Finding induced paths of given parity in claw-free graphs*

Pim van 't Hof[1], Marcin Kamiński[2] and Daniël Paulusma[1]

[1]Department of Computer Science, University of Durham,
Science Laboratories, South Road, Durham DH1 3LE, England.
{pim.vanthof,daniel.paulusma}@durham.ac.uk
[2]Computer Science Department, Université Libre de Bruxelles,
Boulevard du Triomphe CP212, B-1050 Brussels, Belgium
marcin.kaminski@ulb.ac.be

Abstract. The Parity Path problem is to decide if a given graph G contains both an even length and an odd length induced path between two specified vertices s and t. Although the Parity Path problem is NP-complete in general, it has been shown to be solvable in polynomial time for several classes of graphs. In the related Odd Path problem, the goal is to determine whether an induced path of odd length between two specified vertices exists, and if so, to find one. We present a polynomial-time algorithm for the Odd Path problem for the class of claw-free graphs. A slight modification of our algorithm also allows us to solve the Even Path problem, and as a consequence the Parity Path problem, for claw-free graphs in polynomial time.

1 Introduction

Finding a shortest path, a maximum stable set or a hamiltonian cycle in a graph are just a few examples of the wide spectrum of problems dealing with finding a subgraph (or induced subgraph) with some particular property. In this context, simplest subgraphs, such as paths, trees and cycles, with some prescribed property are often studied. The following problem has been extensively studied in the context of perfect graphs. (The *length* of a path refers to its number of edges, and a path is said to be *odd* (respectively *even*) it has odd (respectively even) length.)

Parity Path
Instance: A graph G and two vertices s, t of G.
Question: Does there exist an induced path of odd length and an induced path of even length between s and t in G?

In this paper we focus on the closely related problem of finding an induced path of given parity between a pair of vertices. In particular, we study the following problem.

Odd Path
Instance: A graph G and two vertices s, t of G.
Question: Decide whether there exists an induced path of odd length from s to t in G, and if so, find one.

* This work has been supported by EPSRC (EP/D053633/1) and the Actions de Recherche Concertées (ARC) fund of the Communauté française de Belgique.

The Odd Path problem was shown to be NP-hard by Bienstock [5]. Several authors however have identified a number of graph classes which admit polynomial-time algorithms. Below we survey those results, as well as results on the Parity Path problem and related problems, before stating our results.

Not necessarily induced paths. The problem of finding a (not necessarily induced) path of given parity between a pair of vertices was considered by LaPaugh and Papadimitriou [15]. They mention a solution of the problem due to Edmonds, using a reduction to matching, and propose a faster and more elementary one of linear time complexity. However, the advantage of the original approach by Edmonds is that it can find the shortest induced odd and/or even path, even in a weighted graph. Interestingly, as they also show in the paper, the problem of finding a directed path of given parity is NP-complete for directed graphs.

Papadimitriou with other authors [1] generalized the result of [15] and designed a linear-time algorithm deciding if all (not necessarily induced) paths between two specified vertices are of length $P \bmod Q$, for some integers P and Q.

Perfect graphs. First interest in induced paths of given parity comes from the theory of perfect graphs. Two non-adjacent vertices are called an *even pair* if every induced path between them is even. The interest in even pairs was sparked by an observation of Fonlupt and Uhry [13]: if a graph is perfect and contains an even pair, then the graph obtained by identifying the vertices which form the even pair is also perfect. Later Meyniel showed that minimal non-perfect graphs contain no even pair [17]. Those two facts triggered a series of theoretical and algorithmic results which are surveyed in [10] and its updated version [11].

Parity Path and Group Path. Arikati, in a series of papers with different coauthors, developed polynomial-time algorithms for the Parity Path problem in different classes of graphs. Chordal graphs are considered in [2], where the authors present a linear-time algorithm for the Group Path problem, a generalization of the Odd Path problem. In the Group Path problem edges of the input graph are weighted with elements of some group $\mathcal{G}$ and the problem is to find induced path of given weigth. (The weight of the path is the sum of weight of the edges along the path.) They present an $\mathcal{O}(|\mathcal{G}|m+n)$ algorithm for the Group Path problem on chordal graphs using a perfect elimination ordering.

The topic of [4] is Parity Path on circular-arc graphs. The authors show how to reduce the problem to interval graphs by recursively applying a set of reductions. Since interval graphs are chordal, the algorithm of [2] can be used to obtain the solution. This way they obtain an $\mathcal{O}(n \cdot m)$ algorithm for circular-arc graphs and an $\mathcal{O}(n+m)$ algorithm for proper circular-arc graphs. In [21] polynomial-time algorithms for the Parity Path problem on comparability and cocomparability graphs, and a linear-time algorithm for permutation graphs are given. A polynomial-time algorithm for Parity Path on perfectly orientable graphs is presented in [3]. Sampaio and Sales [19] obtain a polynomial-time algorithm for planar perfect graphs. The authors of [12] characterize even and odd pairs in comparability and P_4-comparability graphs and give polynomial-time algorithms for the Parity Path problem in those classes.

Our results. Our interest in the Odd Path problem was in part stirred by studying Bienstock's NP-completeness reduction in [5]. He builds a graph out of a 3-Sat formula and shows that the formula is satisfiable if and only if there exists an odd induced path between a certain pair of vertices. This is also shown to be equivalent to the existence of two disjoint induced paths (with no edges between the two paths) between certain pairs of vertices in the construction. Finding such two paths is then NP-hard in general but has been proved solvable in polynomial time for claw-free graphs [16]. A natural question to ask is whether the Odd Path problem can also be solved in polynomial time for this class of graphs. In this paper, we answer this question in the affirmative.

As we saw earlier in this section, the Parity Path problem has been extensively studied in different graph classes. However, a polynomial-time algorithm for claw-free graphs has never been proposed; somewhat surprising, since claw-free graphs form a large and important class. We show that a slight modification of our algorithm for the Odd Path problem allows us to solve the Even Path problem, and as a consequence the Parity Path problem, for claw-free graphs in polynomial time. In addition, we can decide in polynomial time whether a claw-free graph contains an odd (or even) hole passing through a given vertex, and if so, find one. (A *hole* is an induced cycle on at least 4 vertices.) This problem was also proved to be NP-hard for general graphs by Bienstock [5].

2 Finding an odd induced path in a claw-free graph

All graphs in this paper are undirected, finite, and have no loops or multiple edges. We refer to [8] for terminology not defined below.

In this section, we present an algorithm that decides in polynomial time whether a claw-free graph has an odd induced path between two specified vertices s and t, and finds such a path if one exists. Our algorithm first preprocesses the input graph and creates a number of smaller instances with certain desirable properties (Section 2.1). Then the algorithm solves Odd Path for each of the smaller graphs and returns an affirmative answer to the original problem if and only if (at least) one of the smaller graphs yields an affirmative answer to the corresponding problem. If one of the small graphs is non-perfect, then we show that it is a Yes-instance (Section 2.2). Otherwise, we use a structural result on perfect claw-free graphs to obtain a solution in this case (Section 2.3).

2.1 Preprocessing the input graph

Let $G = (V, E)$ be a claw-free graph and let G, together with two of its vertices s and t, form an instance of the Odd Path problem.

Step 1. We create a set $\mathcal{G}$ of induced subgraphs of G as follows: for each pair of neighbors s' of s and t' of t, we add the graph $G[V \backslash (N_G(s) \cup N_G(t)) \cup \{s', t'\}]$ to $\mathcal{G}$. This way we obtain a set of $\mathcal{O}(|V|^2)$ instances in which s and t are of degree 1; this convenient property will be used in other steps of our algorithm. Since any induced path from s to t in G contains exactly one vertex of $N_G(s)$ and exactly one vertex of $N_G(t)$, it is obvious that G is a Yes-instance of Odd Path if and only if some graph $G' \in \mathcal{G}$ is a Yes-instance. It is clear that we can perform Step 1 in polynomial time.

Step 2. We "clean" each of the graphs in $\mathcal{G}$ by repeatedly removing irrelevant vertices. Let $G' = (V', E')$ be a graph in $\mathcal{G}$. A vertex $v \in V'$ is called *irrelevant (for vertices s and t)* if v does not lie on any induced path from s to t, and we say that G' is *clean* if none of its vertices is irrelevant. Let G'' denote the graph obtained from G' by repeatedly removing vertices that are irrelevant. It is easy to see that, for any irrelevant vertex $v \in V'$, G' is a YES-instance of ODD PATH if and only if $G'[V' \backslash \{v\}]$ is a YES-instance. Hence, G' is a YES-instance of the ODD PATH problem if and only if G'' is a YES-instance.

We now show that we can perform Step 2 in polynomial time by showing that we can identify irrelevant vertices in polynomial time. We do this by applying a result due to Chudnovsky and Seymour [7] on the THREE-IN-A-TREE problem, which can be stated as follows.

THREE-IN-A-TREE
Instance: A graph G and three vertices v_1, v_2, v_3 of G.
Question: Does there exist an induced subgraph of G which is a tree containing v_1, v_2 and v_3?

Theorem 1 ([7]). *The* THREE-IN-A-TREE *problem is solvable in polynomial time.*

Lemma 1. *The problem of deciding whether a vertex v of a claw-free graph G is irrelevant for two vertices s and t of degree 1 in G can be solved in polynomial time.*

Proof. We claim that there exists an induced path in G between s and t containing v if and only if G together with s, t, v is a YES-instance of the THREE-IN-A-TREE problem. By Theorem 1, this proves that we can decide in polynomial time if v is irrelevant.

If there exists a path in G between s and t containing v, then the path is an induced tree containing all three vertices. Now suppose that there exists an induced subgraph of G which is a tree containing s, t and v. Notice that any induced subgraph of a claw-free graph which is a tree is in fact a path, and since vertices s and t are of degree 1, they must be the endpoints of that path. □

After preprocessing the input graph G we have obtained in polynomial time a set $\mathcal{G}$ of $\mathcal{O}(|V|^2)$ subgraphs of G such that G is a YES-instance of ODD PATH if and only if G' is a YES-instance for some graph $G' \in \mathcal{G}$, and such that each graph $G' \in \mathcal{G}$ satisfies the following two conditions: (1) both s and t have degree 1 in G'; (2) G' is clean. For convenience, we assume from now on that the input graph G already satisfied properties (1) and (2), which prevents us from having to refer to the set $\mathcal{G}$ in the rest of the paper.

The algorithm now distinguishes two cases, depending on whether or not G is perfect.

2.2 G is not perfect

Let G be a graph that is clean for two vertices s and t of degree 1, and suppose G is not perfect. Then G contains an odd hole or an odd antihole by virtue of the Strong Perfect Graph Theorem [6]. (An *antihole* is the complement of a hole. The *length* of an antihole is the number of edges in its complement.) The following lemma states that G cannot contain a long odd antihole.

Lemma 2. *Let G be a claw-free graph that is clean for two vertices s and t of degree 1 in G. Then G does not contain an odd antihole of length more than 5.*

Proof. Suppose G contains an odd antihole X such that $\overline{X} = x_1 x_2 \dots x_k x_1$ is an induced cycle of length $k \geq 7$. Since we assume that both s and t have degree one, neither s nor t belongs to X. Let P be an induced path from s to a vertex of X such that $|V(P)|$ is minimum. Note that such a path P exists since G is clean. Without loss of generality assume that $V(P) \cap V(X) = \{x_1\}$. Let s' be the first vertex we encounter when traversing P from x_1 to s. Suppose $s' = s$. Then, as s has degree 1, $\{s, x_1, x_3, x_4\}$ induces a claw in G. Hence s' has besides x another neighbor s'' on P. We note that $s'' = s$ is possible.

Vertex s' is not adjacent to x_2 nor to x_k, since then G would contain a claw induced by $\{s', s'', x_1, x_2\}$ or $\{s', s'', x_1, x_k\}$, respectively. We claim that for $3 \leq i \leq k-2$, s' is adjacent to exactly one of x_i, x_{i+1}, and we prove this claim by contradiction. If s' is adjacent to neither x_i nor x_{i+1}, then the set $\{s', x_1, x_i, x_{i+1}\}$ induces a claw in G. If s' is adjacent to both x_i and x_{i+1}, then the set $\{s, s'', x_i, x_{i+1}\}$ induces a claw. Hence, for $3 \leq i \leq k-1$, s' is either adjacent to every x_i with i even or else to every x_i with i odd. In the first case, this means that s' is not adjacent to x_3 and is adjacent to x_{k-1}. But then the set $\{s', x_2, x_3, x_{k-1}\}$ induces a claw in G. In the second case we find that $\{s', x_3, x_{k-2}, x_{k-1}\}$ induces a claw in G. This finishes the proof of Lemma 2. $\square$

The following lemma implies that G is a Yes-instance of the Odd Path problem if G contains an odd hole.

Lemma 3. *Let G be a claw-free graph that is clean for two vertices s and t of degree 1 in G. If G contains an odd hole, then there exists both an odd and an even induced path from s to t.*

Proof. Let C be an odd hole of G. Let P be an induced path from s to a vertex p of C and let Q be an induced path from t to a vertex q of C, such that there is no edge in G connecting a vertex in $P[V(P)\backslash\{p\}]$ to a vertex in $Q[V(Q)\backslash\{q\}]$ and such that $|V(P)| + |V(Q)|$ is minimum. Note that such paths P and Q exist since G is clean. Let s' and s'' be the first and second vertex we encounter when traversing P from p to s; t' and t'' are defined similarly. We claim that the vertices s'' and t'' indeed exist. Suppose, for contradiction, that s'' (respectively t'') does not exist, i.e., suppose that sp (respectively tq) is an edge in G. Since s (respectively t) has degree 1 in G, the set $\{p, p^-, p^+, s\}$ (respectively $\{q, q^-, q^+, t\}$) induces a claw in G, contradicting the claw-freeness of G. Here, p^- (respectively p^+) denotes the neighbor of p on C when we traverse C in counter-clockwise (respectively clockwise) order.

Claim 1. *Both s' and t' are adjacent to exactly two adjacent vertices of C.*

Suppose p is the only vertex of C that is adjacent to s'. Then the set $\{p, p^-, p^+, s'\}$ induces a claw in G, contradicting the claw-freeness of G. Hence s' must be adjacent to at least two vertices of C. Suppose there exists a set $D \subseteq V(C)$ such that $|D| \geq 3$ and s' is adjacent to every vertex in D. Since C is an induced cycle, we know that D contains two vertices d_1 and d_2 that are not adjacent. Vertex s'' is not adjacent to any vertex of C due to the minimality of $|V(P)|+|V(Q)|$, which means the set $\{s', s'', d_1, d_2\}$ induces a claw in G. This contradiction finishes the proof of Claim 1 for vertex s'. It is clear that the claim also holds for vertex t'.

We assume, without loss of generality, that $N_G(s') \cap V(C) = \{p, p^+\}$ and $N_G(t') \cap V(C) = \{q, q^+\}$. We distinguish three cases. Suppose $|\{p, p^+\} \cap \{q, q^+\}| = 0$. Since the induced path $s'p^+\overrightarrow{C}qt'$ and the induced path $s'p\overleftarrow{C}q^+t'$ have different parity, there exists both an odd and an even induced path from s to t in G. Suppose $|\{p, p^+\} \cap \{q, q^+\}| = 1$. Without loss of generality, suppose $p^+ = q$. Then the path $s'qt'$ is an even induced path from s' to t', and the path $s'p\overleftarrow{C}q^+t'$ is an odd induced path from s' to t'. Since by definition there is no edge connecting a vertex in $P[V(P)\backslash\{p\}]$ to a vertex in $Q[V(Q)\backslash\{q\}]$, this means there exists both an odd and an even induced path from s to t in G. Suppose $|\{p, p^+\} \cap \{q, q^+\}| = 2$. By Claim 1, neither s' nor t' is adjacent to p^-.

Since s' and t' are not adjacent by the choice of P and Q, the set $\{p, p^-, s', t'\}$ induces a claw in G. This contradiction finishes the proof of Lemma 3. $\square$

To prove the main result of this section, we use a result by Parthasarathy and Ravindra [18]. They call a graph *critical* if it is not perfect and if removing any vertex makes the graph perfect.

Theorem 2 ([18]). *If H is a claw-free critical graph, then H is an odd hole or an odd antihole.*

Theorem 3. *The* Odd Path *problem can be solved in polynomial time for the class of non-perfect claw-free graphs that are clean for two specified vertices of degree 1.*

Proof. Let G be a non-perfect claw-free graph that is clean for two vertices s and t of degree 1. By Lemma 2, G does not contain an odd antihole of length more than 5. Since an odd antihole of length 5 is also an odd hole of length 5, G contains an odd hole as a result of the Strong Perfect Graph Theorem. It is clear from the proof of Lemma 3 that to prove Theorem 3 it suffices to show that we can find an odd hole of G in polynomial time. We can do this as follows. We remove a vertex from G and use the perfect claw-free graph recognition algorithm from [9] to check in polynomial time if the obtained graph G' is perfect. If so, we restore the vertex and repeat the procedure on G, removing another vertex. If not, we repeat the whole procedure on the smaller graph G'. This way we find a critical induced subgraph H of G. Combining Lemma 2 with Theorem 2 implies that H is an odd hole, which means we have found an odd hole of G in polynomial time. $\square$

2.3 G is perfect

In this section, we first define the concepts of elementary and peculiar graphs, and we show that we can solve the Odd Path problem in polynomial time for graphs that are either elementary or peculiar. Using a well-known clique cutset decomposition, we then prove that the Odd Path problem is solvable in polynomial time for the class of perfect claw-free graphs that are clean for two specified vertices of degree 1. This leads to the main result of the paper, which states that the Odd Path problem, as well as the Even Path problem and the Parity Path problem, can be solved in polynomial time for the class of claw-free graphs.

Definition 1. *A graph H is* elementary *if its edges can be colored with two colors such that every induced path on three vertices has its two edges colored differently. We call such a coloring an* elementary coloring *of H.*

See Figure 1 for an example of an elementary graph with an elementary coloring, where the light edges are colored 0 and the heavy edges are colored 1. The following lemma is easy to prove (see also e.g. [9]), and we use it to prove Lemma 5.

Lemma 4. *We can determine in polynomial time if a graph H is elementary. If it is, an elementary coloring of H can be found in polynomial time.*

Lemma 5. *The* Odd Path *problem can be solved in polynomial time for every induced subgraph of an elementary graph.*

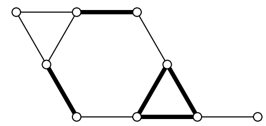

Fig. 1. An elementary graph with an elementary coloring.

Proof. Let H be an induced subgraph of an elementary graph, and let H together with two of its vertices u and v be an instance of the ODD PATH problem. It is easy to see that every induced subgraph of an elementary graph is elementary, which means that H is elementary. If u and v are adjacent, then the observation that the odd induced path uv is the only induced path from u to v trivially solves the ODD PATH problem. Hence we may assume that u and v are not adjacent. Suppose that u and v have a common neighbor w. It is clear that the path uwv is the only induced path from u to v that contains w; in particular, w cannot lie on an odd induced path from u to v. Hence we may delete all common neighbors of u and v before checking whether H contains an odd induced path from u to v using the procedure described below.

We observe that any induced path from u to v in H contains exactly one vertex from $N_H(u)$ and exactly one vertex from $N_H(v)$. We also observe that in any elementary coloring of H, any two consecutive edges of any induced path will be colored differently. Hence if there exists an odd induced path from u to v, then the first and the last edge of that path have the same color. Using these observations, we can check in polynomial time whether H contains an odd induced path from u to v as follows.

We first find an elementary coloring $\varphi : E(H) \to \{0,1\}$ of H; we can find such a coloring φ in polynomial time by Lemma 4. For every pair $u' \in N_H(u)$ and $v' \in N_H(v)$ with $\varphi(uu') = \varphi(vv')$, we act as follows. First we define $H_{u'v'}$ to be the graph obtained from H by deleting the set $(N_H(u) \cup N_H(v) \cup \{u,v\}) \backslash \{u',v'\}$. Note that this graph $H_{u'v'}$ is well-defined, since we may assume that u and v are not adjacent and have no common neighbors. We then check whether there exists a path from u' to v' in $H_{u'v'}$, i.e., whether u' and v' are in the same component of $H_{u'v'}$. If so, then there also exists an induced path P' from u' to v' in $H_{u'v'}$. Adding the vertices u and v as well as the edges uu' and vv' to P' yields an induced path from u to v in H. Since P is induced and φ is an elementary coloring, the colors 0 and 1 alternate on P. Since $\varphi(uu') = \varphi(vv')$, P is an odd induced path from u to v in H. It is clear that the algorithm finds P in polynomial time. If none of the pairs u', v' with $\varphi(uu') = \varphi(vv')$ yields an odd induced path, then the answer to the ODD PATHS problem is NO. □

Definition 2. *A graph H is* peculiar *if it can be obtained from a complete graph K as follows. Partition $V(K)$ into six mutually disjoint non-empty sets $A_i, B_i, i = 1,2,3$. For each $i = 1,2,3$, remove at least one edge with one end-vertex in A_i and the other end-vertex in B_{i+1}, where the subscripts are taken modulo 3. Finally, add three new mutually disjoint non-empty complete graphs $D_i, i = 1,2,3$, such that for each $i = 1,2,3$ each vertex in D_i is besides all vertices in D_i adjacent to all vertices in $V(K)\backslash(A_i \cup B_i)$ and to no other vertices.*

The smallest possible peculiar graph is depicted in Figure 2. The straightforward proofs of the following two lemmas are omitted due to page restrictions.

Lemma 6. *Every peculiar graph is P_6-free but not P_5-free.*

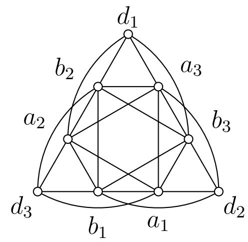

Fig. 2. The smallest possible peculiar graph.

Lemma 7. *The* ODD PATH *problem can be solved in polynomial time for every induced subgraph of a peculiar graph.*

A set $X \subseteq V(H)$ is a *clique cutset* of a graph H if X is a clique and if the graph $H[V(H)\backslash X]$ is disconnected. The following result is due to Chvátal and Sbihi [9].

Theorem 4 ([9]). *A perfect claw-free graph with no clique cutset is either elementary or peculiar.*

Now we have shown that the ODD PATH problem can be solved in polynomial time for (induced subgraphs of) elementary and peculiar graphs, we return to the perfect claw-free graph G and the vertices s and t that form the original instance of the ODD PATH problem; recall that we may assume that both s and t have degree 1 in G, and that G is clean (see Section 2.1).

A *clique cutset decomposition* of G is a collection $\mathcal{C}$ of induced subgraphs of G without a clique cutset such that for every pair $G_i, G_j \in \mathcal{C}$ the set $X_{i,j} := V(G_i) \cap V(G_j)$ is either empty or a clique cutset of $G[V(G_i) \cup V(G_j)]$. A simple lemma of Gavril [14] states that every non-empty $X_{i,j}$ is also a clique cutset of G. Several authors show that that a clique cutset decomposition of a graph can be found in polynomial time (see e.g. [22] or [23] for details).

Theorem 5. *A clique cutset decomposition of a graph can be obtained in polynomial time.*

We are now ready to prove the following theorem.

Theorem 6. *The* ODD PATH *problem can be solved in polynomial time for the class of perfect claw-free graphs that are clean for two specified vertices of degree 1.*

Proof. Let G be a non-perfect claw-free graph that is clean for two vertices s and t of degree 1. Let $\mathcal{C} = \{G_1, \ldots, G_p\}$ be a clique cutset decomposition of G, such that $X_{i,j} := V(G_i) \cap V(G_j)$ is either empty or a clique cutset of G for every $1 \leq i < j \leq p$. The assumption that G is clean, together with the observation that any induced path from s to t may only pass each clique cutset $X_{i,j}$ once, implies that we may without loss of generality assume that $X_{i,j} \neq \emptyset$ if and only if $i \in \{1, \ldots, p-1\}$ and $j = i+1$, and that $s \in V(G_1)$ and $t \in V(G_p)$. Let s' (respectively t') be the unique neighbor of s (respectively t) in G. Then we may without loss of generality assume that $G_1 = G[\{s, s'\}]$, $X_{1,2} = \{s'\}$, $G_p = G[\{t, t'\}]$ and $X_{p-1,p} = \{t'\}$, since both s' and t' are cut vertices of G. We observe that any induced path from s to t contains at most two vertices of each $X_{i,i+1}$, since each set $X_{i,i+1}$ is a clique.

Define $X_{0,1} := \emptyset$, $X_{p,p+1} := \emptyset$ and $V_i := V(G_i)\backslash(X_{i-1,i} \cup X_{i,i+1})$ for $i = 1, \ldots, p$. Let $W_i := V_1 \cup \cdots \cup V_i \cup X_{1,2} \cup \cdots \cup X_{i-1,i} = V(G_1) \cup \ldots \cup V(G_{i-1}) \cup (V(G_i)\backslash X_{i,i+1})$ for $i = 1, \ldots, p$. See Figure 3 for a schematic representation of graph G with respect to the clique cutset decomposition $\mathcal{C}$.

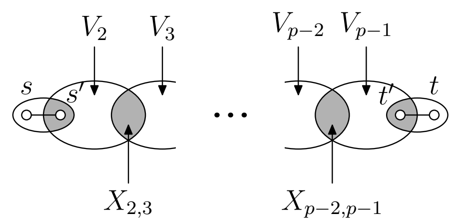

Fig. 3. Structure of the graph G with respect to the clique cutset decomposition $\mathcal{C}$.

We now run the following algorithm for increasing $i = 1, \ldots, p-1$. For each vertex v in $X_{i,i+1}$ we perform the following two steps.

Step 1. Check if $G[W_i \cup \{v\}]$ contains an odd induced path from s to v, and if it contains an even induced path from s to v.
Step 2. For each $v' \in X_{i,i+1}\backslash\{v\}$, check if $G[W_i \cup \{v, v'\}]$ contains an odd induced path from s to v using edge $v'v$, and if it contains an even induced path from s to v using $v'v$.

We claim that we can perform Steps 1 and 2 in polynomial time, keeping track of all the answers in both steps; the details are omitted due to page restrictions. Hence we can determine in polynomial time if G contains an odd induced path from s to t, and we find such a path if one exists. This finishes the proof of Theorem 6. □

In Section 2.1, we showed that for any claw-free graph G and vertices s and t that form an instance of the Odd Path problem, we may assume that both s and t have degree 1 and that G is clean for s and t. This, together with Theorem 3 and Theorem 6, implies the next two results. We omit the straightforward proofs due to page restrictions.

Theorem 7. *The* Odd Path, Even Path *and* Parity Path *problems can be solved in polynomial time for the class of claw-free graphs.*

Corollary 1. *We can decide in polynomial time whether a claw-free graph contains an odd (or even) hole passing through a given vertex, and if so, find one.*

3 Conclusions and open problems

We have presented a polynomial-time algorithm for the Odd Path problem for claw-free graphs. We can also solve the Even Path problem, and as a consequence the Parity Path problem, for claw-free graphs in polynomial time. In addition, we can find an odd (or even) hole through a given vertex in a claw-free graph in polynomial time, if such a hole exists. We conclude with two problems for further research.

Problem 1. Does there exist a polynomial-time algorithm for the Shortest Odd Path problem for claw-free graphs?

Problem 2. Does there exist a polynomial-time algorithm for the Odd Path problem for planar graphs?

References

1. E.M. Arkin, C.H. Papadimitriou, and M. Yannakakis. Modularity of cycles and paths in graphs. *Journal of the ACM*, 38(2):255–274, 1991.
2. S.R. Arikati and U.N. Peled. A linear algorithm for the group path problem on chordal graphs. *Discrete Applied Mathematics*, 44(1-3):185–190, 1993.
3. S.R. Arikati and U.N. Peled. A polynomial algorithm for the parity path problem on perfectly orientable graphs. *Discrete Applied Mathematics*, 65(1):5–20, 1996.
4. S.R. Arikati, C.P. Rangan, and G.K. Manacher. Efficient reduction for path problems on circular-arc graphs. *BIT*, 31(2):182–193, 1991.
5. D. Bienstock. On the complexity of testing for odd holes and induced odd paths. *Discrete Mathematics*, 90(1):85–92, 1991.
6. M. Chudnovsky, N. Robertson, P.D. Seymour, and R. Thomas. The strong perfect graph theorem. *Annals of Mathematics*, 164:51–229, 2006.
7. M. Chudnovsky and P.D. Seymour. The three-in-a-tree problem. Submitted, manuscript available at `www.math.princeton.edu/~pds/papers/threeinatree/paper.ps`.
8. R. Diestel. *Graph Theory.* (3rd Edition). Springer-Verlag Heidelberg, 2005.
9. V. Chvátal and N. Sbihi. Recognizing claw-free perfect graphs. *Journal of Combinatorial Theorie, Series B*, 44:154–176, 1988.
10. H. Everett, C.M.H. de Figueiredo, C.L. Sales, F. Maffray, O. Porto, and B.A. Reed. Path parity and perfection. *Discrete Mathematics*, 165-166:233–252, 1997.
11. H. Everett, C.M.H. de Figueiredo, C. Linhares Sales, F. Maffray, O. Porto, B. Reed. In: L. Ramirez-Alfonsin and B.A. Reed (Eds.), *Perfect Graphs*, Wiley, 67–92, 2001.
12. C.M.H. de Figueiredo, J.G. Gimbel, C.P. Mello, and J.L. Szwarcfiter. Even and odd pairs in comparability and in P_4-comparability graphs. *Discrete Applied Mathematics*, 91(1-3):293–297, 1999.
13. J. Fonlupt and J.P. Uhry. Transformations which preserve perfectness and H-perfectness of graphs. *Annals of Discrete Mathematics*, 16:83–85, 1982.
14. F. Gavril, Algorithms on clique separable graphs. *Discrete Math.*, 19:159–165, 1977.
15. A.S. LaPaugh and C.H. Papadimitriou. The even-path problem for graphs and digraphs. *Networks*, 14:507–513, 1984.
16. B. Lévêque, D.Y. Lin, F. Maffray, and N. Trotignon. Detecting induced subgraphs. *Electronic Notes in Discrete Mathematics*, 29:207–211, 2007.
17. H. Meyniel. A new property of critical imperfect graphs and some consequences. *European Journal of Combinatorics*, 8:313–316, 1987.
18. K.R. Parthasarathy and G. Ravindra. The strong perfect-graph conjecture is true for $K_{1,3}$-free graphs. *Journal of Combinatorial Theory, Series B*, 21:212–223, 1976.
19. R.M. Sampaio and C.L. Sales. On the complexity of finding even pairs in planar perfect graphs. In: Brazilian Symposium on Graphs, Algorithms and Combinatorics 2001, Fortaleza, *Electronic Notes in Discrete Mathematics*, 7:186–189, 2001.
20. C.L. Sales and F. Maffray. Even pairs in claw-free perfect graphs. *Journal of Combinatorial Theory, Series B*, 74:169–191, 1998.
21. C.R. Satyan and C. Pandu Rangan. The parity path problem on some subclasses of perfect graphs. *Discrete Applied Mathematics*, 68(3):293–302, 1996.
22. R.E. Tarjan. Decomposition by clique separators. *Discrete Math.*, 55:221–232, 1985.
23. S.H. Whitesides. A method for solving certain graph recognition and optimization problems, with applications to perfect graph. *Annals of Discrete Mathematics*, 21:281–297, 1984.

Appendix

This appendix contains all the proofs and statements that were omitted from the paper or not presented in detail due to page restrictions.

Lemma 4. *We can determine in polynomial time if a graph H is elementary. If it is, an elementary coloring of H can be found in polynomial time.*

Proof. Let H be a graph. We construct in polynomial time a graph $Gal(H)$, called the *Gallai graph* of H, as follows: $V_{Gal(H)} = E_H$ and two vertices in $Gal(H)$ are adjacent if and only if the corresponding edges in G induce a P_3. It is easy to see that H is elementary if and only if $Gal(H)$ is bipartite. Hence we can check in polynomial time if H is elementary. Since any 2-coloring of $Gal(H)$ corresponds to an elementary coloring of H, we can also find an elementary coloring of H in polynomial time. □

Lemma 6. *Every peculiar graph is P_6-free but not P_5-free.*

Proof. Let H be a peculiar graph, and let A_i, B_i, D_i be a partitioning of $V(H)$ as mentioned in Definition 2. The set $V(H)$ can be partitioned into three cliques, namely $X_1 := A_2 \cup B_1 \cup B_2 \cup D_3$, $X_2 := D_1$ and $X_3 := A_1 \cup A_3 \cup B_3 \cup D_2$. This immediately implies that H is P_7-free, as any induced path in H contains at most two vertices from any clique. The P_6-freeness of H follows from the observation that for every pair $x, y \in X_2$ we have $N_H(x) = N_H(y)$, which implies that any induced path in H containing vertices of $X_1 \cup X_3$ can only contain at most one vertex from X_2.

It follows from Definition 2 that every peculiar graph contains the graph in Figure 2 as an induced subgraph. The graph in Figure 2 contains an induced P_5, namely $d_3a_2d_1b_3d_2$. Hence every peculiar graph contains an induced P_5. □

Lemma 7. *The* Odd Path *problem can be solved in polynomial time for every induced subgraph of a peculiar graph.*

Proof. Let H be an induced subgraph of a peculiar graph, and let H together with two of its vertices u and v be an instance of the Odd Path problem. The problem can trivially be solved if u and v are adjacent, so we assume they are not. Since every peculiar graph is P_6-free by Lemma 6 and every induced subgraph of a P_6-free graph is P_6-free, H is P_6-free. We check for every possible set $S \subseteq V(H) \backslash \{u, v\}$ with $|S| \in \{1, 3\}$ whether $H[S \cup \{u, v\}]$ induces a path in H from u to v. Since we can perform all checks in polynomial time and we only have to perform $\mathcal{O}(|V(H)|^3)$ checks, we can solve the Odd Path problem in polynomial time. It is clear that we also find an odd induced path from u to v in case such a path exists. □

Theorem 6. *The* Odd Path *problem can be solved in polynomial time for the class of perfect claw-free graphs that are clean for two specified vertices of degree 1.*

Proof. We only need to prove that we can perform Steps 1 and 2 in polynomial time. The case $i = 1$ is trivial, since we only have to perform Step 1 on the graph $G[W_1 \cup \{v\}] = G[\{s, s'\}]$. Suppose $i = 2$, and recall that $X_{1,2} = \{s'\}$. To execute Step 1, we first test if the graph $G' := G[V_2 \cup \{s', v\}]$ contains an odd (even) induced path from s' to v. The graph G_2 is a perfect claw-free graph without a clique cutset, so G_2 is either elementary or peculiar by Theorem 4. Since G' is an induced subgraph of G_2, we can perform the above test in polynomial time as a result of Lemma 5 (in case G_2 is elementary) or Lemma 7 (in case G_2 is peculiar). In case of a positive answer, we combine this answer with the trivial answer to the question whether $G[\{s, s'\}]$ contains an odd (even) induced path from s to s'. We perform Step 2 by checking if the graph $G[V_2 \backslash N_G(v) \cup \{s', v, v'\}]$ contains an odd (even) induced path from s' to v; note that such a path must use edge $v'v$ as all other neighbors of v have been removed. This can be done in polynomial time for the same reasons as before. In both steps we keep track of all answers. For each positive answer we keep track of the path itself as well, which we can do as a result of Lemma 5 and Lemma 7.

Now suppose $3 \leq i \leq p-2$. To perform the two checks of Step 1, we act as follows. First we check for all $u \in X_{i-1,i}$ if the graph $G[V_i \cup \{u, v\}]$ contains an odd (even) induced path from u to v. Again, this can be done in polynomial time for the same reasons as before. We combine each positive answer with the answer to the question whether $G[W_{i-1} \cup \{u\}]$ contains an odd (even) induced path from s to u. If this does not lead to two positive answers yet, we check for each $u' \in X_{i-1,i} \backslash \{u\}$ if the graph $G[V_i \backslash N_G(u') \cup \{u, v\}]$ contains an even (odd) induced path from u to v. We combine each positive answer with the answer to the question whether $G[W_{i-1} \cup \{u, u'\}]$ contains an even (odd) induced path from s to u using edge $u'u$. For step 2 we perform similar checks in polynomial time. Again, we keep track of all answers. For each positive answer we keep track of the path itself as well.

For $i = p-1$, we only have to perform Step 1 since $X_{p-1,p} = \{t'\}$. The assumption that $V(G_p) = \{t, t'\}$ means that we can easily combine our two answers for t' to get an answer to the question whether there exists an odd induced path in G from s to t. It is clear that our algorithm also finds such a path in case it exists. □

Theorem 7. *The* Odd Path, Even Path *and* Parity Path *problems can be solved in polynomial time for the class of claw-free graphs.*

Proof. Let G together with two vertices s and t be an input of the Odd Path problem. As we observed in Section 2.1, we may assume that both s and t have degree 1 in G, and that G is clean. This, together with Theorem 3 and Theorem 6, immediately implies that we can solve the Odd Path problem for claw-free graphs in polynomial time. To solve the Even Path problem for a claw-free graph G and two vertices s and t, we first preprocess G as described in Section 2.1 to obtain the set $\mathcal{G}$. We then add a new vertex s' and an edge ss' to every vertex in $\mathcal{G}$, and solve the Odd Path problem for each of those graphs. Note that adding the edge ss' does not create a claw, since s has degree 1 after preprocessing. The Parity Path problem can be solved by running both the Odd Path problem and the Even Path problem. □

Corollary 1. *We can decide in polynomial time whether a claw-free graph contains an odd (or even) hole passing through a given vertex, and if so, find one.*

Proof. Let G be a claw-free graph and let v be a vertex of G. We can find an odd hole through v, or decide that such a hole does not exist, as follows. For each pair u, w of non-adjacent neighbors of v, let $G_{u,w}$ denote the graph obtained from G by removing v and all its neighbors, apart from u and w, from G. We run the Odd Path problem with the graph $G_{u,w}$ and the vertices u and w as input. Clearly, G contains an odd hole through v if and only if the graph $G_{u,w}$ contains an odd induced path from u to w for some pair of non-adjacent neighbors u, w of v. Finding an even hole through a given vertex of a claw-free graph can be done in a similar way, using the algorithm to solve the Even Path problem. □