume that G contains no reducible configurations.

Next we show that $X_3 = \emptyset$ as well. Suppose first that there exist $x, y \in X_3$ adjacent to each other. Let $x_2 \in X_2$ be adjacent to x. Since G is triangle-free, $x_2y \notin E(G)$. By the definition of X_2, there is $x_1 \in X_1$ adjacent to x_2. But now combining $y - x - x_2 - x_1$ with $P(x_1)$ we obtain an induced seven-vertex path in G, a contradiction. Thus X_3 is a stable set. Next, let $x_3 \in X_3$. Then $N(x_3) \subseteq X_2$ and $N(x_3)$ is a stable set, since G is triangle-free. Let $x_2 \in N(x_3)$, and let $x_1 \in X_1$ be a neighbor of x_2. If x_1 is complete to $N(x_3)$, then $N(x_3) \subseteq N(x_1)$, and G contains a reducible configuration, a contradiction. So x_1 has a non-neighbor $x_2' \in N(x_3)$. But now combining $x_1 - x_2 - x_3 - x_2'$ with $P(x_1)$ we obtain an induced seven-vertex path in G, a contradiction. This proves that $X_3 = \emptyset$.

Next let us analyze the structure of X_2. First observe that if $G|X_2$ contains an odd cycle D (which therefore has length at least 5), then for every $x_1 \in X_1$ with a neighbor in D, there exist $u, v, w \in V(D)$ such that $x_1 - u - v - w$ is an induced path in G, and combining this path with $P(x_1)$, we obtain an induced seven-vertex path in G, a contradiction. This proves that $G|X_2$ is bipartite. Let F be a connected component of $G|X_2$, and suppose that $|V(F)| > 1$. Let (A, B) be a bipartition of F. We claim that (A, B) is a homogeneous pair of stable sets in G. Suppose not; then we may assume that there exist $x_1 \in X_1$ and $a_1, a_2 \in A$ such that x_1 is adjacent to a_1 and not to a_2. Choosing a_1 and a_2 with this property and subject to that at minimum distance in F, we may assume that there is $b \in B$ adjacent to both a_1 and a_2. Since G is triangle-free, $x_1 - a_1 - b - a_2$ is an induced path in G, and combining it with $P(x_1)$ we obtain an induced seven-vertex path in G, a contradiction. This proves the claim that (A, B) is a homogeneous pair of stable sets in G, and since G contains no reducible configurations, we deduce that $|A| = |B| = 1$. To summarize, we showed that every component of $G|X_2$ consists either of a single vertex, or of two adjacent vertices. Let $d_1, \dots, d_k$ be the vertices of the singleton component, and let $\{a_1, b_1\}, \dots, \{a_m, b_m\}$ be the vertex sets of the components of size two.

At this point we recall the outline of the algorithm we described at the start of Section 5. In our current set-up, C is not a dominating set of G, however, the set of vertices of $V(G) \setminus C$ that are anticomplete to C (namely X_2) is well under control. We proceed by considering all possible colorings of C, and assigning lists of size one to vertices of C, lists of size at most two to vertices of X_1, and lists $\{1, 2, 3\}$ to vertices of X_2. Thus we have so far replaced our original problem by at most 3^7 list-coloring instances, each of which can "almost" be handled by 4.1: only vertices of X_2 may have lists of size three, but we know a lot about the structure of X_2. The rest of the proof consists of removing the "almost" in the previous sentence. We will not be able to explain this completely here, but let us show a few more steps.

To deal with $d_1, \ldots, d_k$, we "guess" (by examining all possibilities) a few (constantly many) vertices with certain properties and their colors, thus creating a set dominating all of $d_1, \ldots, d_k$. This allows us to reduce the sizes of the lists of $d_1, \ldots, d_k$ to two. Please note that while until now we only guessed colors of certain fixed vertices, thus branching into a constant number of list-coloring problems, at this stage we also guess the vertices that we pre-color, which creates polynomially many sub-problems.

Now we deal with $\{a_1, b_1\}, \ldots, \{a_m, b_m\}$. For simplicity, let us assume that every two vertices of X_1 have a common neighbor in C. Justifying this assumption requires additional arguments, but we will skip them here, referring the reader to [4]. Let X be the set of vertices of X_1 that have neighbors in $\{a_1, b_1, \ldots, a_m, b_m\}$. Let $V = \{v_1, \ldots, v_m\}$ be a set of new vertices. We now construct a new bipartite graph H, with bipartition (X, V) in which $x \in X$ is adjacent to $v_i \in V$ if and only if x has a neighbor in $\{a_i, b_i\}$. Suppose that for some $x, y \in X$ and $i, j \in \{1, \ldots, m\}$ we have $xv_i, yv_j \in E(H)$ and $xv_j, yv_i \notin E(H)$. We may assume that, in G, x is adjacent to a_i, and y to a_j. Now choosing $c \in C$ to be a common neighbor of x and y, we obtain that $b_i - a_i - x - c - y - a_j - b_j$ is a seven-vertex path in G, a contradiction. This proves that no such $x, y \in X$ and $i, j \in \{1, \ldots, m\}$ exist, which implies that H has a very special structure. In particular, it is not difficult to see that some vertex $x_0 \in X$ is complete in H to V. In G this means (up to renaming some vertices) that x_0 is complete to $\{a_1, \ldots, a_m\}$. We can now find such a vertex x_0 in polynomial time, and examine all possibilities for the color of x_0 (by branching into sub-problems). Fixing the color of x_0, in turn, allows us to reduce the size of $L(a_i)$ to two for all $i \in \{1, \ldots, m\}$. At this point, in each of the sub-problems we are considering, only $b_1, \ldots, b_m$ have lists of size three. These lists can be dealt with similarly to those of $d_1, \ldots, d_k$ by "guessing" and pre-coloring a few more "important" vertices, thus arriving at a situation where each sub-problem can be handled by 4.1.

7. 3-coloring P_7-free graphs: using triangles

The goal of this section is to discuss the algorithm of 3.5 when the input graph contains a triangle. The main idea here is to take advantage of the simple fact that all three-colorings of a triangle are the same (up to permuting colors), and, moreover, starting with the coloring of a triangle, the colors of certain other vertices are forced.

Let G be a P_7-free graph that contains a triangle. A *tripod* in G is a triple (A_1, A_2, A_3) of pairwise disjoint subsets of $V(G)$ such that

- $A_1 \cup A_2 \cup A_3 = \{v_1, \dots, v_t\}$, where $t \geq 3$
- $v_i \in A_i$ for $i \in \{1, 2, 3\}$
- $v_1v_2v_3$ is a triangle
- Let $i \in \{1, 2, 3\}$, $\{j, k\} = \{1, 2, 3\} \setminus \{i\}$, and $p \in \{4, \dots, t\}$. If $v_p \in A_i$, then v_p has a neighbor in $\{v_1, \dots, v_{p-1}\} \cap A_j$ and a neighbor in $\{v_1, \dots, v_{p-1}\} \cap A_k$.

The first step of the algorithm is to choose a maximal tripod (A_1, A_2, A_3). It is easy to see that in every 3-coloring of G, each of the sets A_1, A_2, A_3 is monochromatic, thus if one of A_1, A_2, A_3 is not a stable set, the algorithm stops and outputs a determination that no 3-coloring exists. Next we observe that, by the maximality of the tripod, every $v \in V(G) \setminus (A_1 \cup A_2 \cup A_3)$ has neighbors in at most one of A_1, A_2, A_3. Let X_i be the set of all $v \in V(G) \setminus (A_1 \cup A_2 \cup A_3)$ with a neighbor in A_i. As in the previous section, classify the remaining vertices of G by their distance from $A_1 \cup A_2 \cup A_3$. Let Y_i be the set of vertices at distance i (so $Y_1 = X_1 \cup X_2 \cup X_3$; Y_2 is anticomplete to $A_1 \cup A_2 \cup A_3$, but every vertex of Y_2 has a neighbor in $X_1 \cup X_2 \cup X_3$; etc). Observe that for every $i \in \{1, 2, 3\}$ and $x \in X_i$, and $j \in \{1, 2, 3\} \setminus \{i\}$, there is $a \in A_i$ and $b \in A_j$ such that $x - a - b$ is an induced path in G. This implies that $Y_k = \emptyset$ for every $i \geq 5$.

Once again we may assume that G has no reducible configuration. Applying arguments similar to those in the previous section (using also the connected neighborhood reducible configuration this time), we further deduce that $Y_4 = \emptyset$.

Next we prove that there exists $S \subseteq Y_1 \cup Y_2 \cup Y_3$, such that $|S| \leq 100$ and "almost" all vertices of $Y_2 \cup Y_3$ have neighbors in S. Ignoring the "almost" qualification, we are now done: since the coloring of $A_1 \cup A_2 \cup A_3$ is unique up to permuting colors, there are only constantly many possible 3-colorings of $Z = A_1 \cup A_2 \cup A_3 \cup S$, and Z is ("almost") a dominating set in G. Thus we can analyze all possible colorings of Z, obtaining a new list-coloring problem from each of them, and each of these new problems can be solved using 4.1. As in the previous section, in order to complete the proof, we guess a few more vertices that need to be added to Z to create a dominating set in G, thus branching into polynomially many sub-problems.

8. 4-coloring

In this section we briefly discuss the ideas of the proof of 3.7, some of which may extend to the more general question of 4-coloring P_6-free graphs. Let G be a P_6-free graph; then G contains no induced cycles of length at least seven. We may assume that $\omega(G) \leq 4$, and that G does not contain C_9^c, for otherwise G is not 4-colorable. This implies that G^c contains no odd cycle of length at least nine. As discussed in Section 6, we may assume that G is not perfect, and so by 6.1, G contains either a C_5 or a C_7^c.

From now on we restrict our attention to the C_5-free case, which is the subject of 3.7. Let C be the vertex set of a C_7^c in G, let X be the set of vertices of G with a neighbor in C, and $Y = V(G) \setminus (C \cup X)$. It is now easy to check that vertices in X come in two "flavors":

- $x \in X$ such that $N(x) \cap C$ contains a triangle; we call such vertices *major*, and
- $x \in X$ for which there exist $a, b, c \in C$ such that $x - a - b - c$ is a path in G; we call such vertices *minor*.

As in the previous two sections, our strategy here is to analyze all possible colorings of C. Having fixed a coloring of C, we update the lists of the vertices of X; let L be the function describing the lists. Then $|L(x)| = 1$ for every major vertex x, and $|L(x)| \in \{2, 3\}$ for every minor vertex x. Let us from now on ignore the existence of vertices in X with lists of size three (in fact, in [6] they are treated similarly to vertices of Y, rather than of X).

A *clique cutset* of a graph G is a clique K of G such that $G \setminus K$ is not connected. Clique cutsets can be readily used in coloring algorithms [19], due to the fact the $G|K$ has a unique coloring (up to permuting colors). Now, if for some component D of $G|Y$, all vertices of X with a neighbor in D are major, then we can treat G in a way similar to a graph with a clique cutset (because G has a cutset with a unique coloring).

Let us next discuss minor vertices. It is an easy fact that if $x \in X$ is minor, and D is a component of $G|Y$, then x is either complete or anticomplete to $V(D)$. An *anticomponent* of a graph is a maximal connected induced subgraph H of G such that H^c is connected (thus H^c is a component of G^c). A more difficult, but very useful observation is that if X' is the set of all minor vertices of X with a neighbor in Y, and A is an anticomponent of $G|X'$, then $L(a) = L(b)$ for all $a, b \in V(A)$.

Let D be a component of $G|Y$, and let $A \neq \emptyset$ be the set of minor vertices of X with a neighbor in D. Then A is complete to D. To illustrate our approach, let us assume that D contains a triangle. Then A is a stable set (since $\omega(G) \leq 4$), and in particular, $G^c|A$ is connected. We may therefore assume that $L(a) = \{1, 2\}$ for all $a \in A$. Moreover, A is monochromatic in every 4-coloring of G.

For every $d \in D$, let

$$L_1(d) = \{2,3,4\} \setminus \bigcup_{x \in N(d),\text{ and } x \text{ is major}} L(x),$$

and

$$L_2(d) = \{1,3,4\} \setminus \bigcup_{x \in N(d),\text{ and } x \text{ is major}} L(x).$$

Now each of the problems (D, L_1) and (D, L_2) can be solved efficiently by 4.3.

Next consider $G' = G \setminus D$. For $i = 1, 2$, if (D, L_i) is not colorable, let $L'(a) = L(a) \setminus \{i\}$ for all $a \in A$. For $v \in V(G') \setminus A$, let $L'(v) = L(v)$. Now (G, L) is colorable if and only if (G', L') is colorable, with the additional constraint that the set A is monochromatic. Applying similar arguments to the remaining components of Y, we reduce the original problem to a problem that can be solved efficiently using 4.2.

9. Open problems

In this section, for the reader's convenience, we repeat the open problems mentioned earlier in the paper.

Question 9.1. *(first mentioned as 4.4) Given a pair (G, L) where G is a P_7-free graph, and $L(v) \subseteq \{1, 2, 3\}$ for every $v \in V(G)$, what is the complexity of determining whether (G, L) is colorable?*

Question 9.2. *(first mentioned as 5.1) Which P_7-free graphs G have a dominating set of size at most $\log |V(G)|$? What about a constant size dominating set?*

Question 9.3. *(first mentioned in Section 3) What is the complexity of 3-coloring P_t-free graphs where $t \geq 8$?*

Question 9.4. *(first mentioned in Section 3) What is the complexity of 4-coloring P_6-free graphs?*

10. Acknowledgments

This paper is based on the author's joint work with Peter Maceli, Juraj Stacho and Mingxian Zhong. The author is grateful to them, and to Irena Penev, for their careful reading of the present manuscript, and for many helpful suggestions. Juraj Stacho's help in surveying previously known results was invaluable. The author also thanks Alex Scott for telling her about the beautiful questions that motivated the research described here.

References

[1] H. Broersma, F. Fomin, P. Golovach, and D. Paulusma, Three complexity results on coloring P_k-free graphs, *European journal of combinatorics* **34** (3). pp. 609-619.

[2] M. Chudnovsky, G. Cornuéjols, X. Liu, P. Seymour, K. Vušković, Recognizing Berge graphs, *Combinatorica* **25** (2005), 143–187.

[3] M. Chudnovsky and P. Maceli, unpublished.

[4] M. Chudnovsky, P. Maceli, and M. Zhong, Three-coloring graphs with no induced seven-vertex path I : the triangle-free case, manuscript.

[5] M. Chudnovsky, P. Maceli, and M. Zhong, Three-coloring graphs with no induced seven-vertex path II : using a triangle, in preparation.

[6] M. Chudnovsky, P. Maceli, J. Stacho and M. Zhong, Four-coloring graphs with no induced six-vertex path and no induced five-wheel, in preparation.

[7] M. Chudnovsky, N.Robertson, P.Seymour, and R.Thomas, The strong perfect graph theorem, *Annals of Math* **164** (2006), 51-229.

[8] K. Edwards, The complexity of colouring problems on dense graphs, *Theoret. Comput. Sci.* **43** (1986), 337-343.

[9] M. Gröstchel, L. Lovász, and A. Schrijver. Geometric Algorithms and Combinatorial Optimization. Springer Verlag, 1988.

[10] 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** (2010), 74–81.

[11] S. Huang, Improved Complexity Results on k-Coloring P_t-Free Graphs, *Proc. MFCS 2013, LNCS*, to appear.

[12] I. Holyer, The NP-completeness of edge coloring, *SIAM J. Comput.* **10** (1981), 718–720.

[13] M. Kamiński and V.V. Lozin, Coloring edges and vertices of graphs without short or long cycles. *Contrib. Discrete. Math.* **2** (2007), 61–66.

[14] R. M. Karp, Reducibility Among Combinatorial Problems, Complexity of Computer Computations, New York: Plenum., 85–103.

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

[16] B. Randerath and I. Schiermeyer, 3-Colorability $\in P$ for P_6-free graphs, *Discrete Appl. Math.* **136** (2004), 299–313.

[17] D. Seinsche, On a property of the class of n-colorable graphs, *J. Comb. Theory B* **16** (1974), 191–193.

[18] L. Stockmeyer, Planar 3-colorability is polynomial complete, *SIGACT News* (1973), 19–25.

[19] R.E. Tarjan, Decomposition by clique separators, *Discrete Math.* **55** (1985), 221–232.

Maria Chudnovsky
E-mail: mchudnov@columbia.edu