LIST-k-COLORING H-FREE GRAPHS FOR ALL $k > 4$

MARIA CHUDNOVSKY*II, SEPEHR HAJEBI §, AND SOPHIE SPIRKL§‖

Abstract. Given an integer $k > 4$ and a graph H, we prove that, assuming P≠NP, the List-k-Coloring Problem restricted to H-free graphs can be solved in polynomial time if and only if either every component of H is a path on at most three vertices, or removing the isolated vertices of H leaves an induced subgraph of the five-vertex path. In fact, the "if" implication holds for all $k \geq 1$.

1. Introduction

Graphs in this paper have finite vertex sets, no loops and no parallel edges. Let $G = (V(G), E(G))$ be a graph. An *induced subgraph* of G is the graph $G \setminus X$ for some $X \subseteq V(G)$, that is, the graph obtained from G by removing the vertices in X. For $X \subseteq V(G)$, we use both X and $G[X]$ to denote the subgraph of G induced on X, which is the same as $G \setminus (V(G) \setminus X)$. We also say G *contains* a graph H if H is isomorphic to an induced subgraph of G; otherwise, we say G is *H-free*.

For an integer $k \geq 1$, we write $[k] = \{1, \ldots, k\}$. Given a graph G, a *proper k-coloring* of G is a map $\varphi : V(G) \to [k]$ such that for every edge $uv \in E(G)$, we have $\varphi(u) \neq \varphi(v)$. A *list-$k$-assignment* for G is a map $L : V(G) \to 2^{[k]}$. Given a list-k-assignment L for G, an *L-coloring* of G is a proper k-coloring φ of G such that $\varphi(v) \in L(v)$ for all $v \in V(G)$. The k-Coloring Problem is to decide, for a graph G, whether G admits a k-coloring, and the List-k-Coloring Problem asks, for a graph G and a list-k-assignment L of G, whether G admits an L-coloring.

The complexity of coloring problems on graphs with forbidden patterns is a subject of great interest at the intersection of structural and algorithmic

---

Date: February 23, 2024.

*Princeton University, Princeton, NJ, USA.

§Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada.

II Supported by NSF-EPSRC Grant DMS-2120644 and by AFOSR grant FA9550-22-1-0083.

‖ We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), [funding reference number RGPIN-2020-03912]. Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), [numéro de référence RGPIN-2020-03912]. This project was funded in part by the Government of Ontario. This research was conducted while Spirkl was an Alfred P. Sloan Fellow.

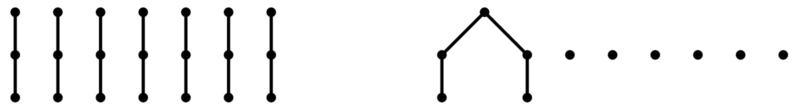

FIGURE 1. The graphs $7P_3$ (left) and $P_5 + 6P_1$ (right).

graph theory (we refer the reader to [3] for a survey). In certain problems such as COLORING and k-LIST-COLORING (see [3] for the definitions), the complexity while restricted to H-free graphs has been characterized. However, the analogous results for k-COLORING and LIST-k-COLORING have remained out of reach.

Specifically, the LIST-2-COLORING PROBLEM can be solved in polynomial time via a reduction to 2SAT [2], whereas the 3-COLORING PROBLEM is famously known to be NP-hard [8]. In fact, the 3-COLORING PROBLEM remains NP-hard in the class of H-free graphs except possibly for some rather restricted choices of H:

Theorem 1.1 (Holyer [5], Kamiński and Lozin [7], Leven and Galil [9])**.** *Let $k \geq 3$ be an integer and let H be a graph with at least one component which is not a path. Then the k-COLORING PROBLEM restricted to H-free graphs is NP-hard.*

The converse to Theorem 1.1 is wide open. In general, for the k-COLORING PROBLEM, no value of $k \geq 3$ is known for which the "easy" choices of H are completely distinguished from the "hard" ones. The situation with the LIST-k-COLORING PROBLEM was also the same until recently, when the last two authors together with Li [4] settled the case $k = 5$. For integers $r, s \geq 1$, we denote by rP_s the graph obtained from the disjoint union of r copies of the s-vertex path, and we write P_s instead of $1Ps$. For graphs H_1, H_2, we write $H_1 + H_2$ to denote the disjoint union of H_1 and H_2 (see Figure 1):

Theorem 1.2 (Hajebi, Li, Spirkl [4])**.** *Suppose that P≠NP. Let H be a graph. Then the LIST-5-COLORING PROBLEM restricted to H-free graphs can be solved in polynomial time if and only if for some integer $r \geq 1$, either rP_3 or $P_5 + rP_1$ contains H.*

In this paper, we extend the conclusion of Theorem 1.2 to all $k > 4$. Like [4], our main contribution is to show that for every $r \geq 1$, excluding (an induced subgraph of) rP_3 results in a polynomial-time solvable case, which also happens to be true for all $k \geq 1$:

Theorem 1.3. *Let $k, r \geq 1$ be fixed integers. Then the LIST-k-COLORING PROBLEM restricted to rP_3-free graphs can be solved in polynomial time.*

As shown in Theorem 1.5 below, Theorem 1.3 along with a number of results from the literature (collected in Theorem 1.4) yields a full dichotomy for the LIST-k-COLORING PROBLEM on H-free graphs for all $k > 4$.

Theorem 1.4. *Let $k \geq 1$ be integers. Then the* List-k-Coloring Problem *restricted to H-free graphs can be solved in polynomial time if*

- $H = P_5 + rP_1$ *for some* $r \geq 1$ (Couturier, Golovach, Kratsch and Paulusma [1]);

and remains NP*-hard if either*

- $H = P_6$ *and* $k > 3$ (Golovach, Paulusma and Song [3])*; or*
- $H = P_4 + P_2$ *and* $k > 4$ (Couturier, Golovach, Kratsch and Paulusma [1]) .

Thus, our main result is the following.

Theorem 1.5. *Let $k > 4$ be an integer and let H be a graph. Then the* List-k-Coloring Problem *restricted to H-free graphs can be solved in polynomial time if for some integer $r \geq 1$, either rP_3 or $P_5 + rP_1$ contains H. Otherwise, the* List-k-Coloring Problem *on H-free graphs is* NP*-hard.*

Proof (assuming Theorem 1.3). If rP_3 contains H for some integer $r \geq 1$, then the result follows from Theorem 1.3, and if $P_5 + rP_1$ contains H for some integer $r \geq 1$, then the result follows from the first bullet of Theorem 1.4. So we may assume that neither holds. Our goal is then to show that the List-k-Coloring Problem on H-free graphs is NP-hard.

By Theorem 1.1, we may assume that each component of H is a path. Since rP_3 does not contain H for any $r \geq 1$, it follows that H contains P_4. This, combined with the assumption that $P_5 + rP_1$ does not contain H for any $r \geq 1$, implies that H contains either P_6 or $P_4 + P_2$. But then the result follows from the second and the third bullet of Theorem 1.4. ∎

It remains to prove Theorem 1.3, which we do in the next section. We also remark that for $k = 3, 4$, a full dichotomy of the List-k-Coloring Problem on H-free graphs remains unknown.

2. The algorithm

We begin by providing some context. The proof of Theorem 1.2 in [4] consists of two steps. The first one, which works for general k, reduces the problem in polynomial time to polynomially many instances in which no three vertices with a common color in their lists induce a path. The second step, confined to the case $k = 5$, renders an intricate analysis within radius-two balls around vertices with list-size more than two, eventually reducing the problem to lists of size at most two (and so to 2SAT). In our proof of Theorem 1.3, the first step remains untouched, but the second step is superseded by Theorem 2.1 below, which has a significantly less technical proof, and holds true for all k.

For integers $k, r \geq 1$, by a (k, r)*-instance* we mean a pair (G, L) where G is an rP_3-free graph and L is a list-k-assignment of G. We say that a (k, r)-instance (G, L) is *admissible* if G admits an L-coloring.

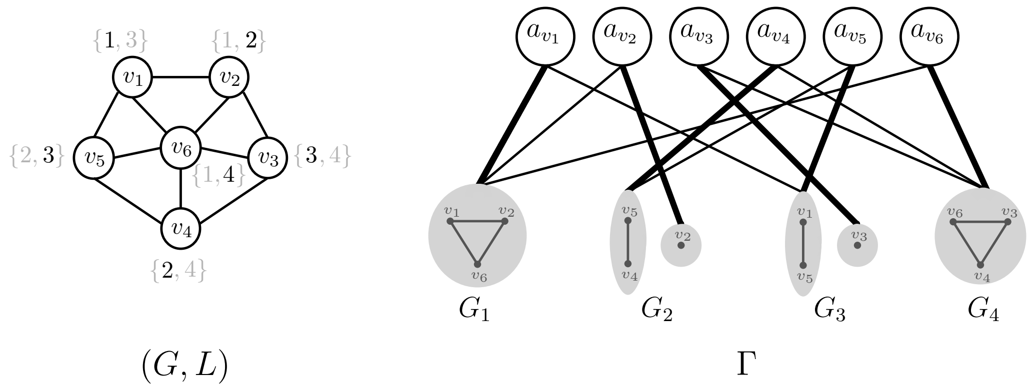

FIGURE 2. A $(4, 2)$-instance (G, L) (left) and the graph Γ (right).

Theorem 2.1. *Let $k, r \geq 1$ be fixed integers. Let (G, L) be a (k, r)-instance where G has $n \geq 1$ vertices. Assume that for every 3-subset $\{x, y, z\}$ of $V(G)$ inducing a path, we have $L(x) \cap L(y) \cap L(z) = \emptyset$. Then it can be decided in time $\mathcal{O}(n^{5/2})$ whether (G, L) is admissible.*

Proof. For every $i \in [k]$, let $G_i = G[\{v \in V(G) : i \in L(v)\}]$. By our assumption, we have that G_i is P_3-free, and so it follows that every component of G_i is a clique of G. Let $\mathcal{C}_i$ be the set of all components of G_i.

We construct a bipartite graph Γ with bipartition (A, B) where the vertices in A and B are labelled as

$$A = \{a_v : v \in V(G)\};$$

$$B = \{b^i_C : i \in [k], C \in \mathcal{C}_i\}$$

such that

$$E(\Gamma) = \bigcup_{i=1}^{k} \bigcup_{C \in \mathcal{C}_i} \{a_v b^i_C : v \in C\}.$$

See Figure 2. It follows that $|V(\Gamma)| = n + |\mathcal{C}_1| + \cdots + |\mathcal{C}_k| \leq (k+1)n$. Moreover, we have:

(1) *(G, L) is admissible if and only if Γ has a matching which covers all vertices in A.*

To see the "only if" implication, assume that G admits an L-coloring φ. Then, for every vertex $v \in V(G)$, we have $v \in V(G_{\varphi(v)})$, and so there exists a unique component $C_v \in \mathcal{C}_{\varphi(v)}$ such that $v \in C_v$. This, along with the definition of Γ, implies that for every $v \in V(G)$, we have $a_v b^{\varphi(v)}_{C_v} \in E(\Gamma)$. Let $M = \{a_v b^{\varphi(v)}_{C_v} : v \in V(G)\}$; then we have $|M| = |A|$. We claim that M is a matching in Γ. Clearly, no two edges in M share an end in A. Also no two edges in M share an end in B; for otherwise there are distinct vertices $u, v \in V(G)$ as well as $i \in [k]$ and $C \in \mathcal{C}_i$ such that $\varphi(u) = \varphi(v) = i$ and $C_u = C_v = C$. But this violates the fact that φ is a proper coloring and C

is a clique of G. The claim follows, and so does the "only if" implication of (1).

For the "if" implication, assume that there exists a matching $M \subseteq E(\Gamma)$ in Γ which covers A. From the definition of Γ, it follows that there exists a map $\varphi : V(G) \to [k]$, as well as a component $C_v \in \mathcal{C}_{\varphi(v)}$ for each $v \in V(G)$, such that $M = \{a_v b^{\varphi(v)}_{C_v} : v \in V(G)\}$. We claim that φ is an L-coloring of G. Assume that $u, v \in V(G)$ are distinct and there exists $i \in [k]$ such that $\varphi(u) = \varphi(v) = i$. Then we have $C_u, C_v \in \mathcal{C}_i$. Also, since M is a matching in Γ, it follows that $b^i_{C_u}$ and $b^i_{C_v}$ are distinct, which in turn implies that C_u and C_v are distinct components of G_i. This, combined with the fact that G_i is an induced subgraph of G, implies that $u \in C_u$ and $v \in C_v$ are not adjacent in G. Thus, φ is a proper k-coloring of G. Moreover, for every vertex $v \in V(G)$, since $a_v b^{\varphi(v)}_{C_v} \in M \subseteq E(\Gamma)$, it follows from the definition of Γ that $v \in C_v \in \mathcal{C}_{\varphi(v)}$, and so $v \in V(G_{\varphi(v)})$. Therefore, we have that $\varphi(v) \in L(v)$ for every $v \in V(G)$. This proves (1).

By (1), Theorem 2.1 is immediate from a well-known result of Hopcroft and Karp [6] that the cardinality of a maximum matching in an n-vertex bipartite graph can be computed in time $\mathcal{O}(n^{5/2})$. ■

Let us turn to the "first step" as discussed at the beginning of this section. For integers $k, r \geq 1$ and a (k, r)-instance (G, L), by a (G, L)-*profile* we mean a set $\mathcal{I}$ of pairs (G', L') where G' is an induced subgraph of G and L' is a list-k-assignment for G' such that $L'(v) \subseteq L(v)$ for all $v \in V(G')$. In particular, if $\mathcal{I}$ is a (G, L)-profile for a (k, r)-instance (G, L), then every pair $(G', L') \in \mathcal{I}$ is a (k, r)-instance, as well.

Theorem 2.2 (Hajebi, Li and Spirkl, see Theorem 5.1 in [4])**.** *Let $k, r \geq 1$ be fixed integers. Then there exists an integer $p = p(k, r) \geq 1$ such that for every (k, r)-instance (G, L) with $|V(G)| = n \geq 1$, there is a (G, L)-profile $\mathcal{I}$ with the following specifications.*

- *$|\mathcal{I}| \leq \mathcal{O}(n^p)$ and $\mathcal{I}$ can be computed from (G, L) in time $\mathcal{O}(n^p)$.*
- *For every $(G', L') \in \mathcal{I}$ and every 3-subset $\{x, y, z\}$ of $V(G')$ inducing a path, we have $L'(x) \cap L'(y) \cap L'(z) = \emptyset$.*
- *(G, L) is admissible if and only if some $(G', L') \in \mathcal{I}$ is admissible.*

Finally, we merge Theorems 2.1 and 2.2 to deduce Theorem 1.3, restated as follows:

Theorem 2.3. *For all fixed integers $k, r \geq 1$, there exists an algorithm which, given a (k, r)-instance (G, L), decides in polynomial time whether (G, L) is admissible.*

Proof. The algorithm is as follows. Given a (k, r)-instance (G, L):

1. Compute the (G, L)-profile $\mathcal{I}$ as in Theorem 2.2.
2. For each (k, r)-instance $(G', L') \in \mathcal{I}$, decide whether (G', L') is admissible.

3. If there exists a (k, r)-instance $(G', L') \in \mathcal{I}$ which is admissible, then return "(G, L) is admissible." Otherwise, return "(G, L) is not admissible."

The correctness of the above algorithm is immediate from the third bullet of Theorem 2.2. Also, the first two bullets of Theorem 2.2 combined with Theorem 2.1 imply that the above algorithm runs in polynomial time. This completes the proof of Theorem 2.3. ■

References

[1] J.-F. Couturier, P. A. Golovach, D. Kratsch, and D. Paulusma. List coloring in the absence of a linear forest. *Algorithmica*, 71(1):21–35, 2015.

[2] K. Edwards. The complexity of colouring problems on dense graphs. *Theoret. Comput. Sci.*, 43(2-3):337–343, 1986.

[3] P. A. Golovach, M. Johnson, D. Paulusma, and J. Song. A survey on the computational complexity of coloring graphs with forbidden subgraphs. *J. Graph Theory*, 84(4):331–363, 2017.

[4] S. Hajebi, Y. Li, and S. Spirkl. Complexity dichotomy for list-5-coloring with a forbidden induced subgraph. *SIAM J. Discrete Math.*, 36(3):2004–2027, 2022.

[5] I. Holyer. The NP-completeness of edge-coloring. *SIAM J. Comput.*, 10(4):718–720, 1981.

[6] J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2:225–231, 1973.

[7] M. Kamiński and V. Lozin. Coloring edges and vertices of graphs without short or long cycles. *Contrib. Discrete Math.*, 2(1):61–66, 2007.

[8] R. Karp. Reducibility among combinatorial problems (1972). In *Ideas that created the future—classic papers of computer science*, pages 349–356. MIT Press, Cambridge, MA, [2021] ©2021.

[9] D. Leven and Z. Galil. NP completeness of finding the chromatic index of regular graphs. *J. Algorithms*, 4(1):35–44, 1983.