Few Induced Disjoint Paths for *H*-Free Graphs

Barnaby Martin[1(✉)], Daniël Paulusma[1], Siani Smith[1], and Erik Jan van Leeuwen[2]

[1] Department of Computer Science, Durham University, Durham, UK
{barnaby.d.martin,daniel.paulusma,siani.smith}@durham.ac.uk
[2] Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands
e.j.vanleeuwen@uu.nl

Abstract. Paths $P^1, \ldots, P^k$ in a graph $G = (V, E)$ are mutually induced if any two distinct P^i and P^j have neither common vertices nor adjacent vertices. For a fixed integer k, the k-Induced Disjoint Paths problem is to decide if a graph G with k pairs of specified vertices (s_i, t_i) contains k mutually induced paths P^i such that each P^i starts from s_i and ends at t_i. Whereas the non-induced version is well-known to be polynomial-time solvable for every fixed integer k, a classical result from the literature states that even 2-Induced Disjoint Paths is NP-complete. We prove new complexity results for k-Induced Disjoint Paths if the input is restricted to H-free graphs, that is, graphs without a fixed graph H as an induced subgraph. We compare our results with a complexity dichotomy for Induced Disjoint Paths, the variant where k is part of the input.

Keywords: Induced disjoint paths · H-free graph · Complexity dichotomy

1 Introduction

We consider problems related to finding paths connecting pre-specified pairs of vertices. A path between vertices s and t in an undirected graph G is an *s-t path* with *terminals* s and t. Terminal pairs $(s_1, t_1), \ldots, (s_k, t_k)$ are *pairwise disjoint* if $\{s_i, t_i\} \cap \{s_j, t_j\} = \emptyset$ for $i \neq j$. The well-known problem k-Disjoint Paths is to decide for a graph G and pairwise disjoint terminal pairs $(s_1, t_1) \ldots, (s_k, t_k)$, if there are pairwise vertex-disjoint paths $P^1, \ldots, P^k$ such that P^i is an (s_i, t_i)-path for $i \in \{1, \ldots, k\}$; here k is *fixed*, that is, k is not part of the input.

Shiloach [18] proved that 2-Disjoint Paths is polynomial-time solvable. Robertson and Seymour [17] even gave a polynomial-time algorithm for k-Disjoint Paths for every integer $k \geq 2$. In contrast, Disjoint Paths, the variant where k is part of the input, appeared on Karp's list of NP-complete problems.

Our Focus. We consider the *induced* variant of k-Disjoint Paths. We say that paths $P^1, \ldots, P^k$ in a graph $G = (V, E)$ are *mutually induced* if any two distinct

P^i and P^j have neither common vertices nor adjacent vertices, that is, if $i \neq j$ then $V(P^i) \cap V(P^j) = \emptyset$ and $uv \notin E$ for every $u \in V(P^i)$ and $v \in V(P^j)$. This leads to the following problem, where k is a fixed constant.

k-INDUCED DISJOINT PATHS
Instance: a graph G and pairwise disjoint terminal pairs $(s_1, t_1) \ldots, (s_k, t_k)$.
Question: Does G have mutually induced paths $P^1, \ldots, P^k$ such that P^i is an s_i-t_i path for $i \in \{1, \ldots, k\}$?

In contrast to the previous setting, even 2-INDUCED DISJOINT PATHS is NP-complete, as shown both by Bienstock [3] and Fellows [4]. Restricting the input to some special graph class might help improve our understanding of the hardness of the problem. To do this systematically we focus on hereditary graph classes.

A class of graphs is *hereditary* if it is closed under vertex deletion. This is a natural property and non-surprisingly hereditary graph classes provide a framework that captures many well-known graph classes. In particular, it is not difficult to see that a graph class $\mathcal{G}$ is hereditary if and only if it can be characterized by a (unique) set $\mathcal{F}_{\mathcal{G}}$ of forbidden induced subgraphs. For example, if $\mathcal{G}$ is the class of bipartite graphs, then $\mathcal{F}_{\mathcal{G}}$ is the set of all odd cycles.

The characterization by $\mathcal{F}_{\mathcal{G}}$ allows for a systematic study, which usually starts with the case where $\mathcal{F}_{\mathcal{G}}$ has size 1, say $\mathcal{F}_{\mathcal{G}} = \{H\}$ for some graph H. A graph is *H-free* if it cannot be modified to H by a sequence of vertex deletions, and if $\mathcal{F}_{\mathcal{G}} = \{H\}$ we obtain the class of *H-free graphs*, which we consider in our paper.

1.1 Related Work

We first discuss existing results for INDUCED DISJOINT PATHS (where k is part of the input). All the positive results hold for a slightly more general problem definition (see Sect. 4). Golovach et al. [7,8] proved that INDUCED DISJOINT PATHS is linear-time solvable for circular-arc graphs and polynomial-time solvable for AT-free graphs, respectively. Belmonte et al. [2] showed the latter for chordal graphs, and Jaffke et al. [9] did so for any graph class of bounded mim-width. In contrast, INDUCED DISJOINT PATHS stays NP-complete even for claw-free graphs [5], line graphs of triangle-free chordless graphs [16] and thus for (theta, wheel)-free graphs, and for planar graphs; to prove the latter, use a result of Lynch [14] (see [8]).

The following recent dichotomy is immediately relevant for our paper. Let $G_1 + G_2$ be the disjoint union of two vertex-disjoint graphs G_1 and G_2, and let sG denote the disjoint union of s copies of a graph G. We write $F \subseteq_i G$ if F is an *induced* subgraph of a graph G, that is, F can be obtained from G by a sequence of vertex deletions. We let P_r denote the path on r vertices. A *linear forest* is the disjoint union of one or more paths.

Theorem 1 ([15]). *For a graph H,* INDUCED DISJOINT PATHS *on H-free graphs is polynomial-time solvable if $H \subseteq_i sP_3 + P_6$ for some $s \geq 0$;* NP*-complete if H is not a linear forest; and quasipolynomial-time solvable otherwise.*

We return to Theorem 1 later, and we now fix k. Radovanović et al. [16] proved that k-Induced Disjoint Paths is polynomial-time solvable for (theta, wheel)-free graphs. Fiala et al. [5] proved the same result for claw-free graphs. Note that both results complement the aforementioned hardness results when k is part of the input. Golovach et al. [6] showed that Induced Disjoint Paths is even FPT with parameter k for claw-free graphs. The same holds for planar graphs [10], and even for graph classes of bounded genus, as shown by Kobayashi and Kawarabayashi [12]. Let C_r denote the r-vertex cycle. It follows (using Lemma 3) from a result of Leveque et al. [13] that 2-Induced Disjoint Paths is NP-complete for H-free graphs if $H = C_r$ for every $r \geq 3$ with $r \neq 6$.

The generalization from paths to connected subgraphs joining sets of terminals instead of pairs has also been considered, but these results do not impact upon our work in this paper; we refer to [15] for further details. Moreover, the restriction to H-free graphs has also been studied for Disjoint Paths (recall that if k is fixed this problem is polynomial in general [17]); see [11] for a complexity classification of Disjoint Paths for H-free graphs, subject to a set of three unknown cases.

1.2 Our Results

To explain our results we first introduce some extra terminology. For $r \geq 1$, the graph $K_{1,r}$ is the $(r+1)$-vertex *star*, i.e., the graph with vertices $x, y_1, \ldots, y_r$ and edges xy_i for $i = 1, \ldots, r$. The graph $K_{1,3}$ is known as the *claw*. The *subdivision* of an edge uw removes uw and replaces it with a new vertex v and edges uv, vw. A *subdivided claw* is a tree with one vertex x of degree 3 and exactly three leaves. For $1 \leq h \leq i \leq j$, let $S_{h,i,j}$ be the subdivided claw whose three leaves are of distance h, i and j from the vertex of degree 3. Note that $S_{1,1,1} = K_{1,3}$. The graph $S_{1,1,2}$ is called the *chair* (or *fork*). Let $\mathcal{S}$ be the set of graphs, each connected component of which is a path or a subdivided claw.

Using the above terminology we can now present our main theorem, which we prove in Sect. 3.

Theorem 2. *Let $k \geq 2$. For a graph H, k-*Induced Disjoint Paths *is polynomial-time solvable if H is a subgraph of the disjoint union of a linear forest and a chair, and it is* NP*-complete if H is not in $\mathcal{S}$.*

Comparing Theorems 1 and 2 shows that the problem becomes tractable for an infinite family of graphs H after fixing k. As the class of claw-free graphs is contained in the class of chair-free graphs, Theorem 2 extends the aforementioned polynomial-time result of Fiala et al. [5] for claw-free graphs. Moreover, the case $H = C_6$ (the 6-vertex cycle) fills a gap in the aforementioned result of Leveque et al. [13]. As we shall explain in Sect. 3, the NP-hardness construction relies on their gadget but also requires significant additional work. Before doing this we first prove the polynomial-time part of Theorem 2 in Sect. 2. In Sect. 4 we summarize our findings and give a number of relevant open problems.

2 Polynomial-Time Algorithms

In this section we prove the polynomial-time part of Theorem 2. We need the following lemma (proof omitted).

Lemma 1. *For every linear forest F, if the k-INDUCED DISJOINT PATHS problem is polynomial-time solvable for H-free graphs for some graph H, then it is so for $(F+H)$-free graphs.*

We need two known results for proving the polynomial part of Theorem 2 in Lemma 2.

Theorem 3 ([5]). *For every $k \geq 2$, k-INDUCED DISJOINT PATHS is polynomial-time solvable for claw-free graphs.*

Theorem 4 ([1]). *If a connected chair-free graph G contains an induced claw and an induced path P on at least eight vertices, then G has a vertex adjacent to all vertices of P.*

Lemma 2. *Let $k \geq 2$. For every linear forest F, k-INDUCED DISJOINT PATHS is polynomial-time solvable for $(F +$ chair$)$-free graphs.*

Proof. By Lemma 1, it remains to consider chair-free graphs. Let (G, T) be an instance of INDUCED DISJOINT PATHS, where G is a chair-free graph on n vertices and $T = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ is a set of terminal pairs. Let $(P^1, \ldots, P^k)$ be a solution for (G, T) (if it exists). We call a path P^i *long* if it has at least eight vertices; else we call it *short*. We first guess which of the paths of a solution for (G, T) will be short. There are 2^k options for doing this, which is a constant number as k is a constant. We will consider each of these options one by one.

Suppose we consider the option where $T' \subseteq T$ is the subset of terminal pairs that will be in short solution paths. Let $|T'| = k' \leq k$. We guess all $O(n^{5k'}) = O(n^{5k})$ options of choosing the inner vertices of the solution paths for the terminal pairs in T'. We discard an option if two of the guessed solution paths contain an edge between them or if a guessed solution path contains a vertex with a neighbour in some $(s_i, t_i) \notin T'$. Otherwise, we continue as follows.

We first delete all vertices of the guessed solution paths and also their neighbours from G. We denote the new instance by (G, T) again and also write $T = \{(s_1, t_1), \ldots, (s_k, t_k)\}$. Assuming our guess was correct, (G, T) only has solutions $(P^1, \ldots, P^k)$ in which each P^i is long. Hence, from G, we can safely remove for every $i \in \{1, \ldots, k\}$, every vertex that is adjacent to both s_i and t_i.

We now check in polynomial time if there are two terminal s_i and t_i that belong to different connected components of the resulting graph G'. If so, then we can discard this branch. Else, we let $(G'_1, T'_1), \ldots, (G'_r, T'_r)$ be the connected components of G', together with the terminal pairs subsets of T they contain.

We consider each (G'_j, T'_j) as a separate instance. If G'_j has an induced claw, consider a path P^i in a solution. As P^i must be long, Theorem 4 tells us that G'_j must contain a vertex adjacent to all vertices of P^i. However, by construction,

G'_j contains no vertices adjacent to both s_i and t_i which both belong to P^i, a contradiction. We now check in polynomial time if G'_j is claw-free. If it is not, then we may discard the branch, as just argued. Otherwise, we apply Theorem 3 to check in polynomial time if (G'_j, T'_j) has a solution. If for some (G'_j, T'_j) no solution exists, then we move to the next branch; otherwise, we return a yes-answer.

As the number of branches is polynomial and processing each branch takes polynomial time, the total running time of our algorithm is polynomial. □

3 Completing the Proof of Theorem 2

We first prove the NP-completeness part of Theorem 2. We base our proof on a hardness result of Leveque et al. [13] for 2-Induced Cycle, which is to decide if a graph has an induced cycle (which may be assumed, without loss of generality, to be a *hole*, that is, on at least four vertices) containing two pre-specified vertices x and y. Namely, we derive the following relation (proof omitted).

Lemma 3. *An instance (G, x, y) of* 2-Induced Cycle*, where x and y have degree* 2*, can be transformed in polynomial time into an instance of* 2-Induced Disjoint Paths *on a graph G'. Any vertex that is introduced has degree at most* 3 *and its incident edges can be subdivided an arbitrary number of times.*

Our first two results (proof details omitted) require a single change to the construction of [13]. The first rectifies a potential issue with the same claim made in [6] by using Lemma 3.

Lemma 4. 2-Induced Disjoint Paths *is* NP*-complete for $K_{1,4}$-free graphs.*

Lemma 5. *For every $s \geq 3$ with $s \neq 6$,* 2-Induced Disjoint Paths *is* NP*-complete for C_s-free graphs.*

Our third result requires a significant overhaul of the construction in [13].

3.1 Omitting "H"-Graphs and Six-Vertex Cycles

Let H_1 be the "H"-graph on six vertices formed by an edge joining the middle vertices of two paths on three vertices. For $\ell \geq 2$, let H_ℓ be the graph obtained from H_1 by subdividing the crossing edge (i.e., the edge whose endpoints both have degree 3) $\ell - 1$ times.

We prove that for every $\ell \geq 1$, 2-Induced Disjoint Paths is NP-complete for (C_6, H_ℓ)-free graphs. To this end, we consider the hardness reduction by Leveque et al. [13] for 2-Induced Cycle in more detail. We very closely follow their notation and the proof of our main Lemma 6 mimics the proof of their Lemma 2.6. We show how their construction can be modified so that it becomes H_ℓ-free for any fixed $\ell \geq 1$ and C_6-free.

Let ϕ be an instance of 3-Satisfiability consisting of m clauses $C_1, \ldots, C_m$ on n variables $z_1, \ldots, z_n$. For each clause C_j of the form $y_{3j-2} \vee y_{3j-1} \vee y_{3j}$ then

y_i, $i \in [3m]$, is a literal from $\{z_1, \ldots, z_n, \overline{z}_1, \ldots, \overline{z}_n\}$. Let $\ell \geq 1$ be given. We will construct a graph G^ℓ_ϕ with two specified vertices x and y of degree 2 so that G^ℓ_ϕ has a hole containing x and y if and only if there is a truth assignment satisfying ϕ.

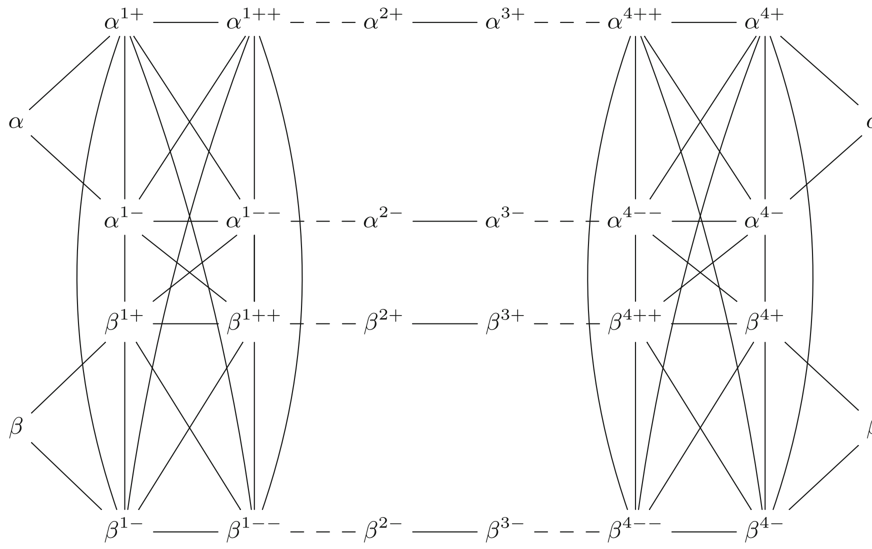

Fig. 1. The literal gadget (dashed lines indicate paths of length ℓ).

For each literal y_j, prepare a graph $G^\ell(y_j)$ as drawn in Fig. 1 where the corresponding labelled vertices inherit a subscript j. Numerous vertices on paths will remain unlabelled. Our literal gadget is more elaborate than that in [13] as we need to forbid, as induced subgraphs, the C_6 and for any fixed ℓ, every H_ℓ.

For each clause C_j, prepare a graph $G^\ell(C_j)$ as drawn in the bottom of Fig. 2 where the corresponding labelled vertices inherit a subscript j. Numerous vertices on paths will remain unlabelled. Our clause gadget is exactly the same as in [13] except we replaced the edges by paths.

For each variable z_i, prepare a graph $G^\ell(z_i)$ as in Fig. 3 consisting of two internally disjoint paths P_i^+ (top) and P_i^- (bottom). The idea in Fig. 3 is that full edges and dashed edges alternate on this diagram and the length is enough for m full edges. The end points of the full edges are labelled $(p^+_{i,1}, p^{++}_{i,1})$, ..., $(p^+_{i,2m}, p^{++}_{i,2m})$ on the top; and $(p^-_{i,1}, p^{--}_{i,1})$, ..., $(p^-_{i,2m}, p^{--}_{i,2m})$ on the bottom. Our variable gadget is exactly the same as in [13] except we lengthened some paths.

The final graph G^ℓ_ϕ is constructed in a manner similar to Leveque et al. [13] from the disjoint union of all the graphs $G^\ell(y_j)$ (literals), $G^\ell(C_j)$ (clauses) and $G^\ell(x_i)$ (variables) with the modifications as below. We indicate specifically where the modifications go beyond the construction of Lemma 2.6 in [13]. The top of

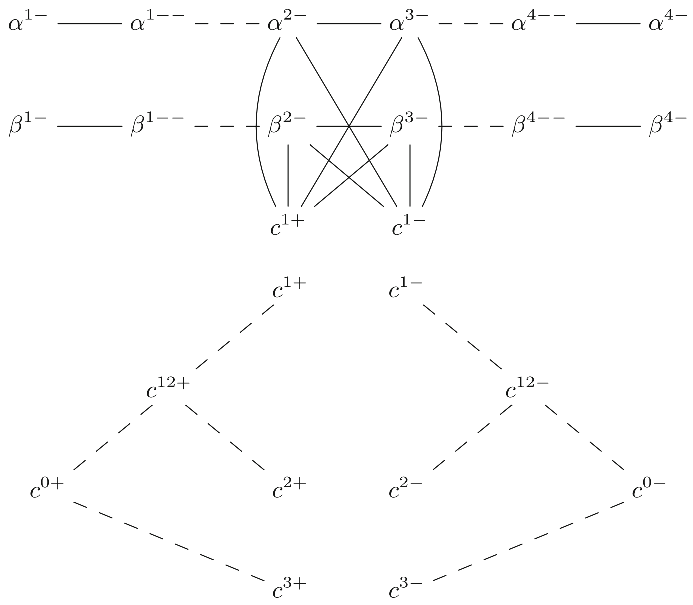

Fig. 2. The clause gadget together with its interface with the literal gadget (drawn above). Dashed lines indicate paths of length ℓ.

Fig. 2 shows how a clause gadget interacts with a literal gadget. Note that a variable gadget interacts with a clause gadget in a similar way.

1. In [13], for $j = 1, \dots, 3m-1$, they added the edges $\alpha'_j\alpha_{j+1}$ and $\beta'_j\beta_{j+1}$. We will instead add paths of length ℓ in place of these edges.
2. In [13], for $j = 1, \dots, m-1$, they added the edges $c_j^{0-}c_{j+1}^{0+}$. We will instead add paths of length ℓ in place of these edges.
3. In [13], for $i = 1, \dots, m-1$, they add the edges $d_i^- d_{i+1}^+$. We will instead add paths of of length ℓ in place of these edges.
4. For $i = 1, \dots, n$, let $y_{n_1}, \dots y_{n_{z_i^-}}$ be the occurrences of $\overline{z}_i$ over all literals. We have slightly different vertex names from [13]. For $j = 1, \dots, z_i^-$, delete the edge $p_{i,j}^+ p_{i,j}^{++}$ and add the four edges $p_{i,j}^+\alpha_{n_j}^{2+}$, $p_{i,j}^+\beta_{n_j}^{2+}$, $p_{i,j}^{++}\alpha_{n_j}^{3+}$, $p_{i,j}^{++}\beta_{n_j}^{3+}$. Additionally to these edges, which were in [13], we also add: $p_{i,j}^+\alpha_{n_j}^{3+}$, $p_{i,j}^+\beta_{n_j}^{3+}$, $p_{i,j}^{++}\alpha_{n_j}^{2+}$, $p_{i,j}^{++}\beta_{n_j}^{2+}$.
5. For $i = 1, \dots, n$, let $y_{n_1}, \dots y_{n_{z_i^+}}$ be the occurrences of z_i over all literals. We have slightly different vertex names from [13]. For $j = 1, \dots, z_i^+$, delete the edge $p_{i,j}^- p_{i,j}^{--}$ and add the four edges $p_{i,j}^-\alpha_{n_j}^{2+}$, $p_{i,j}^-\beta_{n_j}^{2+}$, $p_{i,j}^{--}\alpha_{n_j}^{3+}$, $p_{i,j}^{--}\beta_{n_j}^{3+}$. Additionally to these edges, which were in [13], we also add: $p_{i,j}^-\alpha_{n_j}^{3+}$, $p_{i,j}^-\beta_{n_j}^{3+}$, $p_{i,j}^{--}\alpha_{n_j}^{2+}$, $p_{i,j}^{--}\beta_{n_j}^{2+}$.

6. For $i = 1, \ldots, m$ and $j = 1, 2, 3$, add the edges $\alpha^{2-}_{3(i-1)+j}c^{j+}_i$, $\alpha^{3-}_{3(i-1)+j}c^{j-}_i$, $\beta^{2-}_{3(i-1)+j}c^{j+}_i$, $\beta^{3-}_{3(i-1)+j}c^{j-}_i$. Additionally to these edges, which were in [13], we also add: $\alpha^{3-}_{3(i-1)+j}c^{j+}_i$, $\alpha^{2-}_{3(i-1)+j}c^{j-}_i$, $\beta^{3-}_{3(i-1)+j}c^{j+}_i$, $\beta^{2-}_{3(i-1)+j}c^{j-}_i$.
7. In [13], they add the edges $\alpha'_{3m}d^+_1$ and $\beta'_{3m}c^{0+}_1$. Instead we will add a path of length ℓ.
8. Add the vertex x. In [13], they add the edges $x\alpha_1$ and $x\beta_1$. Instead we will add paths of length ℓ.
9. Add the vertex y. In [13], they add the edges yc^{0-}_m and yd^-_n. Instead we will add paths of length ℓ.

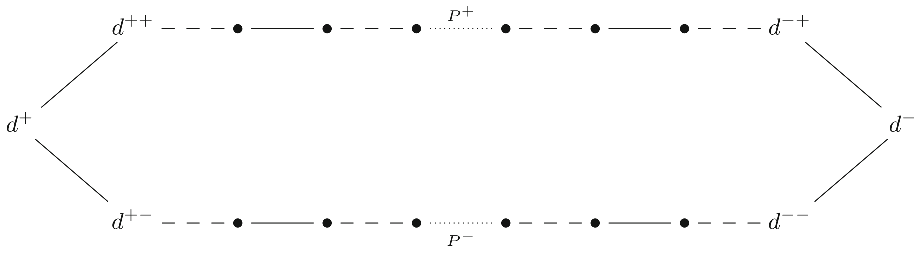

Fig. 3. The variable gadget. Dashed lines indicate paths of length ℓ. Dotted lines indicate a continuation of the gadget.

Claim 1. *ϕ is satisfied by a truth assignment if and only if G^ℓ_ϕ contains a hole passing through x and y.*

Proof. The idea is that any hole emanating from x and moving rightwards towards y (see Fig. 1) must traverse the literal gadgets in precisely one of two ways (upper path on top and bottom; or bottom path on the top and bottom). Now, subsequently the paths building the hole may return to these literal gadgets but they can never leave them as each $\alpha, \beta, \alpha', \beta'$ are already traversed. Hence, the paths do indeed not return to the literal gadgets to ensure their consistent evaluation with the variables and that one in each clause is true.

More formally, first assume that ϕ is satisfied by a truth assignment $\xi \in \{0,1\}^n$. We pick a set of vertices that induce a hole containing x and y.

1. Pick vertices x and y.
2. For $i = 1, \ldots, 3m$, pick $\alpha_i, \alpha'_i, \beta_i, \beta'_i$.
3. For $i = 1, \ldots, 3m$, if y_i is satisfied by ξ, then pick α^{1+}_i, α^{1++}_i, α^{2+}_i, α^{3+}_i,α^{4++}_i, α^{4+}_i and any vertices on a direct path between these. Else, pick α^{1-}_i,α^{1--}_i, α^{2-}_i,α^{3-}_i,α^{4--}_i,α^{4-}_i and any vertices on a direct path between these.
4. For $i = 1, \ldots, n$, if $\xi(i) = 1$, then pick all vertices of P^+_i and all the neighbours of the vertices in P^+_i of the form α^{2+}_k (or one could choose α^{3+}_k, but only one among the two) for any k. Additionally pick any vertices on a direct path between these.

5. For $i = 1, \dots, n$, if $\xi(i) = 0$, then pick all the vertices of P_i^- and all the neighbours of the vertices in P_i^- of the form α_k^{2+} (or one could choose α_k^{3+}, but only one among the two) for any k. Additionally pick any vertices on a direct path between these.
6. For $i = 1, \dots, m$, pick the vertices c_i^{0+} and c_i^{0-}. Choose any $j \in \{3i-2, 3i-1, 3i\}$ such that ξ satisfies y_j. Pick vertices α_j^{2-} and α_j^{3-}.
   - If $j = 3i-2$, then pick $c_i^{12+}, c_i^{1+}, c_i^{1-}, c_i^{12-}$ as well as all vertices on a path between: c^{0+} and c_i^{12+}; c_i^{12+} and c_i^{1+}; c^{0-} and c_i^{12-}; c_i^{12-} and c_i^{1-}.
   - If $j = 3i-1$, then pick $c_i^{12+}, c_i^{2+}, c_i^{2-}, c_i^{12-}$ as well as all vertices on a path between: c^{0+} and c_i^{12+}; c_i^{12+} and c_i^{2+}; c^{0-} and c_i^{12-}; c_i^{12-} and c_i^{2-}.
   - If $j = 3i$, then pick c_i^{3+}, c_i^{3-} as well as all vertices on a path between: c^{0+} and c_i^{3+}; c_i^{0-} and c_i^{3-}.

It suffices to show that the chosen vertices induce a hole in G_ϕ^ℓ containing x and y. The only potential problem is that for some k, one of the vertices $\alpha_k^{2+}, \alpha_k^{3+}, \alpha_k^{2-}, \alpha_k^{3-}$ was chosen more than once. If α_k^{2+} and α_k^{3+} were picked in Step 3, then y_k is satisfied by ξ. Therefore, α_k^{2+} and α_k^{3+} were not chosen in Step 4 or Step 5. Similarly, if α_k^{2-} and α_k^{3-} were picked in Step 6, then y_k is satisfied by ξ. Therefore, α_k^{2-} and α_k^{3-} were not chosen in Step 3. Thus, the chosen vertices induce a hole in G_ϕ^ℓ containing x and y.

Now assume that G_ϕ^ℓ has a hole including x and y. The hole must contain α_1 and β_1 since they are the only neighbours of x. Next, either both α_1^{1+} and β_1^{1+} are in the hole or both α_1^{1-} and β_1^{1-}. W.l.o.g., let α_1^{1+} and β_1^{1+} be in the hole (the same reasoning will apply in the other case). Since α_1^{1-}, β_1^{1-}, α_1^{1--}, β_1^{1--} are all neighbours of two vertices in the hole, they cannot themselves be in the hole. Thus, $\alpha_1^{2^+}$, β_1^{2+}, and the paths that lead to them, must be in the hole. Since $\alpha_1^{2^+}$, β_1^{2+} have the same neighbourhood outside of $G(y_1)$ it follows that $\alpha_1^{3^+}$, β_1^{3+} must be in the hole. Indeed, so must also α_1^{4++}, β_1^{4++}, $\alpha_1^{4^+}$, β_1^{4+} and the path in between. Note that $\alpha_1^{4^-}$, β_1^{4-} are not in the hole, as they are adjacent to both α_1^{4++} and β_1^{4++}. So it must contain instead α_1', β_1', α_2, β_2. By induction, we see for $i \in [3m]$ that the hole must contain α_i, β_i, α_i', β_i'. Also, for each i, the hole must contain $\alpha_i^{1+}, \alpha_i^{1++}, \dots, \alpha_i^{2+}, \alpha_i^{3+}, \dots, \alpha_i^{4++}, \alpha_i^{4+}$ or $\alpha_i^{1-}, \alpha_i^{1--}, \dots, \alpha_i^{2-}, \alpha_i^{3-}, \dots, \alpha_i^{4--}, \alpha_i^{4-}$. Hence, the hole contains d_1^+ and c_1^{0+}.

By symmetry we may assume the hole contains d_1^{++}, and the path to $p_{1,1}^+$, and α_k^{2+} for some k. As α_k^{1++} is adjacent to two vertices in the hole, the hole must contain one of α_k^{2+} and α_k^{3+}. Similarly, the hole cannot proceed on a path to α_k^{4++}, so it must contain $p_{1,2}^+$ and $p_{1,2}^{++}$. By induction, we see that the hole contains $p_{1,i}^+, p_{1,i}^{++}$, for $i \in [n]$, and d_1^-. If the hole contains d_1^{--}, then the hole must contain $p_{1,i}^-, p_{1,i}^{--}$, for $i \in [n]$, and eventually d_1^{+-}, a contradiction. Thus, the hole must contain d_2^+. By induction, for $i \in [n]$, the hole contains all the vertices of the path P_i^+ or P_i^- and, by symmetry, we assume that the hole contains neighbours of the vertices in P_i^+ or P_i^-, one among α_k^{2+} and α_k^{3+}, for each k.

Similarly, for $i \in [m]$, it follows that the hole must contain c_i^{0+} and c_i^{0-}. The hole also contains one of the following:

- c_i^{12+}, c_i^{1+}, c_i^{1-}, c_i^{12-}, and the paths between, and either one of α_j^{2-}, α_j^{3-}; or one of β_j^{2-}, β_j^{3-}.
- c_i^{12+}, c_i^{2+}, c_i^{2-}, c_i^{12-}, and the paths between, and either one of α_j^{2-}, α_j^{3-}; or one of β_j^{2-}, β_j^{3-}.
- c_i^{3+}, and the path between, and either one of α_j^{2-}, α_j^{3-}; or one of β_j^{2-}, β_j^{3-}.

For $i \in [n]$, set $\xi(i) = 1$ if the vertices of P_i^+ are in the hole; otherwise set $\xi(i) = 0$. By construction, at least one literal in every clause is satisfied by ξ. $\square$

Claim 2. *The graph G_ϕ^ℓ is C_6-free and H_i-free for every $i \in [\ell]$.*

Proof. Owing to the length of the ℓ paths that populate our construction and are drawn as dashed edges in our figures, we need only verify the omission of the relevant graphs on the connected components of the graph G_ϕ after the removal of these ℓ paths that are dashed edges. That would suffice for C_6, but H_i has a pendant edge, so for these we must leave a pendant edge from the corresponding connected component at the extremities of an instance of these ℓ paths that are drawn as dashed edges. In this fashion, we only need to check for omission of the given graphs in the non-trivial cases drawn in Fig. 4. It can be readily observed that these graphs are P_7 free (but they are not P_6-free). Hence, we need not test beyond H_3. This task was accomplished by a program testing subgraph isomorphism whose code we provide a link to.[1]

We will give an explicit argument for the case of C_6-freeness, which is simpler as C_6 has numerous symmetries (a transitive automorphism group). Let us begin with the graph depicted on the left-hand side of Fig. 4. This graph has an automorphism that swaps α and β at the same time as $+$ and $-$. It also has an automorphism that only swaps $+$ and $-$. Any subgraph that induces a C_6 cannot contain any of the unlabelled vertices, nor α nor β. This leaves eight vertices that may be involved. We will consider the case where the C_6 contains α^{1+}. Owing to the two automorphisms we have described, this argument would equally apply to α^{1-} and β^{1-}. But any C_6 must involve one of these vertices as there were only eight to choose from. Thus, when we have considered this case, our work is done:

Subcase A. The C_6 contains α^{1+} and α^{1++}. All other neighbours of α^{1+} (except α) are adjacent to α^{1++}. No C_6 can be formed here.
Subcase B. The C_6 contains α^{1+} and α^{1--}. Any C_6 involving a path α^{1--} to α^{1+} must next go to β^{1-}. We cannot continue this cycle.
Subcase C. The C_6 contains α^{1+} and β^{1--}. Any C_6 involving a path β^{1--} to α^{1+} must next go to α^{1-} or α^{1--}. We cannot continue this cycle.
Subcase D. The C_6 contains α^{1+} and β^{1-}. Now, β^{1-} can have as the next in the cycle either of β^{1+} or β^{1++}. We cannot continue this cycle.
Subcase E. The C_6 contains α^{1+} and α^{1-}. Now, α^{1-} can have as the next in the cycle either of β^{1+} or β^{1++}. We cannot continue this cycle.

[1] See https://github.com/barnabymartin/InducedSubgraph.

Now we consider the graph depicted on the right-hand side of Fig. 4. Any C_6 cannot contain any of the unlabelled vertices. It follows that it must use all six remaining vertices. But this induced graph has a triangle, so we are finished. □

We note that the construction in [13] omits all cycles other than C_6, and they note specifically this lacuna, which we have remedied.

We now prove our result and afterwards we prove Theorem 2.

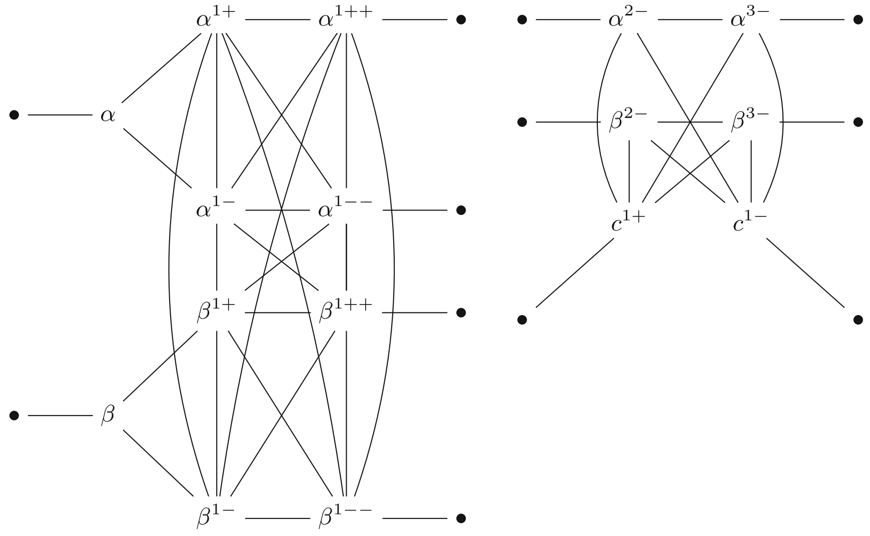

Fig. 4. Cases that need to be checked for omission of the graphs C_6 and H_i ($1 \leq i \leq \ell$).

Lemma 6. *For every integer $\ell \geq 1$,* 2-Induced Disjoint Paths *is* NP-*complete for (C_6, H_ℓ)-free graphs.*

Proof. We give a reduction from an instance ϕ of 3-Satisfiability. First, we construct G_ϕ^ℓ. By Claim 1, G_ϕ^ℓ has a hole through x and y if and only if ϕ is satisfiable. Moreover, G_ϕ^ℓ is (C_6, H_ℓ)-free by Claim 2. We now apply the reduction of Lemma 3. As we can subdivide any number of times the edges incident on the newly created vertices, the resulting graph is still (C_6, H_ℓ)-free. □

Theorem 2 (restated). *Let $k \geq 2$. For a graph H, k-*Induced Disjoint Paths *is polynomial-time solvable if H is a subgraph of the disjoint union of a linear forest and a chair, and it is* NP-*complete if H is not in $\mathcal{S}$.*

Proof. If H has a cycle C_s, apply Lemma 5 for $s \neq 6$ or Lemma 6 for $s = 6$. Then we may assume H is a forest. If H has a vertex of degree at least 4, then every $K_{1,4}$-free graph is H-free, so apply Lemma 4. Suppose H has maximum degree at most 3. If H has a connected component with at least two vertices of degree 3, then H has an induced H_ℓ, so apply Lemma 6 again. Else, H is in $\mathcal{S}$. If H is a subgraph of the disjoint union of a linear forest and a chair, apply Lemma 2. □

4 Conclusions

We showed new tractable and hard results for k-Induced Disjoint Paths for H-free graphs and extended a number of known results in this way. The open cases all involve graphs H that are not the disjoint union of some linear forest and the chair but that do belong to the family $\mathcal{S}$; we recall that $\mathcal{S}$ consists of all graphs, every connected component of which is a path P_r or a subdivided claw $S_{h,i,j}$. In order to make further progress on k-Induced Disjoint Paths and these other problems we must better understand $S_{h,i,j}$-free graphs.

In some previous works, a slightly more general definition is used (see also Sect. 1). Given a graph G, *vertex-disjoint* paths $P^1, \ldots, P^k$, for some integer $k \geq 1$ are *flexibly mutually induced paths* of G if there is no edge between two vertices from different P^i and P^j except possibly between the endpoints of the paths. If k is in the input, the complexity of the corresponding decision problem and ours is most likely different for P_r-free graphs. Namely, Flexibly Induced Disjoint Paths is NP-complete for P_{14}-free graphs [15], whilst Induced Disjoint Path is quasipolynomial-time solvable for P_{14}-free graphs by Theorem 1. However, it is readily seen that all polynomial-time results in Theorem 2 (so, for fixed $k \geq 2$) also hold for Flexible k-Induced Disjoint Paths.

References

1. Alekseev, V.E.: Polynomial algorithm for finding the largest independent sets in graphs without forks. Discrete Appl. Math. **135**, 3–16 (2004)
2. Belmonte, R., Golovach, P.A., Heggernes, P., van't Hof, P., Kaminski, M., Paulusma, D.: Detecting fixed patterns in chordal graphs in polynomial time. Algorithmica **69**, 501–521 (2014). https://doi.org/10.1007/s00453-013-9748-5
3. Bienstock, D.: On the complexity of testing for odd holes and induced odd paths. Discrete Math. **90**, 85–92 (1991)
4. Fellows, M.R.: The Robertson-Seymour theorems: a survey of applications. In: Proceedings of AMS-IMS-SIAM Joint Summer Research Conference (1989). Contemp. Math. **89**, 1–18
5. Fiala, J., Kamiński, M., Lidický, B., Paulusma, D.: The k-in-a-path problem for claw-free graphs. Algorithmica **62**, 499–519 (2012). https://doi.org/10.1007/s00453-010-9468-z
6. Golovach, P.A., Paulusma, D., van Leeuwen, E.J.: Induced disjoint paths in claw-free graphs. SIAM J. Discrete Math. **29**, 348–375 (2015)
7. Golovach, P.A., Paulusma, D., van Leeuwen, E.J.: Induced disjoint paths in circular-arc graphs in linear time. Theor. Comput. Sci. **640**, 70–83 (2016)
8. Golovach, P.A., Paulusma, D., van Leeuwen, E.J.: Induced disjoint paths in AT-free graphs. J. Comput. Syst. Sci. **124**, 170–191 (2022)

9. Jaffke, L., Kwon, O., Telle, J.A.: Mim-width I. induced path problems. Discrete Appl. Math. **278**, 153–168 (2020)
10. Kawarabayashi, K., Kobayashi, Y.: A linear time algorithm for the induced disjoint paths problem in planar graphs. J. Comput. Syst. Sci. **78**, 670–680 (2012)
11. Kern, W., Martin, B., Paulusma, D., Smith, S., van Leeuwen, E.J.: Disjoint paths and connected subgraphs for H-free graphs. Theor. Comput. Sci. **898**, 59–68 (2022)
12. Kobayashi, Y., Kawarabayashi, K.: Algorithms for finding an induced cycle in planar graphs and bounded genus graphs. In: Proceedings of the SODA 2009, pp. 1146–1155 (2009)
13. Lévêque, B., Lin, D.Y., Maffray, F., Trotignon, N.: Detecting induced subgraphs. Discrete Appl. Math. **157**, 3540–3551 (2009)
14. Lynch, J.: The equivalence of theorem proving and the interconnection problem. SIGDA Newslett. **5**, 31–36 (1975)
15. Martin, B., Paulusma, D., Smith, S., van Leeuwen, E.J.: Induced disjoint paths and connected subgraphs for H-free graphs. In: Bekos, M.A., Kaufmann, M. (eds.) WG 2022. LNCS, vol. 13453, pp. 398–411. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-15914-5_29
16. Radovanović, M., Trotignon, N., Vušković, K.: The (theta, wheel)-free graphs Part IV: induced paths and cycles. J. Comb. Theory Ser. B **146**, 495–531 (2021)
17. Robertson, N., Seymour, P.D.: Graph minors. XIII. The disjoint paths problem. J. Comb. Theory Ser. B **63**, 65–110 (1995)
18. Shibi, N.: Algorithme de recherche d'un stable de cardinalité maximum dans un graphe sans étoile. Discrete Math. **29**, 53–76 (1980)