On Cycle Transversals and Their Connected Variants in the Absence of a Small Linear Forest*

Carl Feghali[1][0000−0001−6727−7213], Matthew Johnson[2][0000−0002−7295−2663],
Giacomo Paesani[2][0000−0002−2383−1339], and
Daniël Paulusma[2][0000−0001−5945−9287]

[1] Department of Informatics, University of Bergen, Norway
carl.feghali@uib.no
[2] Department of Computer Science, Durham University, UK
{matthew.johnson2,giacomo.paesani,daniel.paulusma}@durham.ac.uk

Abstract. A graph is H-free if it contains no induced subgraph isomorphic to H. We prove new complexity results for the two classical cycle transversal problems Feedback Vertex Set and Odd Cycle Transversal by showing that they can be solved in polynomial time for (sP_1+P_3)-free graphs for every integer $s \geq 1$. We show the same result for the variants Connected Feedback Vertex Set and Connected Odd Cycle Transversal. For the latter two problems we also prove that they are polynomial-time solvable for cographs; this was known already for Feedback Vertex Set and Odd Cycle Transversal.

1 Introduction

We consider three well-known graph transversals. To define the notion of a graph transversal, let $\mathcal{H}$ be a family of graphs, $G = (V, E)$ be a graph and $S \subseteq V$ be a subset of vertices of G. The graph $G - S$ is obtained from G by removing all vertices of S. We say that S is an *$\mathcal{H}$-transversal* of G if $G - S$ is *$\mathcal{H}$-free*, that is, $G - S$ contains no induced subgraph isomorphic to some graph of $\mathcal{H}$. In other words, S intersects every induced copy of every graph of $\mathcal{H}$ in G.

Due to their generality, graph transversals play a central role in Theoretical Computer Science. In this paper we focus on three classical transversal problems. Let C_r and P_r denote the cycle and path on r vertices, respectively, and let $G = (V, E)$ be a graph with a subset $S \subseteq V$. Then S is a *vertex cover*, *feedback vertex set*, or *odd cycle transversal* if S is an $\mathcal{H}$-transversal for, respectively, $\mathcal{H} = \{P_2\}$ (that is, $G - S$ is edgeless), $\mathcal{H} = \{C_3, C_4, \ldots\}$ (that is, $G - S$ is a forest), or $\mathcal{H} = \{C_3, C_5, \ldots\}$ (that is, $G - S$ is bipartite).

Usually the goal is to find a transversal of minimum size in some given graph. The corresponding decision problems for the three transversals given above are the classical Vertex Cover, Feedback Vertex Set and Odd Cycle Transversal problems, which are to decide if a given graph has a vertex cover, feedback vertex set or odd cycle transversal, respectively, of size

* The research was supported by the Leverhulme Trust (RPG-2016-258).

at most k for some given positive integer k. Each of these three problems are well-studied and are well-known to be NP-complete.

We may add further constraints to a transversal. In particular, we may require a transversal of a graph G to be *connected*, that is, to induce a connected subgraph of G. The corresponding decision problems for the three above transversals are then called Connected Vertex Cover, Connected Feedback Vertex Set and Connected Odd Cycle Transversal, respectively. Garey and Johnson [14] proved that Connected Vertex Cover is NP-complete even for planar graphs of maximum degree 4 (see, for example, [12,30,34] for NP-completeness results for other graph classes). Grigoriev and Sitters [16] proved that Connected Feedback Vertex Set is NP-complete for planar graphs with maximum degree 9. Chiarelli et al. [8] proved that Connected Odd Cycle Transversal is NP-complete for graphs of arbitrarily large girth and for line graphs.

As all three decision problems and their connected variants are NP-complete, we may want to restrict the input to some special graph class in order to achieve tractability. Note that this approach is in line with the aforementioned results in the literature, where NP-completeness was proven for special graph classes, and also with, for instance, polynomial-time results for Connected Vertex Cover by Escoffier, Gourvès and Monnot [11] (for chordal graphs) and Ueno, Kajitani and Gotoh [33] (for graphs of maximum degree at most 3 and trees).

Just as in most of these papers, we consider *hereditary* graph classes, that is, graph classes closed under vertex deletion. Hereditary graph classes form a rich framework that captures many well-studied graph classes. It is not difficult to see that every hereditary graph class $\mathcal{G}$ can be characterized by a (possibly infinite) set $\mathcal{F}_{\mathcal{G}}$ of forbidden induced subgraphs. If $|\mathcal{F}_{\mathcal{G}}| = 1$, say $\mathcal{F} = \{H\}$, then $\mathcal{G}$ is said to be *monogenic*, and every graph $G \in \mathcal{G}$ is said to be H*-free*. Considering monogenic graph classes can be seen as a natural first step for increasing our knowledge on the complexity of an NP-complete problem in a *systematic* way.

The general strategy for obtaining complexity results for problems restricted to H-free graphs is to first try to prove that the restriction of each problem to H-free graphs is NP-complete whenever H contains a cycle or a claw. If this is the case, then we are left to consider the situation where H does not contain a cycle, implying that H is a forest, and does not contain a claw either, implying that H is a *linear forest*, that is, the disjoint union of one or more paths.

Indeed, when H contains a cycle or a claw, the problems Connected Vertex Cover [25], Feedback Vertex Set (respectively, via a folklore trick, see [3,23], and due to hardness for the subclass of line graphs of planar cubic bipartite graphs [25]), Connected Feedback Vertex Set [8], Odd Cycle Transversal [8] and Connected Odd Cycle Transversal [8] are all NP-complete for H-free graphs. Hence, for these five problems, we are then left to consider only the case where H is a linear forest. We note that the situation for Vertex Cover is different. It follows from Poljak's construction [29] that Vertex Cover is NP-complete for graphs of arbitrarily large girth, and thus for H-free graphs if H contains a cycle. However, Vertex Cover is polynomial-time solvable for claw-free graphs [22,32].

In this paper we focus on proving new complexity results for Feedback Vertex Set, Connected Feedback Vertex Set, Odd Cycle Transversal and Connected Odd Cycle Transversal for H-free graphs. From the above we may assume that H is a linear forest. Below we first discuss the known polynomial cases. As we will use algorithms for Vertex Cover and Connected Vertex Cover as subroutines for our new algorithms, we include these two problems in our discussion.

For each $s \geq 1$, Vertex Cover (by combining the results of [1,31]) and Connected Vertex Cover [8] are polynomial-time solvable for sP_2-free graphs.[3] Moreover, Vertex Cover is also polynomial-time solvable for $(sP_1 + P_6)$-free graphs, for every $s \geq 0$ [17], whereas Connected Vertex Cover is so for $(sP_1 + P_5)$-free graphs [20]. Their complexity for P_r-free graphs is unknown for $r \geq 7$ and $r \geq 6$, respectively.

The Feedback Vertex Set and Odd Cycle Transversal problems are polynomial-time solvable for permutation graphs [4], and thus for P_4-free graphs. Recently, Okrasa and Rzążewski [26] proved that Odd Cycle Transversal is NP-complete for P_{13}-free graphs. A small modification of their construction yields the same result for Connected Odd Cycle Transversal. The complexity of Feedback Vertex Set and Connected Feedback Vertex Set is unknown, when restricted to P_r-free graphs for $r \geq 5$. For every $s \geq 1$, both problems and their connected variants are polynomial-time solvable for sP_2-free graphs [8], using the price of connectivity for feedback vertex set [2,19].[4]

1.1 Our Results

We prove in Section 3 that Connected Feedback Vertex Set and Connected Odd Cycle Transversal are polynomial-time solvable for P_4-free graphs, just as Feedback Vertex Set and Odd Cycle Transversal are [4]. We then prove, in Section 4, that, for every $s \geq 1$, these four problems are all polynomial-time solvable for $(sP_1 + P_3)$-free graphs.

To prove our results, we rely on two proof ingredients. The first one is that we use known algorithms for Vertex Cover and Connected Vertex Cover restricted to H-free graphs as subroutines in our new algorithms. The second one is that we consider the connected variant of the transversal problems in a more general form. For Connected Vertex Cover this variant is defined as follows:

[3] The graph $G + H$ is the disjoint union of graphs G and H and sG is the disjoint union of s copies of G; see Section 2.

[4] The price of connectivity concept was introduced by Cardinal and Levy [7] for vertex cover; see also, for example, [6,10].

Connected Vertex Cover Extension
Instance: a graph $G = (V, E)$, a subset $W \subseteq V$ and a positive integer k.
Question: does G have a connected vertex cover S_W with $W \subseteq S_W$ and $|S_W| \leq k$?

Note that Connected Vertex Cover Extension becomes the original problem if $W = \emptyset$. In the same way we define the problems Connected Feedback Vertex Set Extension and Connected Odd Cycle Transversal Extension. In fact we will prove all our results for connected feedback vertex sets and connected odd cycle transversals for the extension versions. This is partially out of necessity – the extension versions sometimes serve as auxiliary problems for some of our inductive arguments and may do so for future results as well – but it does also lead to slightly stronger results.

Remark 1. Note that one could also define extension versions for any original transversal problem. However, such extension versions will be polynomially equivalent. In particular we could solve any of them on input (G, W, k) by considering the original problem on input $(G - W, k - |W|)$ and adding W to the solution. However, due to the connectivity condition, we cannot use this approach for the connected variants and need to follow a more careful approach.

Remark 2. It is known that Vertex Cover is polynomial-time solvable for $(P_1 + H)$-free graphs whenever it is so for H-free graphs. This follows from a well-known observation. see, e.g., [24]: one can solve the complementary problem of finding a maximum independent set in a $(P_1 + H)$-free graph by solving this problem on each H-free graph obtained by removing a vertex and all its neighbours. However, this trick does not work for Connected Vertex Cover. Moreover, it does not work for Feedback Vertex Set and Odd Cycle Transversal and their connected variants either.

2 Preliminaries

Let $G = (V, E)$ be a graph. For a set $S \subseteq V$, the graph $G[S]$ denotes the subgraph of G induced by S. We say that S is *connected* if $G[S]$ is connected. We write $G - S$ for the graph $G[V \setminus S]$. A subset $D \subseteq V$ is a *dominating* set of G if every vertex of $V \setminus D$ is adjacent to at least one vertex of D. An edge uv of a graph $G = (V, E)$ is *dominating* if $\{u, v\}$ is a dominating set. The *complement* of G is the graph $\overline{G} = (V, \{uv \mid uv \notin E \text{ and } u \neq v\})$. The *neighbourhood* of a vertex $u \in V$ is the set $N(u) = \{v \mid uv \in E\}$ and for $U \subseteq V$, we let $N(U) = \bigcup_{u \in U} N(u) \setminus U$. We denote the *degree* of a vertex $u \in V$ by $\deg(u) = |N(u)|$.

Let $G = (V, E)$ be a graph and let $S \subseteq V$. Then S is a *clique* if all vertices of S are pairwise adjacent and an *independent set* if all vertices of S are pairwise non-adjacent. A graph is *complete* if its vertex set is a clique. We let K_r denote the complete graph on k vertices. Let $T \subseteq V$ with $S \cap T = \emptyset$. Then S is *complete* to T if there is an edge between every vertex of S and every vertex of T, and S is *anti-complete* to T if there are no edges between S and T. In the first case we also say that S is *complete* to $G[T]$ and in the second case anticomplete to $G[T]$.

A graph is *bipartite* if its vertex set can be partitioned into at most two independent sets. A bipartite graph is *complete* if its vertex set can be partitioned into two independent sets X and Y such that there is an edge between every vertex of X and every vertex of Y. Note that every edge of a complete bipartite graph is dominating.

Let G_1 and G_2 be two vertex-disjoint graphs. The *union* operation creates the *disjoint union* $G_1 + G_2$ of G_1 and G_2, that is, the graph with vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2)$. We denote the disjoint union of r copies of G_1 by rG_1. The *join* operation adds an edge between every vertex of G_1 and every vertex of G_2. A graph G is a *cograph* if G can be generated from K_1 by a sequence of join and union operations. A graph is a cograph if and only if it is P_4-free (see, for example, [5]).

The following lemma is well-known.

Lemma 1. *Every connected P_4-free graph on at least two vertices has a spanning complete bipartite subgraph which can be found in polynomial time.*

Let $G = (V, E)$ be a graph. The *contraction* of an edge $uv \in E$ deletes the vertices u and v and replaces them by a new vertex made adjacent to precisely those vertices that were adjacent to u or v in G (without introducing self-loops or multiple edges). Recall that a linear forest is the disjoint union of one or more paths. The following lemma is a straightforward observation.

Lemma 2. *Let H be a linear forest and let G be a connected H-free graph. Then the graph obtained from G after contracting an edge is also connected and H-free.*

Recall that Grzesik et al. [17] proved that Vertex Cover is polynomial-time solvable for P_6-free graphs. Using the folklore trick mentioned in Remark 2 (see also, for example, [20,24]) their result can be formulated as follows.

Theorem 1 ([17]). *For every $s \geq 0$,* Vertex Cover *can be solved in polynomial time for $(sP_1 + P_6)$-free graphs.*

We recall also that Connected Vertex Cover is polynomial-time solvable for $(sP_1 + P_5)$-free graphs [20]. We will need the extension version of this result. Its proof, which we omit, is based on a straightforward adaption of the proof for Connected Vertex Cover on $(sP_1 + P_5)$-free graphs [20].

Theorem 2 ([20]). *For every $s \geq 0$,* Connected Vertex Cover Extension *can be solved in polynomial time for $(sP_1 + P_5)$-free graphs.*

3 The Case $\mathbf{H = P_4}$

Recall that Brandstädt and Kratsch [4] proved that Feedback Vertex Set and Odd Cycle Transversal can be solved in polynomial time for permutation graphs, which form a superclass of the class of P_4-free graphs. Hence, we obtain the following proposition.

Proposition 1 ([4]). Feedback Vertex Set *and* Odd Cycle Transversal *can be solved in polynomial time for P_4-free graphs.*

In this section, we prove that the (extensions versions of the) connected variants of Feedback Vertex Set and Odd Cycle Transversal are polynomial-time solvable on P_4-free graphs as well. We use Proposition 1 for the proofs.

Theorem 3. Connected Feedback Vertex Set Extension *can be solved in polynomial time for P_4-free graphs.*

Proof. Let $G = (V, E)$ be a P_4-free graph and $W \subseteq V$. We may assume without loss of generality that G is connected. We search for a smallest connected feedback vertex set S_W of G that contains W. By Lemma 1, we find in polynomial time a spanning complete bipartite subgraph $G' = (X, Y, E')$, so every edge in G' is dominating. The set S_W that we are looking for can be distributed over X and Y in various ways. So we first compute, in Case 1, a smallest feedback vertex set of G that contains both vertices of X and Y. Then, in Case 2, we compute a smallest feedback vertex set of G that is a subset of X, and then a smallest one that is a subset of Y. Afterwards, we take the smallest sets over all sets computed as our final output; note that some sets may not exist. However, as $S = V$ is a feedback vertex set of G, at least one set will be computed.

Case 1. $S_W \cap X \neq \emptyset$ and $S_W \cap Y \neq \emptyset$. In this case, $G[S_W]$ will contain an edge uv of G' and hence S_W will be connected. Otherwise, in order to ensure connectivity and to satisfy the condition of the case, we "guess" an edge uv with $u \in X \cap S_W$ and $v \in Y \cap S_W$, respectively. As we need to consider all possibilities of choosing this edge, this extra step adds an $O(n^2)$-time factor to the running time. We are now left to find a smallest feedback vertex set S' in $G - (W \cup \{u, v\})$. This takes polynomial time due to Proposition 1. We remember $S' \cup W \cup \{u, v\}$.

Case 2. $S_W \subseteq X$ or $S_W \subseteq Y$. We first consider the possibility that $S_W \subseteq X$. Then we must have that $W \subseteq X$; otherwise this possibility will not happen. We start by examining the situation where $S_W = X$. This can only happen if $G[Y]$ is a forest, in which case we remember $|S_W| = X$.

We now examine the situation where $S_W \subsetneq X$. Then Y must be independent, as otherwise $G - S_W$ contains a triangle. So, if Y is not independent, then we discard this option. Assume that Y is an independent set. If $|Y| = 1$, then $G[X] - S_W$ is an independent set, as otherwise $G - S_W$ contains a triangle. Hence, we must compute a smallest connected vertex cover of $G[X]$ that contains W. We can do this in polynomial time due to Theorem 2. We remember the output. If $|Y| \geq 2$, then $|X \setminus S_W| = 1$, as otherwise $G[Y \cup (X \setminus S_W)]$ contains a cycle. Hence, we check in polynomial time if there exists a vertex $x \in X \setminus W$, such that $X \setminus \{x\}$ is connected. If so we remember the size $|X| - 1$.

We now repeat the same procedure to examine the possibility that $S_W \subseteq Y$. In the end we then take the output of minimum size.

Finally, as mentioned, we compare the size of the set computed in Case 1 with the size of the one computed in Case 2, and we return the smallest set as a smallest connected feedback vertex set of G that contains W. □

The second result of this section can be proven in exactly the same way as Theorem 3.

Theorem 4. Connected Odd Cycle Transversal Extension *can be solved in polynomial time for P_4-free graphs.*

Proof. We do the same as in the proof of Theorem 3. The differences are the following. In Case 1, we need to compute a smallest odd cycle transversal S' in $G - (W \cup \{u, v\})$ (which can be done using Proposition 1 as well). In Case 2 we again start by examining the situation where $S_W = X$. This can only happen if $G[Y]$ is bipartite, in which case we remember $|S_W| = X$. We then consider the situation where $S_W \subsetneq X$ in the same as in the proof of Theorem 3 except that we no longer distinguish between $|Y| = 1$ and $|Y| \geq 2$, that is, we follow the approach used in the proof of Theorem 3 for the case where $|Y| = 1$ for all values of $|Y|$. □

4 The Case $\mathbf{H = sP_1 + P_3}$

We will prove that Feedback Vertex Set and Odd Cycle Transversal and their connected variants can be solved in polynomial time for $(sP_1 + P_3)$-free graphs. In order to do this we need a structural result first.

Lemma 3. *For every $s \geq 0$, let G be a bipartite $(sP_1 + P_3)$-free graph. If the smallest connected component of G contains at least c vertices where*

$$c = \begin{cases} 3 & \text{if } s \leq 1 \\ 2s - 1 & \text{if } s \geq 2 \end{cases}$$

then G has only one component.

Proof. Assume that G has two connected components C_1 and C_2 that each contain at least c vertices. As C_1 is bipartite and contains at least $2s-1$ vertices, it contains a set of s independent vertices that induce sP_1. As $c \geq 3$, there is a vertex v in C_2 of degree at least 2, and, as C_2 is bipartite, the neighbours of v are independent so v and two of its neighbours induce a P_3. Thus G is not $(sP_1 + P_3)$-free. This contradiction completes the proof. □

We now state our four results. For the connected variants we can show the extension versions. We only include the proof of Theorem 8, which is the most involved and shows all our techniques, and omit the other proofs.

Theorem 5. *For every $s \geq 0$,* Feedback Vertex Set *can be solved in polynomial time for $(sP_1 + P_3)$-free graphs.*

Theorem 6. *For every $s \geq 0$,* Connected Feedback Vertex Set Extension *can be solved in polynomial time for $(sP_1 + P_3)$-free graphs.*

Theorem 7. *For every $s \geq 0$,* Odd Cycle Transversal *can be solved in polynomial time for $(sP_1 + P_3)$-free graphs.*

Theorem 8. *For every $s \geq 0$,* Connected Odd Cycle Transversal Extension *can be solved in polynomial time for $(sP_1 + P_3)$-free graphs.*

Proof. Let $G = (V, E)$ be an $(sP_1 + P_3)$-free graph on n vertices and $W \subseteq V$. If G is bipartite and W is connected, then W is the unique minimum connected odd cycle transversal that contains W. If G is bipartite and W is not connected, then in polynomial time we find the smallest set $U \supset W$ such that $G[U]$ is connected by adding vertices of shortest paths connecting the different components of $G[W]$ (assuming that all vertices of W belong to the same component as G; otherwise we return a no-answer). If G is disconnected, then each of its connected components except for one must be bipartite; otherwise we return a no-answer. From now on, assume that G is non-bipartite and connected. This means that V is a connected odd cycle transversal of G. We can determine in polynomial time whether V is the minimum size connected odd cycle transversal that contains W by checking, for each vertex $u \in V$, whether or not $V \setminus \{u\}$ is a connected odd cycle transversal of G that contains W. Thus from we assume that V is not a minimum connected odd cycle transversal of G that contains W.

If $s = 0$, then we can use Theorem 4. So we assume that $s \geq 1$ and that we can solve Connected Feedback Vertex Set Extension in polynomial time for $((s-1)P_1 + P_3)$-free graphs. We show that we can find, in polynomial time, a smallest connected odd cycle transversal S_W of G that contains W. In fact, we shall solve the equivalent problem of finding, in polynomial time, a bipartite subgraph B_W of G such that $B_W \cap W = \emptyset$, $G - B_W$ is connected and, subject to these conditions, B_W is of maximum size. To do this, we consider two cases. Let $c = 3$ if $s = 1$ and $c = 2s - 1$ otherwise (the constant c comes from Lemma 3, which we will apply in Case 2). Our two cases derive from assuming, or not, that at least one connected component of B_W has size at most $c - 1$. In each case, we attempt to find, subject to our assumption, a bipartite subgraph B of G such that $B \cap W = \emptyset$ and $G - B$ is connected, and, if such a set B exists, we find the solution of maximum size. We will see that, taken together, we cover all possible cases so the largest B found has size $|B_W|$. In particular, we note that B_W is not empty by our assumption that $S_W \neq V$.

Case 1. At least one connected component L of B_W has size at most $c - 1$.
In this case we take every possible choice for L under consideration, discarding all sets that do not induce a bipartite graph, or whose removal disconnects the graph, or that intersect W (as none of these sets can be a candidate set for L). As $|V(L)| \leq c - 1$, there are at most $O(n^{c-1})$ choices. For each choice of L we do as follows.

Let U be the set of neighbours of the vertices of L that belong to $G - L$. Note that U must belong to S_W if we guessed it correctly, and so we may contract any edge inside $G[U]$ to modify U into an independent set. This takes polynomial time and, by Lemma 2, the resulting graph, which we denote by G again, is still $(sP_1 + P_3)$-free. Moreover, $G - L$ is still connected.

As L contains at least one vertex and G is $(sP_1 + P_3)$-free, $G - (L \cup U)$ is $((s-1)P_1 + P_3)$-free. Let S be a connected odd cycle transversal that contains U. As U is an independent set, each of its vertices has at least one neighbour in $S \setminus U$. Thus there are sets in $S \setminus U$ that dominate U. Let R be a smallest such set.

We consider each possible choice If $|U| = 1$, then $|R| \leq 1$. Suppose $|U| \geq 2$. As U is an independent set on at least two vertices, S_W must contain three vertices of $G - L$ that form an induced path, which we denote by P. As G is $(sP_1 + P_3)$-free and U is independent, $V(P)$ must dominate all but at most $s-1$ vertices of U. Let U' be the subset of vertices of U that are not dominated by $V(P)$. So $|U'| \leq s-1$. Consider a set that contains $V(P)$ and, for each vertex u in U', a neighbour of u in S_W. This set dominates U so is at least the size of R. Thus $|R| \leq |P| + |U'| \leq 3 + s - 1 = s + 2$.

Hence there at most $O(n^{s+2})$ possible choices for R. We consider each possible choice, and for each we compute the size of a smallest odd cycle transversal S_R in $G - (L \cup U)$ that contains $R \cup (W \setminus U)$. (Recall that $W \cap V(L) = \emptyset$, so $W \setminus U$ belongs to $G - (L \cup U)$.) As $G - (L \cup U)$ is $((s-1)P_1 + P_3)$-free, we can find S_R in polynomial time using our algorithm for $((s-1)P_1 + P_3)$-free graphs. Then $S_R \cup U$ is a smallest connected odd cycle transversal of G that contains $U \cup R \cup W$.

Now, over all choices for R, we keep the smallest $S_R \cup U$, which we denote by S_L. Then S_L is a smallest connected odd cycle transversal of G that contains W such that $G - S$ has L as a connected component.

Finally, from all sets S_L, we keep the smallest set found, which we denote by S_1. Then S_1 is a smallest connected odd cycle transversal S of G that contains W such that $G - S$ has a connected component of size at most $c - 1$. We find S_1 in polynomial time, as the number of choices for R and L is polynomially bounded and each choice can be processed in polynomial time. Let $B_1 = G - S$.

Case 2. Every connected component of B_W has size at least c.
In this case we will compute the largest induced bipartite graph B of G such that $B \cap W = \emptyset$ and $S = V(G) \setminus V(B)$ is connected, and, moreover, every connected component of B has size at least c. Then by Lemma 3, the subgraph B we are looking for is connected.

Dealing with the case where one partition class is small.
First we suppose that B has a bipartition (X, Y) such that $|X| \leq s - 1$. To find the best solution in this case, we consider each of the $O(n^{s-1})$ sets X of at most $s - 1$ vertices of G. For every such X, we check whether X is an independent set (in constant time) and whether X is disjoint from W. If both conditions are satisfied, we wish to find Y, the largest possible independent set that is in $G - X$ and disjoint from W such that $G - (X \cup Y)$ is connected. By Theorem 2 we can do this in polynomial time by computing a minimum connected vertex cover S_X of $G - X$ that contains W. Then we can let $G - (X \cup S_X)$ be Y. Note that $X \cup Y$ might not be connected, so we may have duplicated some polynomial-time work. We pick the best solution B, and set $B_2 = B$ (we will update B_2 in the remainder of Case 2 and afterwards compare its size with the size of B_1).

Dealing with the case where both bipartition classes are large.
Now we suppose that B is connected, contains at least $c \geq 3$ vertices and has a bipartition in which each partition class contains at least s vertices. In particular B contains an induced P_3. We consider each of the $O(n^{2s})$ pairs of disjoint sets X' and Y' each containing s of the vertices of G. We check whether X' and Y' are both independent sets and are disjoint from W and whether $G[X' \cup Y']$ has an induced P_3. If these conditions are not satisfied, we discard the case. Otherwise, our aim will be to try to construct from X' and Y', a bipartite graph $B = (X, Y)$ such that $X' \subseteq X$, $Y' \subseteq Y$, $(X \cup Y) \cap W = \emptyset$ and $G - B$ is connected and, subject to these conditions, B is of maximum size. We now define (see also Fig. 1) a partition of $V \setminus (X' \cup Y') = U \cup V_X \cup V_Y \cup Z$ where

$$\begin{aligned} U &= (N(X') \cap N(Y')) \cup W, \\ V_X &= N(X') \setminus (Y' \cup N(Y') \cup W), \\ V_Y &= N(Y') \setminus (X' \cup N(X') \cup W), \\ Z &= V \setminus (X' \cup Y' \cup N(X') \cup N(Y') \cup W). \end{aligned}$$

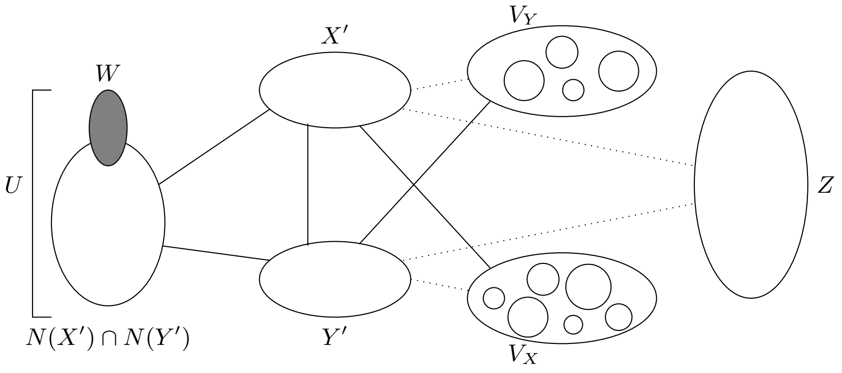

Fig. 1. The decomposition of G in Case 2, where a straight edge between two sets indicates that at least one edge must exist, a dotted edge indicates that no edges between the two sets exist, and the absence of an edge indicates that edges between the two sets could exist. The circles in V_X and V_Y represent disjoint unions of complete graphs.

No vertex $u \in U$ can be a member of B, as either u has at least one neighbour in X' and at least one neighbour in Y', or u belongs to W. We also know that $G[V_X]$ is P_3-free, as otherwise $Y' \cup V_X$ would induce an $sP_1 + P_3$. By the same argument, $G[V_Y]$ is also P_3-free. This means that both $G[V_X]$ and $G[V_Y]$ are the disjoint union of a set of complete graphs. Moreover, Z does not contain an independent set of size greater than $s - 1$ as otherwise, since $G[X' \cup Y']$ contains an induced P_3, we find that $X' \cup Y' \cup Z$ contains a subset of vertices that induce an $sP_1 + P_3$.

Step 1. Reduce Z to the empty set.
We are going to reduce Z to the empty set via some branching. We consider all the possible ways the vertices of Z might be included in B. As Z does not contain an independent set of size greater than $s-1$, every partition class of B contains at most $s-1$ vertices of Z. Hence, we consider each of the $O(n^{2s-2})$ pairs of disjoint sets Z_X and Z_Y of size at most $s-1$ in Z. We check whether Z_X and Z_Y are independent sets. If they both are, we define X' to be $X' \cup Z_X$ and Y' to be $Y' \cup Z_Y$. We redefine U by adding to it the vertices of $Z \setminus (Z_X \cup Z_Y)$; note that U still contains W. Moreover, vertices of V_X with a neighbour in Z_Y cannot belong to Y' (recall that they cannot be in X' either). Similarly, the set of vertices of V_X with a neighbour in Z_X cannot be members of X' or Y' either. Thus we redefine U again by adding all these vertices to it, and V_X and V_Y by removing the vertices we placed in U.

We now have a partition $X' \cup Y' \cup U \cup V_X \cup V_Y$ where $G[X', Y']$ is bipartite, U contains vertices that have neighbours in both X' and Y' or vertices that belong to W, the vertices of V_X have neighbours in X' but not in Y', and the vertices of V_Y have neighbours in Y' but not in X'. Moreover, $G[V_X]$ and $G[V_Y]$ are still the disjoint union of a set of complete graphs.

Step 2. Reduce U to a singleton set.
We are going to reduce U to a singleton set via some branching. Recall that no vertex of U will be placed in the final partition classes X and Y of the bipartite graph B we are searching for. We contract every edge between two vertices in $G[U]$. In the new graph, which we denote by G again, U is an independent set. By Lemma 2, G is still $(sP_1 + P_3)$-free and connected.

As U belongs to the connected complement of the bipartite graph $B = (X, Y)$ we are searching for, the vertices of U need to be made connected to each other via paths in $G - (X' \cup Y')$. Following the same arguments as in Case 1, there must exist a set R of size at most $s+2$ in $G - (X' \cup Y')$ that dominates U (if not then we can discard the case). We guess R by considering all $O(n^{s+2})$ possibilities. If needed we consider all possibilities of making $R \cup U$ connected via adding shortest paths connecting vertices of R. As every connected $(sP_1 + P_3)$-free graph has diameter at most $2s+2$, we need to guess a total of $(|R|-1)2s \leq 2s^2 + 2s$ additional vertices, so must consider $O(n^{2s^2+2s})$ possibilities. In each branch we contract all edges in $G[R \cup U]$ into a single vertex, which we denote by u. By Lemma 2 the resulting graph, which we denote by G again, is $(sP_1 + P_3)$-free and connected.

We redefine V_X and V_Y by removing the vertices that were added to U. Then $G[V_X]$ and $G[V_Y]$ are still the disjoint unions of complete graphs, which we denote by, respectively, $K_X^1, \ldots, K_X^q$ (if $V_X \neq \emptyset$) and $K_Y^1, \ldots, K_Y^r$ (if $V_Y \neq \emptyset$).

Step 3. Adding vertices from V_X to Y' and from V_Y to X'.
To complete the construction of B, we need only describe how to add (in polynomial time) as many vertices in total from V_X to Y' and from V_Y to X' in such a way that the new sets X and Y remain independent sets and the graph induced

by $V \setminus (X \cup Y)$ is connected. In particular $V \setminus (X \cup Y)$ contains the vertex u, to which all vertices of W have been contracted to. Note that from each K_X^h and each K_Y^i we can add at most one vertex to $X' \cup Y'$, as otherwise we create a triangle in B. However, we must be careful, as by adding a vertex from $V_X \cup V_Y$ to $X' \cup Y'$, we may lose connectivity of the graph $G - (X \cup Y)$; recall that X and Y are the sets we are trying to construct. Recall that $S = V(G) \setminus (X \cup Y)$ is the corresponding connected odd cycle transversal that we are trying to create. We analyse the possible structure of S by distinguishing two cases.

Case (i). The graph $G[S]$ contains no edges between $V_X \cap S$ and $V_Y \cap S$. Recall that X and Y can contain at most one vertex from each K_X^h and K_Y^i. Hence, u must be adjacent to at least one vertex of every K_X^h of size at least 2 and at least one vertex of every K_Y^i of size at least 2. If not, then we can discard our current choice, as it will not lead to a set S that is connected. If u is complete to a set $V(K_X^h)$, for $h \in \{1, \ldots, q\}$, we pick an arbitrary vertex of $V(K_X^h)$, and else a non-neighbour of u, and add it to Y'. We do the same thing when considering $V(K_Y^i)$, for $i \in \{1, ..., r\}$, and add a vertex to X'. We also put the vertices of all singleton connected components of $G[V_X]$ and $G[V_Y]$ in Y' and X', respectively. If the resulting set $X' \cup Y'$ is larger than B_2, then we let $B_2 = X' \cup Y'$.

Case (ii). The graph $G[S]$ has an edge xy where $x \in V_X \cap S$ and $y \in V_Y \cap S$. We consider all $O(n^2)$ possibilities for choosing the edge xy. Let xy be such a choice. By definition, x has a neighbour $v_x \in X$ and y has a neighbour $v_y \in Y$. As no vertex of V_Y has a neighbour in X, the vertices y, x, v_x induce a P_3. As G is $(sP_1 + P_3)$-free, x must then be complete to all but at most $s - 1$ graphs K_Y^i. Similarly, y must be complete to all but at most $s - 1$ graphs K_X^h. A graph K_X^h or K_Y^i is *bad* if it is not complete to y or x, respectively, and *good* otherwise.

We first consider the bad complete graphs. Note that x, y could be in a bad complete graph. For each bad complete graph, we "guess" at most one vertex distinct from x and y that we will move to X' or Y' (so we update X' and Y'). This leads to $O(n^{2s-2})$ possible cases and we consider each of them as follows.

We first check if the remaining vertices from the bad complete graphs, the vertex u and the vertices x, y all belong to the same connected component of $G - (X' \cup Y')$. This must hold in order for all these vertices to end up in the connected graph $G-(X \cup Y)$ we are looking for (if the branch under consideration is correct, then all vertices of $G-(X \cup Y)$ belong to $G-(X' \cup Y')$). So, if this does not hold, then we discard the case. Otherwise we try to connect the remaining vertices of the bad components, u, x, and y by considering all possibilities for choosing a smallest connected set in $G - (X' \cup Y')$ that contains all of them.

Before doing this, we first contract any edges between vertices that belong to the union of the bad complete graphs and the set $\{u, x, y\}$. As xy is an edge, this leads to an independent set of size at most $2(s - 1) + 1 + 1 = 2s$. By Lemma 2 the resulting graph is $(sP_1 + P_3)$-free again, so the connected component that we are looking for has diameter at most $2s + 2$. This means that we need to "guess" at most $(2s - 1)(2s + 1) = 4s^2 - 1$ vertices. Hence, the total number of possible choices is $O(n^{4s^2-1})$. We consider each choice. For each choice, it remains to pick for every good complete graph an arbitrary vertex (if it exists) that was not involved in the guessing and put it in X' or Y' in an appropriate way. We may pick these vertices arbitrarily, as we can only pick one vertex from each complete graph and all remaining vertices of the good complete graphs are adjacent to one of x, y, ensuring connectivity. If the resulting set $X' \cup Y'$ is larger than B_2, then we let $B_2 = X' \cup Y'$. This completes the description of Case 2.

Note that from all the bipartite graphs $B = (X, Y)$ we kept track of in Case 2, we stored a largest one B_2. We compare B_2 with B_1, picking the largest as B_W. Then $S_W = V(G) - B_W$ is a smallest connected odd cycle transversal of G that contains W. The correctness of our algorithm follows from its description. Moreover, as the number of branches is polynomial and each branch is processed in polynomial time, the running time of our algorithm is polynomial. □

5 Conclusions

We proved polynomial results for Feedback Vertex Set and Odd Cycle Transversal and their connected variants for H-free graphs, where $H = P_4$ or $H = sP_1 + P_3$. Natural cases for future work are the cases where $H = sP_1 + P_4$ for $s \geq 1$ and $H = P_r$ for $r \geq 5$. Note that Lemma 3 does not hold for (sP_1+P_4)-free graphs: the disjoint union of any number of arbitrarily large stars is even P_4-free. We also lose the crucial property of a connected (sP_1+P_3)-free graph G that any independent set U is dominated by a set $R \subseteq V(G) \setminus U$ with $|R| \leq s+2$. Recall that Vertex Cover and Connected Vertex Cover are polynomial-time solvable even for (sP_1+P_6)-free graphs [17] and (sP_1+P_5)-free graphs [20] for every $s \geq 0$. Recall that Odd Cycle Transversal and Connected Odd Cycle Transversal are known to be NP-complete for P_{13}-free graphs [26]. However, no integer r is known, for which any of the other four problems is NP-complete for P_r-free graphs.

Independent transversals. A similar complexity study is also undertaken for the independent variants of the problems Feedback Vertex Set and Odd Cycle Transversal.[5] In particular, Independent Feedback Vertex Set and Independent Odd Cycle Transversal are polynomial-time solvable for P_5-free graphs [3], but their complexity status is unknown for P_6-free graphs. No integer r is known either such that Independent Feedback Vertex Set and Independent Odd Cycle Transversal are NP-complete for P_r-free graphs. Hence, to make any further progress, we must understand the structure of P_r-free graphs better. This topic has been well-studied in recent years, see also for example [15,18]. However, more research and new approaches are needed.

Two other generalizations. A well-known way of generalizing Feedback Vertex Set and Odd Cycle Transversal is to pre-specify a set T of *terminal* vertices in a graph $G = (V, E)$ and to ask for a corresponding transversal of size at most k that intersects all cycles or odd cycles, respectively, that contain at least one terminal vertex from T. These problems are called Subset Feed-

[5] Independent Vertex Cover is 2-Colouring, which is polynomially solvable.

back Vertex Set and Subset Odd Cycle Transversal (choose $T = V$ to get the original problem back).

Both Subset Feedback Vertex Set and Subset Odd Cycle Transversal are NP-complete for H-free graphs if H contains a cycle or claw, due to the aforementioned NP-completeness for the original problems. Moreover, Subset Feedback Vertex Set is polynomial-time solvable for sP_1-free graphs [28] and for permutation graphs [27], and thus for P_4-free graphs, but NP-complete for split graphs [13], or equivalently, $(C_4, C_5, 2P_2)$-free graphs, and thus for P_5-free graphs.[6] It would be interesting to obtain full complexity dichotomies for Subset Feedback Vertex Set and Subset Odd Cycle Transversal for H-free graphs. For the former problem it remains to solve the cases where $H = P_r + sP_1$ for every pair (r, s) with $r = 2, s \geq 2$ or $3 \leq r \leq 4, s \geq 1$. The latter problem seems to be mainly studied from a parameterized point of view [21].

References

1. E. Balas and C. S. Yu, On graphs with polynomially solvable maximum-weight clique problem, Networks 19 (1989) 247–253.
2. R. Belmonte, P. van 't Hof, M. Kaminski and D. Paulusma, The price of connectivity for feedback vertex set, Discrete Applied Mathematics 217 (2017) 132–143.
3. M. Bonamy, K.K. Dabrowski, C. Feghali, M. Johnson and D. Paulusma, Independent feedback vertex set for P_5-free graphs, Algorithmica 81 (2019) 1342–1369.
4. A. Brandstädt and D. Kratsch, On the restriction of some NP-complete graph problems to permutation graphs. Proc. FCT 1985, Lecture Notes in Computer Science 199 (1985) 53–62.
5. A. Brandstädt, V.B. Le and J. Spinrad, Graph Classes: A Survey, SIAM Monographs on Discrete Mathematics and Applications, Society for Industrial and Applied Mathematics (SIAM), 1999.
6. E. Camby, J. Cardinal, S. Fiorini and O. Schaudt, The price of connectivity for vertex cover, Discrete Mathematics & Theoretical Computer Science 16 (2014) 207–224.
7. J. Cardinal and E. Levy, Connected vertex covers in dense graphs, Theoretical Computer Science 411 (2010) 2581–2590.
8. N. Chiarelli, T.R. Hartinger, M. Johnson, M. Milanic and D. Paulusma, Minimum connected transversals in graphs: new hardness results and tractable cases using the price of connectivity, Theoretical Computer Science 705 (2018) 75–83.
9. D. G. Corneil and J. Fonlupt, The complexity of generalized clique covering, Discrete Applied Mathematics 22 (1988) 109–118.
10. E. Camby and O. Schaudt, The price of connectivity for dominating set: Upper bounds and complexity, Discrete Applied Mathematics 17 (2014) 53–59.
11. B. Escoffier, L. Gourvès and J. Monnot, Complexity and approximation results for the connected vertex cover problem in graphs and hypergraphs, Theoretical Computer Science 8 (2010) 36–49.
12. H. Fernau and D. Manlove, Vertex and edge covers with clustering properties: complexity and algorithms, Journal of Discrete Algorithms 7 (2009) 149–167.
13. F.V. Fomin, P. Heggernes, D. Kratsch, C. Papadopoulos and Y. Villanger, Enumerating minimal subset feedback vertex sets, Algorithmica 69 (2014) 216–231.

[6] Feedback Vertex Set is polynomial-time solvable even for sP_2-free graphs [8].

14. M.R. Garey, D.S. Johnson, The rectilinear Steiner tree problem is NP-complete, SIAM Journal on Applied Mathematics 32 (1977) 826–834.
15. P.A. Golovach, M. Johnson, D. Paulusma and J. Song, A survey on the computational complexity of colouring graphs with forbidden subgraphs, Journal of Graph Theory 84 (2017) 331–363.
16. A. Grigoriev and R. Sitters, Connected feedback vertex set in planar graphs, Proc. WG 2009, LNCS 5911 (2010) 143–153.
17. A. Grzesik, T. Klimošová, M. Pilipczuk, and M. Pilipczuk, Polynomial-time algorithm for maximum weight independent set on P_6-free graphs, Proc. SODA 2019 (2019) 1257–1271.
18. C. Groenland, K. Okrasa, P. Rzążewski, A. Scott, P. Seymour, and S. Spirkl, H-colouring P_t-free graphs in subexponential time, Discrete Applied Mathematics, to appear.
19. T.R. Hartinger, M. Johnson, M. Milanic and D. Paulusma, The price of connectivity for transversals, European Journal of Combinatorics 58 (2016) 203–224.
20. M. Johnson, G. Paesani and D. Paulusma, Connected vertex cover for $(sP_1 + P_5)$-free graphs, Proc. WG 2018, Lecture Notes in Computer Science 11159 (2018) 279–291.
21. D. Lokshtanov, P. Misra, M.S. Ramanujan, S. Saurabh, Hitting selected (odd) cycles, SIAM Journal on Discrete Mathematics 31 (2017) 1581–1615.
22. G.J. Minty, On maximal independent sets of vertices in claw-free graphs, Journal of Combinatorial Theory, Series B 28 (1980) 284–304.
23. N. Misra, G. Philip, V. Raman, and S. Saurabh. On parameterized independent feedback vertex set. Theoretical Computer Science 461 (2012) 65–75.
24. R. Mosca, Stable sets for $(P_6, K_{2,3})$-free graphs, Discussiones Mathematicae Graph Theory 32 (2012) 387–401.
25. A. Munaro, Boundary classes for graph problems involving non-local properties, Theoretical Computer Science 692 (2017) 46–71.
26. K. Okrasa and P. Rzążewski, Subexponential algorithms for variants of homomorphism problem in string graphs, Proc. WG 2019, LNCS, to appear.
27. C. Papadopoulos and S. Tzimas, Polynomial-time algorithms for the subset feedback vertex set problem on interval graphs and permutation graphs, Proc. FCT 2017, Lecture Notes in Computer Science 10472 (2017) 381–394.
28. C. Papadopoulos and S. Tzimas, Subset feedback vertex set on graphs of bounded independent set size, Proc. IPEC 2018, LIPIcs 115 (2018) 20:1–20:14.
29. S. Poljak, A note on stable sets and colorings of graphs, Commentationes Mathematicae Universitatis Carolinae. 15 (1974) 307–309.
30. P.K. Priyadarsini and T. Hemalatha, Connected vertex cover in 2-connected planar graph with maximum degree 4 is NP-complete, International Journal of Mathematical, Physical and Engineering Sciences. 2 (2008) 51–54.
31. S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa, A new algorithm for generating all the maximal independent sets, SIAM Journal on Computing 6 (1977) 505–517.
32. N. Sbihi, Algorithme de recherche d'un stable de cardinalité maximum dans un graphe sans étoile, Discrete Mathematics 29 (1980) 53–76.
33. S. Ueno, Y. Kajitani and S. Gotoh, On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three, Discrete Mathematics 72 (1988) 355–360.
34. T. Wanatabe, S. Kajita and K. Onaga, Vertex covers and connected vertex covers in 3-connected graphs, Proc. IEEE International Symposium on Circuits and Systems 1991, 1017–1020.