

Computing Subset Transversals in H -Free Graphs [†]

Nick Brettell¹, Matthew Johnson², Giacomo Paesani², and Daniël Paulusma²

¹ School of Mathematics and Statistics, Victoria University of Wellington, New Zealand
nick.brettell@vuw.ac.nz

² Department of Computer Science, Durham University, UK
{matthew.johnson2,giacomo.paesani,daniel.paulusma}@durham.ac.uk

Abstract. We study the computational complexity of two well-known graph transversal problems, namely SUBSET FEEDBACK VERTEX SET and SUBSET ODD CYCLE TRANSVERSAL, by restricting the input to H -free graphs, that is, to graphs that do not contain some fixed graph H as an induced subgraph. By combining known and new results, we determine the computational complexity of both problems on H -free graphs for every graph H except when $H = sP_1 + P_4$ for some $s \geq 1$. As part of our approach, we introduce the SUBSET VERTEX COVER problem and prove that it is polynomial-time solvable for $(sP_1 + P_4)$ -free graphs for every $s \geq 1$.

Keywords. feedback vertex; odd cycle transversal; hereditary graph class; H -free, complexity dichotomy

1 Introduction

The central question in Graph Modification is whether or not a graph G can be modified into a graph from a prescribed class \mathcal{G} via at most k graph operations from a prescribed set S of permitted operations such as vertex or edge deletion. The *transversal* problems VERTEX COVER, FEEDBACK VERTEX SET and ODD CYCLE TRANSVERSAL are classical problems of this kind. For example, the VERTEX COVER problem is equivalent to asking if one can delete at most k vertices to turn G into a member of the class of edgeless graphs. The problems FEEDBACK VERTEX SET and ODD CYCLE TRANSVERSAL ask if a graph G can be turned into, respectively, a forest or a bipartite graph by deleting vertices.

We can relax the condition on belonging to a prescribed class to obtain some related *subset transversal* problems. We state these formally after some definitions. For a graph $G = (V, E)$ and a set $T \subseteq V$, a T -cycle is a (not necessarily induced) cycle of G that intersects T . A T -cycle is *odd* if it has an odd number of vertices. A set $S_T \subseteq V$ is a T -vertex cover of G if S_T contains at least one of the two end-vertices for every edge incident to a vertex of T . A set $S_T \subseteq V$ is a T -feedback vertex set or an *odd T -cycle transversal* of G if S_T contains at least one vertex of every T -cycle, or every odd T -cycle, respectively. For example, let G be a star with center vertex c , whose leaves form the set T . Then, both $\{c\} = V \setminus T$ and T are T -vertex covers of G but the first is considerably smaller than the second. See Figures 1 and 2 for some more examples.

SUBSET VERTEX COVER

Instance: a graph $G = (V, E)$, a subset $T \subseteq V$ and a positive integer k .

Question: does G have a T -vertex cover S_T with $|S_T| \leq k$?

SUBSET FEEDBACK VERTEX SET

Instance: a graph $G = (V, E)$, a subset $T \subseteq V$ and a positive integer k .

Question: does G have a T -feedback vertex set S_T with $|S_T| \leq k$?

[†] This paper received support from the Leverhulme Trust (RPG-2016-258). An extended abstract of it appeared in the proceedings of WG 2020 [9].

SUBSET ODD CYCLE TRANSVERSAL

Instance: a graph $G = (V, E)$, a subset $T \subseteq V$ and a positive integer k .

Question: does G have an odd T -cycle transversal S_T with $|S_T| \leq k$?

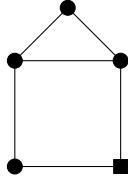


Fig. 1. The house in which we let T consist of the square vertex. The empty set is not an odd T -cycle transversal, as there exists a non-induced odd cycle (containing all the vertices of the graph) passing through the square vertex. Hence, the size of a minimum odd T -cycle transversal is 1, and in particular, this example shows that looking for a minimum odd T -cycle transversal (the problem we consider) or a minimum odd *induced* T -cycle transversal are two different problems.

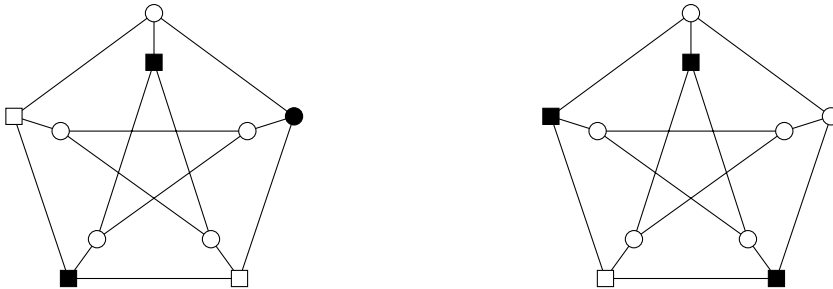


Fig. 2. In both examples, the square vertices of the Petersen graph form a set T and the black vertices form an odd T -cycle transversal S_T , which is also a T -feedback vertex set. In the left example, $S_T \cap (V \setminus T) \neq \emptyset$, and in the right example, $S_T \subseteq T$.

The SUBSET FEEDBACK VERTEX SET and SUBSET ODD CYCLE TRANSVERSAL problems are well known. The SUBSET VERTEX COVER problem is introduced in this paper, and we are not aware of past work on this problem (see Section 6 for open problems). On general graphs, SUBSET VERTEX COVER is polynomially equivalent to VERTEX COVER: to solve SUBSET VERTEX COVER remove edges in the input graph that are not incident to any vertex of T to yield an equivalent instance of VERTEX COVER. However, this equivalence no longer holds for graph classes that are *not* closed under edge deletion.

As the three problems are NP-complete, we consider the restriction of the input to special graph classes in order to better understand which graph properties cause the computational hardness. Instead of classes closed under edge deletion, we focus on classes of graphs closed under vertex deletion. Such classes are called *hereditary*. The reasons for this choice are threefold. First, hereditary graph classes capture many well-studied graph classes. Second, every hereditary graph class \mathcal{G} can be characterized by a (possibly infinite) set $\mathcal{F}_{\mathcal{G}}$ of forbidden induced subgraphs. This enables us to initiate a *systematic* study, starting from the case where $|\mathcal{F}_{\mathcal{G}}| = 1$. Third, we aim to extend and strengthen existing complexity results (that are for hereditary graph classes). If $\mathcal{F}_{\mathcal{G}} = \{H\}$ for some graph H , then \mathcal{G} is *monogenic*, and every $G \in \mathcal{G}$ is *H-free*. Our research question is:

How does the structure of a graph H influence the computational complexity of a subset transversal problem for input graphs that are H -free?

As a general strategy one might first try to prove that the restriction to H -free graphs is NP-complete if H contains a cycle or an induced claw (the 4-vertex star). This is usually done by showing, respectively, that the problem is NP-complete on graphs of arbitrarily large girth (the length of a shortest cycle) and on line graphs, which form a subclass of claw-free graphs. If this is the case, then it remains to consider the case where H has no cycle, and has no claw either. So H is a *linear forest*, that is, the disjoint union of one or more paths.

Existing Results. As NP-completeness results for transversal problems carry over to subset transversal problems, we first discuss results on FEEDBACK VERTEX SET and ODD CYCLE TRANSVERSAL for H -free graphs. By Poljak's construction [37], FEEDBACK VERTEX SET is NP-complete for graphs of girth at least g for every integer $g \geq 3$. The same holds for ODD CYCLE TRANSVERSAL [11]. Moreover, FEEDBACK VERTEX SET [34] and ODD CYCLE TRANSVERSAL [11] are NP-complete for line graphs and thus for claw-free graphs. Hence, both problems are NP-complete for H -free graphs if H has a cycle or claw. Both problems are polynomial-time solvable for P_4 -free graphs [6], for sP_2 -free graphs for every $s \geq 1$ [11] and for $(sP_1 + P_3)$ -free graphs for every $s \geq 1$ [16]. In addition, ODD CYCLE TRANSVERSAL is NP-complete for $(P_2 + P_5, P_6)$ -free graphs [16]. Very recently, Abrishami et al. showed that FEEDBACK VERTEX SET is polynomial-time solvable for P_5 -free graphs [1]. We summarize as follows ($F \subseteq_i G$ means that F is an induced subgraph of G ; see Section 2 for the other notation used).

Theorem 1. *For a graph H , FEEDBACK VERTEX SET on H -free graphs is polynomial-time solvable if $H \subseteq_i P_5$, $H \subseteq_i sP_1 + P_3$ or $H \subseteq_i sP_2$ for some $s \geq 1$, and is NP-complete if $H \supseteq_i C_r$ for some $r \geq 3$ or $H \supseteq_i K_{1,3}$.*

Theorem 2. *For a graph H , ODD CYCLE TRANSVERSAL on H -free graphs is polynomial-time solvable if $H = P_4$, $H \subseteq_i sP_1 + P_3$ or $H \subseteq_i sP_2$ for some $s \geq 1$, and is NP-complete if $H \supseteq_i C_r$ for some $r \geq 3$, $H \supseteq_i K_{1,3}$, $H \supseteq_i P_6$ or $H \supseteq_i P_2 + P_5$.*

We note that no integer r is known such that FEEDBACK VERTEX SET is NP-complete for P_r -free graphs. This situation changes for SUBSET FEEDBACK VERTEX SET which is, unlike FEEDBACK VERTEX SET, NP-complete for split graphs (that is, $(2P_2, C_4, C_5)$ -free graphs), as shown by Fomin et al. [19]. Papadopoulos and Tzimas [35,36] proved that SUBSET FEEDBACK VERTEX SET is polynomial-time solvable for sP_1 -free graphs for any $s \geq 1$, co-bipartite graphs, interval graphs and permutation graphs, and thus P_4 -free graphs. Some of these results were generalized by Bergougnoux et al. [3], who solved an open problem of Jaffke et al. [25] by giving an $n^{O(w^2)}$ -time algorithm for SUBSET FEEDBACK VERTEX SET given a graph and a decomposition of this graph of mim-width w . This does not lead to new results for H -free graphs: a class of H -free graphs has bounded mim-width if and only if $H \subseteq_i P_4$ [8].

We are not aware of any results on SUBSET ODD CYCLE TRANSVERSAL for H -free graphs, but note that this problem generalizes ODD MULTIWAY CUT, just as SUBSET FEEDBACK VERTEX SET generalizes NODE MULTIWAY CUT, another well-studied problem. We refer to a large body of literature [2,12,15,19,20,22,24,27,28,29,30] for further details, in particular for parameterized and exact algorithms for SUBSET FEEDBACK VERTEX SET and SUBSET ODD CYCLE TRANSVERSAL. These algorithms are beyond the scope of this paper.

Our Results. We significantly extend the known results for SUBSET FEEDBACK VERTEX SET and SUBSET ODD CYCLE TRANSVERSAL on H -free graphs. These new results lead us to the following two almost-complete dichotomies:

Theorem 3. *Let H be a graph with $H \neq sP_1 + P_4$ for all $s \geq 1$. Then SUBSET FEEDBACK VERTEX SET on H -free graphs is polynomial-time solvable if $H = P_4$ or $H \subseteq_i sP_1 + P_3$ for some $s \geq 1$, and is NP-complete otherwise.*

Theorem 4. *Let H be a graph with $H \neq sP_1 + P_4$ for all $s \geq 1$. Then SUBSET ODD CYCLE TRANSVERSAL on H -free graphs is polynomial-time solvable if $H = P_4$ or $H \subseteq_i sP_1 + P_3$ for some $s \geq 1$ and NP-complete otherwise.*

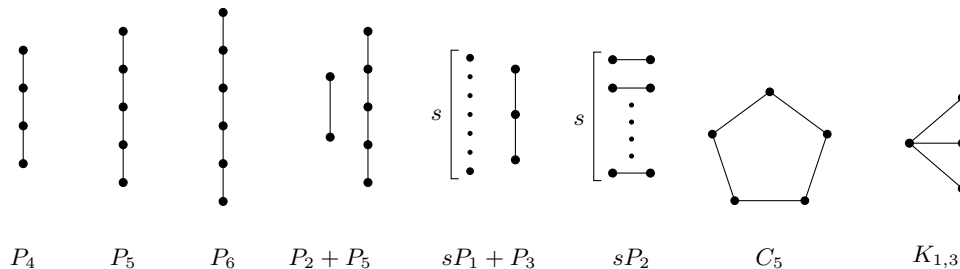


Fig. 3. The forbidden graphs of Theorems 1–4.

Though the proved complexities of SUBSET FEEDBACK VERTEX SET and SUBSET ODD CYCLE TRANSVERSAL are the same on H -free graphs, the algorithm that we present for SUBSET ODD CYCLE TRANSVERSAL on $(sP_1 + P_3)$ -free graphs is more technical compared to the algorithm for SUBSET FEEDBACK VERTEX SET, and considerably generalizes the transversal algorithms for $(sP_1 + P_3)$ -free graphs of [16]. There is further evidence that SUBSET ODD CYCLE TRANSVERSAL is a more challenging problem than SUBSET FEEDBACK VERTEX SET. For example, the best-known parameterized algorithm for SUBSET FEEDBACK VERTEX SET runs in $O^*(4^k)$ time [24], but the best-known run-time for SUBSET ODD CYCLE TRANSVERSAL is $O^*(2^{O(k^3 \log k)})$ [30]. Moreover, it is not known if there is an XP algorithm for SUBSET ODD CYCLE TRANSVERSAL in terms of mim-width in contrast to the known XP algorithm for SUBSET FEEDBACK VERTEX SET [3].

In Section 2 we introduce our terminology. In Section 3 we present some results for SUBSET VERTEX COVER: the first result shows that SUBSET VERTEX COVER is polynomial-time solvable for $(sP_1 + P_4)$ -free graphs for every $s \geq 1$, and we later use this as a subroutine to obtain a polynomial-time algorithm for SUBSET ODD CYCLE TRANSVERSAL on P_4 -free graphs. We present our results on SUBSET FEEDBACK VERTEX SET and SUBSET ODD CYCLE TRANSVERSAL in Sections 4 and 5, respectively. In Section 6 on future work we discuss SUBSET VERTEX COVER in more detail.

2 Preliminaries

We consider undirected, finite graphs with no self-loops and no multiple edges. Let $G = (V, E)$ be a graph, and let $S \subseteq V$. The graph $G[S]$ is the subgraph of G induced by S . We write $G - S$ to denote the graph $G[V \setminus S]$. Recall that for a graph F , we write $F \subseteq_i G$ if F is an induced subgraph of G . The cycle and path on r vertices are denoted C_r and P_r , respectively. We say that S is *independent* if $G[S]$ is edgeless, and that S is a *clique* if $G[S]$ is *complete*, that is, contains every possible edge between two vertices. We let K_r denote the complete graph on r vertices, and sP_1 denote the graph whose vertices form an independent set of size s . A (*connected*) *component* of G is a maximal connected subgraph of G . The graph $\overline{G} = (V, \{uv \mid uv \notin E \text{ and } u \neq v\})$ is the *complement* of G . The *neighbourhood* of a vertex $u \in V$ is the set $N_G(u) = \{v \mid uv \in E\}$. For $U \subseteq V$, we let $N_G(U) = \bigcup_{u \in U} N(u) \setminus U$. The *closed neighbourhoods* of u and U are denoted by $N_G[u] = N_G(u) \cup \{u\}$ and $N_G[U] = N_G(U) \cup U$, respectively. We omit subscripts when there is no ambiguity.

Let $T \subseteq V$ be such that $S \cap T = \emptyset$. Then S is *complete* to T if every vertex of S is adjacent to every vertex of T , and S is *anti-complete* to T if there are no edges between S

and T . In the first case, S is also said to be *complete* to $G[T]$, and in the second case we say it is *anti-complete* to $G[T]$.

We say that G is a *forest* if it has no cycles, and, furthermore, that G is a *linear forest* if it is the disjoint union of one or more paths. The graph G is *bipartite* if V can be partitioned into at most two independent sets. A graph is *complete bipartite* if its vertex set can be partitioned into two independent sets X and Y such that X is complete to Y . We denote such a graph by $K_{|X|,|Y|}$. If X or Y has size 1, the complete bipartite graph is a *star*; recall that $K_{1,3}$ is also called a *claw*. A graph G is a *split graph* if it has a bipartition (V_1, V_2) such that $G[V_1]$ is a clique and $G[V_2]$ is an independent set. A graph is *split* if and only if it is $(C_4, C_5, 2P_2)$ -free [18].

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 (recall that $G_1 + G_2$ is the graph with vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2)$). The *join* operation adds an edge between every vertex of G_1 and every vertex of G_2 . The 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, [7]). It is also well known [13] that a graph G is a cograph if and only if G allows a unique tree decomposition called the *cotree* T_G of G , which has the following properties:

1. The root r of T_G corresponds to the graph $G_r = G$.
2. Each leaf x of T_G corresponds to exactly one vertex of G , and vice versa. Hence x corresponds to a unique single-vertex graph G_x .
3. Each internal node x of T_G has at least two children, is labelled \oplus or \otimes , and corresponds to an induced subgraph G_x of G defined as follows:
 - if x is a \oplus -node, then G_x is the disjoint union of all graphs G_y where y is a child of x ;
 - if x is a \otimes -node, then G_x is the join of all graphs G_y where y is a child of x .
4. Labels of internal nodes on the (unique) path from any leaf to r alternate between \oplus and \otimes .

Note that T_G has $O(n)$ vertices. We modify T_G into a *modified cotree* T'_G in which each internal node has exactly two children by applying the following well-known procedure (see for example [4]). If an internal node x of T_G has more than two children y_1 and y_2 , remove the edges xy_1 and xy_2 and add a new vertex x' with edges xx' , $x'y_1$ and $x'y_2$. If x is a \oplus -node, then x' is a \oplus -node. If x is a \otimes -node, then x' is a \otimes -node. Applying this rule exhaustively yields T'_G . As T_G has $O(n)$ vertices, constructing T'_G from T_G takes linear time. This leads to the following result, due to Corneil, Perl and Stewart, who proved it for cotrees.

Lemma 1 ([14]). *Let G be a graph with n vertices and m edges. Then deciding whether or not G is a cograph, and constructing a modified cotree T'_G (if it exists) takes time $O(n + m)$.*

We also consider optimization versions of subset transversal problems, in which case we have instances (G, T) (instead of instances (G, T, k)). We say that a set $S \subseteq V(G)$ is a *solution* for an instance (G, T) if S is a T -transversal (of whichever kind we are concerned with). A solution S is *smaller* than a solution S' if $|S| < |S'|$, and a solution S is *minimum* if (G, T) does not have a solution smaller than S , and it is *maximum* if there is no larger solution. We will use the following general lemma, which was implicitly used in [36].

Lemma 2. *Let S be a minimum solution for an instance (G, T) of a subset transversal problem. Then $|S \setminus T| \leq |T \setminus S|$.*

Proof. For contradiction, assume that $|S \setminus T| > |T \setminus S|$. Then $|T| < |S|$ (see also Figure 4). This means that T is a smaller solution than S , a contradiction. \square

Let $T \subseteq V$ be a vertex subset of a graph $G = (V, E)$. Recall that a cycle is a T -cycle if it contains a vertex of T . A subgraph of G is a T -forest if it has no T -cycles. Recall also that a

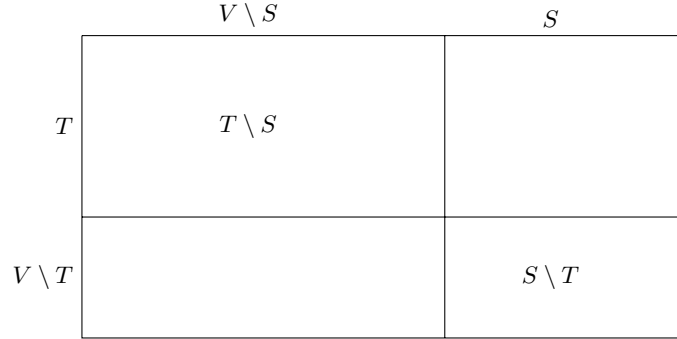


Fig. 4. For a minimum solution S for an instance (G, T) of a subset transversal problem, it must hold that $|S \setminus T| \leq |T \setminus S|$ (see the proof of Lemma 2).

cycle is odd if it has an odd number of edges. A subgraph of G is T -bipartite if it has no odd T -cycles. We recall that a set $S_T \subseteq V$ is a T -vertex cover of G if S_T has at least one vertex of every edge incident to a vertex of T . We also recall that a set $S_T \subseteq V$ is a T -feedback vertex set or an odd T -cycle transversal of G if S_T has at least one vertex of every T -cycle or every odd T -cycle, respectively. Note that S_T is a T -feedback vertex set if and only if $G[V \setminus S_T]$ is a T -forest, and S_T is an odd T -cycle transversal if and only if $G[V \setminus S_T]$ is T -bipartite. A T -path is a path that contains a vertex of T . A T -path is *odd* (or *even*) if the number of edges in the path is odd (or even, respectively).

We will use the following easy lemma, which proves that T -forests and T -bipartite graphs can be recognized in polynomial time. It combines results claimed but not proved in [30,36].

Lemma 3. *Let $G = (V, E)$ be a graph and $T \subseteq V$. Then deciding whether or not G is a T -forest or T -bipartite takes $O(n + m)$ time.*

Proof. A block of G is a maximal 2-connected subgraph of G and is *non-trivial* if it contains a cycle, or, equivalently, at least three vertices. Suppose that we have a block decomposition of G ; it is well known that this can be found in $O(n + m)$ time (see for example [23]). It is clear that G is a T -forest if and only if no non-trivial block contains a vertex of T . We claim that G is T -bipartite if and only if no non-bipartite block contains a vertex of T . To see this note first that the sufficiency is obvious. We will show that if a vertex t of T belongs to a block B that contains an odd cycle C , then t belongs to an odd cycle. If t is in C , we are done. Otherwise find two paths P and P' from t to, respectively, distinct vertices u and u' in C . We can assume that the paths contain no other vertex of C (else we truncate them) and that, as B is 2-connected, they contain no common vertex other than t . We can form two cycles that contain t by adding to $P + P'$ each of the two paths between u and u' in C . As C is an odd cycle, the lengths of these two paths, and therefore the lengths of the two cycles, have distinct parity. Thus t belongs to an odd cycle. Finally we note that the checks of the block decomposition needed to decide whether or not G is a T -forest or T -bipartite can be done in $O(n + m)$ time. \square

3 Subset Vertex Cover

In this section we present some results on SUBSET VERTEX COVER, some of which we will need later on.

Lemma 4. SUBSET VERTEX COVER can be solved in polynomial time for P_4 -free graphs.

Proof. Let G be a cograph with n vertices and m edges. First construct a modified cotree T'_G and then consider each node of T'_G starting at the leaves of T'_G and ending at the root r . Let x be a node of T'_G . We let S_x denote a minimum $(T \cap V(G_x))$ -vertex cover of G_x .

If x is a leaf, then G_x is a 1-vertex graph. Hence, we can let $S_x = \emptyset$. Now suppose that x is a \oplus -node. Let y and z be the two children of x . Then, as G_x is the disjoint union of G_y and G_z , we can let $S_x = S_y \cup S_z$. Finally suppose that x is a \otimes -node. Let y and z be the two children of x . As G_x is the join of G_y and G_z we observe the following: if $V(G_x) \setminus S_x$ contains a vertex of $T \cap V(G_y)$, then $V(G_z) \subseteq S_x$. Similarly, if $V(G_x) \setminus S_x$ contains a vertex of $T \cap V(G_z)$, then $V(G_y) \subseteq S_x$. Hence, we let S_x be the smallest set of $S_y \cup V(G_z)$, $S_z \cup V(G_y)$ and $T \cap V(G_x)$.

Constructing T'_G takes $O(n+m)$ time by Lemma 1. As $T_{G'}$ has $O(n)$ nodes and processing a node takes $O(1)$ time, the total running time is $O(n+m)$. \square

The following lemma generalizes a corresponding well-known observation for VERTEX COVER.

Lemma 5. *Let H be a graph. If SUBSET VERTEX COVER is polynomial-time solvable for H -free graphs, then it is for $(P_1 + H)$ -free graphs as well.*

Proof. Let $G = (V, E)$ be a $(P_1 + H)$ -free graph and let $T \subseteq V$. Let S_T be a minimum T -vertex cover of G . For each vertex $u \in T$ we consider the option that u belongs to $V \setminus S_T$. If so, then $N(u)$ belongs to S_T . Let $G' = G - N[u]$ and let $T' = T \setminus N[u]$. As G' is H -free, we find a minimum T' -vertex cover $S_{T'}$ of G' in polynomial time. We remember the smallest set $S_{T'} \cup N(u)$ and compare it with the size of T to find S_T (or some other minimum solution for (G, T)). \square

Lemma 4, combined with s applications of Lemma 5, yields the following result.

Theorem 5. *For every integer $s \geq 1$, SUBSET VERTEX COVER can be solved in polynomial time for $(sP_1 + P_4)$ -free graphs.*

4 Subset Feedback Vertex Set

In this section we prove Theorem 3. Our contribution is Theorem 6, which is the case where $H = sP_1 + P_3$. In the next section, we present an analogous result for SUBSET ODD CYCLE TRANSVERSAL. The proofs are similar in outline, but the latter requires additional insights.

We require two lemmas. In the first lemma, note that the bound of $4s - 2$ is not necessarily tight, but is sufficient for our needs.

Lemma 6. *Let s be a non-negative integer, and let R be an $(sP_1 + P_3)$ -free tree. Then either*

- (i) $|V(R)| \leq \max\{7, 4s - 2\}$, or
- (ii) R has precisely one vertex r of degree more than 2 and at most $s - 1$ vertices of degree 2, each adjacent to r ; moreover, r has at least $3s - 1$ neighbours.

Proof. If R has no vertices of degree more than 2, then R is a path, so $|V(R)| \leq 2s + 2 \leq \max\{7, 4s - 2\}$, otherwise R has an induced $sP_1 + P_3$ subgraph. Now let r be a vertex of degree more than 2, and let x, y and z be distinct neighbours of r . We view r as the root of the tree, and for $v \in V(R)$ we use T_v to denote the subtree rooted at v .

Suppose that T_x has a vertex of degree at least 2. Then T_x has an induced P_3 , so $R - (V(T_x) \cup \{r\})$ is sP_1 -free. As a tree is bipartite, this means that this subtree consists of at most $2(s - 1)$ vertices. Likewise, $R[\{y, r, z\}] \cong P_3$, so $T_x - x$ is sP_1 -free, and hence consists of at most $2(s - 1)$ vertices. Thus $|V(R)| \leq 2(s - 1) + 2(s - 1) + 2 = 4s - 2$.

We may now assume that for each $v \in N(r)$, the subtree T_v has no vertices of degree at least 2; that is, either $T_v \cong P_1$ or $T_v \cong P_2$. It remains to show that when (i) does not hold, at

most $s - 1$ of the T_v subgraphs are isomorphic to P_2 . Towards a contradiction, suppose that R has s vertices at distance 2 from r , and $|V(R)| > \max\{7, 4s - 2\}$. Since $|V(R)| > 2(s + 1) + 1$ for any non-negative integer s , the vertex r has at least $s + 2$ neighbours. Without loss of generality, label the neighbours of r as $v_1, v_2, \dots, v_{deg(r)}$ such that $T_{v_i} \cong P_2$ for each $i \in \{1, \dots, s\}$. Then $R[v_{s+1}, r, v_{s+2}] \cong P_3$, and $T_{v_i} - \{v_i\} \cong P_1$ for each $i \in \{1, \dots, s\}$; a contradiction.

Finally, $|N_R(r)| + (s - 1) + 1 \geq |V(R)| \geq 4s - 1$, so $|N_R(r)| \geq 3s - 1$. \square

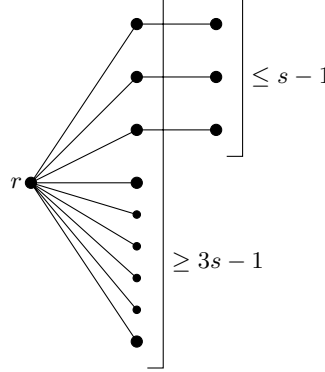


Fig. 5. The structure of an $(sP_1 + P_3)$ -free tree, as given by Lemma 6, when (i) does not hold.

Recall that for a graph $G = (V, E)$ and a set $S \subseteq V$, we let $G - S$ denote the graph $G[V \setminus S]$. We can extend “partial” solutions to full solutions in polynomial time as follows.

Lemma 7. *Let $G = (V, E)$ be a graph with a set $T \subseteq V$. Let $V' \subseteq V$ and $S'_T \subseteq V'$ such that S'_T is a T -feedback vertex set of $G[V']$, and let $Z = V \setminus V'$. Suppose that $G[Z]$ is P_3 -free, and $|N_{G-S'_T}(Z)| \leq 1$. Then there is a polynomial-time algorithm that finds a minimum T -feedback vertex set S_T of G such that $S'_T \subseteq S_T$ and $V' \setminus S'_T \subseteq V \setminus S_T$.*

Proof. Since $G[Z]$ is P_3 -free, it is a disjoint union of complete graphs. Let $G' = G - S'_T$. Suppose that C is a T -cycle in G' . Then C contains at least one vertex of Z . If $N_{G'}(Z) = \emptyset$, then C is contained in a component of $G[Z]$. On the other hand, if $N_{G'}(Z) = \{y\}$, say, then y is a cutvertex of G' , so there exists a component $G[U]$ of $G[Z]$ such that C is contained in $G[U \cup \{y\}]$. Hence, we can consider each component of $G[Z]$ independently: for each component $G[U]$ it suffices to find the maximum subset U' of U such that $G[U' \cup N_{G'}(U)]$ contains no T -cycles. Then $U' \subseteq F_T$ and $U \setminus U' \subseteq S_T$, where $F_T = V \setminus S_T$. So, S_T will be the union of S'_T and the vertex sets $U \setminus U'$, for every component $G[U]$ of $G[Z]$. Hence, it remains to prove how to find the sets U' in polynomial time; we show this below.

Let $U \subseteq Z$ such that $G[U]$ is a component of $G[Z]$. Either $N_{G'}(U) \cap T = \emptyset$, or $N_{G'}(U) = \{y\}$ for some $y \in T$. First, consider the case where $N_{G'}(U) \cap T = \emptyset$. We find a set U' that is a maximum subset of U such that $G[U' \cup N_{G'}(U)]$ has no T -cycles. Clearly if $|U| = 1$, then we can set $U' = U$. If $|U'| \geq 3$, then, since U' is a clique, $U' \subseteq V \setminus T$. Thus, if $|U \setminus T| \geq 2$, then we set $U' = U \setminus T$. So it remains to consider when $|U| \geq 2$ but $|U \setminus T| \leq 1$. If there is some $u \in U$ that is anti-complete to $N_{G'}(U)$, then we can set U' to be any 2-element subset of U containing u . Otherwise $N_{G'}(U) = \{y\}$ and y is complete to U . In this case, for any $u \in U$, we set $U' = \{u\}$.

Now we may assume that $N_{G'}(U) = \{y\}$ and $y \in T$. Again, we find a set U' that is a maximum subset of U such that $G[U' \cup \{y\}]$ has no T -cycles. Partition U into $\{U_0, U_1\}$ where

$u \in U_1$ if and only if u is a neighbour of y . Since $y \in V' \setminus S'_T$, observe that U' contains at most one vertex of U_1 , otherwise $G[U' \cup \{y\}]$ has a T -cycle. Since U' is a clique, if $|U'| \geq 3$ then $U' \subseteq U \setminus T$. So if $|U_0 \setminus T| \geq 2$ and there is an element $u \in U_1 \setminus T$, then we can set $U' = \{u\} \cup (U_0 \setminus T)$. If $|U_0 \setminus T| \geq 2$ but $U_1 \setminus T = \emptyset$, then we can set $U' = U_0 \setminus T$. So we may now assume that $|U_0 \setminus T| \leq 1$. If $U_0 \neq \emptyset$ and $|U| \geq 2$, then we set U' to any 2-element subset of U containing some $u \in U_0$. Clearly if $|U| = 1$, then we can set $U' = U$. So it remains to consider when $U_0 = \emptyset$ and $|U_1| \geq 2$. In this case, we set $U' = \{u\}$ for an arbitrary $u \in U_1$. \square

We now prove the main result of this section.

Theorem 6. *For every integer $s \geq 0$, SUBSET FEEDBACK VERTEX SET 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 for some $s \geq 0$, and let $T \subseteq V$. We describe a polynomial-time algorithm for the optimization version of the problem on input (G, T) . Let $S_T \subseteq V$ such that S_T is a minimum T -feedback vertex set of G , and let $F_T = V \setminus S_T$, so $G[F_T]$ is a maximum T -forest. Note that $G[F_T \cap T]$ is a forest. We consider three cases: either

1. $G[F_T \cap T]$ has at least $2s$ components;
2. $G[F_T \cap T]$ has fewer than $2s$ components, and each of these components consists of at most $\max\{7, 4s - 2\}$ vertices; or
3. $G[F_T \cap T]$ has fewer than $2s$ components, one of which consists of at least $\max\{8, 4s - 1\}$ vertices.

We describe polynomial-time subroutines that find a set F_T such that $G[F_T]$ is a maximum T -forest in each of these three cases, giving a minimum solution $S_T = V \setminus F_T$ in each case. We obtain an optimal solution by running each of these subroutines in turn: of the (at most) three potential solutions, we output the one with minimum size.

Case 1: $G[F_T \cap T]$ has at least $2s$ components.

We begin by proving a sequence of claims that describe properties of a maximum T -forest F_T , when in Case 1. Since G is $(sP_1 + P_3)$ -free, $F_T \cap T$ induces a P_3 -free forest, so $G[F_T \cap T]$ is a disjoint union of graphs isomorphic to P_1 or P_2 . Let $A \subseteq F_T \cap T$ such that $G[A]$ consists of precisely $2s$ components. Note that $|A| \leq 4s$. We also let $Y = N(A) \cap F_T$, and partition Y into $\{Y_1, Y_2\}$ where $y \in Y_1$ if y has only one neighbour in A , whereas $y \in Y_2$ if y has at least two neighbours in A .

Claim 1: $|Y_2| \leq 1$.

Proof of Claim 1. Let $v \in Y_2$. Then v has neighbours in at least $s + 1$ of the components of $G[A]$, otherwise $G[A \cup \{v\}]$ contains an induced $sP_1 + P_3$. Note also that v has at most one neighbour in each component of $G[A]$, otherwise $G[F_T]$ has a T -cycle. Now suppose that Y_2 contains distinct vertices v_1 and v_2 . Then, of the $2s$ components of $G[A]$, the vertices v_1 and v_2 each have some neighbour in $s + 1$ of these components. So there are at least two components of $G[A]$ containing both a vertex adjacent to v_1 , and a vertex adjacent to v_2 . Let A' and A'' be the vertex sets of two such components. Then $A' \cup A'' \cup \{v_1, v_2\} \subseteq F_T$, but $G[A' \cup A'' \cup \{v_1, v_2\}]$ has a T -cycle; a contradiction. \diamond

Claim 2: $|Y| \leq 2s + 1$.

Proof of Claim 2. By Claim 1, it suffices to prove that $|Y_1| \leq 2s$. We argue that each component of $G[A]$ has at most one neighbour in Y_1 , implying that $|Y_1| \leq 2s$. Indeed, suppose that there is a component $G[C_A]$ of $G[A]$ having two neighbours in Y_1 , say u_1 and u_2 . Then $G[V(C_A) \cup \{u_1, u_2\}]$ contains an induced P_3 that is anti-complete to $A \setminus V(C_A)$, contradicting that G is $(sP_1 + P_3)$ -free. \diamond

Claim 3: Y_1 is independent, and no component of $G[A]$ of size 2 has a neighbour in Y_1 .

Proof of Claim 3. Suppose that there are adjacent vertices u_1 and u_2 in Y_1 . Let a_i be the unique neighbour of u_i in A for $i \in \{1, 2\}$. Note that $a_1 \neq a_2$, for otherwise $G[F_T]$ has a T -cycle. Then $\{a_1, u_1, u_2\}$ induces a P_3 , so $G[\{u_1, u_2\} \cup A]$ contains an induced $sP_1 + P_3$, which is a contradiction. We deduce that Y_1 is independent.

Now let $\{a_1, a_2\} \subseteq A$ such that $G[\{a_1, a_2\}]$ is a component of $G[A]$, and suppose that $u_1 \in Y_1$ is adjacent to a_1 . Then a_1 is the unique neighbour of u_1 in A , so $G[\{u_1, a_1, a_2\}] \cong P_3$. Thus $G[\{u_1\} \cup A]$ contains an induced $sP_1 + P_3$, which is a contradiction. \diamond

Claim 4: Let $Z = V \setminus N[A]$. Then $N(Z) \cap F_T \subseteq Y_2$.

Proof of Claim 4. Suppose that there exists $y \in Y_1$ that is adjacent to a vertex $c \in Z$. Let a be the unique neighbour of y in A . Then $G[\{c, y\} \cup A]$ contains an induced $sP_1 + P_3$, which is a contradiction. So Y_1 is anti-complete to Z . Now, if $c \in Z$ is adjacent to a vertex in $N[A] \cap F_T$, then c is adjacent to y_2 where $Y_2 = \{y_2\}$. \diamond

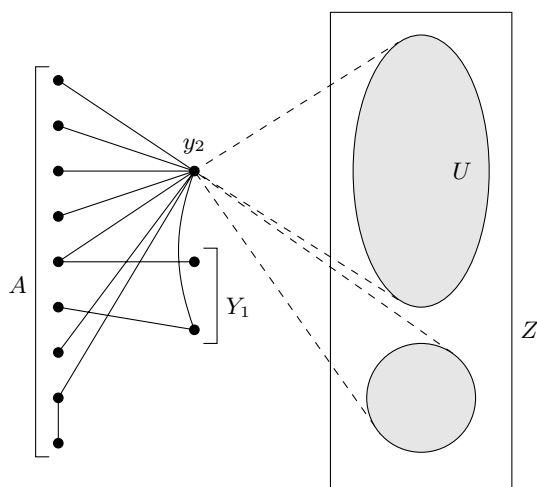


Fig. 6. An example of the structure obtained in Case 1 when $Y_2 = \{y_2\}$.

We now describe the subroutine that finds an optimal solution in Case 1. In this case, for any maximum forest F_T , there exists some set $A \subseteq T$ of size at most $4s$ such that $A \subseteq F_T$, and $G[A]$ consists of exactly $2s$ components, each isomorphic to either P_1 or P_2 . Since $G[A]$ consists of components of $G[F_T \cap T]$, there is such an A for which $N(A) \cap T \subseteq S_T$. Thus we guess a set $A' \subseteq T$ in $O(n^{4s})$ time, discarding those sets that do not induce a forest with exactly $2s$ components, and those that induce a component consisting of more than two vertices.

For any such F_T and A' , the set $N(A') \cap F_T$ has size at most $2s + 1$, by Claim 2. Thus, in $O(n^{2s+1})$ time, we guess $Y' \subseteq N(A')$ with $|Y'| \leq 2s + 1$, and assume that $Y' \subseteq F_T$ whereas $N(A') \setminus Y' \subseteq S_T$. Let Y'_2 be the subset of Y' that contains vertices that have at least two neighbours in A' . We discard any sets Y' that do not satisfy Claims 1 or 3, or those sets for which $G[A' \cup Y']$ has a T -cycle on three vertices, one of which is the unique vertex of Y'_2 .

Let $Z = V \setminus N[A']$ (for example, see Figure 6). Since $G[A']$ contains an induced sP_1 , the subgraph $G[Z]$ is P_3 -free. Now $N(Z) \cap F_T \subseteq Y'_2$ by Claim 4, where $|Y'_2| \leq 1$ by Claim 1. Thus, by Lemma 7, we can extend a partial solution $S'_T = N[A'] \setminus (A' \cup Y')$ of $G[N[A']]$ to a solution S_T of G , in polynomial time.

Case 2: $G[F_T \cap T]$ has at most $2s - 1$ components, each of size at most $\max\{7, 4s - 2\}$.

We guess sets $F \subseteq T$ and $S \subseteq V \setminus T$ such that $F_T \cap T = F$ and $S_T \setminus T = S$. Since F has size at most $(2s-1) \max\{7, 4s-2\}$ vertices, there are $O(n^{\max\{14s-7, 8s^2-8s+2\}})$ possibilities for F . By Lemma 2, we may assume that $|S_T \setminus T| \leq |F|$. So for each guessed F , there are at most $O(n^{\max\{14s-7, 8s^2-8s+2\}})$ possibilities for S . For each S and F , we set $S_T = (T \setminus F) \cup S$ and check, in $O(n+m)$ -time by Lemma 3, if $G - S_T$ is a T -forest. In this way we exhaustively find all solutions satisfying Case 2, in $O(n^{\max\{14s-7, 8s^2-8s+2\}})$ time; we output the one of minimum size.

Case 3: $G[F_T \cap T]$ has at most $2s-1$ components, one of which has size at least $\max\{8, 4s-1\}$.

By Lemma 6, there is some subset $B_T \subseteq F_T \cap T$ such that $|B_T| \geq \max\{8, 4s-1\}$, and $G[B_T]$ is a component of $G[F_T \cap T]$ that is a tree satisfying Lemma 6(ii), as illustrated in Figure 5. In particular, there is a unique vertex $r \in B_T$ such that r has degree more than 2 in $G[B_T]$. Moreover, $G[F_T]$ has a component $G[D]$ that contains B_T , where $G[D]$ is a tree that also satisfies Lemma 6(ii). Note that there are at most $s-1$ vertices in $N_{G[B_T]}(r)$ having a neighbour in $V \setminus T$.

We guess a set $B' \subseteq T$ such that $|B'| = \max\{8, 4s-1\}$. We also guess a set $L' \subseteq V \setminus T$ such that $|L'| \leq s-1$. Let $D' = B' \cup L'$. We check that $G[D']$ has the following properties:

- $G[D']$ is a tree,
- $G[D']$ has a unique vertex r' of degree more than 2, with $r' \in B'$,
- $G[D']$ has at most $s-1$ vertices with distance 2 from r' , and each of these vertices has degree 1, and
- each vertex $v \in L'$ has degree 1 in $G[D']$, and distance 2 from r' .

We assume that D' induces a subtree of the large component $G[D]$, where $r = r'$, and D' contains r , all neighbours of r with degree 2 in $G[D]$, and all vertices at distance 2 from r . In other words, $G[D']$ can be obtained from $G[D]$ by deleting some subset of the leaves of $G[D]$ that are adjacent to r . In particular, $D' \subseteq F_T$. We also assume that L' is the set of all vertices of $D \setminus T$ that have distance 2 from r .

It follows from these assumptions that $N(D' \setminus \{r\}) \setminus \{r\} \subseteq S_T$. Let $Z = V \setminus N[D' \setminus \{r\}]$, and observe that each $z \in Z$ has at most one neighbour in D' (if it has such a neighbour, this neighbour is r). So $N(Z) \cap F_T \subseteq \{r\}$.

In order to apply Lemma 7, it remains to show that $G[Z]$ is P_3 -free. Let $B_1 = B' \cap N(r)$. As r has at least $3s-1$ neighbours in $G[B']$, by Lemma 6, $G[B_1]$ contains an induced sP_1 . Moreover, $N(B_1) \cap F_T \subseteq D'$. Since G is $(sP_1 + P_3)$ -free, $G[Z]$ is P_3 -free. Thus, by Lemma 7, we can extend a partial solution $S'_T = N(D' \setminus \{r\}) \setminus \{r\}$ of $G[N[D' \setminus \{r\}]]$ to a solution S_T of G , in polynomial time. \square

We are now ready to prove Theorem 3.

Theorem 3 (restated). *Let H be a graph with $H \neq sP_1 + P_4$ for all $s \geq 1$. Then SUBSET FEEDBACK VERTEX SET on H -free graphs is polynomial-time solvable if $H = P_4$ or $H \subseteq_i sP_1 + P_3$ for some $s \geq 1$, and is NP-complete otherwise.*

Proof. If H has a cycle or claw, we use Theorem 1. The cases $H = P_4$ and $H = 2P_2$ follow from the corresponding results for permutation graphs [35] and split graphs [19]. The remaining case $H \subseteq_i sP_1 + P_3$ follows from Theorem 6. \square

5 Subset Odd Cycle Transversal

At the end of this section we prove Theorem 4. We need three new results to combine with existing knowledge. Our first result uses the reduction of [35] which proved the analogous result for SUBSET FEEDBACK VERTEX SET.

Theorem 7. SUBSET ODD CYCLE TRANSVERSAL is NP-complete for the class of split graphs (or equivalently, $(C_4, C_5, 2P_2)$ -free graphs).

Proof. We observe that the problem belongs to NP by Lemma 3. To show NP-hardness, we reduce from VERTEX COVER. Let a graph $G = (V, E)$ and a positive integer k be an instance of VERTEX COVER. From G , we construct a graph G' as follows. Let $V(G') = V \cup E$. Add an edge between $e \in E$ and $v \in V$ in G' if and only if v is an end-vertex of e in G . Add edges so that V induces a clique of G' . Hence, G' is a split graph with independent set E and clique V . For example, when $G = P_4$, see Figure 7. Let $T = E$. We show that G has a vertex cover of size at most k if and only if G' has an odd T -cycle transversal of size at most k .

First suppose that G has a vertex cover S of size at most k . Then S is an odd T -cycle transversal of G' . Now suppose that G' has an odd T -cycle transversal S_T of size at most k . As every vertex of E in G' has degree 2, we can replace every vertex of E that belongs to S_T by one of its neighbours to obtain an odd T -cycle transversal of the same size as S_T . Hence we may assume, without loss of generality, that $S_T \cap E = \emptyset$. As a vertex of E and its two neighbours in V form a triangle, this means that S_T contains at least one neighbour of every $e \in E$. Hence, S_T is a vertex cover of G . \square

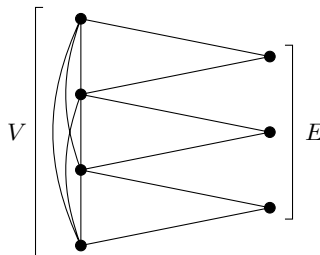


Fig. 7. The graph P'_4 : an example of the construction in the proof of Theorem 7.

Theorem 8. SUBSET ODD CYCLE TRANSVERSAL can be solved in polynomial time for P_4 -free graphs.

Proof. Let G be a cograph with n vertices and m edges, and let $T \subseteq V(G)$. First construct the modified cotree T'_G and then consider each node of T'_G starting at the leaves of T'_G and ending in its root r . Let x be a node of T'_G . We let S_x denote a minimum odd $(T \cap V(G_x))$ -cycle transversal of G_x .

If x is a leaf, then G_x is a 1-vertex graph. Hence, we can let $S_x = \emptyset$. Now suppose that x is a \oplus -node. Let y and z be the two children of x . Then, as G_x is the disjoint union of G_y and G_z , we let $S_x = S_y \cup S_z$.

Finally suppose that x is a \otimes -node. Let y and z be the two children of x . Let $T_y = T \cap V(G_y)$ and $T_z = T \cap V(G_z)$. Let $B_x = V(G_x) \setminus S_x$. As G_x is the join of G_y and G_z we observe the following. If $B_x \cap V(G_y)$ contains two adjacent vertices, at least one of which belongs to T_x , then $B_x \cap V(G_z) = \emptyset$ (as otherwise $G[B_x]$ has a triangle containing a vertex of T) and thus $V(G_z) \subseteq S_x$. In this case we may assume that $S_x = S_y \cup V(G_z)$. Similarly, if $B_x \cap V(G_z)$ contains two adjacent vertices, at least one of which belongs to T_z , then $B_x \cap V(G_y) = \emptyset$ and thus $V(G_y) \subseteq S_x$. In this case we may assume that $S_x = S_z \cup V(G_y)$. From the two sets $S_y \cup V(G_z)$ and $S_z \cup V(G_y)$ we remember the smallest one.

It remains to examine the case where $B_x \cap V(G_y)$ and $B_x \cap V(G_z)$ induce subgraphs of G in which the vertices of $T_y \cap B_x$ and $T_z \cap B_x$, respectively, are singleton components.

First suppose that $T_y \cap B_x$ and $T_z \cap B_x$ are both non-empty. Then $B_x \cap V(G_y)$ and $B_x \cap V(G_z)$ are both independent sets, as otherwise $G[B_x]$ would contain a T -triangle. We examine this situation by computing a largest independent set I_y in G_y and a largest independent set I_z in G_z ; it is well-known that this can be done in polynomial time (for example, it follows from Lemma 4). We remember $V(G_x) \setminus (I_y \cup I_z)$. Since the union of two independent sets induces a bipartite graph, this is an odd T -cycle transversal.

Now suppose that $T_y \cap B_x$ is non-empty, but $T_z \cap B_x$ is empty. Then $B_x \cap V(G_z)$ must be an independent set, as otherwise we obtain a T -triangle by taking a vertex of $T_y \cap B_x$ and two adjacent vertices of $B_x \cap V(G_z)$. First assume that $B_x \cap V(G_z)$ has size at least 2. We observe that $(B_x \cap V(G_y)) \setminus T_y$ is also an independent set; otherwise two adjacent vertices of $(B_x \cap V(G_y)) \setminus T_y$, two vertices of $B_x \cap V(G_z)$ and one vertex of $T_y \cap B_x$ would form a T -cycle on five vertices. Hence, both $B_x \cap V(G_y)$ and $B_x \cap V(G_z)$ are independent sets, and we already dealt with this case above.

Now assume that $B_x \cap V(G_z)$ has size at most 1. In this case $B_x \cap V(G_y)$ is a minimum T_y -vertex cover of G_y . We can compute a minimum T_y -vertex cover S of G_y in polynomial time by Lemma 4. We remember $S \cup (V(G_z) \setminus \{z'\})$ where z' is an arbitrary vertex of $V(G_z) \setminus T_z$ if the latter set is non-empty; otherwise we just remember $S \cup V(G_z)$.

We deal with the case where $T_z \cap B_x$ is non-empty, but $T_y \cap B_x$ is empty in the same way and remember the output. We also consider the possible situation where $T_z \cap B_x = T_y \cap B_x = \emptyset$, in which case we remember $T \cap V(G_x)$. Finally, we take as set S_x a set of minimum size over the sets that we remembered.

Constructing T'_G takes $O(n+m)$ time by Lemma 1. As $T_{G'}$ has $O(n)$ nodes and processing a node takes $O(n+m)$ time (due to the application of Lemma 4), the total running time is $O(n^2 + mn)$. \square

The following result is the main result of this section. Its proof uses the same approach as the proof of Theorem 6 but we need more advanced arguments.

Theorem 9. *For every integer $s \geq 0$, SUBSET ODD CYCLE TRANSVERSAL 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 and let $T \subseteq V$. Note that it suffices to prove the result for $s \geq 3$, since for every $0 \leq s' \leq s$, the class of $(s'P_1 + P_3)$ -free graphs is contained in the class of $(sP_1 + P_3)$ -free graphs. We describe a polynomial-time algorithm to solve the optimization problem on input (G, T) . That is, we describe how to find a smallest odd T -cycle transversal S_T . In fact, we will solve the equivalent problem of finding a maximum size set B_T such that $G[B_T]$ is T -bipartite; so, of course, B_T is the complement of a smallest odd T -cycle transversal, that is $S_T = V \setminus B_T$. We separate into two cases that separately seek to find T -bipartite subgraphs with complementary constraints on the size of the intersection of this subgraph with T . The largest one found overall is the desired output.

Case 1: Compute a largest T -bipartite subgraph $G[B_T]$ such that $|B_T \cap T| \leq 4s - 3$.

Note that $B^* = V \setminus T$ is a candidate solution. We must see if we can find something larger. Consider each set $B' \subseteq T$ of size at most $4s - 3$, discarding any set that does not induce a bipartite graph. There are $O(n^{4s-3})$ possible sets. For each choice of B' , consider all sets $S \subseteq V \setminus T$ of size less than $|B'|$. Then $B' \cup ((V \setminus T) \setminus S)$ is a candidate solution if it induces a T -bipartite subgraph, which is checked in $O(n+m)$ -time by Lemma 3. For each B' , there are $O(n^{4s-3})$ possible choices of S to consider. Note that we do not need to examine larger sets S since then $B' \cup ((V \setminus T) \setminus S)$ is no larger than B^* .

Case 2: Compute a largest T -bipartite subgraph $G[B_T]$ such that $|B_T \cap T| \geq 4s - 2$.

Note that B_T might not exist in which case the output of Case 1 is our result. We make some observations about the subgraph $G[B_T]$ that we seek. As $G[B_T \cap T]$ is a bipartite graph on at least $4s - 2$ vertices, it contains an independent set A of size $2s - 1$. Let $Y = B_T \cap N(A)$

and consider a partition $\{Y_1, Y_2\}$ of Y where y is in Y_1 if y has precisely one neighbour in A , and otherwise y is in Y_2 .

Claim 1: Y_1 is an independent set, no two vertices of Y_1 have a common neighbour in A and $|Y_1| \leq |A|$.

Proof of Claim 1. Suppose that there are adjacent vertices $y, y' \in Y_1$, and let a be the unique neighbour of y in A . Then, according to whether or not y' is adjacent to a , either $\{y, y', a\}$ induces an odd T -cycle, or $G[A \cup \{y, y'\}]$ contains an induced $sP_1 + P_3$ (recall that $s \geq 3$). Both are contradictions. If there are two (non-adjacent) vertices $y, y' \in Y_1$ that have the same neighbour a in A , then, again, $G[A \cup \{y, y'\}]$ contains an induced $sP_1 + P_3$, a contradiction. It follows that no two vertices of Y_1 have a common neighbour in A , and hence $|Y_1| \leq |A|$. \diamond

Claim 2: Y_2 is an independent set, each $y \in Y_2$ has at least s neighbours in A and any two vertices of Y_2 share at least one neighbour in A .

Proof of Claim 2. Let y and y' be two distinct vertices in Y_2 . By definition, y and y' each have at least two neighbours in A . Since $G[A \cup \{y\}]$ is $(sP_1 + P_3)$ -free, y is non-adjacent to at most $s - 1$ vertices of A . So y has at least $2s - 1 - (s - 1) = s$ neighbours in A . Similarly, y' has at least s neighbours in A . As $|A| = 2s - 1$, this means that y and y' have a common neighbour $a \in A$. If y and y' are adjacent, then $\{y, y', a\}$ would induce an odd T -cycle, a contradiction. \diamond

Armed with these definitions and claims, we show how to find B_T in polynomial time. We have two subcases.

Case 2a: Compute a largest T -bipartite subgraph $G[B_T]$ such that $|B_T \cap T| \geq 4s - 2$ and for some choice of A , we have $|Y| \leq 3s$.

Consider each set $A \subseteq T$ of size $2s - 1$ such that A is an independent set. There are $O(n^{2s-1})$ choices for A . For each A , we consider each set $Y_1 \subseteq N(A)$ of vertices that each have a single neighbour in A such that Y_1 satisfies Claim 1. As we require that $|Y_1| \leq |A|$, there are $O(n^{2s-1})$ choices for Y_1 . Then consider each set $Y \subseteq N(A)$ of size at most $3s$ such that $Y_1 \subseteq Y$ and $Y_2 = Y \setminus Y_1$ is a set of vertices that each have at least two neighbours in A and satisfies Claim 2. We also require that $A \cup Y$ does not induce any odd T -cycles, which is checked in $O(n + m)$ -time by Lemma 3. There are $O(n^{3s})$ choices for Y .

Note that $G[A \cup Y]$ is bipartite since $G[Y]$ can contain only even cycles as Y_1 and Y_2 are independent sets, and any odd cycle in $G[A \cup Y]$ must thus contain a vertex of A and so is an odd T -cycle, since $A \subseteq T$. By Claim 2, vertices of Y_2 all belong to the same component of $G[A \cup Y]$ and, as, by definition and Claim 1, each vertex in $G[A \cup Y_1]$ has degree at most 1, we deduce that every vertex of degree at least 2 in $G[A \cup Y]$ belongs to the same component. We denote this component by $G[D]$, or we let D be the empty set if there is no such component (which only occurs when $Y_2 = \emptyset$). See Figure 8 for an illustration.

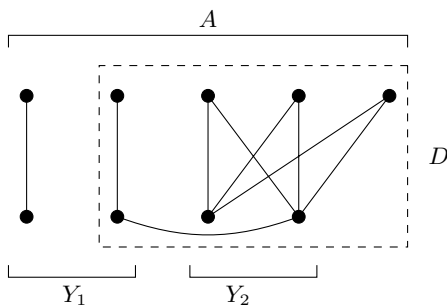


Fig. 8. An example of $G[A \cup Y]$ in Case 2a when $s = 3$.

We now let $Z = V \setminus N[A]$.

Claim 3: $N(Z) \cap B_T \subseteq Y_2$.

Proof of Claim 3. By definition, $N(Z) \cap B_T \subseteq Y$. Suppose that $z \in Z$ is adjacent to a vertex $y \in Y_1$. Let a be the unique neighbour of y in A . Since $|A| = 2s - 1 \geq s + 1$ as $s \geq 3$, it follows that $G[\{z, y\} \cup A]$ contains an induced $sP_1 + P_3$, a contradiction. So Y_1 is anti-complete to Z , and the claim follows. \diamond

The aim in the remainder of this subcase is to find the largest possible set B_T that induces a T -bipartite subgraph and that contains $A \cup Y$ and a subset of Z . Since $G[A]$ contains an induced sP_1 subgraph, $G[Z]$ is P_3 -free, and so is a disjoint union of complete graphs. From Claim 3, it follows that $(A \cup Y) \setminus D$ (which is a subset of $A \cup Y_1$) is anti-complete to $B_T \setminus (A \cup Y)$. Therefore, we only need to consider odd T -cycles in $G[Z \cup D]$ from now on.

For a component $G[U]$ of $G[Z]$, let U^+ be the subset of U that consists of each vertex u of U such that $G[A \cup Y \cup \{u\}]$ does not contain an odd T -cycle through u , which is checked in $O(n + m)$ -time by Lemma 3. Clearly for each component $G[U]$ in $G[Z]$, any vertex that might be in B_T must belong to U^+ . We shall see later that we can consider each component of $G[Z]$ independently and that it suffices to find for each the maximum size subset of U^+ that can be added to B_T . We note that by Claim 3, the neighbours of U^+ in $A \cup Y$ belong to Y_2 and, as $Y_2 \subseteq D$, these neighbours therefore also belong to D . We first investigate the possible edges between U^+ and D .

Claim 4: Either $|N(U^+ \cap B_T) \cap D| \leq 1$ or $|N(D) \cap (U^+ \cap B_T)| \leq 1$.

Proof of Claim 4. We can assume that there are two distinct vertices u_1, u_2 of $U^+ \cap B_T$ that have a different neighbour in D else the claim follows immediately. Let y_1 be a neighbour of u_1 in D and y_2 be a neighbour of u_2 in D with $y_1 \neq y_2$. By Claim 2, y_1 and y_2 have a common neighbour a in A . Thus we have a path $u_1 y_1 a y_2 u_2$. Recall that $G[Z]$ is a disjoint union of complete graphs. Hence, U^+ is a clique, which means that the 5-vertex path $u_1 y_1 a y_2 u_2$ can be extended to a cycle by the edge $u_1 u_2$, but, as $A \subseteq T$, this is an odd T -cycle, a contradiction. \diamond

Claim 5: For each component $G[U]$ of $G[Z]$, let U^{++} be a subset of U^+ ; and let Z^{++} be the union of each U^{++} over all components $G[U]$ of $G[Z]$. If $G[A \cup Y \cup Z^{++}]$ contains an odd T -cycle, then $G[A \cup Y \cup U^{++}]$ contains an odd T -cycle for some component $G[U]$ of $G[Z]$.

Proof of Claim 5. Suppose that C is an odd T -cycle of $G[A \cup Y \cup Z^{++}]$. First we show that C contains two vertices of some U^{++} . Towards a contradiction, suppose C is a subgraph of $G[A \cup Y \cup Z^*]$, where Z^* is a subset of Z^{++} with at most one vertex from each component of $G[Z]$. Recall that D is a bipartite graph that (if non-empty) is a component of $G[A \cup Y]$. By Claim 3, all neighbours of Z^* are contained in Y_2 , which, in turn, is contained in one side of the bipartition of D . Hence $G[A \cup Y \cup Z^*]$ is bipartite and has no odd T -cycles and, in particular, C is not an odd T -cycle. From this contradiction we deduce that there is some component $G[U]$ of $G[Z]$ such that C contains two vertices of U^{++} .

We can assume that C is not contained in $G[A \cup Y \cup U^{++}]$ else the claim follows immediately. Then there are distinct vertices $u_1 \in U^{++}$ and $u_2 \in U^{++}$ and distinct vertices $y_1 \in N_C(u_1) \cap Y_2$ and $y_2 \in N_C(u_2) \cap Y_2$. By Claim 2, y_1 and y_2 have a common neighbour $a \in A$. Then, as U , and thus U^{++} , is a clique, $u_1 y_1 a y_2 u_2 u_1$ is an odd T -cycle contained in $G[A \cup Y \cup U^{++}]$, a contradiction. \diamond

By Claim 5, to extend $A \cup Y$ to the largest possible T -bipartite graph, for each component $G[U]$ of $G[Z]$, we must find in polynomial time a maximum subset U^{++} of U^+ such that $G[A \cup Y \cup U^{++}]$ has no odd T -cycle.

We find a set U^{++} as follows. We first suppose that for the set we seek $|U^{++}| \geq 3$. Note that in this case we have $U^{++} \cap T = \emptyset$, since U^{++} is a clique. Partition $U^+ \setminus T$ into $\{U_0^+, U_1^+, U_2^+\}$ where $u \in U_0^+$ if $u \in U^+ \setminus T$ has no neighbours in D , $u \in U_1^+$ if u has exactly one neighbour in D , and otherwise $u \in U_2^+$. If U_1^+ is not empty, then let

$N(U_1^+) \cap D = \{d_1, \dots, d_m\}$, for some $m \geq 1$. We partition U_1^+ into classes $\{Q_1, \dots, Q_m\}$ such that $u \in Q_i$ if $N(u) \cap D = \{d_i\}$. Using Claim 4, either $U^{++} \cap U_1^+ = \emptyset$ or $U^{++} \cap U_2^+ = \emptyset$. Moreover, when $U^{++} \cap U_1^+ \neq \emptyset$, then $U^{++} \cap U_1^+ \subseteq Q_i$ for some $i \in \{1, \dots, m\}$; and when $U^{++} \cap U_2^+ \neq \emptyset$, then $|U^{++} \cap U_2^+| = 1$. So, if there exists some $d_i \notin T$, then we choose an i with $d_i \notin T$ that maximises $|Q_i|$, and set $U^{++} = U_0^+ \cup Q_i$. If $U_1^+ \neq \emptyset$ but $d_i \in T$ for all $i \in \{1, \dots, m\}$, then set $U^{++} = U_0^+ \cup \{u\}$ for an arbitrarily chosen $u \in U_1^+$. Now suppose U_1^+ is empty, and recall that in this case $|U_2^+ \cap U^{++}| \leq 1$. If U_2^+ is non-empty, then set $U^{++} = U_0^+ \cup \{u_2\}$ for some arbitrarily chosen $u_2 \in U_2^+$. Finally, if U_2^+ is also empty, then set $U^{++} = U_0^+$. This process finds a maximum U^{++} of size at least 3 if such a set exists. We note that $G[A \cup Y \cup U^{++}]$ does not contain an odd T -cycle since either the only neighbour of U^{++} in $A \cup Y$ is some $d_i \notin T$, or the only vertex in U^{++} adjacent to $A \cup Y$ does not belong to T .

Now consider the case where $|U^{++}| \leq 2$. We exhaustively check all pairs of vertices in U^+ , of which there are $O(n^2)$. Let u_1, u_2 be such a pair of distinct vertices. We check that $G[A \cup Y \cup \{u_1, u_2\}]$ is T -bipartite; if it is, then we set $U^{++} = \{u_1, u_2\}$. Recall that this check runs in polynomial time, by Lemma 3. If no pair is found, we set U^{++} to be the singleton set consisting of any arbitrarily chosen vertex of U^+ . If U^+ is empty, then we set $U^{++} = \emptyset$. We observe that in all cases we obtained U^{++} in polynomial time, as needed.

Case 2b: Compute a largest T -bipartite subgraph $G[B_T]$ such that $|B_T \cap T| \geq 4s - 2$ and for some choice of A , we have $|Y| \geq 3s + 1$.

Note that as A has size $2s - 1$ and $|Y_1| \leq |A|$, we have that $|Y_2| \geq s + 2$. Let Y'_2 be a subset of Y_2 with $|Y'_2| = s + 2$. Let $A_0 = N(Y'_2) \cap A$. By Claim 2 it follows that $|A_0| \geq s$. As $A_0 \subseteq A$, we also find that $|A_0| \leq 2s - 1$.

Claim 6: Let y and y' be distinct vertices of $N(A_0)$. Then there is an even T -path in $G[A_0 \cup Y'_2 \cup \{y, y'\}]$ between y and y' .

Proof of Claim 6. Assume that y and y' have no common neighbour in A_0 else the claim is immediate. First let us assume at least one of y and y' is in Y'_2 . Without loss of generality, $y \in Y'_2$. Then $y' \in N(A_0) \setminus Y'_2$ by the assumption that y and y' have no common neighbour and Claim 2. Let $a' \in A_0$ be a neighbour of y' . By definition, a' has a neighbour y'' in Y'_2 and $y \neq y''$ else a' is a common neighbour of y and y' . By Claim 2 again, y and y'' have a common neighbour $a'' \in A_0$. Thus $ya''y''a'y'$ is an even T -path in $G[A_0 \cup Y'_2 \cup \{y, y'\}]$ (in particular, note that a' and a'' belong to T , as $A_0 \subseteq A$ belongs to T).

Now we consider the case where $y, y' \in N(A_0) \setminus Y'_2$. Let y^* be a vertex of Y'_2 . By the previous case there are even T -paths in $G[A_0 \cup Y'_2 \cup \{y, y^*\}]$ between y and y^* and between y' and y^* . Consider the subgraph induced by the vertices of these two paths. After removing the edge yy' (if it exists) this subgraph is bipartite, since $G[A_0 \cup Y'_2]$ is bipartite and the neighbours of y and y' are both in A_0 . We then discard edges from (even) cycles until a path between y and y' is obtained. This path is even, and moreover passes through $A_0 \subseteq T$. Hence, it is an even T -path between y and y' in $G[A_0 \cup Y'_2 \cup \{y, y'\}]$. \diamond

Claim 7: $N(A_0) \cap N(Y'_2) \cap B_T = \emptyset$, or equivalently, $N(A_0) \cap N(Y'_2) \subseteq S_T$.

Proof of Claim 7. For contradiction, assume there is a vertex $v \in N(A_0) \cap N(Y'_2) \cap B_T$. By assumption there are vertices $a \in A_0 \cap N(v)$ and $y \in Y'_2 \cap N(v)$. By definition of A_0 , there is a vertex $y' \in Y'_2 \cap N(a)$. If $y' = y$ then $\{a, v, y\}$ induces an odd T -cycle, a contradiction. Suppose now that $y' \neq y$. Then by Claim 6 there is an even T -path P in $G[A_0 \cup Y'_2 \cup \{y, y'\}]$ ($= G[A_0 \cup Y'_2]$) between y and y' . As $Y'_2 \subseteq Y_2$ is an independent set by Claim 2 and A_0 is independent by definition, v does not belong to $A_0 \cup Y'_2$ and hence, v is not on P . If a does not belong to P , then the cycle formed by P and $yvay'$ is an odd T -cycle of B_T . If a does belong to P , then we can divide P into a path P_1 from y to a and a path P_2 from y' to a . As P_1 and P_2 must have the same parity, one of the two cycles formed by P_1 and avy and by P_2 and ay' is an odd T -cycle of B_T . In either case, we have a contradiction. \diamond

By Claim 7, $N(A_0) \cap N(Y'_2) \subseteq S_T$. We let $Y'_0 = N(A_0) \setminus N(Y'_2)$. Note that $Y'_2 \subseteq B_T$ and $Y'_2 \subseteq N(A_0)$. Hence, $Y'_2 \subseteq Y'_0$. Note that Y'_0 contains no neighbours of Y'_2 by definition. Moreover, $Y'_2 \subseteq Y_2$, and Y_2 is an independent set by Claim 2. It follows that the vertices of Y'_2 are isolated in $G[Y'_0]$. Finally, as $|Y'_2| = s + 2$, we observe that $G[Y'_0 \setminus Y'_2]$ is a disjoint union of complete graphs; see also Figure 9.

Let $Z = V \setminus N[A_0]$; see also Figure 9. Recall that A_0 is a subset of A of size at least s and that A is an independent set. Hence, $G[A_0]$ has an induced sP_1 . Consequently, $G[Z]$ is P_3 -free, so $G[Z]$ is a disjoint union of complete graphs. Let $G[U]$ be a component of $G[Z]$; note that U is a clique. Partition the vertices of U into $\{U_0, U_1, U_2\}$, where $u \in U_0$ if u has no neighbours in Y'_0 , whereas $u \in U_1$ if all neighbours of u in Y'_0 are in one component of $G[Y'_0]$, and otherwise $u \in U_2$ (that is, when u has neighbours in distinct components of $G[Y'_0]$).

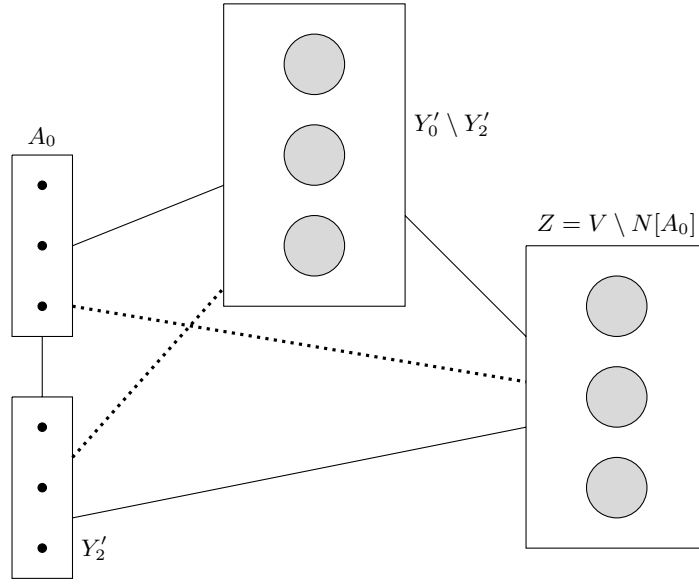


Fig. 9. The decomposition of G in Case 2b excluding the set $N(A_0) \cap N(Y'_2)$ (as this set will belong to S_T). Recall that $A_0 = N(Y'_2) \cap A$ and Y'_2 are independent sets with $s \leq |A_0| \leq 2s - 1$ and $|Y'_2| = s + 2$. Moreover, $Y'_0 = N(A_0) \setminus N(Y'_2)$, so $Y'_0 \setminus Y'_2$ is anti-complete to $Y'_2 \subseteq Y'_0$. The graphs $G[Y'_0]$ and $G[Z]$ are disjoint unions of complete graphs (in particular, the vertices of Y'_2 are isolated in $G[Y'_0]$). Finally, A_0 is anti-complete to $Z = V \setminus N[A_0]$ by definition.

Claim 8: If $u \in U_2$, then u has at least two neighbours in Y'_2 .

Proof of Claim 8. Suppose that $u \in U_2$. By definition, u has neighbours x and y in distinct components of $G[Y'_0]$. By definition, Y'_0 contains no neighbours of Y'_2 . Recall that the vertices of Y'_2 are isolated in $G[Y'_0]$. Hence, $Y'_2 \cup \{x, y\}$ is independent. As G is $(sP_1 + P_3)$ -free and $|Y'_2| = s + 2$, the claim follows. \diamond

Claim 9: Either $U_0 = \emptyset$ or $U_1 = \emptyset$. Moreover, if $U_1 \neq \emptyset$, then $U_1 \cup (N(U_1) \cap Y'_0)$ is a clique.

Proof of Claim 9. For contradiction, assume that U_0 and U_1 are both non-empty. Let $u_0 \in U_0$, $u_1 \in U_1$ and $y \in N(u_1) \cap Y'_0$. Recall that U is a clique, so u_0 and u_1 are adjacent and thus $\{u_0, u_1, y\}$ induces a P_3 . Recall also that Y'_0 contains no neighbours of Y'_2 and that the vertices of Y'_2 form an independent set of size $s + 2 \geq s$ in $G[Y'_0]$. Then $G[\{u_0, u_1, y\} \cup Y'_2]$ contains an induced $sP_1 + P_3$; a contradiction.

Now let u_1 and u'_1 be two distinct vertices in U_1 . Let $y_1 \in N(u_1) \cap Y'_0$ and $y'_1 \in N(u'_1) \cap Y'_0$. If y_1 is not adjacent to u'_1 , then $\{y_1, u_1, u'_1\}$ induces a P_3 . Then, by the same reason as above,

$G[\{u_1, u'_1, y_1\} \cup Y'_2]$ contains an induced $sP_1 + P_3$. Hence, y_1 is adjacent to u'_1 , and by the same arguments, y'_1 is adjacent to u_1 . If $y_1 \neq y'_1$ and y_1 and y'_1 are not adjacent, then $\{y_1, u_1, y'_1\}$ induces a P_3 and thus $G[\{u_1, y_1, y'_1\} \cup Y'_2]$ contains an induced $sP_1 + P_3$; a contradiction. We conclude that $U_1 \cup (N(U_1) \cap Y'_0)$ is a clique. \diamond

Claim 10: $|U_2 \cap B_T| \leq 1$.

Proof of Claim 10. For contradiction, assume there exist vertices $u, u' \in U_2 \cap B_T$ with $u \neq u'$. By Claim 8, we find that u and u' each have at least two neighbours in Y'_2 . Hence, there exist vertices $y, y' \in Y'_2$ such that $y \in N(u)$, $y' \in N(u')$ and $y \neq y'$. By Claim 6, there is an even T -path P in $G[A'_0 \cup Y'_2]$ ($= G[A'_0 \cup Y'_2 \cup \{y, y'\}]$) between y and y' . By using the path $yu u' y'$, we can extend P to an odd T -cycle; a contradiction. \diamond

Claim 11: Suppose that $u_1, u_2 \in B_T$ for some $u_1 \in U_1$ and $u_2 \in U_2$. Let y be a vertex in $N(U_1) \cap Y'_0$, which is a clique. Then $y \in Y'_0 \setminus Y'_2$ and $y \notin B_T$.

Proof of Claim 11. By Claim 8, we find that u_2 has at least two neighbours in Y'_2 . Hence, u_2 has a neighbour $y' \in Y'_2$ such that $y' \neq y$. By Claim 6, there is an even T -path P in $G[A_0 \cup Y'_2 \cup \{y\}]$ between y and y' . As, by Claim 9, $U_1 \cup (N(U_1) \cap Y'_0)$ is a clique, y and u_1 are adjacent. As U is a clique, u_1 and u_2 are adjacent. By using the path $yu_1 u_2 y'$, we can extend P to an odd T -cycle. Since $V(P) \setminus \{y\} \subseteq A_0 \cup Y'_2 \subseteq B_T$ and $u_1, u_2 \in B_T$, this implies that $y \notin B_T$ and thus $y \notin Y'_2$. \diamond

Claim 12: B_T contains at most one vertex of each component of $G[Y'_0]$.

Proof of Claim 12. Recall that $G[Y'_0]$ is the disjoint union of complete graphs. For contradiction, assume that y and y' are vertices in $Y'_0 \cap B_T \subseteq N(A_0) \cap B_T$ that belong to the same complete graph. By Claim 6, there is an even T -path between y and y' that uses only vertices of $A_0 \cup Y'_2 \cup \{y, y'\} \subseteq B_T$. This can be extended to an odd T -cycle by adding the edge between y and y' , a contradiction. \diamond

Claim 13: If the graph $G[A_0 \cup (Y'_0 \cap B_T) \cup Z]$ contains an odd T -cycle C , then there is a component $G[U]$ of $G[Z]$ such that the graph $G[A_0 \cup (Y'_0 \cap B_T) \cup U]$ contains an odd T -cycle C' .

Proof of Claim 13. Recall that A_0 is a subset of B_T . Let C be an odd T -cycle in $G[A_0 \cup (Y'_0 \cap B_T) \cup Z]$. We can assume that C contains vertices from at least two components of $G[Z]$ else we are done. Since $B_T \subseteq A_0 \cup Y'_0 \cup Z$, and Z is anti-complete to A_0 , the cycle C is the concatenation of a number of the following two types of paths on at least two vertices: a path is of *type 1* if it starts and ends in $Y'_0 \cap B_T$ and is contained in $G[A_0 \cup (Y'_0 \cap B_T)]$; a path is of *type 2* if it starts and ends in $Y'_0 \cap B_T$ and all its internal vertices are contained in a component of $G[Z]$. Since A_0 and, by Claim 12, $Y'_0 \cap B_T$ are independent sets, the graph $G[A_0 \cup (Y'_0 \cap B_T)]$ is bipartite and all the subpaths of C of type 1 are even. Moreover, since C is an odd cycle, there is a subpath P of C of type 2 that is odd. Let y and y' be the two end-vertices of P , so y and y' both belong to $Y'_0 \cap B_T$, which is a subset of $N(A_0)$. By Claim 6, there exists an even T -path Q in $G[A_0 \cup Y'_2 \cup \{y, y'\}]$, which is an induced subgraph of $G[A_0 \cup (Y'_0 \cap B_T)]$ as $Y'_2 \cup \{y, y'\} \subseteq Y'_0 \cap B_T$. Let $G[U]$ be the component of $G[Z]$ that contains the internal vertices of P . Joining P and Q yields the desired odd T -cycle C' in $G[A_0 \cup (Y'_0 \cap B_T) \cup U]$. \diamond

Let us now describe how we find the largest possible B_T . Our approach is to first guess a constant number of vertices to be in B_T . To be more precise, we consider each possible pair of sets A_0 , with $s \leq |A_0| \leq 2s - 1$, and Y'_2 , of size $s + 2$, that conform with the definitions of this subcase and Claim 6; note that the latter can be verified in polynomial time as the graph $G[A_0 \cup Y'_2 \cup \{y, y'\}]$ with $y, y' \in N(A_0)$ has constant size. The total number of choices is $O(n^{3s+1})$. For each choice of these two sets we want to find the largest possible B_T that contains $A_0 \cup Y'_2$. Recall again that $N(A_0) \cap N(Y'_2) \subseteq S_T$ by Claim 7. Thus we want to include in B_T as many vertices as possible from the set of vertices that we have not yet assigned to B_T or S_T , that is, from the set $(Y'_0 \setminus Y'_2) \cup Z$.

So our goal is to find a subset $X \subseteq Y'_0 \setminus Y'_2$ and subsets $U' \subseteq U$ for each component $G[U]$ of $G[Z]$ to add to B_T . We will first find each U' and then X . In each case, we will choose the largest possible subset such that the vertices added to B_T so far do not induce an odd T -cycle. We must ask whether this greedy approach will lead us to find the largest possible B_T (given the initial choice of A_0 and Y'_2); that is, does the choice of some U' not affect the later choice of another U' , or of X ?

By Claim 13, we may consider the components of $G[Z]$ independently. Hence, we must just show when we choose a subset U' that this does not affect our later choice of X in a way that will prevent us claiming we have found a maximum size B_T . We will see that sometimes as we choose the largest possible U' , this forces us to put some vertices from $Y'_0 \setminus Y'_2$ into S_T (to avoid $G[B_T]$ containing an odd T -cycle); that is, we remove the possibility that such vertices can later be chosen to be in X . However, when this happens, we will show that in choosing U' , all the vertices of $Y'_0 \setminus Y'_2$ forced into S_T belong to the same component of $G[Y'_0 \setminus Y'_2]$. By Claim 12, any choice of X will contain at most one vertex from this component. Hence, the *cost* of our choice of U' is that X might contain one fewer vertex x than is otherwise possible. We can allow this as long as putting x in B_T prevents us from putting more than one vertex of U in B_T , and we show that this will always be the case.

Before we show how to find U' , let us make one more comment. When U' is found, we must check that, for any choice of X , it holds that $G[A_0 \cup Y'_2 \cup X \cup U']$ does not contain an odd T -cycle. First suppose that $U' \cap U_2 \neq \emptyset$; say U' contains a vertex $u \in U_2$. By Claim 10, we have that U' contains no other vertex of U_2 . By Claim 11, no vertex of $U_1 \cap U'$ has neighbours in $B_T \setminus U'$. Thus u is a cutvertex of $G[B_T]$. Hence, we need only check that each of $G[A_0 \cup Y'_2 \cup X \cup \{u\}]$ and $G[U']$ contains no odd T -cycle. As the former is bipartite, all that we need to confirm is that $G[U']$ does not contain an odd T -cycle, or, equivalently, since U' is a clique, that $|U'| \leq 2$ or $U' \cap T = \emptyset$.

Now suppose that $U' \cap U_2 = \emptyset$. Then U' is a subset of either U_0 or U_1 , as at most one of the latter two sets is non-empty by Claim 9. First suppose that $U' \subseteq U_0$. Then there are no edges from U' to the rest of B_T . So, again, we need only check that $|U'| \leq 2$ or $U' \cap T = \emptyset$. Now suppose that $U' \subseteq U_1$. Then the vertices of U' might have neighbours in $B_T \setminus U'$. These neighbours must lie in Y'_0 , and, by Claim 9, they must form a clique so, by Claim 12, there can only be one such neighbour which we denote y . Thus y is a cutvertex so we need only check that the complete graph $G[U' \cup \{y\}]$ contains no odd T -cycle, or, equivalently, that either $U' \cup \{y\}$ contains only two vertices or no vertex of T .

We now show how to find U' by distinguishing three cases; note that we can check in polynomial time which case applies. We use again that $|U' \cap U_2| \leq 1$ by Claim 10 and at most one of U_0 and U_1 is non-empty by Claim 9. When we have found U' , we must make the checks outlined above.

Case 2bi. $U_1 = \emptyset$.

We build U' by first letting it contain all vertices of $U_0 \setminus T$ and, if $U_2 \setminus T$ is non-empty, one arbitrary vertex from the set $U_2 \setminus T$. If U' now has at least two vertices, then we are done since the only other vertices we could add are in T , but then U' would induce an odd T -cycle. If, however, U' has fewer than two vertices, then we add up to two vertices from $U \cap T$ subject to the constraints that U' can only contain one vertex of U_2 and cannot contain more than two vertices (else, again, we obtain an odd T -cycle). In this way we have made U' as large as possible. As $U' \cap U_1 = \emptyset$, we need only note that U' contains no vertex of T , or has at most two vertices, to confirm that $G[B_T]$ will contain no odd T -cycle using a vertex of U' .

Case 2bii. $U_1 \neq \emptyset$ and $N(U_1) \cap Y'_0$ is a subset of Y'_2 .

By Claim 9, $N(U_1) \cap Y'_0$ is a clique. Hence, in this case, all the vertices of $N(U_1) \cap Y'_0$ belong to the same component of $G[Y'_2]$. As the components of $G[Y'_2]$ are isolated vertices, we have that $N(U_1) \cap Y'_0 = \{y\}$ for some $y \in Y'_2$. As $Y'_2 \subseteq B_T$, we cannot, by Claim 11, add vertices from both U_1 and U_2 to U' . As we can only ever add one vertex from U_2 , we will instead add only vertices from U_1 . If $y \in T$ or $U_1 \subseteq T$, then we must let U' contain only one vertex of U_1

else $U' \cup \{y\}$ will induce an odd T -cycle. Otherwise, we let U' contain all vertices of $U_1 \setminus T$; there is at least one and so we cannot also add any vertex of $U_1 \cap T$ without $U' \cup \{y\}$ inducing an odd T -cycle. We note that we have made U' as large as possible and that $U' \cup \{y\}$ either contains no vertex of T or contains only two vertices.

Case 2biii. $U_1 \neq \emptyset$ and $N(U_1) \cap Y'_0$ is not a subset of Y'_2 .

Again, by Claim 9, $N(U_1) \cap Y'_0$ is a clique. Hence, in this case, all the vertices of $N(U_1) \cap Y'_0$ belong to the same component of $G[Y'_0 \setminus Y'_2]$. We build U' by first letting it contain all vertices of $U_1 \setminus T$ and, if it is non-empty, one vertex from the set $U_2 \setminus T$. If U' now has at least two vertices, then we are done since we cannot add vertices of T . If U' has fewer than two vertices, then we add vertices from $U \cap T$ noting again that U' can only contain one vertex of U_2 and cannot now contain more than two vertices. When we add these vertices, we give preference to vertices from U_1 and so only add a vertex from U_2 if $|U_1| = 1$.

Again we have made U' as large as possible. If U' contains vertices from each of U_1 and U_2 , then note that, by Claim 11, X will contain no vertices from $N(U_1) \cap Y'_0$. Thus these vertices are added to S_T and we will never add one of them to X . Similarly, suppose that $U' \subseteq U_1$ and $|U'| \geq 2$: then X can contain no vertex from $N(U_1) \cap Y'_0$ that is in T and, moreover, can contain none at all from $N(U_1) \cap Y'_0$ if one of the vertices of U' is in T . Again we add vertices to S_T as needed. Note that, considering how U' was built, we cannot avoid this without choosing a smaller U' : if it was possible to find a U' of the same size without using a vertex of U_2 or in T we would have done so. As at most one of the vertices we move to S_T could ever have been added to B_T , there is no cost to this.

It only remains to note that if U' contains a vertex of U_2 , then it contains no vertex of T unless it contains only two vertices, and that if $U' \cap U_2 = \emptyset$, then, for each neighbour $y \in Y'_0$ of U' that has not been placed in S_T , the set $\{y\} \cap U'$ contains no vertex of T unless it has only two vertices.

We have completed the choice of each U' . It now only remains to note that we can build X by choosing one arbitrary vertex from each component of $G[Y'_0]$ excluding, of course, those vertices that we moved to S_T in the process of finding each U' (recall that we cannot add two vertices from a component of $G[Y'_0]$ due to Claim 12).

This completes our proof. We conclude that we found a largest T -bipartite subgraph $G[B_T]$ in polynomial time. Hence, $S_T = V \setminus B_T$ is a smallest odd T -cycle transversal of G . \square

We are now ready to prove our almost-complete classification.

Theorem 4 (restated). *Let H be a graph with $H \neq sP_1 + P_4$ for all $s \geq 1$. Then SUBSET ODD CYCLE TRANSVERSAL on H -free graphs is polynomial-time solvable if $H = P_4$ or $H \subseteq_i sP_1 + P_3$ for some $s \geq 1$, and is NP-complete otherwise.*

Proof. If H has a cycle or claw, we use Theorem 2. The cases $H = P_4$ and $H = 2P_2$ follow from Theorems 7 and 8, respectively. The remaining case, where $H \subseteq_i sP_1 + P_3$, follows from Theorem 9. \square

6 Conclusions

We gave almost-complete classifications of the complexity of SUBSET FEEDBACK VERTEX SET and SUBSET ODD CYCLE TRANSVERSAL for H -free graphs. The only open case in each classification is when $H = sP_1 + P_4$ for some $s \geq 1$, which is also open for FEEDBACK VERTEX SET and ODD CYCLE TRANSVERSAL for H -free graphs. Our proof techniques for $H = sP_1 + P_3$ do not carry over and new structural insights are needed in order to solve the missing cases.

Open Problem 1 *Determine the complexity of (SUBSET) FEEDBACK VERTEX SET and (SUBSET) ODD CYCLE TRANSVERSAL for $(sP_1 + P_4)$ -free graphs, when $s \geq 1$.*

One of the main obstacles to solving Open Problem 1 is the case where there is a solution S such that $G - S$ is a forest that contains (many) arbitrarily large stars. In particular, Lemma 6 no longer holds.

The vertex-weighted version of SUBSET FEEDBACK VERTEX SET has also been studied for H -free graphs. Papadopoulos and Tzimas [36] proved that WEIGHTED SUBSET FEEDBACK VERTEX SET is polynomial-time solvable for $4P_1$ -free graphs but NP-complete for $5P_1$ -free graphs (in contrast to the unweighted version). Bergougnoux et al. [3] recently proved that WEIGHTED SUBSET FEEDBACK VERTEX SET is polynomial-time solvable for every class of bounded mim-width and thus for P_4 -free graphs. Combining these results with Theorem 3 and two recent results, polynomial-time solvable cases when $H \in \{P_1 + P_3, 3P_1 + P_2\}$ [10], leaves three gaps (see also [10]).

Open Problem 2 *Determine the complexity of WEIGHTED SUBSET FEEDBACK VERTEX SET for H -free graphs when $H \in \{2P_1 + P_3, P_1 + P_4, 2P_1 + P_4\}$.*

For the weighted variant, a vertex in T may have a large weight that prevents it from being deleted in any solution; in particular, Lemma 2, which plays a crucial role in our proofs, no longer holds.

As shown in [10], the NP-completeness proof given by Papadopoulos and Tzimas for WEIGHTED SUBSET FEEDBACK VERTEX SET on $5P_1$ -free graphs [36] can also be used to show that the weighted version of SUBSET ODD CYCLE TRANSVERSAL is NP-complete for $5P_1$ -free graphs. In the same paper [10] it is proven that WEIGHTED SUBSET ODD CYCLE TRANSVERSAL is polynomial-time solvable for H -free graphs if $H \in \{P_1 + P_3, 3P_1 + P_2, P_4\}$. Combining these results with Theorem 4 leads to the same three open cases as listed in Open Problem 2.

Open Problem 3 *Determine the complexity of WEIGHTED SUBSET ODD CYCLE TRANSVERSAL for H -free graphs when $H \in \{2P_1 + P_3, P_1 + P_4, 2P_1 + P_4\}$.*

As previously mentioned, Bergougnoux et al. [3] gave an XP algorithm for WEIGHTED SUBSET FEEDBACK VERTEX SET parameterized by mim-width.

Open Problem 4 *Does there exist an XP algorithm for SUBSET ODD CYCLE TRANSVERSAL and WEIGHTED SUBSET ODD CYCLE TRANSVERSAL parameterized by mim-width?*

We also introduced the SUBSET VERTEX COVER problem and showed that this problem is polynomial-time solvable on $(sP_1 + P_4)$ -free graphs for every $s \geq 0$. Lokshantov et al. [31] proved that VERTEX COVER is polynomial-time solvable for P_5 -free graphs. Grzesik et al. [21] extended this result to P_6 -free graphs.

Open Problem 5 *Determine the complexity of SUBSET VERTEX COVER for P_5 -free graphs.*

Open Problem 6 *Determine whether there exists an integer $r \geq 5$ such that SUBSET VERTEX COVER is NP-complete for P_r -free graphs.*

By Poljak's construction [37], VERTEX COVER is NP-complete for H -free graphs if H has a cycle. However, VERTEX COVER becomes polynomial-time solvable on $K_{1,3}$ -free graphs [32,38]. We did not research the complexity of SUBSET VERTEX COVER on $K_{1,3}$ -free graphs and also leave this as an open problem for future work.

Open Problem 7 *Determine the complexity of SUBSET VERTEX COVER for $K_{1,3}$ -free graphs.*

Finally, several related transversal problems have been studied but not yet for H -free graphs. For example, the parameterized complexity of EVEN CYCLE TRANSVERSAL and SUBSET EVEN CYCLE TRANSVERSAL has been addressed in [33] and [27], respectively. Moreover, several variants for transversal problems have been studied for H -free graphs, but not the subset version: for example, CONNECTED VERTEX COVER, CONNECTED FEEDBACK VERTEX SET and CONNECTED ODD CYCLE TRANSVERSAL, and also for INDEPENDENT FEEDBACK VERTEX SET and INDEPENDENT ODD CYCLE TRANSVERSAL; see [5,11,17,26] for a number of recent results. It would be interesting to solve the subset versions of these transversal problems for H -free graphs and to determine the connections amongst all these problems in a more general framework.

Acknowledgments. We thank three anonymous reviewers for helpful comments that improved the presentation of several proofs.

References

1. T. Abrishami, M. Chudnovsky, M. Pilipczuk, P. Rzażewski, and P. Seymour. Induced subgraphs of bounded treewidth and the container method. *Proc. SODA 2021*, 1948–1964, 2021.
2. B. Bergougnoux, É. Bonnet, N. Brettell, and O. Kwon. Close relatives of feedback vertex set without single-exponential algorithms parameterized by treewidth. *Proc. IPEC 2020, LIPIcs*, 180:3:1–3:17, 2020.
3. B. Bergougnoux, C. Papadopoulos, and J. A. Telle. Node multiway cut and subset feedback vertex set on graphs of bounded mim-width. *Proc. WG 2020, LNCS*, 12301:388–400, 2020.
4. H. L. Bodlaender and R. H. Möhring. The pathwidth and treewidth of cographs. *SIAM Journal on Discrete Mathematics*, 6:181–188, 1993.
5. M. Bonamy, K. K. Dabrowski, C. Feghali, M. Johnson, and D. Paulusma. Independent feedback vertex set for P_5 -free graphs. *Algorithmica*, 81:1342–1369, 2019.
6. A. Brandstädt and D. Kratsch. On the restriction of some NP-complete graph problems to permutation graphs. *Proc. FCT 1985, LNCS*, 199:53–62, 1985.
7. A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph Classes: A Survey*, volume 3 of *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, 1999.
8. N. Brettell, J. Horsfield, A. Munaro, G. Paesani, and D. Paulusma. Bounding the mim-width of hereditary graph classes. *Journal of Graph Theory*, 99:117–151, 2022.
9. N. Brettell, M. Johnson, G. Paesani, and D. Paulusma. Computing subset transversals in H -free graphs. *Proc. WG 2020, LNCS*, 12301:187–199, 2020.
10. N. Brettell, M. Johnson, and D. Paulusma. Computing weighted subset transversals in H -free graphs. *Proc. WADS 2021, LNCS*, 12808:229–242, 2021.
11. N. Chiarelli, T. R. Hartinger, M. Johnson, M. Milanič, and D. Paulusma. Minimum connected transversals in graphs: New hardness results and tractable cases using the price of connectivity. *Theoretical Computer Science*, 705:75–83, 2018.
12. R. Chitnis, F. V. Fomin, D. Lokshtanov, P. Misra, M. S. Ramanujan, and S. Saurabh. Faster exact algorithms for some terminal set problems. *Journal of Computer and System Sciences*, 88:195–207, 2017.
13. D. G. Corneil, H. Lerchs, and L. S. Burlingham. Complement reducible graphs. *Discrete Applied Mathematics*, 3:163–174, 1981.
14. D. G. Corneil, Y. Perl, and L. K. Stewart. A linear recognition algorithm for cographs. *SIAM Journal on Computing*, 14:926–934, 1985.
15. M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. O. Wojtaszczyk. Subset feedback vertex set is fixed-parameter tractable. *SIAM Journal on Discrete Mathematics*, 27:290–309, 2013.
16. K. K. Dabrowski, C. Feghali, M. Johnson, G. Paesani, D. Paulusma, and P. Rzażewski. On cycle transversals and their connected variants in the absence of a small linear forest. *Algorithmica*, 82:2841–2866, 2020.
17. K. K. Dabrowski, M. Johnson, G. Paesani, D. Paulusma, and V. Zamaraev. On the price of independence for vertex cover, feedback vertex set and odd cycle transversal. *Proc. MFCS 2018, LIPIcs*, 117:63:1–63:15, 2018.
18. S. Földes and P. L. Hammer. Split graphs. *Congressus Numerantium*, 19:311–315, 1977.

19. F. V. Fomin, P. Heggernes, D. Kratsch, C. Papadopoulos, and Y. Villanger. Enumerating minimal subset feedback vertex sets. *Algorithmica*, 69:216–231, 2014.
20. P. A. Golovach, P. Heggernes, D. Kratsch, and R. Saei. Subset feedback vertex sets in chordal graphs. *Journal of Discrete Algorithms*, 26:7–15, 2014.
21. A. Grzesik, T. Klímašová, M. Pilipczuk, and M. Pilipczuk. Polynomial-time algorithm for maximum weight independent set on P_6 -free graphs. *Proc. SODA 2019*, 1257–1271, 2019.
22. E. C. Hols and S. Kratsch. A randomized polynomial kernel for subset feedback vertex set. *Theory of Computing Systems*, 62:63–92, 2018.
23. J. E. Hopcroft and R. E. Tarjan. Algorithm 447: Efficient algorithms for graph manipulation. *Communications of the ACM*, 16:372–378, 1973.
24. Y. Iwata, M. Wahlström, and Y. Yoshida. Half-integrality, lp-branching, and FPT algorithms. *SIAM Journal on Computing*, 45:1377–1411, 2016.
25. L. Jaffke, O. Kwon, and J. A. Telle. Mim-width II. the feedback vertex set problem. *Algorithmica*, 82:118–145, 2020.
26. M. Johnson, G. Paesani, and D. Paulusma. Connected vertex cover for $(sP_1 + P_5)$ -free graphs. *Algorithmica*, 82:20–40, 2020.
27. N. Kakimura, K. Kawarabayashi, and Y. Kobayashi. Erdős-Pósa property and its algorithmic applications: parity constraints, subset feedback set, and subset packing. *Proc. SODA 2012*, 1726–1736, 2012.
28. K. Kawarabayashi and Y. Kobayashi. Fixed-parameter tractability for the subset feedback set problem and the S-cycle packing problem. *Journal of Combinatorial Theory, Series B*, 102:1020–1034, 2012.
29. S. Kratsch and M. Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. *Journal of the ACM*, 67:16:1-16:50, 2020.
30. D. Lokshtanov, P. Misra, M. S. Ramanujan, and S. Saurabh. Hitting selected (odd) cycles. *SIAM Journal on Discrete Mathematics*, 31:1581–1615, 2017.
31. D. Lokshtanov, M. Vatshelle, and Y. Villanger. Independent set in P_5 -free graphs in polynomial time. *Proc. SODA 2014*, 570–581, 2014.
32. G. J. Minty. On maximal independent sets of vertices in claw-free graphs. *Journal of Combinatorial Theory, Series B*, 28:284–304, 1980.
33. P. Misra, V. Raman, M. S. Ramanujan, and S. Saurabh. Parameterized algorithms for even cycle transversal. *Proc. WG 2012*, 7551:172–183, 2012.
34. A. Munaro. On line graphs of subcubic triangle-free graphs. *Discrete Mathematics*, 340:1210–1226, 2017.
35. C. Papadopoulos and S. Tzimas. Polynomial-time algorithms for the subset feedback vertex set problem on interval graphs and permutation graphs. *Discrete Applied Mathematics*, 258:204–221, 2019.
36. C. Papadopoulos and S. Tzimas. Subset feedback vertex set on graphs of bounded independent set size. *Theoretical Computer Science*, 814:177–188, 2020.
37. S. Poljak. A note on stable sets and colorings of graphs. *Commentationes Mathematicae Universitatis Carolinae*, 15:307–309, 1974.
38. N. Sbihi. Algorithme de recherche d’un stable de cardinalité maximum dans un graphe sans étoile. *Discrete Mathematics*, 29:53–76, 1980.