Computing Weighted Subset Transversals in H-Free Graphs*

Nick Brettell[1][0000−0002−1136−418X], Matthew Johnson[2][0000−0002−7295−2663], and Daniël Paulusma[2][0000−0001−5945−9287]

[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,daniel.paulusma}@durham.ac.uk

Abstract. For the Odd Cycle Transversal problem, the task is to find a small set S of vertices in a graph that intersects every cycle of odd length. The Subset Odd Cycle Transversal requires S to intersect only those odd cycles that include a vertex of a distinguished vertex subset T. If we are given weights for the vertices, we ask instead that S has small weight: this is the problem Weighted Subset Odd Cycle Transversal. We prove an almost-complete complexity dichotomy for Weighted Subset Odd Cycle Transversal for graphs that do not contain a graph H as an induced subgraph. Our general approach can also be used for Weighted Subset Feedback Vertex Set, which enables us to generalize a recent result of Papadopoulos and Tzimas.

1 Introduction

For a *transversal* problem, one seeks to find a small set of vertices within a given graph that intersects every subgraph of a specified kind. Two problems of this type are Feedback Vertex Set and Odd Cycle Transversal, where the objective is to find a small set S of vertices that intersects, respectively, every cycle and every cycle with an odd number of vertices. Equivalently, when S is deleted from the graph, what remains is a forest or a bipartite graph, respectively.

For a *subset transversal* problem, we are also given a vertex subset T and we must find a small set of vertices that intersects every subgraph of a specified kind *that also contains a vertex of* T. An *(odd)* T*-cycle* is a cycle of the graph (with an odd number of vertices) that intersects T. A set $S_T \subseteq V$ is a T*-feedback vertex set* or an *odd* T*-cycle transversal* of a graph $G = (V, E)$ if S_T has at least one vertex of, respectively, every T-cycle or every odd T-cycle; see also Fig. 1. A *(non-negative) weighting* of G is a function $w : V \to \mathbb{R}^+$. For $v \in V$, $w(v)$ is the *weight* of v, and for $S \subseteq V$, the weight $w(S)$ of S is the sum of the weights of the vertices in S. In a *weighted subset transversal* problem the task is to find a transversal whose weight is less than a prescribed bound. We study:

* The research in this paper received support from the Leverhulme Trust (RPG-2016-258).

Weighted Subset Feedback Vertex Set
Instance: a graph G, a subset $T \subseteq V(G)$, a non-negative vertex weighting w of G and an integer $k \geq 1$.
Question: does G have a T-feedback vertex set S_T with $w(S_T) \leq k$?

Weighted Subset Odd Cycle Transversal
Instance: a graph G, a subset $T \subseteq V(G)$, a non-negative vertex weighting w of G and an integer $k \geq 1$.
Question: does G have an odd T-cycle transversal S_T with $w(S_T) \leq k$?

Both problems are NP-complete even when the weighting function is 1 and $T = V$. We continue a systematic study of transversal problems on hereditary graph classes, focusing on the weighted subset variants. *Hereditary* graph classes can be characterized by a set of forbidden induced subgraphs. We begin with the case where this set has size 1: the class of graphs that, for some graph H, do not contain H as an induced subgraph; such a graph is said to be *H-free*.

Past Results. We first note some NP-completeness results for the special case where $w \equiv 1$ and $T = V$, which corresponds to the original problems Feedback Vertex Set and Odd Cycle Transversal. These results immediately imply NP-completeness for the weighted subset problems. By Poljak's construction [14], for every integer $g \geq 3$, Feedback Vertex Set is NP-complete for graphs of finite girth at least g (the girth of a graph is the length of its shortest cycle). There is an analogous result for Odd Cycle Transversal [4]. It has also been shown that Feedback Vertex Set [10] and Odd Cycle Transversal [4] are NP-complete for line graphs and, therefore, also for claw-free graphs. Thus the two problems are NP-complete for the class of H-free graphs whenever H contains a cycle or claw. Of course, a graph with no cycle is a forest, and a forest with no claw has no vertex of degree at least 3. Hence, we need now only focus on the case where H is a *linear forest*, that is, a collection of disjoint paths.

There is no linear forest H for which Feedback Vertex Set on H-free graphs is known to be NP-complete, but for Odd Cycle Transversal we can take $H = P_2 + P_5$ or $H = P_6$, as the latter problem is NP-complete even for $(P_2 + P_5, P_6)$-free graphs [5]. It is known that Subset Feedback Vertex Set [6] and

Fig. 1. Two examples (from [3]) of the Petersen graph with the set T indicated by square vertices. The set S_T of black vertices forms both an odd T-cycle transversal and a T-feedback vertex set. On the left, $S_T \cap T \neq \emptyset$. On the right, $S_T \subseteq T$.

Subset Odd Cycle Transversal [3], which are the special cases with $w \equiv 1$, are NP-complete for $2P_2$-free graphs; in fact, these results were proved for split graphs which form a proper subclass of $2P_2$-free graphs. For the weighted subset problems, there is just one additional case of NP-completeness currently known, from the interesting recent work of Papadopoulos and Tzimas [13] as part of the following dichotomy.

Theorem 1 ([13]). Weighted Subset Feedback Vertex Set *on sP_1-free graphs is polynomial-time solvable if $s \leq 4$ and* NP*-complete if $s \geq 5$.*

The unweighted version of Subset Feedback Vertex Set can be solved in polynomial time for sP_1-free graphs for every $s \geq 1$ [13]. In contrast, for many transversal problems, the complexities on the weighted and unweighted versions for H-free graphs align; see, for example Vertex Cover [7], Connected Vertex Cover [8] and (Independent) Dominating Set [9].

The other known polynomial-time algorithm for Weighted Subset Feedback Vertex Set on H-free graphs is for the case where $H = P_4$. This can be proven in two ways: Weighted Subset Feedback Vertex Set is polynomial-time solvable for permutation graphs [12] and also for graphs for which we can find a decomposition of constant mim-width [2]; both classes contain the class of P_4-free graphs. To the best of our knowledge, algorithms for Weighted Subset Odd Cycle Transversal on H-free graphs have not previously been studied.

We now mention the polynomial-time results on H-free graphs for the unweighted subset variants of the problems (which do not imply anything for the weighted subset versions). Both Subset Feedback Vertex Set and Subset Odd Cycle Transversal are polynomial-time solvable on H-free graphs if $H = P_4$ or $H = sP_1 + P_3$ [3, 12]. Additionally, Feedback Vertex Set is polynomial-time solvable on P_5-free graphs [1] and sP_3-free graphs for every integer $s \geq 1$ [11], and both Feedback Vertex Set and Odd Cycle Transversal are polynomial-time solvable on sP_2-free graphs for every $s \geq 1$ [4].

Our Results. Our main result is the following almost-complete dichotomy. We write $H \subseteq_i G$, or $G \supseteq_i H$ to say that H is an induced subgraph of G.

Theorem 2. *Let H be a graph with $H \notin \{2P_1 + P_3, P_1 + P_4, 2P_1 + P_4\}$. Then* Weighted Subset Odd Cycle Transversal *on H-free graphs is polynomial-time solvable if $H \subseteq_i 3P_1 + P_2$, $P_1 + P_3$, or P_4, and is* NP*-complete otherwise.*

As a consequence, we obtain a dichotomy analogous to Theorem 1.

Corollary 1. *The* Weighted Subset Odd Cycle Transversal *problem on sP_1-free graphs is polynomial-time solvable if $s \leq 4$ and is* NP*-complete if $s \geq 5$.*

For the hardness part of Theorem 2 it suffices to show hardness for $H = 5P_1$; this follows from the same reduction used by Papadopoulos and Tzimas [13] to prove Theorem 1. The three tractable cases, where $H \in \{P_4, P_1 + P_3, 3P_1 + P_2\}$, are all new. Out of these cases, $H = 3P_1 + P_2$ is the most involved. For this case we use a different technique to that used in [13]. Although we also reduce

	polynomial-time	unresolved	NP-complete
FVS	$H \subseteq_i P_5$ or sP_3 for $s \geq 1$	$H \supseteq_i P_1 + P_4$	none
OCT	$H = P_4$ or $H \subseteq_i sP_1 + P_3$ or sP_2 for $s \geq 1$	$H = sP_1 + P_5$ for $s \geq 0$ or $H = sP_1 + tP_2 + uP_3 + vP_4$ for $s, t, u \geq 0$, $v \geq 1$ with $\min\{s, t, u\} \geq 1$ if $v = 1$, or $H = sP_1 + tP_2 + uP_3$ for $s, t \geq 0$, $u \geq 1$ with $u \geq 2$ if $t = 0$	$H \supseteq_i P_6$ or $P_2 + P_5$
SFVS, SOCT	$H = P_4$ or $H \subseteq_i sP_1 + P_3$ for $s \geq 1$	$H = sP_1 + P_4$ for $s \geq 1$	$H \supseteq_i 2P_2$
WSFVS, WSOCT	$H \subseteq_i P_4, P_1 + P_3$, or $3P_1 + P_2$	$H \in \{2P_1 + P_3, P_1 + P_4, 2P_1 + P_4\}$	$H \supseteq_i 5P_1$ or $2P_2$

Table 1. The complexity of Feedback Vertex Set (FVS), Odd Cycle Transversal (OCT), and their subset (S) and weighted subset (WS) variants, when restricted to H-free graphs for linear forests H. All problems are NP-complete for H-free graphs when H is not a linear forest. The four blue cases (two for WSFVS, two for WSOCT) are the *algorithmic* contributions of this paper; see also Theorems 2 and 3.

to the problem of finding a minimum weight vertex cut that separates two given terminals, our technique relies less on explicit distance-based arguments, and we devise a method for distinguishing cycles according to parity. Our technique also enables us to extend the result of [13] on Weighted Subset Feedback Vertex Set from $4P_1$-free graphs to $(3P_1 + P_2)$-free graphs, leading to the same almost-complete dichotomy for Weighted Subset Feedback Vertex Set.

Theorem 3. *Let H be a graph with $H \notin \{2P_1 + P_3, P_1 + P_4, 2P_1 + P_4\}$. Then* Weighted Subset Feedback Vertex Set *on H-free graphs is polynomial-time solvable if $H \subseteq_i 3P_1 + P_2$, $P_1 + P_3$, or P_4, and is* NP*-complete otherwise.*

We refer to Table 1 for an overview of the current knowledge of the problems, including the results of this paper.

2 Preliminaries

Let $G = (V, E)$ be a graph. If $S \subseteq V$, then $G[S]$ denotes the subgraph of G induced by S, and $G - S$ is the graph $G[V \setminus S]$. The path on r vertices is denoted P_r. the *union* operation $+$ creates the disjoint union $G_1 + G_2$ having vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2)$. By sG, we denote the disjoint union of s copies of G. Thus sP_1 denotes the graph whose vertices form an independent set of size s. A *(connected) component* of G is a maximal connected subgraph 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 $N_G(U) = \bigcup_{u \in U} N(u) \setminus U$. Let S and T be two disjoint vertex sets of a graph G. 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.

3 General Framework of the Polynomial Algorithms

We first explain our general approach with respect to odd cycle transversals. Afterwards we modify our terminology for feedback vertex sets, but we note that our approach can be easily extended to other kinds of transversals as well. So, consider an instance (G, T, w) of Weighted Subset Odd Cycle Transversal. A subgraph of G with no odd T-cycles is *T-bipartite*. Note that a subset $S_T \subseteq V$ is an odd T-cycle transversal if and only if $G[V \setminus S_T]$ is T-bipartite. A *solution* for (G, T, w) is an odd T-cycle transversal S_T. From now on, whenever S_T is defined, we let $B_T = V(G) \setminus S_T$ denote the vertex set of the corresponding T-bipartite graph. If $u \in B_T$ belongs to at least one odd cycle of $G[B_T]$, then u is an *odd* vertex of B_T. Otherwise, when $u \in B_T$ is not in any odd cycle of $G[B_T]$, we say that u is an *even* vertex of B_T. Note that by definition every vertex in $T \cap B_T$ is even. We let $O(B_T)$ and $R(B_T)$ denote the sets of odd and even vertices of B_T (so $B_T = O(B_T) \cup R(B_T)$). A solution S_T is *neutral* if B_T consists of only even vertices; in this case S_T is an odd cycle transversal of G. We say that S_T is *T-full* if B_T contains no vertex of T. If S_T is neither neutral nor T-full, then S_T is a *mixed* solution. We can now outline our approach to finding minimum weight odd T-cycle transversals:

1. Compute a neutral solution of minimum weight.
2. Compute a T-full solution of minimum weight.
3. Compute a mixed solution of minimum weight.
4. From the three computed solutions, take one of overall minimum weight.

As mentioned, a neutral solution is a minimum-weight odd cycle transversal. Hence, in Step 1, we will use existing polynomial-time algorithms from the literature for computing such an odd cycle transversal (these algorithms must be for the weighted variant). Step 2 is trivial: we can just set $S_T := T$ (as w is non-negative). Hence, most of our attention will go to Step 3. For Step 3, we analyse the structure of the graphs $G[R(B_T)]$ and $G[O(B_T)]$ for a mixed solution S_T and how these graphs relate to each other.

For Weighted Subset Feedback Vertex Set we follow exactly the same approach, but we use slightly different terminology. A subgraph of a graph $G = (V, E)$ is a *T-forest* if it has no T-cycles. Note that a subset $S_T \subseteq V$ is a T-feedback vertex set if and only if $G[V \setminus S_T]$ is a T-forest. We write $F_T = V \setminus S_T$ in this case. If $u \in F_T$ belongs to at least one cycle of $G[F_T]$, then u is a *cycle vertex* of F_T. Otherwise, if $u \in F_T$ is not in any cycle of $G[F_T]$, we say that u is a *forest vertex* of F_T. By definition every vertex in $T \cap F_T$ is a forest vertex.

We obtain our results for Weighted Subset Feedback Vertex Set by a simplification of our algorithms for Weighted Odd Cycle Transversal. Hence, to explain our approach fully, we will now give a polynomial-time algorithm for Weighted Odd Cycle Transversal for $(3P_1 + P_2)$-free graphs.

4 Applying Our Framework on $(3\mathbf{P_1} + \mathbf{P_2})$-free Graphs

We let $G = (V, E)$ be a $(3P_1 + P_2)$-free graph with a vertex weighting w, and let $T \subseteq V$. For Step 1, we need the polynomial-time algorithm of [4] for Odd

Cycle Transversal on sP_2-free graphs ($s \geq 1$), and thus on $(3P_1 + P_2)$-free graphs (take $s = 4$). The algorithm in [4] was for the unweighted case, but it can be easily adapted for the weighted case.[3]

Lemma 1. *For every integer $s \geq 1$,* Weighted Odd Cycle Transversal *is polynomial-time solvable for sP_2-free graphs.*

As Step 2 is trivial, we focus on Step 3. We will reduce to a classical problem, well known to be polynomial-time solvable by standard network flow techniques.

Weighted Vertex Cut
Instance: a graph $G = (V, E)$, two distinct non-adjacent terminals t_1 and t_2, and a non-negative vertex weighting w.
Task: determine a set $S \subseteq V \setminus \{t_1, t_2\}$ of minimum weight such that t_1 and t_2 are in different connected components of $G - S$.

For a mixed solution S_T, we let $O = O(B_T)$ and $R = R(B_T)$; recall that $O \neq \emptyset$ and $R \cap T \neq \emptyset$. For our reduction to Weighted Vertex Cut, we need some **structural lemmas**. We first bound the number of components of $G[O]$.

Lemma 2. *Let $G = (V, E)$ be a $(3P_1 + P_2)$-free graph, and let $T \subseteq V$. For every mixed solution S_T, the graph $G[O]$ has at most two connected components.*

We now prove that $|R| \leq 8$. If $G[O]$ is disconnected, then even $|R| \leq 2$, as shown in Lemma 3. Otherwise we use Lemma 4 and the fact that $G[R]$ is bipartite.

Lemma 3. *Let $G = (V, E)$ be a $(3P_1 + P_2)$-free graph, and let $T \subseteq V$. For every mixed solution S_T, if $G[O]$ is disconnected, then R is a clique with $|R| \leq 2$.*

Lemma 4. *Let $G = (V, E)$ be a $(3P_1 + P_2)$-free graph and let $T \subseteq V$. For every mixed solution S_T, every independent set in $G[R]$ has size at most 4.*

We say that a vertex in O is a *connector* if it has a neighbour in R.

Lemma 5. *Let $G = (V, E)$ be a $(3P_1 + P_2)$-free graph, and let $T \subseteq V$. For every mixed solution S_T, if $G[O]$ has two connected components D_1 and D_2, then D_1 and D_2 each have at most one connector.*

Proof. By Lemma 3, R is a clique of size at most 2. For contradiction, suppose that, say, D_1 has two distinct connectors v_1 and v_2. Then v_1 and v_2 each have at most one neighbour in R, else the vertices of R would be in an odd cycle in $G[B_T]$, as R is a clique. Let u_1 be the neighbour of v_1 in R, and let u_2 be the neighbour of v_2 in R; note that $u_1 = u_2$ is possible.

An edge on a path P from v_1 to v_2 in D_1 does not belong to an odd cycle in $G[D_1]$; else there would be a path P' from v_1 to v_2 in $G[O]$ with a different parity than P and one of the cycles $u_1v_1Pv_2u_2u_1$ or $u_1v_1P'v_2u_2u_1$ is odd, implying that u_1 and u_2 would not be even.

[3] Proofs of Lemmas 1–4 are omitted for space reasons. A full version of this paper can be found at https://arxiv.org/abs/2007.14514

By definition, v_1 and v_2 belong to at least one odd cycle, which we denote by C_1 and C_2, respectively. Then $V(C_1) \cap V(C_2) = \emptyset$ and there is no edge between a vertex of C_1 and a vertex of C_2 except from possibly the edge v_1v_2; else there would be a path from v_1 to v_2 in $G[O]$ with an edge that belongs to an odd cycle (C_1 or C_2), a contradiction with what we found above. Note also that u_1 has no neighbours in $V(C_1)$ other than v_1; otherwise $G[B_T]$ would have an odd cycle containing u_1. Moreover, u_1 has no neighbours in $V(C_2)$ either, except v_2 if $u_1 = u_2$; otherwise $G[B_T]$ would contain an odd cycle containing u_1 and u_2.

We now let w_1 and x_1 be two adjacent vertices of C_1 that are not adjacent to u_1. Let w_2 be a vertex of C_2 not adjacent to u_1. Then, we found that $\{u_1, w_2, w_1, x_1\}$ induces a $2P_1 + P_2$ (see Figure 2).

We continue by considering D_2, the other connected component of $G[O]$. By definition, D_2 has an odd cycle C'. As $|R| \leq 2$ and each vertex of R can have at most one neighbour on an odd cycle in $G[B_T]$, we find that C' contains a vertex v' not adjacent to any vertex of R, so v' is not adjacent to u_1. As v' and the vertices of $\{w_2, w_1, x_1\}$ belong to different connected components of $G[O]$, we find that v' is not adjacent to any vertex of $\{w_2, w_1, x_1\}$ either. However, now $\{u_1, v', w_2, w_1, x_1\}$ induces a $3P_1 + P_2$ (see also Figure 2), a contradiction. □

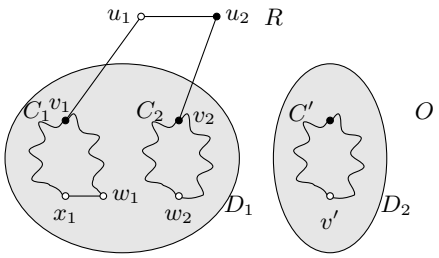

Fig. 2. An illustration for the proof of Lemma 5: the white vertices induce a $3P_1 + P_2$.

We need one more structural lemma about connectors, in the case where $G[O]$ is connected. Let R consist of two adjacent vertices u_1 and u_2. Let O (with $O \cap T = \emptyset$) be the disjoint union of two complete graphs K and L, each on an odd number of vertices that is at least 3, plus a single additional edge, such that:

1. u_1 is adjacent to exactly one vertex v_1 in K and to no vertex of L;
2. u_2 is adjacent to exactly one vertex v_2 in L and to no vertex of K; and
3. v_1 and v_2 are adjacent.

Note that $G[B_T]$ is indeed T-bipartite. We call the corresponding mixed solution S_T a *2-clique solution* (see Figure 3).

Lemma 6. *Let $G = (V, E)$ be a $(3P_1 + P_2)$-free graph and let $T \subseteq V$. For every mixed solution S_T that is not a 2-clique solution, if $G[O]$ is connected, then O has no two connectors with a neighbour in the same connected component of $G[R]$.*

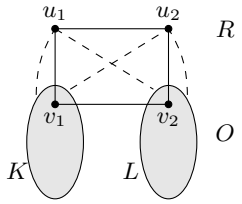

Fig. 3. The structure of B_T corresponding to a 2-clique solution S_T.

Proof. For some $p \geq 1$, let $F_1, \ldots, F_p$ be the set of components of $G[R]$. For contradiction, assume O has two distinct connectors v_1 and v_2, each with a neighbour in the same F_i, say, F_1. Let $u_1, u_2 \in V(F_1)$ be these two neighbours, where $u_1 = u_2$ is possible. Let Q be a path from u_1 to u_2 in F_1 (see Figure 4). We make an important claim: *All paths from v_1 to v_2 in $G[O]$ have the same parity.* The reason is that if there exist paths P and P' from v_1 to v_2 in $G[O]$ that have different parity, then either the cycle $u_1v_1Pv_2u_2Qu_1$ or the cycle $u_1v_1P'v_2u_2Qu_1$ is odd. This would mean that u_1 and u_2 are not even.

By definition, v_1 and v_2 each belong to at least one odd cycle, which we denote by C_1 and C_2, respectively. We choose C_1 and C_2 such that they have minimum length. We note that $V(C_1) \cap V(C_2) = \emptyset$ and that there is no edge between a vertex of C_1 and a vertex of C_2 except possibly the edge v_1v_2; otherwise there would be paths from v_1 to v_2 in $G[O]$ that have different parity, a contradiction with the claim above.

We also note that v_1 is the only neighbour of u_1 on C_1; otherwise u_1 would belong to an odd cycle of $G[B_T]$. Similarly, v_2 is the only neighbour of u_2 on C_2. Moreover, u_1 has no neighbour on C_2 except v_2 if $u_1 = u_2$, and u_2 has no neighbour on C_1 except v_1 if $u_1 = u_2$. This can be seen as follows. For a contradiction, first suppose that, say, u_1 has a neighbour w on C_2 and $w \neq v_2$. As C_2 is an odd cycle, there exist two vertex-disjoint paths P and P' on C_2 from w to v_2 of different parity. Using the edges u_1w and u_2v_2 and the path Q from u_1 to u_2, this means that u_1 and u_2 are on odd cycle of $G[B_T]$. However, this is not possible as u_1 and u_2 are even. Hence, u_1 has no neighbour on $V(C_2) \setminus \{v_2\}$. By the same reasoning, u_2 has no neighbour on $V(C_1) \setminus \{v_1\}$. Now suppose that u_1 is adjacent to v_2 and that $u_1 \neq u_2$. Then u_1 is not adjacent to u_2, otherwise the vertices u_1, u_2 and v_2 would form a triangle, and consequently, u_1 and u_2 would not be even. Recall that $V(C_1) \cap V(C_2) = \emptyset$ and that there is no edge between a vertex of C_1 and a vertex of C_2. Hence, we can now take u_1, u_2, a vertex of $V(C_1) \setminus \{v_1\}$, and two adjacent vertices of $V(C_2) \setminus \{v_2\}$ (which exist as C_2 is a cycle) to find an induced $3P_1 + P_2$, a contradiction.

We now claim that C_1 and C_2 each have exactly three vertices. For contraction, assume that at least one of them, C_1 has length at least 5 and that in C_1, we have that x and y are the two neighbours of v_1. As C_1 has minimum length, x and y are not adjacent. Let t_1 and t_2 be adjacent vertices of C_2 distinct from

v_2. Then $\{u_1, x, y, t_1, t_2\}$ induces a $3P_1 + P_2$ in G, a contradiction. Hence, C_1 and C_2 are triangles, say with vertices v_1, w_1, x_1 and v_2, w_2, x_2, respectively.

Now suppose $G[O]$ has a path from v_1 to v_2 on at least three vertices. Let s be the vertex adjacent to v_1 on this path. Then $s \notin \{w_1, x_1, w_2, x_2\}$ and s is not adjacent to any vertex of $\{w_1, x_1, w_2, x_2\}$ either; otherwise $G[O]$ contains two paths from v_1 to v_2 that are of different parity. As u_1 and s are not adjacent (else u_1 belongs to a triangle), we find that $\{s, u_1, w_2, w_1, x_1\}$ induces a $3P_1+P_2$, a contradiction (see also Figure 4). We conclude that as $G[O]$ is connected, v_1 and v_2 must be adjacent.

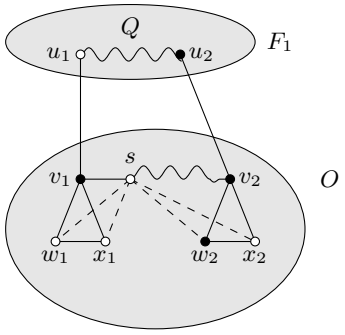

Fig. 4. The white vertices induce a $3P_1 + P_2$.

So far, we found that O contains two vertex-disjoint triangles on vertex sets $\{v_1, w_1, x_1\}$ and $\{v_2, w_2, x_2\}$, respectively, with v_1v_2 as the only edge between them. As v_1 is adjacent to v_2, we find that $u_1 \neq u_2$; otherwise $\{u_1, v_1, v_2\}$ would induce a triangle, which is not possible as $u_1 \in R$. Recall that u_1 is not adjacent to any vertex of $V(C_1) \cup V(C_2)$ except v_1, and similarly, u_2 is not adjacent to any vertex of $V(C_1) \cup V(C_2)$ except v_2. Then u_1 must be adjacent to u_2, as otherwise $\{u_1, u_2, w_1, w_2, x_2\}$ would induce a $3P_1 + P_2$.

Let $z \in O \setminus (V(C_1) \cup V(C_2))$. Suppose u_1 is adjacent to z. First assume z is adjacent to w_1 or x_1, say w_1. Then $u_1zw_1x_1v_1u_1$ is an odd cycle. Hence, this is not possible. Now assume z is adjacent to w_2 or x_2, say w_2. Then $u_1zw_2v_2u_2u_1$ is an odd cycle. This is not possible either. Hence, z is not adjacent to any vertex of $\{w_1, x_1, w_2, x_2\}$. Moreover, z is not adjacent to u_2, as otherwise $\{u_1, u_2, z\}$ induces a triangle in $G[B_T]$. However, $\{u_2, w_2, z, w_1, x_1\}$ now induces a $3P_1+P_2$. Hence, u_1 is not adjacent to z. In other words, v_1 is the only neighbour of u_1 on O. By the same arguments, v_2 is the only neighbour of u_2 on O.

Let K be a maximal clique of O that contains C_1 and let L be a maximal clique of O that contains C_2. Note that K and L are vertex-disjoint, as for example, $w_1 \in K$ and $w_2 \in L$ are not adjacent. We claim that $O = K \cup L$. For contradiction, assume that r is a vertex of O that does not belong to K or L. As u_1 and u_2 are adjacent vertices that have no neighbours in $O \setminus \{v_1, v_2\}$, the

$(3P_1 + P_2)$-freeness of G implies that $G[O \setminus \{v_1, v_2\}]$ is $3P_1$-free. As $K \setminus \{v_1\}$ and $L \setminus \{v_2\}$ induce the disjoint union of two complete graphs on at least two vertices, this means that r is adjacent to every vertex of $K \setminus \{v_1\}$ or to every vertex of $L \setminus \{v_2\}$, say r is adjacent to every vertex of $K \setminus \{v_1\}$. Then r has no neighbour r' in $L \setminus \{v_2\}$, as otherwise the cycle $v_1u_1u_2v_2r'rw_1v_1$ is an odd cycle in $G[B_T]$ that contains u_1 (and u_2). Moreover, as K is maximal and r is adjacent to every vertex of $K \setminus \{v_1\}$, we find that r and v_1 are not adjacent. Recall also that u_2 has v_2 as its only neighbour in O, hence u_2 is not adjacent to r. This means that $\{r, v_1, u_2, w_2, x_2\}$ induces a $3P_1 + P_2$, which is not possible. We conclude that $O = K \cup L$; consequently, both K and L have odd size.

We now consider the graph F_1 in more detail. Suppose F_1 contains another vertex $u_3 \notin \{u_1, u_2\}$. As F_1 is connected and bipartite (as $V(F_1) \subseteq R$), we may assume without loss of generality that u_3 is adjacent to u_1 but not to u_2. If u_3 has a neighbour K, then $G[B_T]$ contains an odd cycle that uses u_1, u_3 and one vertex of K (if the neighbour of u_3 in K is v_1) or three vertices of K (if the neighbour of u_3 in K is not v_1). Hence, u_3 has no neighbour in K. This means that $\{u_2, u_3, w_2, w_1, x_1\}$ induces a $3P_1 + P_2$, so u_3 cannot exist. Hence, F_1 consists only of the two adjacent vertices u_1 and u_2.

Now suppose that $p \geq 2$, that is, F_2 is nonempty. Let $u' \in V(F_2)$. As $u' \in R$, we find that u' is adjacent to at most one vertex of C_1 and to at most one vertex of C_2. Hence, we may without loss of generality assume that u' is not adjacent to w_1 and w_2. Then $\{u', w_1, w_2, u_1, u_2\}$ induce a $3P_1 + P_2$. We conclude that $R = \{u_1, u_2\}$. However, now S_T is a 2-clique solution of G, a contradiction. □

An algorithmic lemma, for finding a 2-clique solution of minimum weight:

Lemma 7. *Let $G = (V, E)$ be a $(3P_1 + P_2)$-free graph with a vertex weighting w, and let $T \subseteq V$. It is possible to find in polynomial time a 2-clique solution for (G, w, T) that has minimum weight.*

Proof. As the cliques K and L in B_T have size at least 3 for a 2-clique solution S_T, there are distinct vertices x_1, y_1 in $K \setminus \{v_1\}$ and distinct vertices x_2, y_2 in $L \setminus \{v_2\}$. The ordered 8-tuple $(u_1, u_2, v_1, v_2, x_1, y_1, x_2, y_2)$ is a *skeleton* of the 2-clique solution. We call the labelled subgraph of B_T that these vertices induce a *skeleton graph*.

In order to find a 2-clique solution of minimum weight in polynomial time, we consider all $\mathcal{O}(n^8)$ possible ordered 8-tuples $(u_1, u_2, v_1, v_2, x_1, y_1, x_2, y_2)$ of vertices of G and further investigate those that induce a skeleton graph. In this case, we note that if these vertices form the skeleton of a 2-clique solution S_T, then $R(B_T) = \{u_1, u_2\}$ and $O(B_T)$ is a subset of $V' = \{v_1, x_1, y_1\} \cup \{v_2, x_2, y_2\} \cup (N(v_1) \cap N(x_1) \cap N(y_1)) \cup (N(v_2) \cap N(x_2) \cap N(y_2))$. We further refine the definition of V' by deleting any vertex that cannot, by definition, belong to $O(B_T)$; that is, we remove every vertex that belongs to $T \cup (N(\{u_1, u_2\}) \setminus \{v_1, v_2\})$ or is a neighbour of both a vertex in $\{v_1, x_1, y_1\}$ and a vertex in $\{v_2, x_2, y_2\}$. We write $G' = G[V']$. Note that u_1 and u_2 are not in G' (as they are not adjacent to any vertex in $\{x_1, x_2, y_1, y_2\}$), whereas $v_1, v_2, x_1, x_2, y_1, y_2$ all are in G'.

We now show that the sets $K' = \{v_1, x_1, y_1\} \cup (N(\{v_1, x_1, y_1\}) \cap V')$ and $L' = \{v_2, x_2, y_2\} \cup (N(\{v_2, x_2, y_2\}) \cap V')$ partition V', and moreover, that K' and L' are cliques. By definition, every vertex of V' either belongs to K' or to L'. By construction, $K' \cap L' = \emptyset$ since every vertex in $K' \setminus \{v_1\}$ is a neighbour of v_1 and every vertex in $L' \setminus \{v_2\}$ is a neighbour of v_2 and no vertex in V' is adjacent to both v_1 and v_2 which are themselves distinct. For a contradiction, suppose K' is not a clique. Then K' contains two non-adjacent vertices t and t'. As $K' \setminus \{v_1, x_1, y_1\}$ is complete to the clique $\{v_1, x_1, y_1\}$, we find that t and t' both belong to $K' \setminus \{v_1, x_1, y_1\}$. By construction of G', we find that $\{t, t'\}$ is anti-complete to $\{u_1, u_2, x_2\}$. By the definition of a skeleton, $\{u_1, u_2\}$ is anti-complete to $\{x_2\}$. Then $\{u_1, u_2, t, t', x_2\}$ induces a $3P_1 + P_2$ in G, a contradiction. By the same arguments, L' is a clique.

In G' we first delete the edge v_1v_2. Second, for $i \in \{1, 2\}$ we replace the vertices v_i, x_i, y_i by a new vertex v_i^* that is adjacent precisely to every vertex that is a neighbour of at least one vertex of $\{v_i, x_i, y_i\}$ in G'. This transforms the graph G' into the graph $G^* = (V^*, E^*)$. Note that in G^* there is no edge between v_1^* and v_2^*. We give each vertex $z \in V^* \setminus \{v_1^*, v_2^*\}$ weight $w^*(z) = w(z)$, and for $i \in \{1, 2\}$, we set $w^*(v_i^*) = w(v_i) + w(x_i) + w(y_i)$. See Figure 5.

The algorithm will now solve Weighted Vertex Cut on (G^*, w^*) with terminals v_1^* and v_2^*; recall that this can be done in polynomial time by standard network flow techniques. Let S^* be the output. Then $G^* - S^*$ has two distinct connected components on vertex sets K^* and L^*, respectively, with $v_1^* \in K^*$ and $v_2^* \in L^*$. We set $K = (K^* \setminus \{v_1^*\}) \cup \{v_1, x_1, y_1\}$ and $L = (L^* \setminus \{v_2^*\}) \cup \{v_2, x_2, y_2\}$ and note that $G' - S^*$ contains $G[K]$ and $G[L]$ as distinct connected components.

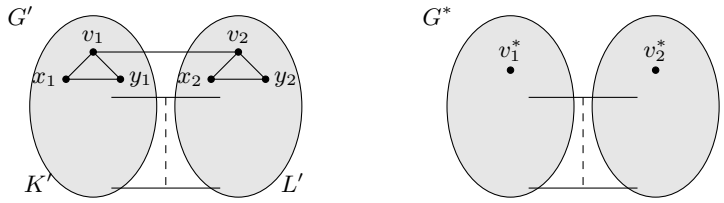

Fig. 5. The graph G' and G^* in the proof of Lemma 7.

As K is a subset of the clique K' and L is a subset of the clique L' and $V' = K' \cup L'$, we find that $G[K]$ and $G[L]$ are the only two connected components of $G' - S'$, and moreover that K and L are cliques. As no vertex of $(K \cup L) \setminus \{v_1, v_2\}$ is adjacent to u_1 or u_2, this means that $S = V \setminus (\{u_1, u_2\} \cup K \cup L)$ is a 2-clique solution for G. Moreover, as S^* is an optimal solution of Weighted Vertex Cut on instance (G^*, w^*) with terminals v_1^* and v_2^*, we find that S has minimum weight over all 2-clique solutions with skeleton $(u_1, u_2, v_1, v_2, x_1, y_1, x_2, y_2)$.

From all the $\mathcal{O}(n^8)$ 2-clique solutions computed in this way, we pick one with minimum weight; we found this 2-clique solution in polynomial time. □

The Algorithm. We are now ready to prove the main result of the section.

Theorem 4. Weighted Subset Odd Cycle Transversal *is polynomial-time solvable for* $(3P_1 + P_2)$*-free graphs.*

Proof. Let G be a $(3P_1 + P_2)$-free graph with a vertex weighting w, and let $T \subseteq V(G)$. We describe a polynomial-time algorithm for the optimization version of the problem on input (G, T, w) using the approach of Section 3. So, in Step 1, we compute a neutral solution of minimum weight, i.e., a minimum weight odd cycle transversal, using polynomial time due to Lemma 1 (take $s = 4$). We then compute, in Step 2, a T-full solution by setting $S_T = T$. It remains to compute a mixed solution S_T of minimum weight (Step 3) and compare its weight with the two solutions found above (Step 4). By Lemma 2 we can distinguish between two cases: $G[O]$ is connected or $G[O]$ consists of two connected components.

Case 1. $G[O]$ is connected.
We first compute in polynomial time a 2-clique solution of minimum weight by using Lemma 7. In the remainder of Case 1, we will compute a mixed solution S_T of minimum weight with connected $G[O]$ that is not a 2-clique solution. By Lemma 4 and the fact that $G[R]$ is bipartite by definition, we find that $|R| \leq 8$. We consider all $\mathcal{O}(n^8)$ possibilities for R. We discard a choice for R if $G[R]$ is not bipartite. If $G[R]$ is bipartite, we compute a solution S_T of minimum weight such that B_T contains R. Let $F_1, \ldots, F_p$ be the components of $G[R]$. By definition, $p \geq 1$. By Lemma 4 $p \leq 4$. By Lemma 6, O has at most $p \leq 4$ connectors.

We now consider all $\mathcal{O}(n^4)$ possible choices for a set D of at most four connectors. For each set D, we first check that $G[D \cup R]$ is T-bipartite and that there are no two vertices in D with a neighbour in the same F_i; if one of these conditions is not satisfied, we discard our choice of D. If both conditions are satisfied we put the vertices of D in O, together with any vertex that is not in T and that is not adjacent to any vertex of R. Then, as $G[D \cup R]$ is T-bipartite and no two vertices in D are adjacent to the same component F_i, the graph $G[R \cup O]$ is T-bipartite. We remember the weight of $S_T = V \setminus (R \cup O)$. In doing the above, we may have computed a set O that is disconnected or that contains even vertices. So we might compute some solutions more than once. However, we can compute each solution in polynomial time, and the total number of solutions we compute in Case 1 is $\mathcal{O}(n^8) \cdot \mathcal{O}(n^4) = \mathcal{O}(n^{12})$, which is polynomial as well. Out of all the 2-clique solutions and other mixed solutions we found, we pick a solution $S_T = V_T \setminus (R \cup O)$ with minimum weight as the output for Case 1.

Case 2. $G[O]$ consists of two connected components D_1 and D_2.
By Lemma 3, R is a clique of size at most 2. We consider all possible $\mathcal{O}(n^2)$ options for R. Each time R is a clique, we proceed as follows. By Lemma 5, both D_1 and D_2 have at most one connector. We consider all $\mathcal{O}(n^2)$ ways of choosing at most one connector from each of them. If we choose two, they must be non-adjacent. We discard the choice if the subgraph of G induced by R and the chosen connector(s) is not T-bipartite. Otherwise we continue. If we chose at most one connector v, we let O consist of v and all vertices that do not belong to T and that do not have a neighbour in R. Then $G[R \cup O]$ is T-bipartite

and we store $S_T = V \setminus (R \cup O)$. Note that O might not induce two connected components consisting of odd vertices, so we may duplicate some work. However, $R \cup O$ induces a T-bipartite graph and we found O in polynomial time, and this is what is relevant (together with the fact that we only use polynomial time).

When the algorithm chooses two (non-adjacent) connectors v and v' we do as follows. We remove any vertex from T and any neighbour of R other than v and v'. Let (G', w') be the resulting weighted graph (w' is the restriction of w to $V(G')$). We solve Weighted Vertex Cut in polynomial time on G', w' with v and v' as terminals. Let S be the output. We let $O = V(G') - S$. Then $G[O]$ has two connected components (as $G[R \cup \{v, v'\}]$ is T-bipartite, this implies that $G[R \cup O]$ is T-bipartite) but $G[O]$ might contain even vertices. However, what is relevant is that $G[R \cup O]$ is T-bipartite, and that we found O in polynomial time. We remember the solution $S_T = V \setminus (R \cup O)$. In the end we remember from all the solutions we computed one with minimum weight as the output for Case 2. The number of solutions is $\mathcal{O}(n^2) \cdot \mathcal{O}(n^2) = \mathcal{O}(n^4)$ and we found each solution in polynomial time so processing Case 2 takes polynomial time.

Correctness of our algorithm follows from the correctness of Cases 1 and 2, which describe all possible mixed solutions due to Lemma 2. As processing Cases 1 and 2 takes polynomial time, we compute a mixed solution of minimum weight in polynomial time. Computing a non-mixed solution of minimum weight takes polynomial time as deduced already. Hence, the running time is polynomial. □

The Proof of Theorems 2 and Theorem 3

We omit the proofs that Weighted Subset Odd Cycle Transversal is polynomial-time solvable for P_4-free graphs and (P_1+P_3)-free graphs. The reduction in [13] for Weighted Subset Feedback Vertex Set for $5P_1$-free graphs yields NP-completeness for $5P_1$-free graphs. Theorem 2 follows from Theorem 4, the above results and the result of [4] that even Odd Cycle Transversal is NP-complete on H-free graphs if H has a cycle or a claw.

We omit the proofs that Weighted Feedback Vertex Set is polynomial-time solvable for sP_2-free graphs for every $s \geq 1$, $(3P_1+P_2)$-free graphs and (P_1+P_3)-free graphs. The problem is polynomial-time solvable for P_4-free graphs [2]. Theorem 3 now follows from the above results and the results that Feedback Vertex Set is NP-complete on H-free graphs if H has a cycle [14] or a claw [10].

5 Conclusions

We determined the complexity of Weighted Subset Odd Cycle Transversal and Weighted Subset Feedback Vertex Set on H-free graphs except when $H \in \{2P_1+P_3, P_1+P_4, 2P_1+P_4\}$. We believe that the case $H = 2P_1+P_3$ is polynomial-time solvable for both problems using our methodology and our algorithms for $H = P_1+P_3$ as a subroutine. The other two cases are open even for Odd Cycle Transversal and Feedback Vertex Set. For these cases we first need to be able to determine the complexity of finding a maximum induced disjoint union of stars in a $(P_1 + P_4)$-free graph. We refer to Table 1 for other

unresolved cases in our framework and note again that our results demonstrate that the classifications of Weighed Subset Odd Cycle Transversal and Subset Odd Cycle Transversal do not coincide for H-free graphs.

References

1. Abrishami, T., Chudnovsky, M., Pilipczuk, M., Rzążewski, P., Seymour, P.: Induced subgraphs of bounded treewidth and the container method. Proc. SODA 2021 pp. 1948–1964 (2021)
2. Bergougnoux, B., Papadopoulos, C., Telle, J.A.: Node Multiway Cut and Subset Feedback Vertex Set on graphs of bounded mim-width. Proc. WG 2020, LNCS **12301**, 388–400 (2020)
3. Brettell, N., Johnson, M., Paesani, G., Paulusma, D.: Computing subset transversals in H-free graphs. Proc. WG 2020, LNCS **12301**, 187–199 (2020)
4. Chiarelli, N., Hartinger, T.R., Johnson, M., Milanič, M., Paulusma, D.: Minimum connected transversals in graphs: New hardness results and tractable cases using the price of connectivity. Theoretical Computer Science **705**, 75–83 (2018)
5. Dabrowski, K.K., Feghali, C., Johnson, M., Paesani, G., Paulusma, D., Rzążewski, P.: On cycle transversals and their connected variants in the absence of a small linear forest. Algorithmica **82**, 2841–2866 (2020)
6. Fomin, F.V., Heggernes, P., Kratsch, D., Papadopoulos, C., Villanger, Y.: Enumerating minimal subset feedback vertex sets. Algorithmica **69**, 216–231 (2014)
7. Grzesik, A., Klimošová, T., Pilipczuk, M., Pilipczuk, M.: Polynomial-time algorithm for maximum weight independent set on P_6-free graphs. Proc. SODA 2019 pp. 1257–1271 (2019)
8. Johnson, M., Paesani, G., Paulusma, D.: Connected Vertex Cover for $(sP_1 + P_5)$-free graphs. Algorithmica **82**, 20–40 (2020)
9. Lozin, V., Malyshev, D., Mosca, R., Zamaraev, V.: Independent domination versus weighted independent domination. Information Processing Letters **156**, 105914 (2020)
10. Munaro, A.: On line graphs of subcubic triangle-free graphs. Discrete Mathematics **340**, 1210–1226 (2017)
11. Paesani, G., Paulusma, D., Rzążewski, P.: Feedback Vertex Set and Even Cycle Transversal for H-free graphs: finding large block graphs. CoRR **abs/2105.02736** (2021)
12. Papadopoulos, C., Tzimas, S.: Polynomial-time algorithms for the subset feedback vertex set problem on interval graphs and permutation graphs. Discrete Applied Mathematics **258**, 204–221 (2019)
13. Papadopoulos, C., Tzimas, S.: Subset feedback vertex set on graphs of bounded independent set size. Theoretical Computer Science **814**, 177–188 (2020)
14. Poljak, S.: A note on stable sets and colorings of graphs. Commentationes Mathematicae Universitatis Carolinae **15**, 307–309 (1974)