List Locally Surjective Homomorphisms in Hereditary Graph Classes

Pavel Dvořák
Tata Institute of Fundamental Research, Mumbai, India
Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

Tomáš Masařík
Institute of Informatics, Faculty of Mathematics, Informatics and Mechanics,
University of Warsaw, Poland

Jana Novotná
Institute of Informatics, Faculty of Mathematics, Informatics and Mechanics,
University of Warsaw, Poland
Department of Applied Mathematics, Faculty of Mathematics and Physics,
Charles University, Prague, Czech Republic

Monika Krawczyk
Faculty of Mathematics and Information Science, Warsaw University of Technology, Poland

Paweł Rzążewski
Faculty of Mathematics and Information Science, Warsaw University of Technology, Poland
Institute of Informatics, Faculty of Mathematics, Informatics and Mechanics,
University of Warsaw, Poland

Aneta Żuk
Faculty of Mathematics and Information Science, Warsaw University of Technology, Poland

Abstract

A *locally surjective homomorphism* from a graph G to a graph H is an edge-preserving mapping from $V(G)$ to $V(H)$ that is surjective in the neighborhood of each vertex in G. In the *list locally surjective homomorphism* problem, denoted by LLSHom(H), the graph H is fixed and the instance consists of a graph G whose every vertex is equipped with a subset of $V(H)$, called list. We ask for the existence of a locally surjective homomorphism from G to H, where every vertex of G is mapped to a vertex from its list. In this paper, we study the complexity of the LLSHom(H) problem in F-free graphs, i.e., graphs that exclude a fixed graph F as an induced subgraph. We aim to understand for which pairs (H, F) the problem can be solved in subexponential time.

We show that for all graphs H, for which the problem is NP-hard in general graphs, it cannot be solved in subexponential time in F-free graphs for F being a bounded-degree forest, unless the ETH fails. The initial study reveals that a natural subfamily of bounded-degree forests F, that might lead to some tractability results, is the family $\mathcal{S}$ consisting of forests whose every component has at most three leaves. In this case, we exhibit the following dichotomy theorem: besides the cases that are polynomial-time solvable in general graphs, the graphs $H \in \{P_3, C_4\}$ are the only connected ones that allow for a subexponential-time algorithm in F-free graphs for every $F \in \mathcal{S}$ (unless the ETH fails).

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Mathematics of computing → Graph algorithms

Keywords and phrases Homomorphism, Hereditary graphs, Subexponential-time algorithms

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2022.30

Related Version *Full Version*: `https://arxiv.org/abs/2202.12438` [9]

Funding *Pavel Dvořák*: Partially supported by EPSRC New Investigator Award EP/V010611/1 and by Czech Science Foundation GAČR grant #22-14872O.

33rd International Symposium on Algorithms and Computation (ISAAC 2022).
Editors: Sang Won Bae and Heejin Park; Article No. 30; pp. 30:1–30:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Tomáš Masařík: Received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme Grant Agreement 948057.
Jana Novotná: Supported by SVV-2020–260578 and GAUK 384321 of Charles University.
Paweł Rzążewski: Supported by Polish National Science Centre grant no. 2018/31/D/ST6/00062.

1 Introduction

Graph coloring is arguably one of the best-studied problems in algorithmic graph theory. It is well-known that k-Coloring is polynomial-time solvable for $k \leq 2$ and NP-hard for every $k \geq 3$ [27]. Furthermore, assuming the Exponential-Time Hypothesis (ETH) [25, 26], the hard cases do not even admit algorithms working in subexponential time.

Coloring F-free graphs. A very natural direction of research is to investigate what restrictions put on the family of input graphs allow us to solve the problem more efficiently than in general graphs. In recent years, a very active topic has been to study the complexity of k-Coloring and related problems in graphs defined by one or more forbidden induced subgraphs. For a family $\mathcal{F}$ of graphs, we say that a graph G is $\mathcal{F}$-free if G does not contain any graph from $\mathcal{F}$ as an induced subgraph. If $\mathcal{F}$ consists of a single graph F, then we say F-free instead of $\{F\}$-free. Note that the class of $\mathcal{F}$-free graphs is *hereditary* i.e., closed under vertex deletion. On the other hand, every hereditary class of graphs can be equivalently defined as $\mathcal{F}$-free graphs for some unique minimal (possibly infinite) family $\mathcal{F}$ of graphs.

It is well-known that for every $k \geq 3$, the k-Coloring problem is NP-hard in F-free graphs, unless F is a *linear forest*, i.e., every connected component of F is a path. Indeed, for every constant g, k-Coloring is NP-hard in graphs of girth (i.e., the length of a shortest cycle) at least g [10]. Setting $g = |V(F)| + 1$, we immediately obtain hardness for every F that is not a forest. On the other hand, k-Coloring is NP-hard in line graphs, which are claw-free [23, 29]. The only forests that are claw-free are linear forests.

The complexity of k-Coloring in P_t-free graphs, where P_t is the path with t vertices, has recently attracted a lot of attention. For $t = 5$, the problem is polynomial-time solvable for every constant k [22]. If $k \geq 5$, then the problem is NP-hard already in P_6-free graphs [24]. The case $k = 4$ is also fully understood: it is polynomial-time solvable for $t \leq 6$ [38] and NP-hard for $t \geq 7$ [24]. The case of $k = 3$ is much more elusive. We know a polynomial-time algorithm for P_7-free graphs [1]. However, for $t \geq 8$, we know neither polynomial-time algorithm nor any hardness result. Some positive results are also known for the case that F is a disconnected linear forest [28, 6, 18].

Let us point out that almost all mentioned algorithmic results also hold for the more general *list* variant of the problem, where each vertex is given a list of admissible colors. The notable exception is List 4-Coloring, which is NP-hard already in P_6-free graphs [19]. Furthermore, all hardness results also imply the nonexistence of subexponential-time algorithms (assuming the ETH).

Some more general positive results can be obtained if we relax our notion of tractability. As observed by Groenland et al. [20], List 3-Coloring can be solved in subexponential time in P_t-free graphs, for every fixed t. This was recently improved by Pilipczuk, Pilipczuk, and Rzążewski [37] who showed a quasipolynomial-time algorithm for this problem. Note that this is strong evidence that the problem is not NP-hard.

Graph homomorphisms. Graph colorings can be seen as a special case of *graph homomorphisms*. A homomorphism from a graph G to a graph H (with possible loops) is a function $h : V(G) \to V(H)$, such that for every $uv \in E(G)$ it holds that $h(u)h(v) \in E(H)$. Note that homomorphisms to K_k are precisely proper k-colorings. By the celebrated result of Hell and Nešetřil [21], determining whether an input graph G admits a homomorphism to a fixed graph H is polynomial-time solvable if H is bipartite or has a vertex with a loop, and NP-hard otherwise. A list variant of the graph homomorphism problem, denoted by LHom(H), has also been considered. It turns out that the problem can be solved in polynomial time if H is a so-called *bi-arc graph*, and otherwise, the problem is NP-hard [12, 13, 14].

The complexity of variants of the graph homomorphism problem in hereditary graph classes was also studied. For example, Chudnovsky et al. [5] showed that LHom(C_k) for $k \in \{5, 7\} \cup [9, \infty)$ is polynomial-time solvable in P_9-free graphs. On the negative side, they showed that for every $k \geq 5$ the problem is NP-hard and cannot be solved in subexponential time (assuming the ETH) in F-free graphs, unless every component of $F \in \mathcal{S}$, where $\mathcal{S}$ consists of graphs whose every connected component is a path or a tree with three leaves (called a *subdivided claw*). This negative result was later extended by Piecyk and Rzążewski [36] who showed that if H is not a bi-arc graph (i.e., LHom(H) is NP-hard in general graphs), then LHom(H) is NP-hard and cannot be solved in subexponential time (assuming the ETH) in F-free graphs, unless $F \in \mathcal{S}$.

The case of forbidden path or subdivided claw was later investigated by Okrasa and Rzążewski [35]. They defined a class of *predacious graphs* and showed that if H is not predacious, then for every H, the LHom(H) problem can be solved in quasipolynomial time in P_t-free graphs (for every t). Otherwise, for every H, there exists t for which LHom(H) cannot be solved in subexponential time in P_t-free graphs unless the ETH fails. They also provided some partial results for the case of forbidden subdivided claws.

The complexity of variants of the graph homomorphism problem in other hereditary graph classes has also been considered [7, 15, 34].

Locally surjective graph homomorphisms. Graph homomorphisms are a very robust notion, which can be easily extended by putting some additional restrictions on the solution. In this paper, we focus on one such variant called *locally surjective homomorphisms*. A homomorphism h from G to H is locally surjective if it is surjective in the neighborhood of each vertex of G. In other words, if $h(v) = a \in V(H)$, then for every neighbor b of a in H (including a, if it has a loop) there is a neighbor v' of v in G, such that $h(v') = b$. The study of locally surjective homomorphisms originates in social sciences, where they can be used to model some social roles (the problem is called *role assignment* [11]). The problem of determining whether an input graph admits a locally surjective homomorphism to a fixed graph H is denoted by LSHom(H). Fiala and Paulusma [17] provided the full complexity dichotomy for LSHom(H). For simplicity, let us consider only connected graphs H, and let K_1° be the one-vertex graph with a loop. We denote $\mathcal{H}_{\text{poly}} := \{K_1, K_1^\circ, K_2\}$. Fiala and Paulusma [17] showed that LSHom(H) is polynomial-time-solvable if $H \in \mathcal{H}_{\text{poly}}$, and otherwise it is NP-hard. Again, the hardness reduction excluded also subexponential time algorithms under the ETH.

Let us point out that LSHom(P_3) is closely related to the well-known hypergraph 2-coloring problem [30] (or, equivalently, Positive NAE SAT). In this problem, we ask whether the input hypergraph admits a 2-coloring of its vertices which makes no edge monochromatic. Consider a hypergraph $\mathbf{H}$ with vertices $\mathcal{V}$ and hyperedges $\mathcal{E}$, and let G be its *incidence graph*, i.e., the bipartite graph with vertex set $\mathcal{V} \cup \mathcal{E}$, where $v \in \mathcal{V}$ is adjacent to $e \in \mathcal{E}$ if and only if $v \in e$. Note that proper 2-colorings of $\mathbf{H}$ are precisely locally surjective homomorphisms of G to P_3 with consecutive vertices $1, 2, 3$, where $\mathcal{V}$ is mapped to $\{1, 3\}$, and $\mathcal{E}$ is mapped to $\{2\}$. As shown by Camby and Schaudt [3], 2-coloring of hypergraphs with P_7-free incidence graph is polynomial-time solvable.

The structural and computational aspects of locally surjective homomorphisms were studied by several authors [4, 16, 2]. However, up to the best of our knowledge, no systematic study of LSHom(H) in hereditary graph classes has been conducted.

Our contribution. In this paper, we consider the complexity of the *list* variant of LSHom(H), called LLSHom(H). First, we observe that if $H \in \mathcal{H}_{\text{poly}}$ (recall, these are the easy cases of LSHom(H)), then also LLSHom(H) can be solved in polynomial time in general graphs.

Then we focus on the complexity of the problem in F-free graphs. In particular, we are interested in determining the pairs (H, F), for which the problem can be solved in subexponential time. Similarly to the case of LHom(H), we split into two cases, depending whether $F \in \mathcal{S}$.

In the first case, we identify two more positive cases: we show that if $H \in \{P_3, C_4\}$, then the problem admits a subexponential-time algorithm for every $F \in \mathcal{S}$. The algorithm itself uses a win-win strategy: we combine branching on a high-degree vertex with a separator theorem that can be used if the maximum degree is bounded. A similar approach was used for various other problems [20, 31], however, the specifics of our problem require a slightly more complicated approach.

We also show that the above cases are the only positive ones for general $F \in \mathcal{S}$, which provides the following dichotomy theorem.

► **Theorem 1.** *Let $H \notin \mathcal{H}_{\text{poly}}$ be a fixed connected graph.*

1. *If $H \in \{P_3, C_4\}$, then for every $F \in \mathcal{S}$, the LLSHom(H) problem can be solved in time $2^{\mathcal{O}((n \log n)^{2/3})}$ in n-vertex F-free graphs.*
2. *Otherwise there is t, such that the LLSHom(H) problem cannot be solved in subexponential time in P_t-free graphs, unless the ETH fails.*

Further, we turn our attention to other forbidden graphs F. We show that whenever the problem is NP-hard for general graphs, i.e., for every $H \notin \mathcal{H}_{\text{poly}}$, then for every $g > 0$ there exists $d = d(H)$, such that LSHom(H) is NP-hard in graphs of degree at most d and girth at least g. This implies the following lower bound.[1]

► **Theorem 2 (♠).** *For every connected $H \notin \mathcal{H}_{\text{poly}}$, there exists $d \in \mathbb{N}$, such that the following holds. For every graph F that is not a forest of maximum degree at most d, the LLSHom(H) problem cannot be solved in time $2^{o(n)}$ in n-vertex F-free graphs of maximum degree d, unless the ETH fails.*

We conclude the paper by discussing the possibilities of improving our theorems in order to fully classify the complexity of LSHom(H) in F-free graphs.

2 Preliminaries

For a graph G and $v \in V(G)$, by $N_G(v)$ we denote the set of neighbors of v in G. For a set $X \subseteq V(G)$, by $N_G[X]$ we denote $X \cup \bigcup_{v \in X} N_G(v)$. If G is clear from the context, then we omit the subscripts. By $G[X]$, where $X \subseteq V(G)$, we denote the subgraph of G induced by the set X.

[1] Proofs of statements marked with (♠) are presented in the full version [9].

For graphs G and H, by $G \times H$ we denote their *direct product* (sometimes called categorical product or Kronecker product), i.e., the graph

$$V(G \times H) = V(G) \times V(H),$$
$$E(G \times H) = \big\{\{(u_1, v_1), (u_2, v_2)\} \mid u_1u_2 \in E(G) \wedge v_1v_2 \in E(H)\big\}.$$

For $t, a, b, c \geq 1$, by P_t we denote the t-vertex path, and by $S_{a,b,c}$ we denote the three-leaf tree with leaves at distance a, b, and c, respectively, from the unique vertex of degree 3, which we call *central*. Every such $S_{a,b,c}$ is called a *subdivided claw*. Recall that by $\mathcal{S}$, we denote the family of graphs whose every connected component is either a path or a subdivided claw.

Let h be a homomorphism from G to H. We say that a vertex $v \in V(G)$ is *happy* (in h) if $h(N_G(v)) = N_H(h(v))$. In other words, for every neighbor y of $h(v)$, some neighbor of v is colored y. We say that a homomorphism h is *locally surjective* if every vertex is happy in h. If h is a locally surjective homomorphism from G to H, then we denote it by $h : G \xrightarrow{s} H$.

For a fixed graph H (with possible loops) we consider the LLSHom(H) problem, whose instance is (G, L) where G is a graph and $L : V(G) \to 2^{V(H)}$ is a list function. We ask whether there exists a homomorphism $h : G \xrightarrow{s} H$, such that for every $v \in V(G)$ it holds that $h(v) \in L(v)$. If h is such a homomorphism, then we denote it by $h : (G, L) \xrightarrow{s} H$.

Observe that if G or H is disconnected, then each component of G must be mapped to some component of H. Thus, the problem can be easily reduced to the case that both G and H are connected. We will assume this from now on.

Recall that the non-list variant of our problem, i.e., LSHom(H), is polynomial time-solvable if $H \in \mathcal{H}_{\text{poly}} := \{K_1, K_2, K_1^\circ\}$ (where K_1° denotes the one-vertex graph with a loop), and NP-hard otherwise [17]. Let us point out that exactly the same dichotomy holds for LLSHom(H).

▶ **Corollary 3** (♠). *If $H \in \mathcal{H}_{\text{poly}}$, then LLSHom($H$) is polynomial-time solvable, and otherwise it is NP-hard.*

2.1 Associated Bipartite Graphs and Associated Instances

Now let us show that in order to identify the hard cases of LLSHom(H) it is sufficient to consider the case that H is bipartite. A similar approach was used to solve LHom(H) [14, 32], but to the best of our knowledge, we are the first to observe that it also works for LLSHom(H).

Let H be a connected bipartite graph with bipartition classes X, Y, and consider an instance (G, L) of LLSHom(H), where G is connected. Note that if G is not bipartite, then (G, L) is clearly a no-instance. Thus, assume that G is bipartite with the bipartition classes A, B. We observe that in every homomorphism $h : G \to H$, either all vertices of A are mapped to X, and all vertices of B are mapped to Y, or all vertices of A are mapped to Y, and all vertices of B are mapped to X. Thus in order to solve (G, L), we can consider these two cases separately. More specifically, we need to solve two instances (G, L_1) and (G, L_2) of LLSHom(H), where

$$L_1(v) = \begin{cases} L(v) \cap X & \text{if } v \in A, \\ L(v) \cap Y & \text{if } v \in B, \end{cases} \qquad L_2(v) = \begin{cases} L(v) \cap Y & \text{if } v \in A, \\ L(v) \cap X & \text{if } v \in B. \end{cases}$$

This motivates the following definition, see also [32].

▶ **Definition 4.** *Let H be a connected bipartite graph with bipartition classes X, Y. We say that an instance (G, L) of LLSHom(H) is consistent, if*

1. *G is connected bipartite with the bipartition classes A, B,*
2. *$L(A) \subseteq X$ and $L(B) \subseteq Y$.*

For a graph $H = (V, E)$, by $H^* := H \times K_2$ we denote its *associated bipartite graph*. In other words, the vertex set of H^* is $\{v', v'' \ : \ v \in V\}$ and the edge set is $\{u'v'' \ : \ uv \in E\}$. We also define $V' := \{v' \ : \ v \in V\}$ and $V'' := \{v'' \ : \ v \in V\}$, i.e., V', V'' are the bipartition classes of H^*.

Note that if H is connected and nonbipartite, then H^* is connected. If H is bipartite, then H^* consists of two disjoint copies of H.

▶ **Lemma 5.** *Let H be a fixed connected nonbipartite graph. Let (G, L') be a consistent instance of* LLSHom(H^*)*. For each $v \in V(G)$, define $L(v) := \{x : \{x', x''\} \cap L'(v) \neq \emptyset\}$. Then (G, L') is a* `yes`*-instance of* LLSHom(H^*) *if and only if (G, L) is a* `yes`*-instance of* LLSHom(H)*.*

Proof. First consider $h^* : (G, L') \xrightarrow{s} H^*$. Define $h : V(G) \to V(H)$ as follows: $h(v) = x$ if and only if $h^*(v) \in \{x', x''\}$. Clearly, h is a homomorphism from G to H, and it respects lists L.

Let us show that h is locally surjective. Consider $v \in V(G)$ such that $h(v) = x$ and some $y \in N_H(x)$. Since $h(v) = x$, we know that $h^*(v) \in \{x', x''\}$ (the actual value depends on the bipartition class where v belongs). By symmetry, assume that $h^*(v) = x'$. Since h^* is locally surjective and $x'y'' \in E(H^*)$, there is $u \in N_G(v)$, such that $h^*(u) = y''$. Then, $h(u) = y$.

Now, consider $h : (G, L) \xrightarrow{s} H$. Let the bipartition classes of G be A, B, such that $L'(A) \subseteq V'$ and $L'(B) \subseteq V''$ (this holds since (G, L') is consistent). We define $h^* : V(G) \to V(H^*)$ as follows. Consider $v \in V(G)$ and let $h(v) = x$. If $v \in A$, then $h^*(v) = x'$ and if $v \in B$, then $h^*(v) = x''$. Again, it is straightforward to verify that h^* is a homomorphism from G to H^*, and it respects lists L', since (G, L') is consistent.

Now, let us argue that h^* is locally surjective. By symmetry, consider $v \in A$, such that $h(v) = x$. Then, $h^*(v) = x'$. Let $y'' \in N_H(x')$. Since h is locally surjective, there is $u \in N_G(v) \subseteq B$, such that $h(u) = y$. Thus, $h^*(u) = y''$. ◀

▶ **Corollary 6 (♠).** *Let $\mathcal{G}$ be a class of graphs and let H be a fixed connected nonbipartite graph. Suppose there is an algorithm A that solves every bipartite instance (G, L) of* LLSHom(H)*, such that $G \in \mathcal{G}$, in time $f(|V(G)|)$. Then there is an algorithm that solves every instance (G, L') of* LLSHom(H^*)*, where $G \in \mathcal{G}$, in time $f(|V(G)|) \cdot |V(G)|^{\mathcal{O}(1)}$.*

Note that Corollary 6 immediately implies the following.

▶ **Corollary 7.** *Let H be a fixed connected nonbipartite graph and let $\mathcal{G}$ be a class of graphs. If* LLSHom(H^*) *cannot be solved in time $2^{o(n)}$ for n-vertex instances in $\mathcal{G}$, then* LLSHom(H) *cannot be solved in time $2^{o(n)}$ for n-vertex instances in $\mathcal{G}$.*

3 Algorithm for F-free Graphs for $F \in \mathcal{S}$

An important tool used in our algorithm is the following structural result about $\{S_{t,t,t}, K_3\}$-free graphs (note that it only appears in the full version of [35]).

▶ **Theorem 8** (Okrasa, Rzążewski [33])**.** *Let $t \geq 2$ be an integer. Given an n-vertex $(K_3, S_{t,t,t})$-free graph G with maximum degree Δ, in time $2^{\mathcal{O}(t \cdot \Delta)} \cdot n$ we can find a tree decomposition of G with width at most $56t\Delta$.*

Equipped by this, we are ready to prove the following algorithmic result.

▶ **Theorem 9.** *Let $a, b, c \geq 1$ be fixed integers. The* LLSHom(P_3) *in n-vertex $S_{a,b,c}$-free graphs can be solved in time $2^{\mathcal{O}((n \log n)^{2/3})}$.*

Proof. Denote the consecutive vertices of P_3 by $1, 2, 3$. Let (G, L) be an instance of LLSHom(P_3), where G has n vertices and is $S_{a,b,c}$-free. Let $t = \max(2, a, b, c)$ and note that G is $S_{t,t,t}$-free. Furthermore, if G is not bipartite, then (G, L) is clearly a no-instance. Thus, we can assume that G is bipartite (and, in particular, triangle-free).

Furthermore, recall that we can safely assume that the instance (G, L) is consistent. Let X and Y denote the bipartition classes of G, such that $L(X) \subseteq \{1, 3\}$ and $L(Y) = \{2\}$. Note that if $|Y| = 0$ or $|X| \leq 1$, then we are clearly dealing with a no-instance. Thus from now on, let us assume otherwise. In particular, it means that every vertex from X is happy in every list homomorphism from G to P_3. Consequently, our task boils down to choosing colors for vertices of X to make each vertex from Y happy.

Actually, we will design a recursive algorithm that solves a slightly more general problem, where we are additionally given a function $\sigma : Y \to 2^{\{1,3\}}$. We are looking for a list homomorphism $h : (G, L) \to P_3$, such that for every $y \in Y$ it holds that $\sigma(y) \subseteq h(N_G(y))$. Initially, we have $\sigma(y) = \{1, 3\}$ for every $y \in Y$. Thus, the returned homomorphism is indeed locally surjective. During the course of the algorithm, we will modify the sets σ to keep track of the colors seen by vertices in Y in the part of the graph that was removed.

Each recursive call starts with a preprocessing phase. First, we exhaustively apply the following steps. If there is some $x \in X$ with $L(x) = \emptyset$, then we immediately terminate the recursive call and report a no-instance. If there is some $y \in Y$ with $\sigma(y) = \emptyset$, then we can safely remove y from the graph. If there is some $x \in X$ with $|L(x)| = 1$, then we remove the element of $L(x)$ from the sets σ of all neighbors of x, and remove x from the graph.

If none of the above steps can be applied and G has an isolated vertex $y \in Y$ (note that $\sigma(y) \neq \emptyset$), then we terminate and report a no-instance. The preprocessing phase can clearly be performed in polynomial time.

Finally, if the graph obtained is disconnected, then we apply the following reasoning to every connected component independently. Let us still denote the instance by (G, L, σ), and assume that G is connected.

We consider two cases. First, suppose that there is $x \in X$ with $\deg x > (n \log n)^{1/3}$. We branch on choosing the color for x, i.e., we perform two recursive calls of the algorithm, in one branch setting $L(x) = \{1\}$, and in the other $L(x) = \{3\}$. Note that at least $\deg x/3 \geq (n \log n)^{1/3}/3$ neighbors of x have the same set σ. Consequently, in at least one branch, the sets σ will be reduced for at least $(n \log n)^{1/3}/3$ vertices during the preprocessing phase. In the other branch, we are guaranteed to have a little progress, too: the vertex x will be removed from the graph. Let us define the measure μ of the instance as $\mu := \sum_{x \in X} |L(x)| + \sum_{y \in Y} |\sigma(y)|$. Clearly $n \leq \mu \leq 2n$. Thus the complexity of this step is given by the following recursive inequality:

$$F(\mu) \leq F(\mu - 2) + F(\mu - (n \log n)^{1/3}/3) = \mu^{\mathcal{O}(\mu/(n \log n)^{1/3})} = 2^{\mathcal{O}((n \log n)^{2/3})}.$$

So now let us assume that for each $x \in X$ it holds that $\deg x < (n \log n)^{1/3}$. Let Y' be the set of vertices $y \in Y$ satisfying $\deg y \geq (n \log n)^{2/3}$. Observe that $|E(G)| \leq |X| \cdot (n \log n)^{1/3} \leq n^{4/3} \log^{1/3} n$. Consequently, $|Y'| \leq |E(G)|/(n \log n)^{2/3} \leq n^{2/3}$.

Consider the graph $G' := G - Y'$. As it is an induced subgraph of G, it is $(K_3, S_{t,t,t})$-free. Furthermore, the maximum degree of G' is at most $(n \log n)^{2/3}$. Consequently by Theorem 8, in time $2^{\mathcal{O}((n \log n)^{2/3})}$ we can find a tree decomposition of G' with width $\mathcal{O}((n \log n)^{2/3})$. Let us modify this tree decomposition by adding the set Y' to every bag – this way we obtain a tree decomposition of G with width $\mathcal{O}((n \log n)^{2/3} + n^{2/3}) = \mathcal{O}((n \log n)^{2/3})$.

Using fairly standard dynamic programming on a tree decomposition, we can solve our auxiliary problem on graphs given with a tree decomposition of width w in time $2^{\mathcal{O}(w)} \cdot n^{\mathcal{O}(1)}$. Indeed, the state of the dynamic programming is the coloring of the vertices from X in the current bag and the colors seen by the vertices from Y in the subgraph induced by the subtree rooted at the current bag (these colors are reflected in sets σ). Thus, the total number of states to consider is at most 3^w (two possibilities for a vertex from X and at most three for a vertex from Y).

Consequently, in the second case we obtain the running time $2^{\mathcal{O}((n\log n)^{2/3})} + 2^{\mathcal{O}((n\log n)^{2/3})} = 2^{\mathcal{O}((n\log n)^{2/3})}$.

Summing up, the overall complexity of the algorithm is $2^{\mathcal{O}((n\log n)^{2/3})}$. This completes the proof. ◀

Note that every P_t-free graph is also, e.g., $S_{t,1,1,}$-free, so Theorem 9 can also be applied to P_t-free graphs. Now, let us show a slight generalization of Theorem 9 to the case that we exclude a forest of paths and subdivided claws.

▶ **Theorem 10.** *For every $F \in \mathcal{S}$, the LLSHOM(P_3) problem in n-vertex F-free graphs can be solved in time $2^{\mathcal{O}((n\log n)^{2/3})}$.*

Sketch of proof. Let $F = F_1 + F_2 + \ldots + F_p$ for some $p \geq 1$, where each F_i is a subdivided claw.

We begin similarly to the algorithm from Theorem 9. Again, we are solving an auxiliary problem with instance (G, L, σ). First, we check if the instance graph is bipartite, and otherwise, we reject it. Let the bipartition classes of G be X and Y, and let $L(X) \subseteq \{1, 3\}$ and $L(Y) = \{2\}$. Then, we perform the preprocessing phase and the branching phase; note that in these phases, we do not assume anything about the forbidden induced graph. The recursion tree has $2^{\mathcal{O}((n\log n^{2/3})}$ leaves, each corresponds to an instance which is (K_3, F)-free and every vertex from X has maximum degree at most $(n\log n)^{1/3}$. Consider one such instance, for simplicity let us call it (G, L).

We continue as in the proof of Theorem 9 by selecting the set $Y' \subseteq Y$ of vertices of degree at least $(n\log n)^{2/3}$. Recall that $|Y'| \leq n^{2/3}$ and the graph $G' - Y'$ is of maximum degree at most $(n\log n)^{2/3}$.

Now for each $i = 1, \ldots, p-1$ we perform the following steps. Let (G', L') be an instance corresponding to a leaf of the recursion tree, with the set Y' removed. We check if G' contains F_i as an induced subgraph, this can be done in polynomial time by the exhaustive enumeration. If not, then G' is (K_3, F_i)-free and we can call the algorithm given by Theorem 8 and continue exactly as in the proof of Theorem 9.

Thus, let us suppose that there is $S \subseteq V(G')$, such that $G'[S] \simeq F_i$. We observe that $|N[S]| = \mathcal{O}((n\log n)^{2/3})$. We exhaustively guess the coloring of $N[S] \cap X$, this results in $2^{\mathcal{O}((n\log n)^{1/3})}$ branches. In each branch, we update the sets σ for the neighbors of colored vertices; in particular we reject if some vertex from $y \in S \cap Y$ does not see some color in $\sigma(y)$. Note that each instance is $(K_3, F_{i+1} + \ldots, +F_p)$-free.

After the last iteration, the instances corresponding to the leaves of the recursion tree are (K_3, F_p)-free, and thus we continue as in the proof of Theorem 9, i.e., use Theorem 8, restore the set Y', and solve the problem by dynamic programming.

The total number of leaves of the recursion tree is at most (here c_1, c_2 are constants)

$$\underbrace{2^{c_1\cdot(n\log n^{2/3})}}_{\substack{\text{branching on}\\ \text{a high-degree vertex in } X}} \cdot \prod_{i=1}^{p-1} \underbrace{2^{c_2\cdot(n\log n^{2/3})}}_{\substack{\text{branching on the neighborhood}\\ \text{of an induced copy of } F_i}} = 2^{\mathcal{O}((n\log n^{2/3})},$$

each of which corresponds to an instance that is (K_3, F_i)-free for some $i \in [p]$, and thus can be solved in time $2^{\mathcal{O}((n\log n^{2/3})}$ as in Theorem 9. ◀

Now, let us show that the algorithm from Theorem 9 can be used to solve LLSHom(C_4) in P_t-free graphs. The proof of the following lemma is based on a similar argument used by Okrasa and Rzążewski [34] in the non-list case.

▶ **Lemma 11.** *Let (G, L) be a consistent instance of LLSHom(C_4). Then the problem can be reduced in polynomial time to solving two consistent instances of LLSHom(P_3).*

Proof. Let the consecutive vertices of C_4 be $1, 2, 3, 4$. Let the bipartition classes of G be X, Y, and let $L(X) \subseteq \{1, 3\}$ and $L(Y) \subseteq \{2, 4\}$.

Define L', L'' as follows:

$$L'(v) = \begin{cases} L(v) & \text{if } v \in X, \\ \{2\} & \text{if } v \in Y, \end{cases} \quad \text{and} \quad L''(v) = \begin{cases} \{1\} & \text{if } v \in X, \\ L(v) & \text{if } v \in Y. \end{cases}$$

We claim that (G, L) is a yes-instance of LLSHom(C_4) if and only if (G, L') is a yes-instance of LLSHom($C_4[1, 2, 3]$) and (G, L'') is a yes-instance of LLSHom($C_4[4, 1, 2]$). Note that both graphs $C_4[1, 2, 3]$ and $C_4[4, 1, 2]$ are induced three-vertex paths.

First, suppose that there is some $h : (G, L) \xrightarrow{s} C_4$. Let us define $h', h'' : V(G) \to \{1, 2, 3, 4\}$ as follows:

$$h'(v) = \begin{cases} h(v) & \text{if } v \in X, \\ 2 & \text{if } v \in Y, \end{cases} \quad \text{and} \quad h''(v) = \begin{cases} 1 & \text{if } v \in X, \\ h(v) & \text{if } v \in Y. \end{cases}$$

Let us argue that $h' : (G, L') \xrightarrow{s} C_4[1, 2, 3]$ and $h'' : (G, L'') \xrightarrow{s} C_4[4, 1, 2]$. We prove only the first claim. The proof of the second one is analogous. First, h' is clearly a homomorphism. Furthermore, it satisfies the lists, as for $x \in X$ we have $h'(x) = h(x) \in L'(x) = L(x)$, and for $y \in Y$ we have $h'(y) = 2 \in \{2\} = L'(y)$. Finally, let us argue that h' is locally surjective. For each $x \in X$, there are some y, y' such that $h(y) = 2$ and $h(y') = 4$, as otherwise h is not locally surjective. Thus, $h'(y) = h'(y') = 2$, which makes x happy in h'. Similarly, for each $y \in Y$, there are some x, x' such that $h(x) = 1$ and $h(x') = 3$, as otherwise h is not locally surjective. Thus, $h'(x) = 1$ and $h'(x') = 3$, which makes y happy in h'.

Now, suppose there are $h' : (G, L') \xrightarrow{s} C_4[1, 2, 3]$ and $h'' : (G, L'') \xrightarrow{s} C_4[4, 1, 2]$. We define $h : V(G) \to \{1, 2, 3, 4\}$ as follows:

$$h(v) = \begin{cases} h'(v) & \text{if } v \in X, \\ h''(v) & \text{if } v \in Y. \end{cases}$$

Let us argue that $h : (G, L) \xrightarrow{s} C_4$.

First, note that h is a homomorphism, as for every edge $xy \in E(G)$, where $x \in X$ and $y \in Y$, we have $h(x) \in \{1, 3\}$ and $h(y) \in \{2, 4\}$.

Now, observe that h respects the lists L. Indeed, for $x \in X$ we have $h(x) = h'(x) \in L'(x) = L(x)$ and for $y \in Y$ we have $h(y) = h''(y) \in L''(y) = L(y)$.

Finally, let us argue that h is locally surjective. Consider some $x \in X$; the argument for vertices in Y is symmetric. Note that $h''(x) = 1$. Since h'' is locally surjective, x has two neighbors y, y', such that $h''(y) = 2$ and $h''(y') = 4$. Consequently, $h(y) = 2$ and $h(y') = 4$, which makes x happy in h. This completes the proof. ◀

Combining Lemma 11 with Theorem 10, we immediately obtain the following.

▶ **Theorem 12.** *For every $F \in \mathcal{S}$, the LLSHom(C_4) in n-vertex F-free graphs can be solved in time $2^{\mathcal{O}((n \log n)^{2/3})}$.*

4 Hardness for F-free Graphs for $F \in \mathcal{S}$

In this section, we will prove the hardness part of Theorem 1, i.e., if H is connected and $H \notin \mathcal{H}_{\text{poly}} \cup \{P_3, C_4\}$ then there is t such that LLSHOM(H) cannot be solved in subexponential time in P_t-free graphs. It easily follows that such H contains at least one of $K_2^{\circ}, K_2^{\circ\circ}, K_3, P_4, K_{1,3}$ as an induced subgraph, where K_2° and $K_2^{\circ\circ}$ are graphs consisting of an edge with a loop at one or both of its endpoint, respectively. First, we will prove the theorem for several base cases for $H \in \{K_{1,3}, P_4, K_2^{\circ\circ}\}$.

▶ **Theorem 13** (♠). *For $H \in \{K_{1,3}, P_4, K_2^{\circ\circ}\}$ the LLSHOM(H) problem cannot be solved in time $2^{o(n)}$ in n-vertex P_{14}-free graphs, unless the ETH fails. Moreover, the problem is hard even for instances where all the lists are of size at most 2, and each vertex with a list of size exactly two has a neighbor with a list of size exactly one.*

Then, we will generalize the result for H containing an induced subgraph $H' \in \{K_{1,3}, P_4, K_2^{\circ\circ}\}$. The last cases when H contains K_2° or K_3 as an induced subgraph will follow from the base cases and Corollary 7 as for such H the graph H^* contains P_4 as an induced subgraph.

4.1 Proof of the Hardness Part of Theorem 1

First, we show a lemma that helps us extend the hardness reductions to all graphs in the second part of Theorem 1.

▶ **Lemma 14.** *Let H be a graph without isolated vertices and let $u, v \in V(H)$. There exists a graph $Z := H \times H$ with lists L and $z \in V(Z)$ with $L(z) = \{u, v\}$ such that there are at least two homomorphism $h_u, h_v : (Z, L) \xrightarrow{s} H$ such that $h_u(z) = u$ and $h_v(z) = v$.*

Proof. First, we define z as $(u, v) \in V(Z)$ and set $L(z)$ appropriately. We do not restrict other lists of $V(Z)$. For vertices $(a, b) \in V(Z)$ we define $h_u((a, b)) := a$ and $h_v((a, b)) := b$. We verify that h_u is indeed a locally surjective list homomorphism. Take an edge $(a, b)(c, d) \in E(Z)$. We infer that $ac \in E(H)$. For $(a, b) \in V(Z)$ there exist $d \in N_H(b)$ as no vertex is isolated. Vertex (a, b) is happy as for each $c \in N_H(a)$ we have an edge $(a, b)(c, d) \in E(Z)$ and so vertex (a, b) is happy. The proof for h_v is analogous. ◀

Now, we are ready to prove the hardness part of Theorem 1. In particular, we will show the following theorem.

▶ **Theorem 15.** *Let $H \notin \mathcal{H}_{\text{poly}} \cup \{P_3, C_4\}$ be a connected graph. Let q be the number of vertices in the longest induced path in $H \times H$. There exists $t \leq 14 + 2q$ such that LLSHOM(H) cannot be solved in time $2^{o(n)}$ in n-vertex P_t-free graphs, unless the ETH fails.*

Proof. Let $H \notin \mathcal{H}_{\text{poly}} \cup \{P_3, C_4\}$ be connected. As we stated above, the graph H contains at least one of $K_2^{\circ}, K_2^{\circ\circ}, K_3, P_4, K_{1,3}$ as an induced subgraph H'. Theorem 13 proved the cases when $H \in \{K_{1,3}, P_4, K_2^{\circ\circ}\}$.

First, suppose that $H' \in \{K_{1,3}, P_4, K_2^{\circ\circ}\}$. We show how to adjust the hardness construction for H' to H. Recall that by Theorem 13 we can assume that the list of each vertex is of size at most two, and moreover, if v has a list of size exactly two, then v has a neighbor with a list of size one. Now, we describe how to modify an instance (G', L') of LLSHOM(H') (called *original one*) into an equivalent instance (G, L) of LLSHOM(H).

Let w be a vertex of G'. First, consider the case that $|L(w)| = 1$ where $\{c_w\} = L(w)$. Let $X_w := (V(H) \setminus V(H')) \cap N_H(c_w)$. In other words, set X_w represents the neighbors of c_w that are only in H but not in H'. We add $|X_w|$ disjoint copies of graph H into G with lists $L(y) = \{y\}$ for $y \in V(H)$. For each $c \in X_w$ we connect c in the c-th copy of H with w. We call c the *contact vertex* of the respective *additional gadget*.

Now, consider the case that $|L(w)| = 2$. Let $\{a, b\} = L(w)$. As observed there is a vertex $q_w \in N_{G'}(w)$ such that $|L(q)| = 1$ and let $\{c_q\} = L(q)$. Let $X_w^{ab} := (V(H) \setminus V(H')) \cap N_H(a) \cap N_H(b)$. Let $X_w^a := (V(H) \setminus V(H')) \cap N_H(a) \setminus X_w^{ab}$. Let $X_w^b := (V(H) \setminus V(H')) \cap N_H(b) \setminus X_w^{ab}$. Finally, let $X_w := X_w^b \cup X_w^a$. In other words, set X_w^{ab} represents common neighbors of a and b that are only in H and not in H'. Similarly, set X_w^a is composed of the neighbors of a which are not the neighbors of b, and the symmetrical is true for set X_w^b.

We add $|X_w^{ab}|$ disjoint copies of graph H into G' with lists $L(y) = \{y\}$ for $y \in V(H)$. We call those copies of H *additional gadgets*. For each $c \in X_w^{ab}$, we connect c in the c-th copy of H with w. We call c the *contact vertex*. We use Lemma 14 and we add $|X_w|$ copies of the graph Z_w into G'. We connect w with a special vertex $z \in V(Z_w)$. For each $c \in X_w$, we define $L(z) := \{c, c_q\}$ in the c-th copy of Z_w. Again, we say that c is the *contact vertex* and a copy of Z_w is a *non-trivial additional gadget*. Consult Figure 1 for an overview of the construction.

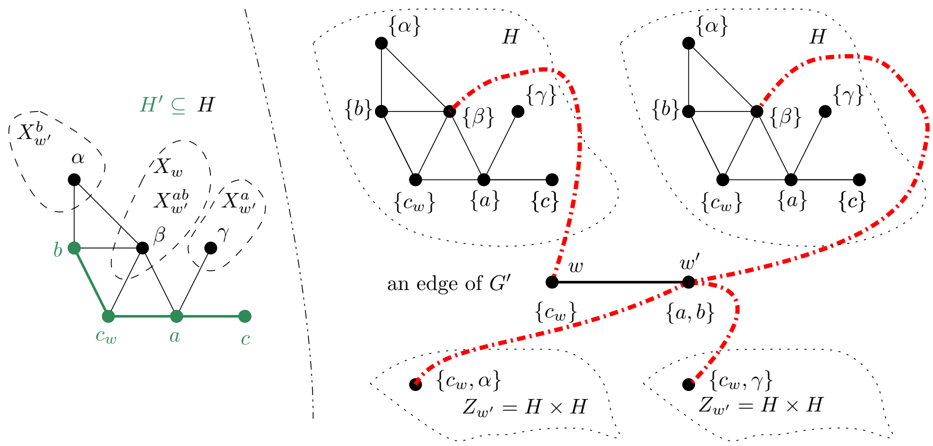

Figure 1 An example of constuction of (G, L) shown on one edge w, w' of (G', L'). The added gadgets are attached using red dash-dotted edges.

It is easy to verify that the additional gadgets always allow us to make the original vertices of the construction happy regarding the vertices outside H' as we added one copy of an additional gadget per vertex v of H' per each color ($c \in N_H(v) \setminus V(H')$) v need to see in its neighborhood in construction of G. The above is the *first property*. On the other hand, the additional gadgets never allow the contact vertices to be mapped to any vertex of H except for the ones that are already seen in the neighborhood within the original construction (which happens only in the case of non-trivial additional gadgets), we call it the *second property*. By the construction, this exception happens only when the original vertex v had a list of size two (in G') $L(v) = \{c_1, c_2\}$ and the color $c_1 \in H'$ had a private neighbor q (with respect to the other color) outside of H', i.e. $q \in N_H(c_1) \setminus N_H(c_1) \setminus V(H')$. In this case, the non-trivial additional gadget may allow mapping its contact vertex to the color of $u \in N_{G'}(v)$ which has the list of size exactly one (such a vertex always exists by the assumptions on the hardness construction of G'). Moreover, observe that regardless of what color is assigned to a vertex v in the `yes`-instance of G', there is always a valid (respecting homomorphism) color to map vertex $u \in V(G) \setminus V(G')$. We call it the *third property*. Note that we do not need to argue about vertices within the additional gadgets as the mapping for them always exists, and it makes all their vertices happy regardless of the mapping of G' by Lemma 14 for non-trivial additional gadgets (and by trivial reasons for the rest).

Whenever we have a yes-instance (G', L') of LLSHom(H'), we obtain a yes-instance (G, L) of LLSHom(H) as a conclusion of the first and third properties allowing all vertices (of G) to be happy and mapped correctly. Conversely, whenever we have a yes-instance (G, L) of LLSHom(H), we obtain a yes-instance (G', L') of LLSHom(H'). Indeed, if we restrict to the vertices of G' only (those are always mapped to H'), they must be already happy without any help from vertices in $V(G) \setminus V(G')$ by the second property (the restricted mapping is a homomorphism trivially). Therefore, we conclude that the newly constructed instance (G, L) is equivalent to the original one, i.e., (G', L').

Observe that the length of the longest induced path in G is the length of the longest path in G' plus twice the length of the longest path in $H \times H$, which we denoted as q.

It remains to show what to do if $H' \in \{K_2^\circ, K_3\}$ is an induced subgraph of H. As H is non-bipartite, we create a connected bipartite graph H^*. Now, observe that if $H' = K_2^\circ$ then H^* contains P_4. Further, if $H' = K_3$ then H^* contains C_6 and so P_4. As in the case of H^* containing a P_4, we already proved hardness, the hardness for H follows by Corollary 7. ◀

5 Concluding Remarks

Let us conclude the paper with discussing some potential ways to strengthen our results. First of all, we believe that the cases covered by the algorithmic statement in Theorem 1 are actually polynomial-time solvable. Furthermore, we think the hardness counterpart of Theorem 1, as well as Theorem 2, hold even in the non-list setting, i.e., for LSHom(H).

Next, recall that in the hardness part of Theorem 1, the length t of the forbidden induced path depends on H. One might wonder if it is possible to find t, such that for every $H \notin \mathcal{H}_{\text{poly}} \cup \{P_3, C_4\}$, the LLSHom($H$) problem is hard in P_t-free graphs.

Suppose that such a t exists and consider $H = P_t$ with consecutive vertices $1, \ldots, t$. Without loss of generality, we may assume that $t \geq 4$. Consider a locally surjective homomorphism h from G to P_t. Note that h is in particular surjective, so there exists a vertex v_1 mapped to 1. By local surjectivity of h, there must be a neighbor v_2 of v_1 mapped to 2, a neighbor v_3 of v_2 mapped to 3, and so on. Note that $v_1, v_2, \ldots, v_t$ is a path in G. Furthermore, this path is induced, as otherwise h is not a homomorphism. Consequently, every yes-instance of LSHom(P_t) (and thus of LLSHom(P_t)) contains an induced t-vertex path. This means that LLSHom(P_t) is polynomial-time solvable (and actually trivial) in P_t-free graphs. On the other hand, $P_t \notin \mathcal{H}_{\text{poly}} \cup \{P_3, C_4\}$, so by Theorem 1 (2.) there exists some t', for which the problem is hard in $P_{t'}$-free graphs.

Moreover, recall that in Theorem 2 the degree bound on F depends on H. Again, one might wonder if this is necessary. However, every yes-instance of LSHom(H) must contain a vertex of degree $\Delta(H)$, as some vertex v of G must be mapped to a maximum-degree vertex a of H, and all vertices from $N_H(a)$ must appear on the set $N_G(v)$. Consequently, we cannot hope for a universal upper bound on the degree of G in the proof of Theorem 2.

The above two examples show that obtaining the full characterization of pairs (H, F), for which LLSHom(H) admits a subexponential-time algorithm in F-free graphs, would be a tedious task. One can probably start with some small graphs F. Let us point out that if $F = P_4$, then LLSHom(H) is polynomial-time solvable for every H. Indeed, P_4-free graphs, also known as cographs, have bounded *cliquewidth* and the result follows from the celebrated meta-theorem for bounded-cliquewidth graphs by Courcelle, Makowsky, and Rotics [8].

References

1 Flavia Bonomo, Maria Chudnovsky, Peter Maceli, Oliver Schaudt, Maya Stein, and Mingxian Zhong. Three-coloring and list three-coloring of graphs without induced paths on seven vertices. *Combinatorica*, 38(4):779–801, 2018. doi:10.1007/s00493-017-3553-8.

2 Laurent Bulteau, Konrad K. Dabrowski, Noleen Köhler, Sebastian Ordyniak, and Daniël Paulusma. An algorithmic framework for locally constrained homomorphisms. *CoRR*, abs/2201.11731, 2022. doi:10.48550/arXiv.2201.11731.

3 Eglantine Camby and Oliver Schaudt. A new characterization of P_k-free graphs. *Algorithmica*, 75(1):205–217, 2016. doi:10.1007/s00453-015-9989-6.

4 Steven Chaplick, Jiří Fiala, Pim van 't Hof, Daniël Paulusma, and Marek Tesař. Locally constrained homomorphisms on graphs of bounded treewidth and bounded degree. *Theoretical Computer Science*, 590:86–95, 2015. doi:10.1016/j.tcs.2015.01.028.

5 Maria Chudnovsky, Shenwei Huang, Pawel Rzążewski, Sophie Spirkl, and Mingxian Zhong. Complexity of C_k-coloring in hereditary classes of graphs. In Michael A. Bender, Ola Svensson, and Grzegorz Herman, editors, *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany*, volume 144 of *LIPIcs*, pages 31:1–31:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ESA.2019.31.

6 Maria Chudnovsky, Shenwei Huang, Sophie Spirkl, and Mingxian Zhong. List 3-coloring graphs with no induced $P_6 + P_3$. *Algorithmica*, 83(1):216–251, 2021. doi:10.1007/s00453-020-00754-y.

7 Maria Chudnovsky, Jason King, Michał Pilipczuk, Paweł Rzążewski, and Sophie Spirkl. Finding large h-colorable subgraphs in hereditary graph classes. *SIAM Journal on Discrete Mathematics*, 35(4):2357–2386, 2021. doi:10.1137/20M1367660.

8 Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150, 2000. doi:10.1007/s002249910009.

9 Pavel Dvořák, Monika Krawczyk, Tomáš Masařík, Jana Novotná, Paweł Rzążewski, and Aneta Żuk. List locally surjective homomorphisms in hereditary graph classes, 2022. doi:10.48550/arXiv.2202.12438.

10 Thomas Emden-Weinert, Stefan Hougardy, and Bernd Kreuter. Uniquely colourable graphs and the hardness of colouring graphs of large girth. *Combinatorics, Probability and Computing*, 7(4):375–386, 1998. doi:10.1017/S0963548398003678.

11 Martin G. Everett and Steve Borgatti. Role colouring a graph. *Mathematical Social Sciences*, 21(2):183–188, 1991. doi:10.1016/0165-4896(91)90080-B.

12 Tomás Feder and Pavol Hell. List homomorphisms to reflexive graphs. *Journal of Combinatorial Theory, Series B*, 72(2):236–250, 1998. doi:10.1006/jctb.1997.1812.

13 Tomás Feder, Pavol Hell, and Jing Huang. List homomorphisms and circular arc graphs. *Combinatorica*, 19(4):487–505, 1999. doi:10.1007/s004939970003.

14 Tomás Feder, Pavol Hell, and Jing Huang. Bi-arc graphs and the complexity of list homomorphisms. *Journal of Graph Theory*, 42(1):61–80, 2003. doi:10.1002/jgt.10073.

15 Tomás Feder, Pavol Hell, and Jing Huang. List homomorphisms of graphs with bounded degrees. *Discrete Mathematics*, 307(3-5):386–392, 2007. doi:10.1016/j.disc.2005.09.030.

16 Jiří Fiala and Jan Kratochvíl. Locally constrained graph homomorphisms - structure, complexity, and applications. *Computer Science Review*, 2(2):97–111, 2008. doi:10.1016/j.cosrev.2008.06.001.

17 Jiří Fiala and Daniël Paulusma. A complete complexity classification of the role assignment problem. *Journal of Computer and System Sciences*, 349(1):67–81, 2005. doi:10.1016/j.tcs.2005.09.029.

18 Petr A. Golovach, Matthew Johnson, Daniël Paulusma, and Jian Song. A survey on the computational complexity of coloring graphs with forbidden subgraphs. *Journal of Graph Theory*, 84(4):331–363, 2017. doi:10.1002/jgt.22028.

19 Petr A. Golovach, Daniël Paulusma, and Jian Song. Closing complexity gaps for coloring problems on H-free graphs. *Information and Computation*, 237:204–214, 2014. doi:10.1016/j.ic.2014.02.004.

20 Carla Groenland, Karolina Okrasa, Paweł Rzążewski, Alex Scott, Paul Seymour, and Sophie Spirkl. H-colouring P_t-free graphs in subexponential time. *Discrete Applied Mathematics*, 267:184–189, 2019. doi:10.1016/j.dam.2019.04.010.

21 Pavol Hell and Jaroslav Nešetřil. On the complexity of H-coloring. *Journal of Combinatorial Theory, Series B*, 48(1):92–110, 1990. doi:10.1016/0095-8956(90)90132-J.

22 Chính T. Hoàng, Marcin Kaminski, Vadim V. Lozin, Joe Sawada, and Xiao Shu. Deciding k-colorability of P_5-free graphs in polynomial time. *Algorithmica*, 57(1):74–81, 2010. doi:10.1007/s00453-008-9197-8.

23 Ian Holyer. The NP-completeness of edge-coloring. *SIAM Journal on Computing*, 10(4):718–720, 1981. doi:10.1137/0210055.

24 Shenwei Huang. Improved complexity results on k-coloring P_t-free graphs. *European Journal of Combinatorics*, 51:336–346, 2016. doi:10.1016/j.ejc.2015.06.005.

25 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

26 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

27 Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. doi:10.1007/978-1-4684-2001-2_9.

28 Tereza Klimošová, Josef Malík, Tomáš Masařík, Jana Novotná, Daniël Paulusma, and Veronika Slívová. Colouring $P_r + P_s$-free graphs. *Algorithmica*, 82(7):1833–1858, 2020. doi:10.1007/s00453-020-00675-w.

29 Daniel Leven and Zvi Galil. NP-completeness of finding the chromatic index of regular graphs. *Journal of Algorithms*, 4(1):35–44, 1983. doi:10.1016/0196-6774(83)90032-9.

30 László Lovász. Coverings and colorings of hypergraphs. In *Proc. 4th Southeastern Conference of Combinatorics, Graph Theory, and Computing*, volume 37 of *Utilitas Math.*, pages 3–12, 1973.

31 Jana Novotná, Karolina Okrasa, Michał Pilipczuk, Paweł Rzążewski, Erik Jan van Leeuwen, and Bartosz Walczak. Subexponential-time algorithms for finding large induced sparse subgraphs. *Algorithmica*, 83(8):2634–2650, 2021. doi:10.1007/s00453-020-00745-z.

32 Karolina Okrasa, Marta Piecyk, and Paweł Rzążewski. Full complexity classification of the list homomorphism problem for bounded-treewidth graphs. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 74:1–74:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ESA.2020.74.

33 Karolina Okrasa and Paweł Rzążewski. Complexity of the list homomorphism problem in hereditary graph classes. *CoRR*, abs/2010.03393, 2020. arXiv:2010.03393.

34 Karolina Okrasa and Paweł Rzążewski. Subexponential algorithms for variants of the homomorphism problem in string graphs. *Journal of Computer and System Sciences*, 109:126–144, 2020. doi:10.1016/j.jcss.2019.12.004.

35 Karolina Okrasa and Paweł Rzążewski. Complexity of the list homomorphism problem in hereditary graph classes. In Markus Bläser and Benjamin Monmege, editors, *38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, March 16-19, 2021, Saarbrücken, Germany (Virtual Conference)*, volume 187 of *LIPIcs*, pages 54:1–54:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.STACS.2021.54.

36 Marta Piecyk and Paweł Rzążewski. Fine-grained complexity of the list homomorphism problem: Feedback vertex set and cutwidth. In Markus Bläser and Benjamin Monmege, editors, *38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, March 16-19, 2021, Saarbrücken, Germany (Virtual Conference)*, volume 187 of *LIPIcs*, pages 56:1–56:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.STACS.2021.56.

37 Marcin Pilipczuk, Michał Pilipczuk, and Paweł Rzążewski. Quasi-polynomial-time algorithm for Independent Set in P_t-free graphs via shrinking the space of induced paths. In Hung Viet Le and Valerie King, editors, *4th Symposium on Simplicity in Algorithms, SOSA 2021, Virtual Conference, January 11-12, 2021*, pages 204–209. SIAM, 2021. doi:10.1137/1.9781611976496.23.

38 Sophie Spirkl, Maria Chudnovsky, and Mingxian Zhong. Four-coloring P_6-free graphs. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1239–1256. SIAM, 2019. doi:10.1137/1.9781611975482.76.