Complexity Framework for Forbidden Subgraphs: When Hardness Is Not Preserved under Edge Subdivision

Barnaby Martin
Durham University, Durham, United Kingdom
barnaby.d.martin@durham.ac.uk

Sukanya Pandey
Utrecht University, Utrecht, The Netherlands
s.pandey1@uu.nl

Daniël Paulusma
Durham University, Durham United Kingdom
daniel.paulusma@durham.ac.uk

Siani Smith
Durham University, Durham, United Kingdom
siani.smith@durham.ac.uk

Erik Jan van Leeuwen
Utrecht University, Utrecht, The Netherlands
e.j.vanleeuwen@uu.nl

Abstract

A graph G is H-subgraph-free if G does not contain H as a (not necessarily induced) subgraph. We make inroads into the classification of three problems for H-subgraph-free graphs that have the properties that they are solvable in polynomial time on classes of bounded treewidth and NP-complete on subcubic graphs, yet NP-hardness is not preserved under edge subdivision. The three problems are k-Induced Disjoint Paths, C_5-Colouring and Hamilton Cycle. Although we do not complete the classifications, we show that the boundary between polynomial time and NP-complete differs for C_5-Colouring from the other two problems.

2012 ACM Subject Classification Mathematics of computing → Graph theory; Theory of computation → Graph algorithms analysis; Theory of computation → Problems, reductions and completeness

Keywords and phrases forbidden subgraph; complexity dichotomy; edge subdivision; treewidth

1 Introduction

Let G and H be two graphs. If H can be obtained from G by a sequence of vertex deletions only, then H is *an induced* subgraph of G; else G is *H-free*. The induced subgraph relation has been well studied in the literature for many classical graph problems, such as Colouring, Feedback Vertex Set, Independent Set, and so on.

In this paper we focus on the subgraph relation. A graph G is said to contain a graph H as a *subgraph* if H can be obtained from G by a sequence of vertex deletions and edge deletions; else G is said to be *H-subgraph-free*. For a set of graphs $\mathcal{H}$, a graph G is *$\mathcal{H}$-subgraph-free* if G is H-subgraph-free for every $H \in \mathcal{H}$; we also write that G is $(H_1, \ldots, H_p)$-subgraph-free. Graph classes closed under edge deletion are also called *monotone* [1, 3].

Complexity classifications for H-subgraph-free graphs have been less well studied in the literature than for H-free graphs; see [2] for complexity classifications for Independent Set, Dominating Set and Longest Path and [5] for a classification for List Colouring; all these classifications hold even for $\mathcal{H}$-subgraph-free graphs, where $\mathcal{H}$ is any finite set of graphs. In [6] a short, alternative proof for the classification for Independent Set for

H-subgraph-free graphs was given. In general, such classifications might be hard to obtain; see, for example, [6] for a partial classification for Colouring for H-subgraph-free graphs. Therefore, in [8] a more systematic approach was followed, namely by introducing a new framework for $\mathcal{H}$-subgraph-free graph classes (finite $\mathcal{H}$) adapting the approach of [6].

To explain the framework of [8] we need to introduce some additional terminology. Firstly, a class of graphs has bounded *treewidth* if there exists a constant c such that every graph in it has treewidth at most c. Now let $G = (V, E)$ be a graph. Then G is *subcubic* if every vertex of G has degree at most 3. The *subdivision* of an edge $e = uv$ of G replaces e by a new vertex w with edges uw and wv. For an integer $k \geq 1$, the *k-subdivision* of G is the graph obtained from G by subdividing each edge of G exactly k times. Let $\mathcal{G}$ be a class of graphs. For an integer k we let $\mathcal{G}^k$ consist of the k-subdivisions of the graphs in $\mathcal{G}$.

The framework of [8] makes a distinction between "efficiently solvable" and "computationally hard", which could for example mean a distinction between "polynomial time" and NP-complete. Let Π be a decision problem that takes as input a (possibly weighted) graph. We say that Π is computationally hard *under edge subdivision of subcubic graphs* if there exists an integer $k \geq 1$ such that the following holds for the class of subcubic graphs $\mathcal{G}$: if Π is computationally hard for $\mathcal{G}$, then Π is computationally hard for $\mathcal{G}^{kp}$ for every integer $p \geq 1$. That is, a graph problem Π is a *C123-problem* (belongs to the framework) if it satisfies the following three conditions:

C1. Π is efficiently solvable for every graph class of bounded treewidth;
C2. Π is computationally hard for the class of subcubic graphs; and
C3. Π is computationally hard under edge subdivision of subcubic graphs.

The *claw* is the 4-vertex star. A *subdivided* claw is a graph obtained from a claw after subdividing each of its edges zero or more times. The *disjoint union* of two vertex-disjoint graphs G_1 and G_2 has vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2)$. The set $\mathcal{S}$ consists of the graphs that are disjoint unions of subdivided claws and paths. As shown in [8], C123-problems allow for full complexity classifications for $\mathcal{H}$-subgraph-free graphs (as long as $\mathcal{H}$ has finite size).

▶ **Theorem 1** ([8]). *Let Π be a C123-problem. For a finite set $\mathcal{H}$, the problem Π on $\mathcal{H}$-subgraph-free graphs is efficiently solvable if $\mathcal{H}$ contains a graph from $\mathcal{S}$ and computationally hard otherwise.*

Examples of C123-problems include Independent Set, Dominating Set, List Colouring, Odd Cycle Transversal, Max Cut, Steiner Tree and Vertex Cover; see [8]. However, there are still many graph problems that are not C123-problems, such as Colouring (whose classification is still open even for H-subgraph-free graphs). Hence, it is a natural question if those problems can still be classified for graph classes defined by some set of forbidden subgraphs.

How do problems that do not satisfy C3 but that do satisfy C1 and C2 behave for H-subgraph-free graphs? Can we still classify their computational complexity?

Let us call such problems C12-problems. We study the problems k-Induced Disjoint Paths, C_5-Colouring and Hamilton Cycle. All of these problems are C12-problems. All of these violate our condition C3, but the manner of this violation is different for each of them. As we will observe later, for $k \geq 3$, C_5-Colouring becomes trivially true under k-subdivision. On the other hand, under k-subdivision (for any k), Hamilton Cycle becomes trivially false (unless we started with a cycle), and k-Induced Disjoint Paths reduces to k-Disjoint Paths, which can be solved in polynomial time. Let us note that when the parameter k is part of the input, Disjoint Paths and Induced Disjoint Paths are C123-problems [8].

Let us make the following observation that is a restriction of Theorem 1.

▶ **Theorem 2** ([8]). *Let Π be a C12-problem. For a finite set $\mathcal{H}$, the problem Π on $\mathcal{H}$-subgraph-free graphs is efficiently solvable if $\mathcal{H}$ contains a graph from $\mathcal{S}$.*

Let H_1 be the "H"-graph, that is, the graph on six vertices which is 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 edge whose endpoints each have degree 3 exactly $\ell - 1$ times. See Figure 1 for two examples. Note that hereonin $H_1, \ldots, H_k$ will never denote arbitrary graphs, but rather the "H"-graphs we just defined. When allied with C2, Condition C3 ensure that C123-problems remain NP-hard on $(H_1, \ldots, H_k)$-subgraph-free graphs (for all k). Note that C123-problems are in P when on $(H_1, H_2, \ldots)$-subgraph-free graphs, as these have bounded treewidth [8].

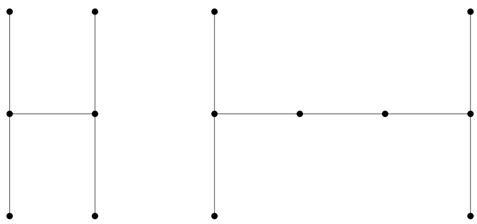

Figure 1 Left: the graph H_1. Right: the graph H_3.

Our results are as follows.

▶ **Theorem 3.** *k-Induced Disjoint Paths is in P for both of the classes of H_1-subgraph-free graphs and H_2-subgraph-free graphs. For all $\ell > 4$, 2-Induced Disjoint Paths is NP-complete for the class of $(H_4, \ldots, H_{\ell-1})$-subgraph-free graphs.*

▶ **Theorem 4.** *C_5-Colouring is in P for (H_1, H_2, H_3)-subgraph-free graphs, but it is NP-complete for (H_1, H_2)-subgraph-free graphs.*

▶ **Theorem 5.** *Hamilton Cycle is in P for the class of H_1-subgraph-free graphs.*

Related Work for the Induced Subgraph Relation

Recall that for some graph H, a graph G is H-free if G can be obtained from H by a sequence that only consists of vertex deletions. There is an almost complete classification for Disjoint Paths, in [9], in which two cases are left open. For Induced Disjoint Paths, there is a complete classification in [12]. For k-Induced Disjoint Paths, there is a partial classification in [13]. For C_5-Coloring, there is a partial classification in [4]. For Hamilton Path, some partial classification can be inferred from [10].

2 k-Induced Disjoint Paths

Let us recall our family of problems.

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 $i \in \{1, \ldots, k\}$?

Note that k-Disjoint Paths does not insist the paths are mutually induced, but only that they are node-disjoint. The versions of these problems in which k is part of the input are denoted Induced Disjoint Paths and Disjoint Paths. Let us note that k-Disjoint Paths is in P for all k [14].

2.1 Tractability for the H_2-subgraph-free Case

The following will be a good warm-up for the more complicated case of H_2.

▶ **Theorem 6.** *For every integer $k \geq 2$, k-Induced Disjoint Paths is in P on H_1-subgraph-free graphs.*

Proof. We prove the result for $k = 2$. The extension to $k \geq 2$ will be straightforward. Let G be an instance of 2-Induced Disjoint Paths together with two terminal pairs (s_1, t_1) and (s_2, t_2). We may assume without loss of generality that there is no edge between s_1 and t_1 and no edge between s_2 and t_2.

We first check if there exists a solution in which one of the paths has length 2. We can do this in polynomial time as follows. We first consider all $O(n)$ options of choosing a vertex to be the middle vertex of one of these paths. We then check if the graph obtained from removing the guessed middle vertex and its two neighbouring terminals s_i and t_i as well all the neighbours of these three vertices has a connected component that contains both terminals s_j and t_j of the other pair. This takes polynomial time.

We now check if there exists a solution in which both paths have length at least 3. We consider all $O(n^4)$ options of choosing the neighbours s_1', t_1', s_2', t_2' of s_1, t_1, s_2, t_2, respectively, on the two solution paths (should a solution exist). We discard a branch if there exists an edge between a vertex of $\{s_1, s_1', t_1, t_1'\}$ and a vertex of $\{s_2, s_2', t_2, t_2'\}$. Suppose this is not the case. We remove s_1, t_1, s_2, t_2 and every neighbour of a vertex in $\{s_1, t_1, s_2, t_2\}$ that does not belong to $\{s_1', t_1', s_2', t_2'\}$. Afterwards, it suffices to solve 2-Disjoint Paths on the resulting graph G' with terminal pairs (s_1', t_1') and (s_2', t_2'). This can be seen as follows. Any solution of 2-Induced Disjoint Paths is a solution of 2-Disjoint Paths. Now suppose we have a solution (P_1, P_2) of 2-Disjoint Paths. If there exist an edge between a vertex of P_1 and a vertex of P_2, then we find the forbidden subgraph H_1 (possibly after adding the vertices s_1, t_1, s_2, t_2 back). Since the number of branches is $O(n^4)$ and each created instances of 2-Disjoint Paths can be solved in polynomial time [14, 15], the running time of this case is polynomial as well. ◀

2.2 Tractability for the H_2-subgraph-free Case

We would like to make some further assumptions about a k-Disjoint Paths (not induced) algorithm that we will call iteratively. We would like that a path between s_i and t_i

$(*)$ avoids neighbours of $\{s_j, t_j\}$ $(i \neq j)$.

Now, we enforce this by preprocessing the input, or rather reducing a single input into multiple inputs that we then solve. Let us consider all paths of length three from each of the terminals $\{s_1, t_1, \ldots, s_k, t_k\}$ (if we meet another terminal this length will be potentially

.

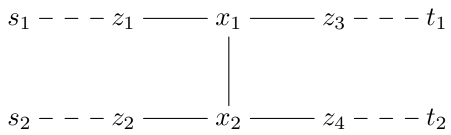

Figure 2 The subgraph resulting from our construction.

less than three but then we either have a no-instance or we reduce to a case involving fewer pairs of terminals). We now consider all combinations of these and we forget about all other vertices at distance at most three from the corresponding terminals by moving to a subgraph. Note that removing vertices cannot introduce an H_2 as a subgraph into the graph. In these preprocessed graphs the terminals all have degree 1 and the vertices at distance one and two all have degree 2, It follows that $(*)$ is enforced but we need to run our algorithm on polynomially many new graphs. Moreover, as discussed, we will have the additional property

(†) terminals have degree 1.

Now, we run an algorithm for k-Disjoint Paths and we either solve k-Induced Disjoint Paths or we end up, due to $(*)$, with a subgraph as shown in Figure 2 where we assume w.l.o.g. that a failure results in paths connecting the first two pairs of terminals.
Let $S = \{z_1, x_1, z_3, z_2, x_2, z_4\}$.

Suppose $z \in \{z_1, z_2, z_3, z_4\}$ has two neighbours outside of S. Then G has H_2 as a subgraph. Thus we may assume (‡) that z has at most one neighbour outside of S.

Suppose there are both the edges (z_1, z_2) and (z_3, z_4). Then G has a H_2 as a subgraph, since we assumed $(*)$, which implies that $\{s_1, t_1, s_2, t_2\} \cap \{z_1, z_2, z_3, z_4\} = \emptyset$. Suppose there are both the edges (z_1, z_4) and (z_2, z_3). Then, by $(*)$, we again have $\{s_1, t_1, s_2, t_2\} \cap \{z_1, z_2, z_3, z_4\} = \emptyset$, which implies we have an H_2 whose middle path runs z_1, x_1, z_3.

Suppose we have one of the edges (z_1, z_2) and (z_3, z_4), and one of $\{x_1, x_2\}$ has a neighbour q outside of S. W.l.o.g. let us assume it is (z_1, z_2) and x_1. Then there is an H_2 with middle path x_1, z_1, z_2 since $z_2 \neq s_2$ by $(*)$. Suppose we have one of the edges (z_1, z_4) and (z_2, z_3), and one of $\{x_1, x_2\}$ has a neighbour q outside of S. W.l.o.g. let us assume it is (z_1, z_4) and x_1. Then there is an H_2 with middle path x_1, z_1, z_4 since $z_4 \neq t_2$ by $(*)$. Thus we are, w.l.o.g., in one of the two situations depicted in Figure 3 and Figure 4. The dotted lines are possible edges and each vertex of $\{z_1, z_2, z_3, z_4\}$ has at most one neighbour outside of S.

▶ **Lemma 7.** *Let G be an instance of k-Induced Disjoint Paths and let G' be that instance after one application of Rule 1. Then G is a yes-instance of k-Induced Disjoint Paths iff G' is a yes-instance of k-Induced Disjoint Paths.*

Proof. Let us address the change we see in Figure 3.

Suppose we have a solution to k-Induced Disjoint Paths in G. If it uses no vertices in S, then it is already a solution to k-Induced Disjoint Paths in G'. Thus, it must use some vertex in S.

If the solution uses both x_1 and x_2, then we can substitute the edge (x_1, x_2) in the solution to k-Induced Disjoint Paths in G with x to obtain a solution to k-Induced Disjoint Paths in G'. It cannot use neither of x_1 or x_2 so, w.l.o.g., suppose it used x_1. We can substitute this for x to obtain a solution to k-Induced Disjoint Paths in G', unless some other solution path runs through a neighbour q of x_2. Note q cannot be a terminal,

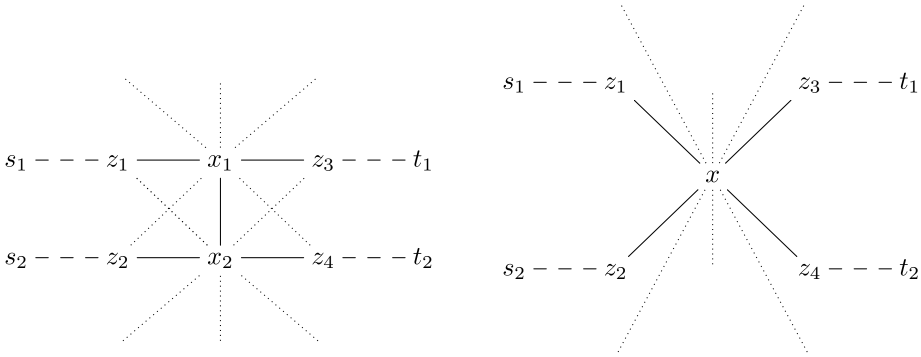

Figure 3 Rule 1. Possible connections in our subgraph (left). What we replace this subgraph with (right).

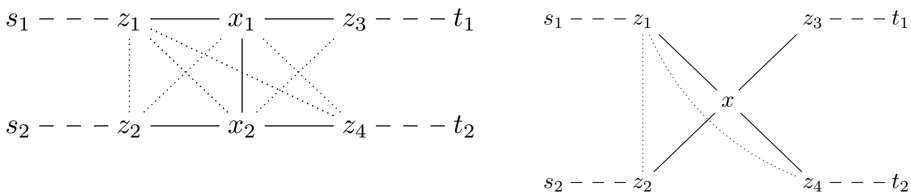

Figure 4 Rule 2. Possible connections in our subgraph (left). What we replace this subgraph with (right).

due to $(*)$, hence it has two neighbours p and r on this other solution path and these are outside of $\{z_1, x_1, z_3\}$ because this path must avoid x_1 and any of its neighbours. But now p, q, r, q, x_2, x_1, z_1, x_1, z_3 forms an H_2.

Suppose we have a solution to k-Induced Disjoint Paths in G'. If this solution does not involve x then it maps to a solution of k-Induced Disjoint Paths in G. Suppose now it does involve x. Suppose mapping x to either of x_1 or x_2 does not produce a solution to k-Induced Disjoint Paths in G. Then mapping x to either the edge (x_1, x_2) (or the symmetric (x_2, x_1)) must produce a solution to k-Induced Disjoint Paths in G. ◀

▶ **Lemma 8.** *Let G be an instance of k-Induced Disjoint Paths and let G' be that instance after one application of Rule 2. Then G is a yes-instance of k-Induced Disjoint Paths iff G' is a yes-instance of k-Induced Disjoint Paths.*

Proof. Let us address the change we see in Figure 4, where we assume (w.l.o.g.) that there was no edge in G from z_3 to z_4 or from z_2 to z_3.

Suppose we have a solution to k-Induced Disjoint Paths in G. If it uses no vertices in S, the it is already a solution to k-Induced Disjoint Paths in G'. Thus, it must use some vertex in S. Recall the assumption (‡). Suppose the solution uses the edge z_1 to z_2. Then it doesn't use any other vertex from S and we can keep this edge to obtain a solution for k-Induced Disjoint Paths in G'. Suppose the solution uses the edge z_1 to z_4. Then it doesn't use any other vertex from S and we can keep this edge to obtain a solution for k-Induced Disjoint Paths in G'.

If the solution uses both x_1 and x_2, then we can substitute the edge (x_1, x_2) in the solution to k-Induced Disjoint Paths in G with x to obtain a solution to k-Induced

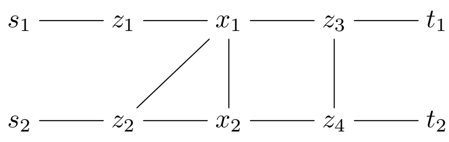

Figure 5 A counterexample to the statement that G has a subgraph H_2 implies that G' has a subgraph H_2.

Disjoint Paths in G'. Suppose it uses neither of x_1 and x_2. Then it uses either the edge (z_1, z_4) or (z_1, z_2) and we are in a previous case.

Now, suppose the solution uses z_1 or z_3, then it must use x_1 or z_2; and if it uses z_2 or z_4, then it must use x_1 or x_2. We assumed it was only one, so let us assume (w.l.o.g.) that it is x_1. Owing to (‡), we can substitute this for x to obtain a solution to k-Induced Disjoint Paths in G'.

Suppose we have a solution to k-Induced Disjoint Paths in G'. If this solution does not involve x then it maps to a solution of k-Induced Disjoint Paths in G. Suppose now it does involve x. Suppose mapping x to either of x_1 or x_2 does not produce a solution to k-Induced Disjoint Paths in G. Then mapping x to either the edge (x_1, x_2) (or the symmetric (x_2, x_1)) must produce a solution to k-Induced Disjoint Paths in G. ◀

▶ **Lemma 9.** *If G omits H_2 as a subgraph then G' omits H_2 as a subgraph.*

Proof. Suppose G' has an H_2 involving x. If x is a leaf vertex in H_2 then it is clear that G' already had this H_2 involving either x_1 or x_2.

Suppose x is a degree 3 vertex in H_2. If the neighbours of x in the H_2 were both neighbours of x_1 or both neighbours of x_2 in G then it is clear that G already had this H_2.

Now suppose one of the neighbours, say z'_1, was adjacent to x_1 and the other, say z'_2, was adjacent to x_2. Let x', x'', z''_1, z''_2 form the remaining vertices of the H_2 where x, x', x'' and z''_1, x'', z''_2 are both paths of length 2 in this H_2. Thus, z'_1, x, z'_2, x, x', x'' and z''_1, x'', z''_2 form the H_2 in G'. *W.l.o.g.* suppose x' was adjacent to x_1 in G. Now it is clear that z'_1, x_1, x_2, x_1, x', x'' and z''_1, x'', z''_2 formed an H_2 in G.

Finally, suppose that x is the degree 2 vertex in H_2. Let z'_1, x', z'_2, x', x, x'', z''_1, x'', z''_2 be the paths that form the H_2 in G'. Suppose, w.l.o.g., that x' was adjacent to x_1 in G. If x'' was also adjacent to x_1 in G then z'_1, x', z'_2, x', x_1, x'', z''_1, x'', z''_2 are paths that form an H_2 in G. Suppose now that x'' was adjacent to x_2 but not x_1 in G and we may also assume that x' is adjacent to x_1 but not x_2. Now z'_1, x', z'_2, x', x_1, x_2, z_2, x_2, z_4 are paths that form the H_2 in G, unless $\{z_2, z_4\} \cap \{z'_1, z'_2\} \neq \emptyset$. W.l.o.g., suppose $z_2 = z'_1$. If $z_2 \neq s_2$ and p is next on the path from t_2 to s_2 after the z_2, then p, z_2, x_2, z_2, x', x_1, z_1, x_1, z_3 is an H_2 in G (note that $\{z_1, z_3\} \cap \{x', z_2, p\} = \emptyset$). Finally, if $z_2 = s_2$ then we violate condition (†). ◀

Let us note that the sequent G has H_2 as a subgraph then G' has H_2 as a subgraph is in general false. A counterexample for G is furnished in Figure 5. The dotted lines are possible edges and each vertex of $\{z_1, z_2, z_3, z_4\}$ has at most one neighbour outside of S.

▶ **Corollary 10.** k-Induced Disjoint Paths *is in* P *on H_2-subgraph-free graphs.*

Proof. We iteratively run our algorithm for k-Disjoint Paths. If it returns no, then it is also a no-instance to k-Induced Disjoint Paths. If it returns a solution, the either this is a solution to k-Induced Disjoint Paths or we use one of the two reduction rules.

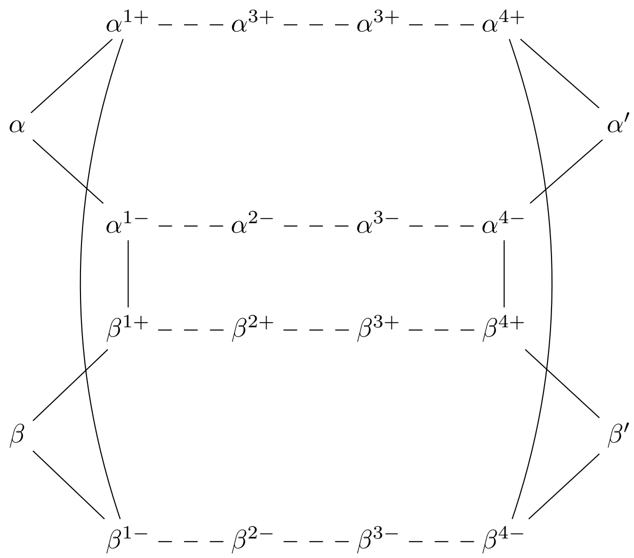

Figure 6 The literal gadget (dashed lines indicate paths of length ℓ).

These make the instance smaller by one vertex, so the procedure will terminate. Note that it follows from Lemma 9 that we will never find an H_2 as a subgraph. ◀

2.3 NP-hardness for the H_4-subgraph-free Case

We follow very closely the argument from Section 2.4 in [11]. It is not possible to take that construction and simply subdivide all edges some fixed number of times. However, some of the edges may be liberally subdivided. Indeed, our gadgets are precisely those from [11] with some edges subdivided $\ell - 1$ times. These edges are drawn in dashed lines in our gadgets in Figures 6, 7 and 8. Thus, the dashed edges represent ℓ-paths.

Let $\ell \geq 1$ be an integer. 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 ($j = 1, \ldots, m$), with $C_j = y_{3j-2} \vee y_{3j-1} \vee y_{3j}$, then y_i ($i = 1, \ldots, 3m$) is a literal from $\{z_1, \ldots, z_n, \overline{z}_1, \ldots, \overline{z}_n\}$.

Let us build a graph G^ℓ_ϕ with two specified vertices x and y of degree 2. There will be a hole containing x and y in G_ϕ if and only if there exists a truth assignment satisfying ϕ.

For each literal y_j ($j = 1, \ldots, 3m$), prepare a graph $G(y_j)$ on 20 named vertices $\alpha, \alpha', \alpha^{1+}, \ldots, \alpha^{4+}, \alpha^{1-}, \ldots, \alpha^{4-}, \beta, \beta', \beta^{1+}, \ldots, \beta^{4+}, \beta^{1-}, \ldots, \beta^{4-}$, together with certain paths in between using unnamed vertices, as drawn in Figure 6. (We drop the subscript j in the labels of the vertices for clarity.)

For $i = 1, 2, 3$ add paths of length ℓ between α^{i+} and $\alpha^{(i+1)+}$; α^{i-} and $\alpha^{(i+1)-}$; β^{i+} and $\beta^{(i+1)+}$; and β^{i-} and $\beta^{(i+1)-}$. Also add the edges $\alpha^{1+}\beta^{1-}$, $\alpha^{1-}\beta^{1+}$, $\alpha^{4+}\beta^{4-}$, $\alpha^{4-}\beta^{4+}$, $\alpha\alpha^{1+}$, $\alpha\alpha^{1-}$, $\alpha^{4+}\alpha'$, $\alpha^{4-}\alpha'$, $\beta\beta^{1+}$, $\beta\beta^{1-}$, $\beta^{4+}\beta'$, $\beta^{4-}\beta'$.

For each clause C_j ($j = 1, \ldots, m$), prepare a graph $G(C_j)$ with 10 named vertices $c^{1+}, c^{2+}, c^{3+}, c^{1-}, c^{2-}, c^{3-}, c^{0+}, c^{12+}, c^{0-}, c^{12-}$, together with certain paths in between using unnamed vertices, as drawn in Figure 7. (We drop the subscript j in the labels of the vertices for clarity.) Add paths of length ℓ between the following pairs of vertices: c^{12+} and c^{1+}; c^{12+} and c^{2+}; c^{12-} and c^{1-}; c^{12-} and c^{2-}; c^{0+} and c^{12+}; c^{0+} and c^{3+}; c^{0-} and c^{12-}; c^{0-} and c^{3-}.

For each variable z_i .($i = 1, \ldots, n$), prepare a graph $G(z_i)$ with $2z_i^- + 2z_i^+$ vertices, where

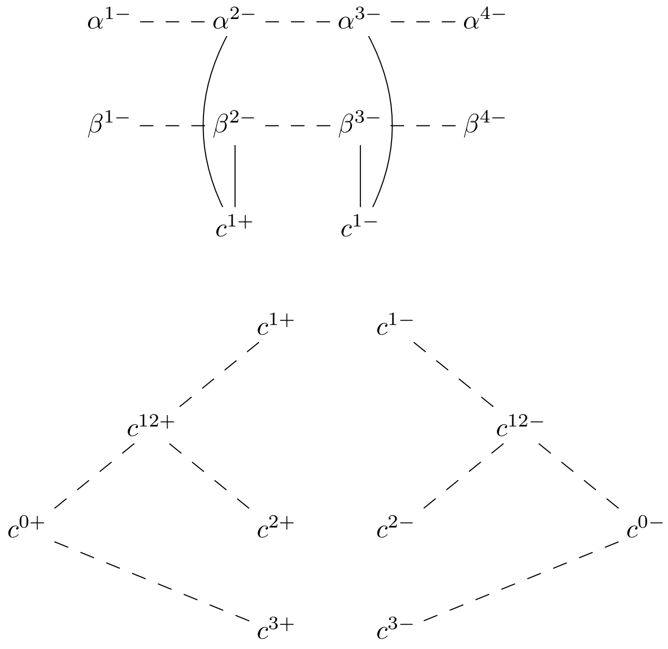

Figure 7 The clause gadget together with its interface with the literal gadget (drawn above). Dashed lines indicate paths of length ℓ.

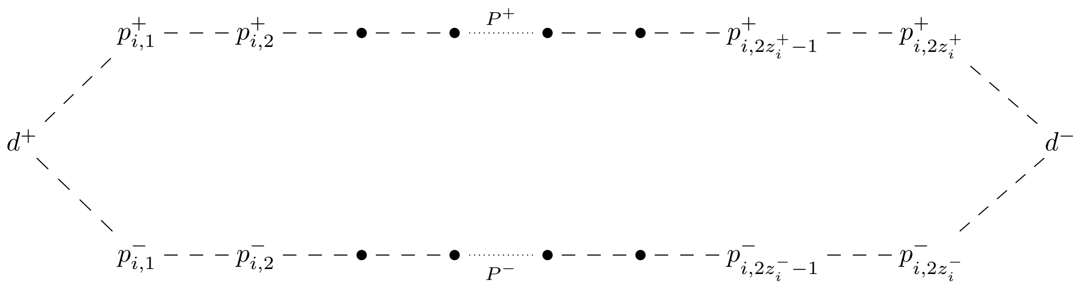

Figure 8 The variable gadget. Dashed lines indicate paths of length ℓ. Dotted lines indicate a continuation of the gadget.

z_i^- is the number of times $\overline{z}_i$ appears in clauses $C_1, \ldots, C_m$ and z_i^+ is the number of times z_i appears in clauses $C_1, \ldots, C_m$.

Let $G(z_i)$ consist of two internally disjoint paths P_i^+ and P_i^- with common endpoints d_i^+ and d_i^- and lengths $1 + (2\ell)z_i^-$ and $1 + (2\ell)z_i^+$, respectively. Label the vertices of P_i^+ and P_i^- as in Figure 8.

The final graph G_ϕ^ℓ will be constructed from the disjoint union of all the graphs $G(y_j)$, $G(C_i)$, and $G(z_i)$ with the following modifications:

- For $j = 1, \ldots, 3m-1$, add paths of length ℓ between the pairs: α'_j and α_{j+1}; β'_j and β_{j+1}.
- For $j = 1, \ldots, m-1$, add a path of length ℓ between c_j^{0-} and c_{j+1}^{0+}.
- For $j = 1, \ldots, n-1$, add a path of length ℓ between d_i^- and d_{i+1}^+.
- For $i = 1, \ldots, n-1$, let $y_{n_1}, \ldots, y_{n_{z_i^-}}$ be the occurrences of $\overline{z}_i$ over all literals. For $j = 1, \ldots, z_i^-$, delete the path between $p_{i,2j-1}^+$ and $p_{i,2j}^+$ and add the four edges $p_{i,2j-1}^+\alpha_{n_j}^{2+}$, $p_{i,2j-1}^+\beta_{n_j}^{2+}$, $p_{i,2j}^+\alpha_{n_j}^{3+}$, $p_{i,2j}^+\beta_{n_j}^{3+}$.
- For $i = 1, \ldots, n-1$, let $y_{n_1}, \ldots, y_{n_{z_i^+}}$ be the occurrences of z_i over all literals. For

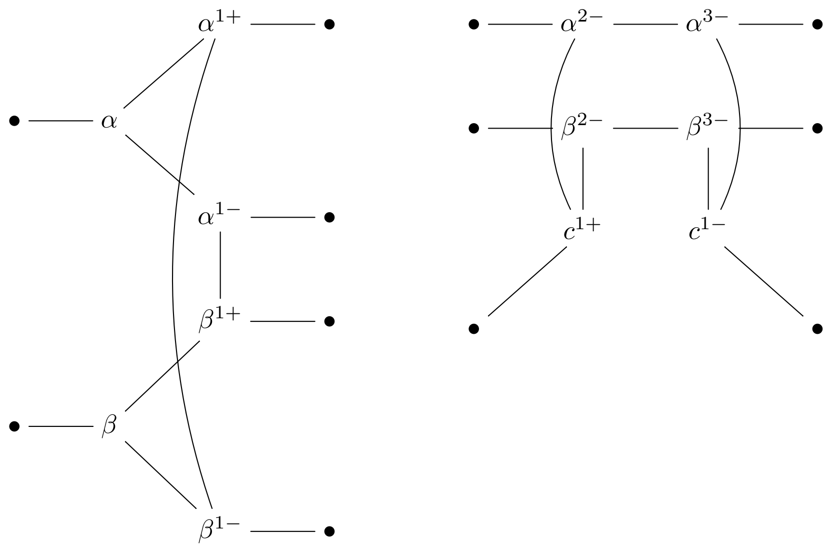

Figure 9 Cases that need to be checked for omission of graphs H_ℓ.

$j = 1, \dots, z_i^+$, delete the path between $p^-_{i,2j-1}$ and $p^-_{i,2j}$ and add the four edges $p^-_{i,2j-1}\alpha^{2+}_{n_j}$, $p^-_{i,2j-1}\beta^{2+}_{n_j}$, $p^-_{i,2j}\alpha^{3+}_{n_j}$, $p^-_{i,2j}\beta^{3+}_{n_j}$.

- For $i = 1, \dots, m$ and $j = 1, 2, 3$, add the edges $\alpha^{2-}_{3(i-1)+j}c_i^{j+}$, $\alpha^{3-}_{3(i-1)+j}c_i^{j-}$, $\beta^{2-}_{3(i-1)+j}c_i^{j+}$, $\beta^{3-}_{3(i-1)+j}c_i^{j-}$.
- Add a path of length ℓ between the pairs of vertices: $\alpha'_{3m}d_1^+$ and d_1^+; $\beta'_{3m}d_1^+$ and c_1^{0+}.
- Add the vertex x and add paths of length ℓ between the pairs of vertices: x and α_1; x and β_1.
- Add the vertex y and add paths of length ℓ between the pairs of vertices: y and c_m^{0-}; y and d_n^-.

It is easy to verify that the maximum degree of G_ϕ^ℓ is 3, that it is polynomial (actually linear) in the size $n + m$ of ϕ, and that x and y are non-adjacent and both have degree two.

▶ **Lemma 11.** *Let $\ell \geq 5$. G_ϕ^ℓ omits as a subgraph $H_4, \dots, H_{\ell-1}$.*

Proof. Owing to the length of the ℓ-paths that populate our construction, we need only verify the omission of these graphs on the connected components of the graph G_ϕ^ℓ after the removal of these paths (except a pendant edge from the corresponding connected component at the extremities of an instance of these paths). In this fashion, we only need to check for omission of the given graphs in the non-trivial cases drawn in Figure 9.

Indeed, the two cases are isomorphic. Let $i = 4, 5$. Any two vertices of degree at least three that are separated by a path of length i must be in the subgraph C_6 at distance $6 - i$ from one another. If $i = 4$ then these vertices have a common neighbour so the H_i can't be completed. If $i = 5$ then these two vertices are adjacent. For $6 \leq i \leq \ell - i$ it is not possible to find two vertices of degree at least thre that are separated by a path of length ℓ. ◀

Note that G_ϕ^ℓ contains H_1, H_2 and H_3.

▶ **Lemma 12.** *ϕ is satisfied by a truth assignment if and only if G_ϕ^ℓ contains a hole passing through x and y.*

Proof. First assume that ϕ is satisfied by a truth assignment $\xi \in \{0, 1\}^n$. We will 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 the vertices $\alpha_i, \alpha'_i, \beta_i, \beta'_i$.
3. For $i = 1, \ldots, 3m$, if y_i is satisfied by ξ , then pick the vertices $\alpha_i^{1+}, \alpha_i^{2+}, \alpha_i^{3+}, \alpha_i^{4+}, \beta_i^{1+}, \beta_i^{2+}, \beta_i^{3+}, \beta_i^{4+}$. Otherwise, pick the vertices $\alpha_i^{1-}, \alpha_i^{2-}, \alpha_i^{3-}, \alpha_i^{4-}, \beta_i^{1-}, \beta_i^{2-}, \beta_i^{3-}, \beta_i^{4-}$.
4. For $i = 1, \ldots, n$, if $\xi(i) = 1$, then pick all the vertices of the path P_i^+ and all the neighbours of the vertices in P_i^+ of the form α_k^{2+} or α_k^{3+} for any k.
5. For $i = 1, \ldots, n$, if $\xi(i) = 0$, then pick all the vertices of the path P_i^- and all the neighbours of the vertices in P_i^- of the form α_k^{2+} or α_k^{3+} for any k.
6. For $i = 1, \ldots, m$, 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 the vertices $c_j^{12+}, c_j^{1+}, c_j^{12-}, c_j^{1-}$. If $j = 3i-1$, then pick the vertices $c_j^{12+}, c_j^{2+}, c_j^{12-}, c_j^{2-}$. If $j = 3i$, then pick the vertices c_j^{3+}, c_j^{3-}.

The given vertices do not yet induce a connected component, because we need to add the vertices of ℓ-paths in between. Thus, if p and q are vertices which we selected that have an ℓ-path between them (drawn as a dashed edge in the associated gadget), then we need to add the interior vertices of this path also.

It suffices to show that the chosen vertices induce a hole 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 and α_k^{2-} and α_k^{3-} were not picked in Step 3. Thus, the chosen vertices induce a hole in G_ϕ^ℓ containing vertices x and y.

Now assume G_ϕ^ℓ contains a hole H passing through x and y. The hole H must contain α_1 and β_1, and the paths leading to them, since they are the only two path neighbours of x. Next, either both α_1^{1+} and β_1^{1+} are in H or both α_1^{1-} and β_1^{1-} are in H.

Without loss of generality, let α_1^{1+} and β^{1+} be in H (the same reasoning that follows will hold true for the other case). Since α_1^{1-} and β^{1-} are both neighbours of two members of H, they cannot be in H. Thus, α_1^{2+} and β_1^{2+}, and the paths to them, must be in H. Since α_1^{2+} and β_1^{2+} have the same neighbours outside $G(y_1)$, it follows that H must contain α_1^{3+} and β_1^{3+}, and the paths that lead to them. Also, H must contain α_1^{4+} and β_1^{4+}, and the paths that lead to them. Suppose that α_1^{4-} and β_1^{4-} are in H. Because α_1^{i-} has the same neighbour as β_1^{i-} outside $G(y_1)$ for $i = 2, 3$, it follows that H must contain α_1^{3-}, α_1^{2-}, α_1^{1-}. But then H is not a hole containing x, a contradiction. Therefore, α_1^{4-} and β_1^{4-} cannot both be in H, so H must contain α'_1, β'_1, α_2, β_2, and the paths to them.

By induction, we see for $i = 1, 2, \ldots, 3m$ that H must contain $\alpha_i, \alpha'_i, \beta_i, \beta'_i$. Also, for each i, either H contains $\alpha_i^{1+}, \alpha_i^{2+}, \alpha_i^{3+}, \alpha_i^{4+}, \beta_i^{1+}, \beta_i^{2+}, \beta_i^{3+}, \beta_i^{4+}$ or H contains $\alpha_i^{1-}, \alpha_i^{2-}, \alpha_i^{3-}, \alpha_i^{4-}, \beta_i^{1-}, \beta_i^{2-}, \beta_i^{3-}, \beta_i^{4-}$.

As a result, H_ϕ^ℓ must also contain d_1^+ and c_1^{0+} and the paths to them. By symmetry, we may assume H_ϕ^ℓ contains $p_{1,1}^+$ and α_k^{2+}, for some k. Since α_k^{1+} is adjacent to two vertices in H, H must contain α_k^{3+} and the path of length ℓ toward it. Similarly, H cannot contain α_k^{4+}, so H contains $p_{1,2}^+$ and $p_{1,3}^+$, as well as the paths through these. By induction, we see that H contains $p_{1,i}^+$ for $i = 1, 2, \ldots, z_i^+$ and d_1^- and the ℓ-paths in between. If H contains $p_{1,z_i^-}^-$, then H must contain $p_{1,i}^-$ for $i = z_i^-, \ldots, 1$, a contradiction. Thus, H must contain d_2^+ and the ℓ-path to it. By induction, for $i = 1, 2, \ldots, n$, we see that H contains all the vertices of the path P_i^+ or P_i^- and by symmetry, we may assume H contains all the neighbours of the vertices in P_i^+ or P_i^- of the form α_k^{2+} or α_k^{3+}, for any k.

Similarly, for $i = 1, 2, \ldots, m$, it follows that H must contain c_i^{0+} and c_i^{0-}. Also, H contains one of the following:

- $c_i^{12+}, c_i^{1+}, c_i^{12-}, c_i^{1-}$ and either α_j^{2-} and α_j^{3-} or β_j^{2-} and β_j^{3-} (where α_j^{2-} is adjacent to c_i^{1+}).
- $c_i^{12+}, c_i^{2+}, c_i^{12-}, c_i^{2-}$ and either α_j^{2-} and α_j^{3-} or β_j^{2-} and β_j^{3-} (where α_j^{2-} is adjacent to c_i^{2+}).
- c_i^{3+}, c_i^{3-} and either α_j^{2-} and α_j^{3-} or β_j^{2-} and β_j^{3-} (where α_j^{2-} is adjacent to c_i^{3+}).

We can recover the satisfying assignment ξ as follows. For $i = 1, 2, \ldots, n$, set $\xi(i) = 1$ if the vertices of P_i^+ are in H and set $\xi(i) = 0$ if the vertices of P_i^- are in H. By construction, it is easy to verify that at least one literal in every clause is satisfied, so ξ is indeed a satisfying assignment. ◀

We are now in a position to prove Theorem 3. We need to borrow one lemma (whose proof is straightforward) from [11], and for which we need to define the problem 2-Induced Cycle. This has as input a graph with two labelled vertices, with yes-instances those inputs where there exists an induced cycle (hole) containing those two labelled vertices.

▶ **Lemma 13** (See Theorem 2.7 in [11]). *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'.*

Proof of Theorem 3. The first part for H_1 appears as Theorem 6 while for H_2 it appears as Corollary 10. Note that the H_1 case is readily seen once the simplification (†) is made, because k-Disjoint Paths must solve k-Induced Disjoint Paths since the input is H_1-subgraph-free.

For the second part, we reason via Lemma 13. We construct G_ϕ^ℓ. By Lemma 12, G_ϕ^ℓ has a hole through x and y if and only if ϕ is satisfiable. Moreover, G_ϕ^ℓ is $(H_4, \ldots, H_{\ell-1})$-subgraph-free by Lemma 11. ◀

3 C_5-Colouring

A *homomorphism* between graphs G and H is a function f from $V(G)$ to $V(H)$ so that, for all $xy \in E(G)$ we have $f(x)f(y) \in E(H)$. Let us recall our problem.

C_5-Colouring	
Instance:	a graph G.
Question:	Does G have a homomorphism to the cycle C_5?

▶ **Lemma 14.** *There exists n_1 so that for all $N \geq n_1$, and for all $x, y \in V(C_5)$, there is a walk of length N in C_5 from x to y.*

Proof. We may take $n_1 = 4$.

Consider $N = 4$. To walk a distance of zero: walk two forward then two back. To walk at distance one (w.l.o.g.) forward: walk four backward. To walk at distance two (w.l.o.g.) forward: walk one back, one forward, and two forward.

Consider $N = 5$. To walk a distance of zero: walk five forward. To walk at distance one (w.l.o.g.) forward: walk two forward, two back and one forward. To walk at distance two (w.l.o.g.) forward: walk one back, one forward, and three back.

Consider $N \geq 6$. Keep moving one forward then one back until one of the two previous cases applies. ◀

▶ **Corollary 15.** *Let G be an instance of C_5-*Colouring *and let G' be the same instances after $n_1 - 1$ subdivisions. Then G' is a trivial yes-instance of C_5-*Colouring*.*

▶ **Corollary 16.** *C_5-Colouring fails C3.*

Let us note that C_5-Colouring fulfills C1 and C2.

▶ **Lemma 17.** *C_5-Colouring is* NP*-complete for (H_1, H_2)-subgraph-free graphs.*

Proof. It is well-known [7] and easy to see that there is a reduction from K_5-colouring to C_5-Colouring that takes an input G and simply subdivides twice each edge. The obtained graph plainly omits both H_1 and H_2 as a subgraph (but generally contains many instances of H_3). ◀

We are now in a position to prove Theorem 4.

Proof of Theorem 4. The first part comes from Lemma 14 with $n_1 = 4$. The point is that any instance which omits each of H_1, H_2 and H_3 as a subgraph must be trivially true, because all paths between vertices of degree at least three are of length at least four. This means that vertices of degree at least three can be mapped anywhere on C_5 and the instance can still be extended to a C_5-colouring.

The second part appears as Lemma 17. ◀

4 Hamilton Cycle

Recall that a Hamilton Cycle in a graph is one which visits every node exactly once. Let us recall our problem.

Hamilton Cycle	
Instance:	a graph G.
Question:	Does G contain a Hamilton Cycle?

This section is devoted to the proof of Theorem 5, whose statement we recall.

Theorem 5. Hamilton Cycle is in P for the class of H_1-subgraph-free graphs.

Proof. Let G be an input to Hamilton Cycle. If G is not connected, it is a no-instance. Else, if G has no vertex of degree > 2, then G is a yes-instance iff it is 2-regular (a cycle). Let v be a vertex of degree > 2. If v has a neighbour of degree 1, then G is a no-instance. If all neighbours of v have degree 2, then we are in the situation depicted in Figure 10, in which we draw the neighbours of v as u_i ($i = 1, 2, 3, \ldots$). Since any Hamilton Cycle that accesses u_i must involve v, for $i = 1, 2, 3$, we derive a contradiction, as we can only come to v once and leave v once. Thus, G must be a no-instance of Hamilton Cycle (note that G need not be a no-instance of Hamilton Path). Let u be a neighbour of v of degree > 2. Consider that u has two neighbours distinct from v, let us call them p, q and v has two neighbours distinct from u, let us call them r, s. Since G is H_1-free, it is not possible that $\{p, q\} \cap \{r, s\} = \emptyset$. Let us branch on two possibilities.

(Diamond case.) Suppose $\{p, q\} = \{r, s\}$ and G contains an induced diamond or K_4. If $\{u, v, p, q\}$ have no neighbours outside of $\{u, v, p, q\}$, then G is a yes-instance (G itself is either a diamond or K_4). If either of the following pairs have distinct neighbours outside of $\{u, v, p, q\}$ then G contains an H_1: $\{p, u\}$, $\{u, q\}$, $\{q, v\}$, $\{v, p\}$.

Suppose u and v have no neighbours outside of $\{u, v, p, q\}$ except that are also neighbours of one of $\{p, q\}$. And now suppose one of p and q has a neighbour outside of $\{u, v, p, q\}$ and the other doesn't. Without loss of generality, suppose it is p.

Suppose there is some vertex adjacent to u or v or both, outside of $\{u, v, p, q\}$, and note that such a vertex must also be adjacent to p. There can be no more than one such vertex as otherwise we have an H_1. If t has a neighbour outside of $\{u, v, p, q\}$ then we have an H_1. So, t has no such neighbour and in fact G has precisely vertices $\{u, v, p, q, t\}$ and is a yes-instance.

Thus, we may assume that there is no vertex adjacent to u or v outside of $\{u, v, p, q\}$. This means G is a no-instance as p or q (whichever has the neighbour outside of $\{u, v, p, q\}$) may only be traversed once.

Now let us assume that p and q have distinct neighbours x and y outside of $\{u, v, p, q\}$. Note that each of them has a unique neighbour (else G contains an H_1). Now, let us build G' from G by contracting $\{u, v, p, q\}$ to a single vertex. We draw this case in Figure 11. We claim G' has a Hamilton Cycle iff G has a Hamilton Cycle.

(Proof of Claim.) The forward direction is trivial. For the backward direction, note that once any Hamilton Cycle visits any of $\{u, v, p, q\}$, necessarily from x or y, then it must visit them all in immediate succession, before leaving via whichever of x and y it didn't enter by.

Suppose now that one or more of u and v have neighbours outside of $\{u, v, p, q\}$ that are not neighbours of $\{p, q\}$. It follows that p and q have no neighbours outside of $\{u, v, p, q\}$. If $\{u, v, p, q\}$ induces a K_4, then we can build G' from G by contracting $\{u, v, p, q\}$ to a single vertex. That G' has a Hamilton Cycle iff G has a Hamilton Cycle follows exactly as in the previous claim (indeed, if we swap $\{p, q\}$ for $\{u, v\}$ we are in the previous case). If $\{u, v, p, q\}$ induces a diamond, then G is a no-instance (we draw this case in Figure 12).

(Bull case.) Suppose $p = r$ but $q \neq s$ and G contains a bull with triangle $\{u, v, p\}$ and pendant edges uq and vs. If there is an edge us or vq (or ps or pq) then we have a diamond and we are in a previous case. (There may or may not be the edge qs.) If u or v has degree > 3 then there is an H_1 so let us assume they have degree exactly 3.

Suppose p has degree 2, then we can contract $\{u, v, p\}$ to a single vertex. We claim that G' has a Hamilton Cycle iff G has a Hamilton Cycle.

(Proof of Claim.) The forward direction is trivial. For the backward direction, note that once any Hamilton Cycle visits any of $\{u, v\}$, necessarily from q or s, then it must visit all of $\{u, p, v\}$ in immediate succession, before leaving via whichever of q and s it didn't enter by.

Suppose p has degree > 2 and note that is must be ≤ 3 to avoid an H_1 so we may assume p has precisely one neighbour outside of $\{u, v\}$ which we will call t. If t has degree 1 then this is a no-instance. If t has degree > 2 then there is an H_1 (Recall that there can be no edge from t to u or v as this would introduce a diamond. It is possible there is an edge from t to q or s). Thus, t has degree 2. Suppose one among q and s has no neighbour outside of $\{u, v, q, s, t\}$. Note that there is no edge qv or su as this would create a diamond. If there is no edge qs this is a no-instance. If there is an edge qs then we can replace G by G' in which we contract $\{u, v, p, t\}$ to a single vertex. We claim that G' has a Hamilton Cycle iff G has a Hamilton Cycle.

(Proof of Claim.) Assume w.l.o.g. that q has no neighbour outside of $\{u, v, q, s, t\}$. For the forward direction we may traverse in the order s, q, u, v, p, t. For the backward direction, note that once any Hamilton Cycle visits t from outside of $\{u, v, q, s, t\}$, then it must visit all of $\{u, v, q, s, t\}$ in immediate succession, before leaving via s.

Suppose now that both q and s have a neighbour outside of $\{u, v, q, s\}$. In this case, there can be no edge qs, as this would introduce an H_1. We claim that G is a no-instance. We draw this case in Figure 13.

(Proof of Claim.) Since any Hamilton Cycle that accesses any one of $\{q, s, t\}$ must involve two among $\{u, v, p\}$, we derive a contradiction, as we can only come to each from $\{u, v, p\}$

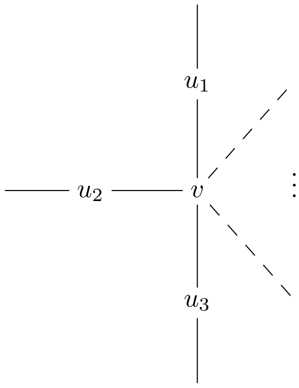

Figure 10 The case in which v has degree ≥ 3 and all its neighours have degree 2.

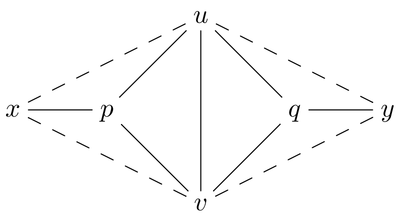

Figure 11 The case in which $\{u, v, p, q\}$ induces a diamond or K_4 and p and q are of degree 3. The dashed lines are edges that may or may not be present.

once. Thus, G must be a no-instance of Hamilton Cycle. ◀

5 Final Remarks

It is well known that H-Colouring is polynomial-time solvable whenever H is a bipartite graph [7]. We can generalise our results from C_5-Colouring to C_{2k+1}-Colouring, but we will be less clear about some of the bounds.

C_{2k+1}-Colouring	
Instance:	a graph G.
Question:	Does G have a homomorphism to the cycle C_{2k+1}?

We omit the proofs of the following results.

▶ **Lemma 18.** *For each $k \geq 2$, there exists n_k so that for all $N \geq n_k$, and for all $x, y \in V(C_{2k+1})$, there is a walk of length N in C_{2k+1} from x to y.*

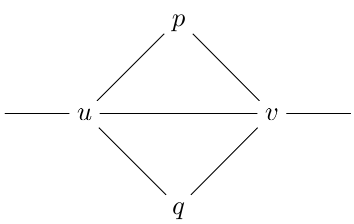

Figure 12 The case in which $\{u, v, p, q\}$ induces a diamond and p and q are of degree 2.

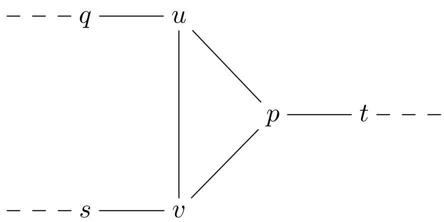

Figure 13 The case in which p has precisely one neighbour outside of $\{u, v\}$ which we will call t

▶ **Corollary 19.** *Let G be an instance of C_{2k+1}-Colouring and let G' be the same instances after $n_k - 1$ subdivisions. Then G' is a trivial yes-instance of C_{2k+1}-Colouring.*

▶ **Corollary 20.** *C_{2k+1}-Colouring fails C3.*

Let us note that C_{2k+1}-Colouring fulfills C1 and C2.

▶ **Lemma 21.** *C_{2k+1}-Colouring is NP-complete for $(H_1, \dots, H_k)$-subgraph-free graphs.*

▶ **Theorem 22.** *C_{2k+1}-Colouring is in P for $(H_1, \dots, H_{n_k-1})$-subgraph-free graphs. C_{2k+1}-Colouring is NP-complete for $(H_1, \dots, H_{k+1})$-subgraph-free graphs.*

Acknowledgements

We are grateful to Matthew Johnson, Jelle J. Oostveen and Hans Bodlaender for useful discussions.

References

1 Vladimir E. Alekseev, Rodica Boliac, Dmitry V. Korobitsyn, and Vadim V. Lozin. NP-hard graph problems and boundary classes of graphs. *Theoretical Computer Science*, 389:219–236, 2007.
2 Vladimir E. Alekseev and Dmitry V. Korobitsyn. Complexity of some problems on hereditary graph classes. *Diskretnaya Matematika*, 2:90–96, 1990.
3 Rodica Boliac and Vadim V. Lozin. On the clique-width of graphs in hereditary classes. *Proc. ISAAC 2022, LNCS*, 2518:44–54, 2002.
4 Maria Chudnovsky, Shenwei Huang, Pawel Rzazewski, Sophie Spirkl, and Mingxian Zhong. Complexity of C_k-coloring in hereditary classes of graphs. *Proc. ESA 2019, LIPIcs*, 144:31:1–31:15, 2019.
5 Petr A. Golovach and Daniël Paulusma. List coloring in the absence of two subgraphs. *Discrete Applied Mathematics*, 166:123–130, 2014.
6 Petr A. Golovach, Daniël Paulusma, and Bernard Ries. Coloring graphs characterized by a forbidden subgraph. *Discrete Applied Mathematics*, 180:101–110, 2015.
7 Pavol Hell and Jaroslav Nesetril. On the complexity of H-coloring. *Journal of Combinatorial Theory, Series B*, 48:92–110, 1990.
8 Matthew Johnson, Barnaby Martin, Jelle J. Oostveen, Sukanya Pandey, Daniël Paulusma, Siani Smith, and Erik Jan van Leeuwen. Complexity framework for forbidden subgraphs. *CoRR*, 2211.12887, 2022.
9 Walter Kern, Barnaby Martin, Daniël Paulusma, Siani Smith, and Erik Jan van Leeuwen. Disjoint paths and connected subgraphs for H-free graphs. *Theoretical Computer Science*, 898:59–68, 2022.
10 Walter Kern and Daniël Paulusma. Contracting to a longest path in H-free graphs. *Proc. ISAAC 2020, LIPIcs*, 181:22:1–22:18, 2020.

11 Benjamin Lévêque, David Y. Lin, Frédéric Maffray, and Nicolas Trotignon. Detecting induced subgraphs. *Discrete Applied Mathematics*, 157:3540–3551, 2009.
12 Barnaby Martin, Daniël Paulusma, Siani Smith, and Erik Jan van Leeuwen. Induced Disjoint Paths and Connected Subgraphs for H-free graphs. *Proc. WG 2022, LNCS*, 13453:398–411, 2022.
13 Barnaby Martin, Daniël Paulusma, Siani Smith, and Erik Jan van Leeuwen. Few induced disjoint paths for H-free graphs. *Theoretical Computer Science*, 939:82–193, 2023.
14 Neil Robertson and Paul D. Seymour. Graph minors .XIII. The Disjoint Paths problem. *Journal of Combinatorial Theory, Series B*, 63:65–110, 1995.
15 Yossi Shiloach. A polynomial solution to the undirected Two Paths problem. *Journal of the ACM*, 27:445–456, 1980.