Computing Weighted Subset Odd Cycle Transversals in H-Free Graphs*

Nick Brettell[1], Matthew Johnson[2], and Daniël Paulusma[2]

[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 problem 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. In particular, our result shows that the complexities of the weighted and unweighted variant do not align on H-free graphs, just as Papadopoulos and Tzimas showed for Subset Feedback Vertex Set.

Keywords. odd cycle transversal, H-free graph, complexity dichotomy

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 can now define the following problems:

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 $w(S_T) \leq k$?

* The paper received support from the Leverhulme Trust (RPG-2016-258). The first author was also supported by a Rutherford Foundation Postdoctoral Fellowship, administered by the Royal Society Te Apārangi. An extended abstract appeared in the proceedings of WADS 2021 [7].

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; a graph in this class is said to be *H-free*.

1.1 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 [29], 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). Exactly the same result holds for Odd Cycle Transversal [8]. It has also been shown that Feedback Vertex Set [22] and Odd Cycle Transversal [8] 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 to 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 [12]. It is known that Subset Feedback Vertex Set [14] and Subset Odd Cycle Transversal [6], 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. Papadopoulos and Tzimas [28] proved the following interesting dichotomy, which motivated our research.

Theorem 1 ([28]). 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 $s \geq 1$ [28]. In contrast, for many transversal problems, the complexities on the weighted and unweighted versions for H-free graphs align; see, for example Vertex Cover [16], Connected Vertex Cover [17] and (Independent) Dominating Set [20]. Thus Subset Feedback Vertex Set is one of the few known problems

Fig. 1. Two examples of a 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, $S_T \setminus T \neq \emptyset$. On the right, $S_T \subseteq T$.

for which, on certain hereditary graph classes, the (unweighted) problem is polynomial-time solvable, but the weighted variant is NP-complete.

Very recently, Paesani et al. [26] completed the complexity classification of Weighted Subset Feedback Vertex Set for H-free graphs by giving a polynomial-time algorithm for $2P_1+P_4$-free graphs. Prior to this, the other known polynomial-time algorithm for Weighted Subset Feedback Vertex Set on H-free graphs was for the case where $H = P_4$. This latter result can be shown in several ways. First, Weighted Subset Feedback Vertex Set is polynomial-time solvable for permutation graphs [27] and also for graphs for which we can find a decomposition of constant mim-width in polynomial time [2]; both classes contain the class of P_4-free graphs. A linear-time algorithm can also be obtained by making the following two observations. First, P_4-free graphs have clique-width 1 and we can construct a 1-expression in linear time by the definition of a cograph (it is well-known that the class of P_4-free graphs coincides with the class of cographs). Second, the property that a subset of vertices S is a T-feedback vertex subset in a graph $G = (V, E)$ for some given set $T \subseteq V$ can be expressed in MSO_1 monadic second-order logic (with S as the only free monadic variable). Hence, we can apply a meta-theorem of Courcelle et al. [11]. 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). It is known that Subset Odd Cycle Transversal is polynomial-time solvable on $(sP_1 + P_3)$-free graphs for every integer $s \geq 0$ [6] and on P_4-free graphs [6]. In Section 6 we show that the latter result can be generalized to the weighted variant in a straightforward way. Besides completing the complexity classification for the weighted version, Paesani et al. [26] also completed the complexity classification of Subset Feedback Vertex Set for H-free graphs by giving a polynomial-time algorithm for the latter problem on (sP_1+P_4)-free graphs for every $s \geq 0$.

We now mention the polynomial-time results on H-free graphs for the weighted variants of the two original problems (which do not imply anything for the weighted subset versions). There are no additional polynomial-time results for the unweighted versions. Abrishami et al. [1] proved that Weighted Feedback Vertex Set is polynomial-time solvable on P_5-free graphs. Paesani et al. [25] recently extended this result to $(sP_1 + P_5)$-free graphs for every $s \geq 1$ [24] and also proved that Weighted Feedback Vertex Set is polynomial-time solvable on sP_3-free graphs for every $s \geq 1$ [24]. In another recent paper [15], Gartland et al. proved that Weighted Feedback Vertex Set is quasipolynomial-time solvable for P_t-free graphs for every $t \geq 1$. Such a result is unlikely for (Weighted) Odd Cycle Transversal due to the aforementioned NP-hardness of this problem for $(P_2 + P_5, P_6)$-free graphs [12]. However, Weighted Odd Cycle Transversal can be solved in polynomial time on sP_2-free graphs for every $s \geq 1$; this follows from a straightforward generalization of the corresponding result for the unweighted variant given in [8]. Finally, Weighted Odd Cycle Transversal is also polynomial-time solvable for (sP_1+P_3)-free graphs; this follows from a straightforward adaptation of the proof for the unweighted variant given in [12].[3]

1.2 Our Results

We enhance the current understanding of Weighted Subset Odd Cycle Transversal on H-free graphs. We highlight that Subset Odd Cycle Transversal is a problem whose weighted variant turns out to be harder than its unweighted variant when restricting the input to H-free graphs. In particular, we show to what extent this holds by giving the following

[3] The result for Odd Cycle Transversal on $(sP_1 + P_3)$-free graphs ($s \geq 1$) from [12] was shown before the corresponding result for Subset Odd Cycle Transversal was proven in [6].

	polynomial-time	unresolved	NP-complete
(W)FVS	$H \subseteq_i sP_1 + P_5$ or sP_3 for $s \geq 1$	$H \supseteq_i P_2 + P_4$ or P_6	none
(W)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	$H = sP_1 + P_4$ for $s \geq 0$	none	$H \supseteq_i 2P_2$
SOCT	$H = P_4$ or $H \subseteq_i sP_1 + P_3$ for $s \geq 0$	$H = sP_1 + P_4$ for $s \geq 1$	$H \supseteq_i 2P_2$
WSFVS	$H \subseteq_i 2P_1 + P_4$	none	$H \supseteq_i 5P_1$ or $2P_2$
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 (Weighted) Feedback Vertex Set ((W)FVS), (Weighted) Odd Cycle Transversal ((W)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 two blue cases for WSOCT are the main *algorithmic* contributions of this paper; see also Theorem 2.

almost-complete dichotomy. We write $H \subseteq_i G$, or $G \supseteq_i H$ to say that H is an *induced* subgraph of G (that is, H can be obtained from G by a sequence of vertex deletions).

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 [28] 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, the case $H = P_4$ easily follows from existing results (as we will explain) and the case $H = 3P_1 + P_2$ is the most involved. For the latter case we use a different technique to that used in [28]. Just as in [28], we do reduce to the problem of finding a minimum weight vertex cut that separates two given terminals. However, our technique relies less on explicit distance-based arguments, and we devise a method for distinguishing cycles according to parity. Our technique can also be used to prove that Weighted Subset Feedback Vertex Set is polynomial-time solvable for $(3P_1 + P_2)$-free graphs. However, we omit a proof of this due to the aforementioned result of Paesani et al. [26] for the superclass of $(2P_1 + P_4)$-free graphs. Their proof for $H = 2P_1 + P_4$ is based on the fact that acyclic graphs (forests) have vertices of degree 1. This is not necessarily true for odd-cycle-free (bipartite) graphs. As such, it appears their approach cannot be adapted for the Weighted Odd Cycle Transversal problem.

We refer to Table 1 for an overview of the current knowledge of the problems on H-free graphs, 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. We say that S is *independent* if $G[S]$ has no edges, and that S is a *clique* and $G[S]$ is *complete* if every pair of vertices in S is joined by an edge.

If G_1 and G_2 are vertex-disjoint graphs, then 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 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 $N_G(U) = \bigcup_{u \in U} N(u) \setminus U$.

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. 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 $G[T]$.

A graph is *bipartite* if its vertex set 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. If X or Y has size 1, the complete bipartite graph is a *star*; recall that $K_{1,3}$ is also called a claw.

3 General Framework of the Algorithms

We first explain our general approach with respect to odd cycle transversals. We note that our approach can be easily extended to other kinds of transversals as well (see also [7]).

So, consider an instance (G, T, w) of Weighted Subset Odd Cycle Transversal. Recall that a cycle is a T*-cycle* if it contains a vertex of T. 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). So, 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.

4 Weighted Subset Odd Cycle Transversal on $(3P_1 + P_2)$-free Graphs

We will prove that Weighted Subset Odd Cycle Transversal is polynomial-time solvable for $(3P_1 + P_2)$-free graphs using the framework from Section 3. 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 [8] 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 [8] was for the unweighted case, but it can be trivially adapted for the weighted case of Lemma 1 which we state without proof.

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 need to 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, by the definition, $O \neq \emptyset$ and $R \cap T \neq \emptyset$ (see also Figure 2). For our reduction to Weighted Vertex Cut, we need some structural results first.

4.1 Structural Lemmas

As O is nonempty, $G[O]$ has at least one connected component. We first bound the number of components of $G[O]$.

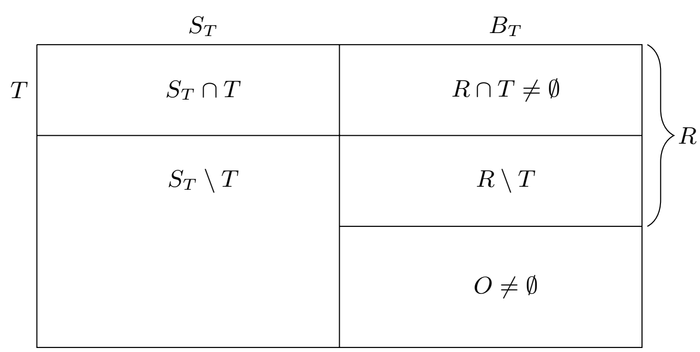

Fig. 2. The decomposition of V when S_T is a mixed solution. The sets $O = O(B_T)$ and $R = R(B_T)$ are the odd and even vertices of B_T, respectively.

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.*

Proof. For contradiction, assume that $G[O]$ has at least three connected components D_1, D_2, D_3. As each D_i contains an odd cycle, each D_i has an edge. Hence, each D_i must be a complete graph, otherwise one D_i, say D_1 has two non-adjacent vertices, which would induce together with a vertex of D_2 and an edge of D_3, a $3P_1 + P_2$.

Recall that, as S_T is mixed, R is nonempty. Let $u \in R$. Then u does not belong to any D_i. Moreover, u can be adjacent to at most one vertex of each D_i, otherwise u and two of its neighbours in D_i would form a triangle (as D_i is complete) and u would not be even. As each D_i is a complete graph on at least three vertices, we can pick two non-neighbours of u in D_1, which form an edge, a non-neighbour of u in D_2 and a non-neighbour of u in D_3. These four vertices, together with u, induce a $3P_1 + P_2$, a contradiction. □

We now prove two lemmas that together will allow us to provide an upper bound on the size of R.

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 $|R| \leq 2$.*

Proof. For contradiction, suppose R contains two non-adjacent vertices u_1 and u_2. Let D and D' be the two connected components of $G[O]$. Then D has an odd cycle C on vertices $v_1, \ldots, v_r$ for some $r \geq 3$ and D' has an odd cycle C' on vertices $w_1, \ldots, w_s$ for some $s \geq 3$.

Now, u_1 and u_2 are adjacent to at most one vertex of C, as otherwise they lie on an odd cycle in $G[B_T]$, which would contradict the fact that they are even vertices. Hence, as $r \geq 3$, we may assume that v_1 is not adjacent to u_1 nor to u_2 (see Figure 3). Hence, at least one of u_1 and u_2 has a neighbour in $\{w_1, w_2\}$, otherwise $\{u_1, u_2, v_1, w_1, w_2\}$ would induce a $3P_1 + P_2$. Say u_1 is adjacent to w_1. Similarly, one of u_1, u_2 has a neighbour in $\{w_2, w_3\}$. As u_1 already has a neighbour in C', we find that u_1 cannot be adjacent to w_2 or w_3, otherwise u_1 would be in an odd cycle of $G[B_T]$, contradicting $u_1 \in R$. Hence, u_2 is adjacent to either w_2 or w_3. So u_1 and u_2 each have a neighbour on C' and these neighbours are not the same.

By the same reasoning, but with the roles of C and C' reversed, we find that u_1 and u_2 also have (different) neighbours on C. However, we now find that there exists an odd cycle using u_1, u_2 and appropriate paths P_C and $P_{C'}$ between their neighbours on C and C', respectively. We conclude that R is a clique, and thus, as $G[R]$ is bipartite, $|R| \leq 2$. □

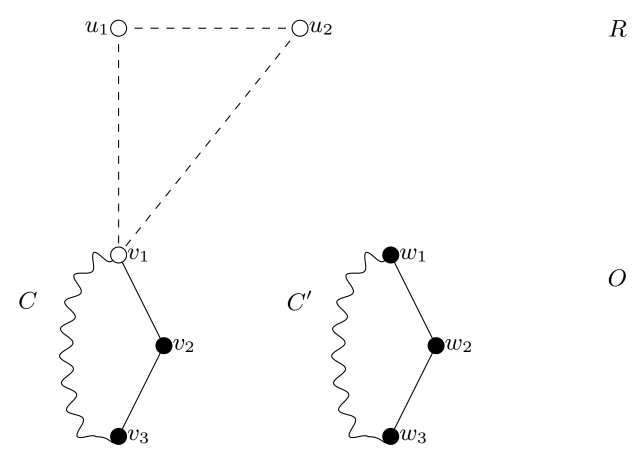

Fig. 3. The situation in Lemma 3 where dotted lines indicate non-edges. Note that not all edges incident with u_1 and u_2 are drawn.

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.*

Proof. Suppose that R contains an independent set $I = \{u_1, \ldots, u_5\}$ of five vertices. As S_T is mixed, O is nonempty. Hence, $G[B_T]$ has an odd cycle C. Let v_1, v_2, v_3 be consecutive vertices of C in that order. As G is $(3P_1+P_2)$-free, $v_1v_2 \in E$ and $\{u_1,u_2,u_3\}$ is independent, one of v_1, v_2 is adjacent to one of u_1,u_2,u_3, say v_1 is adjacent to u_1. Then v_1 must be adjacent to at least two vertices of $\{u_2,u_3,u_4,u_5\}$, otherwise three non-neighbours of v_1 in $\{u_2,u_3,u_4,u_5\}$, together with the edge u_1v_1, would induce a $3P_1+P_2$. Hence, we may assume without loss of generality that v_1 is adjacent to u_2 and u_3.

Let $i \in \{1,2,3\}$. As u_i is adjacent to v_1 and C is odd, u_i cannot be adjacent to v_2 or v_3, otherwise u_i would belong to an odd cycle in $G[B_T]$, so u_i would not be even, contradicting that $u_i \in R$. Hence, $\{u_1,u_2,u_3,v_2,v_3\}$ induces a $3P_1+P_2$, a contradiction. □

Together Lemmas 3 and 4 show that we always have that $|R| \leq 8$: if $G[O]$ is disconnected, then Lemma 3 proves the stronger result that $|R| \leq 2$, and if $G[O]$ is connected, we can use Lemma 4 and the fact that $G[R]$ is bipartite.

Before we continue with our structural analysis, we first prove a simple general lemma that we will need in the next two lemmas.

Lemma 5. *Let v_1 and v_2 be vertices in a graph G. Let P be a path from v_1 to v_2 and let e be an edge of P. If e belongs to an odd cycle C of G, then there is a path P' from v_1 to v_2 such that the parities of P and P' differ.*

Proof. Let u_1 be a vertex incident with e. Let $u_1,\dots,u_t$ be the set of vertices that belong to both P and C in the order that they are reached when traversing the path $Q = C - e$ from u_1. So u_t is also incident with e.

First suppose that, for $i = 1,\dots,t-1$, the parity of the path from u_i to u_{i+1} on P is the same as the parity of the path from u_i to u_{i+1} on Q. Then a walk W on P from u_1 to u_t that visits each of $u_1,\dots,u_t$ in turn has the same parity as Q. Let W' be the walk obtained from W after extending it with the edge u_tu_1. As W' is a walk from a vertex back to itself it is even, since it must traverse each edge an even number of times; the same number of times in each direction. So W' has even parity, and thus W has odd parity. However, the parity of Q is even, as Q is an odd cycle with one edge removed, a contradiction.

Thus there exists an index $i \in \{1,\dots,t-1\}$ such that the parity of the path from u_i to u_{i+1} on P differs from the parity of the path from u_i to u_{i+1} on Q. Let P' be the path formed from P by replacing the subpath from u_i to u_{i+1} with a subpath of Q. □

We will now look into the ways O and R are connected to each other. We say that a vertex in O is a *connector* if it has a neighbour in R. Here is our first structural lemma on connectors.

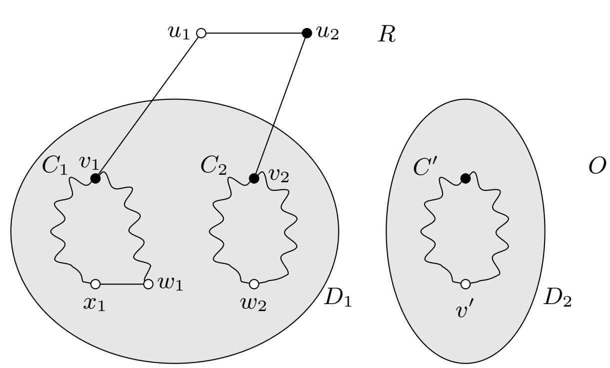

Fig. 4. An illustration for the proof of Lemma 6: the white vertices induce a $3P_1 + P_2$.

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, 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.

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 claim that $V(C_1) \cap V(C_2) = \emptyset$ and there is no edge between a vertex of C_1 and a vertex of C_2 except for possibly the edge v_1v_2. If not, then 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). By Lemma 5 applied to $G[D_1]$, there would be another path P' from v_1 to v_2 in $G[O]$ with a different parity to P. Thus the cycles $u_1Pu_2u_1$ and $u_1P'u_2u_1$ would have different parity (whether or not u_1 and u_2 are distinct) and so one of them would be odd. Hence, u_1 and u_2 would not be even. 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 4).

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 4), a contradiction. □

We need one more structural lemma (Lemma 7) about connectors, in the case where $G[O]$ is connected. In order to be able to make use of this lemma we need to exclude a special kind of mixed solution S_T. 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 with at least three vertices, 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 $G[B_T] = G[O \cup R]$ is indeed T-bipartite. We call the corresponding mixed solution S_T a *2-clique solution* (see Figure 5).

Lemma 7. *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 $G[R]$.*

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 6).

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. As in Lemma 6, we claim that $V(C_1) \cap V(C_2) = \emptyset$ and there is no edge between a vertex of C_1 and a vertex of C_2 except for possibly the edge v_1v_2. If there was such an edge, then there would be a path P from v_1 to v_2 in $G[O]$ with an edge that belongs to an odd cycle. By Lemma 5 applied to

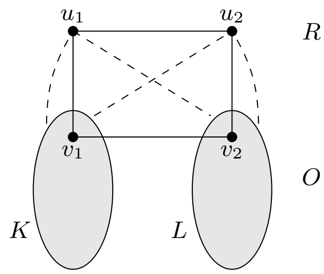

Fig. 5. The structure of B_T corresponding to a 2-clique solution S_T. The subgraphs K and L are each cliques on an odd number of vertices that is at least 3.

$G[D_1]$, there would be another path P' from v_1 to v_2 in $G[O]$ with a different parity than P. Then one of the cycles $u_1Pu_2u_1$ or $u_1P'u_2u_1$ is odd (whether or not u_1 and u_2 are distinct), implying that u_1 and u_2 would not be even.

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 except possibly the edge v_1v_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 contradiction, assume, without loss of generality, that C_1 has length at least 5. Let x and y be the two neighbours of v_1 in C_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 3 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 6). We conclude that as $G[O]$ is connected, v_1 and v_2 must be adjacent.

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

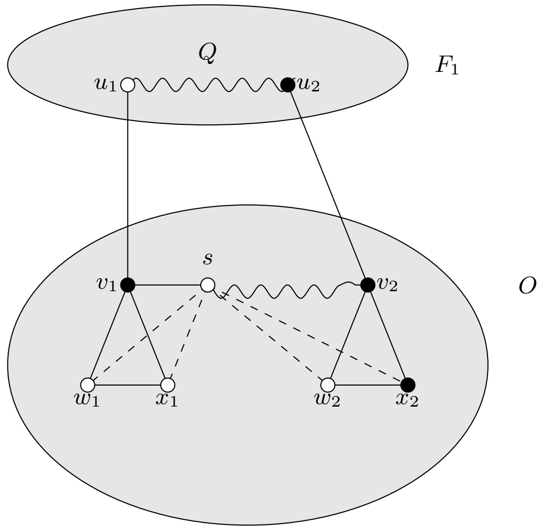

Fig. 6. The white vertices induce a $3P_1 + P_2$.

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; note that K does not intersect C_2 as $V(C_1) \cap V(C_2) = \emptyset$ and v_1v_2 is the only edge between C_1 and C_2. Let L be a maximal clique of $O \setminus K$ that contains C_2. We note that there is no edge e from a vertex of K to a vertex of L other than v_1v_2; else there is an odd cycle in $G[B_T]$ containing e and the three edges $v_1u_1u_2v_2$; if e is incident with v_1 (or v_2), then a 5-cycle is obtained using one further edge within L (or K), and otherwise a 7-cycle can be formed by adding two edges from K to connect e to v_1 and one edge from L to connect e to v_2. We prove below that $O = K \cup L$.

For contradiction, assume that r is a vertex of O that does not belong to $K \cup L$. 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 nonempty complete graphs, this means that r is adjacent to every vertex of $K \setminus \{v_1\}$ or to every vertex of $L \setminus \{v_2\}$; assume 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 $O = K \cup L$.

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 in 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

$\{u', w_1, w_2, u_1, u_2\}$ induces 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. $\square$

4.2 An Algorithmic Lemma

As part of our algorithm we need to be able to find a 2-clique solution of minimum weight in polynomial time. This is shown in the next lemma.

Lemma 8. *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.* (see Figure 7).

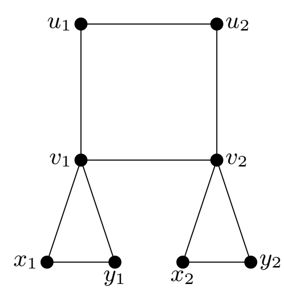

Fig. 7. A skeleton graph.

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. 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'.

Let $K' = \{v_1, x_1, y_1\} \cup (N(\{v_1, x_1, y_1\}) \cap V')$ and $L' = \{v_2, x_2, y_2\} \cup (N(\{v_2, x_2, y_2\}) \cap V')$. We now show that

(i) K' and L' partition V', and
(ii) 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. This shows (i).

We now prove (ii). 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.

We will now continue as follows. 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 8.

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.

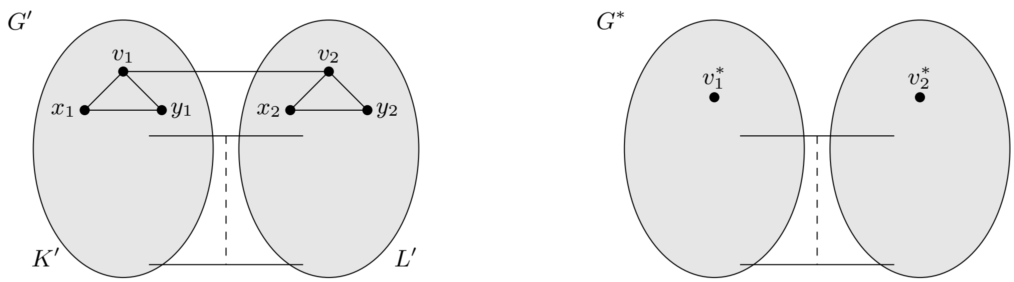

Fig. 8. The graph G' and G^* in the proof of Lemma 8.

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; note that we found this 2-clique solution in polynomial time. □

4.3 The Algorithm

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

Theorem 3. 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. We compute a mixed solution of minimum weight for each type.

Case 1. $G[O]$ is connected.
We first compute in polynomial time a 2-clique solution of minimum weight using Lemma 8. 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 7, 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 6, 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).

In the case where the algorithm chooses two (non-adjacent) connectors v and v' we proceed as follows. We remove any vertex from $R \cup T$ and any neighbour of R other than v and v'. Let (G', w') be the resulting weighted graph (where w' is the restriction of w to $V(G')$). We then solve Weighted Vertex Cut in polynomial time on G', w' and with v and v' as terminals. Let S be the output. We let $O = V(G') - S$.

By construction, v and v' are in different connected components of $G[O]$ and no vertex of R is adjacent to a connected component of $G[O]$ that does not contain v or v'. Together with the fact that $G[R \cup \{v, v'\}]$ is T-bipartite and that $G[O]$ has no vertices of T, this implies that $G[R \cup O]$ is T-bipartite. Note that $G[O]$ might contain even vertices or more than two connected components. However, what is only 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. Note that 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. Hence, processing Case 2 takes polynomial time.

Correctness and Running Time. The 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. □

5 Weighted Subset Odd Cycle Transversal on $(P_1 + P_3)$-free graphs

In this section, we will prove that Weighted Subset Odd Cycle Transversal can be solved in polynomial time for $(P_1 + P_3)$-free graphs (and also explain how the same result can be obtained for P_4-free graphs using existing machinery). We will follow the framework of Section 3 but in a less strict sense.

5.1 Auxiliary Results

We make use of an algorithm that decides the problem on P_4-free graphs. It can be shown that Weighted Subset Odd Cycle Transversal is polynomial-time solvable for P_4-free graphs by an obvious adaptation of the proof of the unweighted variant of Subset Odd Cycle Transversal from [6]. However, we remark that, as is the case for Weighted Subset Feedback Vertex Set discussed in Section 1, the result also follows from a meta-theorem of Courcelle et al. [11] that shows that on graph classes of bounded clique-width, certain optimization problems have linear time algorithms. We do not discuss the details, but it is enough to make the following two observations. First, P_4-free graphs have clique-width 1 and a corresponding 1-expression can be constructed in linear time. Second, the property that a subset of vertices S is an odd T-cycle transversal in a graph $G = (V, E)$ for some given set $T \subseteq V$ can be expressed in MSO_1 monadic second-order logic (with S as the only free monadic variable). Thus, we state without an explicit proof:

Theorem 4. Weighted Subset Odd Cycle Transversal *is polynomial-time solvable for P_4-free graphs.*

In order to prove our main result, we reduce to solving a weighted subset variant of the well-known Independent Set problem and also need the lemma below (Lemma 9). We say that $I_T \subseteq V(G)$ is a *T-independent set* of G if each vertex of $I_T \cap T$ is an isolated vertex in $G[I_T]$.

> Weighted Subset Independent Set
> *Instance:* a graph G, a subset $T \subseteq V(G)$, a non-negative vertex weighting w and an integer $k \geq 1$.
> *Question:* does G have a T-independent set I_T with $w(I_T) \geq k$?

Lemma 9. Weighted Subset Independent Set *is polynomial-time solvable for $3P_1$-free graphs.*

Proof. Let G be a $3P_1$-free graph, and let $T \subseteq V(G)$. Suppose I_T is a T-independent set of G. Observe that $|I_T \cap T| \leq 2$: if I_T contained three vertices of T, then they would form an independent set of size 3, contradicting that G is $3P_1$-free. Thus to find a maximum weight T-independent set we consider each subset T' of T of size at most 2. The maximum weight T-independent set that contains T' is clearly $V(G) \setminus (T \cup N(T'))$. Thus if we compute this collection of $\mathcal{O}(n^2)$ T-independent sets, this collection will contain a maximum-weight T-independent set. □

Finally, we need to define the complementary problem of Weighted Subset Independent Set: the weighted subset variant of Vertex Cover, which we now formally define. For a graph $G = (V, E)$ and a set $T \subseteq V$, a set $S_T \subseteq V$ is a *T-vertex cover* if S_T has at least contains one vertex incident to every edge that is incident to a vertex of T. Note that I_T is a T-independent set if and only if $V(G) \setminus I_T$ is a T-vertex cover.

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

We need a result on this problem for P_4-free graphs, which can be proven in the same way as the unweighted variant of Subset Vertex Cover in [6]. Alternatively, the property that a subset of vertices S is an T-vertex cover in a graph $G = (V, E)$ for some given set $T \subseteq V$ can be expressed in MSO_1 monadic second-order logic (with S as the only free monadic variable) and we can use the meta-theorem of Courcelle et al. [11] again.

Lemma 10. Weighted Subset Vertex Cover *can be solved in polynomial time for P_4-free graphs.*

The *paw* is the graph obtained from a triangle after adding a new vertex that is adjacent to only one vertex of the triangle. Alternatively, the paw is the complement of $P_1 + P_3$ and is therefore denoted $\overline{P_1 + P_3}$. We need the following result on paw-free graphs due to Olariu [23].

Lemma 11 ([23]). *Every connected $(\overline{P_1 + P_3})$-free graph is either triangle-free or $(P_1 + P_2)$-free.*

5.2 The Algorithm

Let G be a graph. We say that a set $X \subseteq V(G)$ *meets* a subgraph H of G if $X \cap V(H) \neq \emptyset$. A subgraph H of G is a *co-component* of G if H is a connected component of $\overline{G}$. We are now ready to prove the main result of this section.

Theorem 5. Weighted Subset Odd Cycle Transversal *is polynomial-time solvable for $(P_1 + P_3)$-free graphs.*

Proof. Let G be a $(P_1 + P_3)$-free graph. We present a polynomial-time algorithm for the optimization problem, where we seek to find $S_T \subseteq V(G)$ such that S_T is a minimum-weight odd T-cycle transversal. Note that for such an S_T, the set $B_T = V(G) \setminus S_T$ is a maximum-weight set such that $G[B_T]$ is a T-bipartite graph.

In $\overline{G}$, each connected component D is $(\overline{P_1 + P_3})$-free. By Lemma 11, D is either triangle-free or $(P_1 + P_2)$-free in $\overline{G}$; that is, D is $3P_1$-free or P_3-free in G. Let $D_1, D_2, \ldots, D_\ell$ be the co-components of G.

Let $B_T \subseteq V(G)$ such that $G[B_T]$ is a T-bipartite graph. For now, we do not require that B_T has maximum weight. We start by considering some properties of such a set B_T. Observe that $G - T$ is a T-bipartite graph, so we may have $B_T \cap T = \emptyset$.

Claim 1. If $B_T \cap T \neq \emptyset$, then $B_T \subseteq V(D_i) \cup V(D_j)$ for some $i, j \in \{1, 2, \ldots, \ell\}$.

We prove Claim 1 as follows. Suppose $u \in B_T \cap T$, say $u \in V(D_i)$ for some $i \in \{1, \ldots, \ell\}$. The claim holds if $B_T \subseteq V(D_i)$, so suppose $v \in B_T \setminus V(D_i)$. Then $v \in V(D_j)$ for some $j \in \{2, \ldots, \ell\}$ with $j \neq i$. If B_T also contains a vertex $v' \in D_{j'}$ for some $j' \in \{2, \ldots, \ell\} \setminus \{i, j\}$, then $\{u, v, v'\}$ induces a triangle in G, since D_i, D_j, and $D_{j'}$ are co-components. As this triangle contains $u \in T$, it is an odd T-cycle of $G[B_T]$, a contradiction. ◇

Note that Claim 1 states that B_T meets at most two co-components of G when $B_T \cap T \neq \emptyset$. The next two claims consider the case when B_T meets precisely two co-components of G.

Claim 2. Suppose $B_T \cap T \neq \emptyset$ and there exist distinct $i, j \in \{1, \ldots, \ell\}$ such that $B_T \cap V(D_i) \neq \emptyset$ and $B_T \cap V(D_j) \neq \emptyset$. If $B_T \cap V(D_i)$ contains a vertex of T, then $B_T \cap V(D_j)$ is an independent set.

We prove Claim 2 as follows. Suppose that $B_T \cap V(D_i)$ contains a vertex $t \in T$ and $G[B_T \cap V(D_j)]$ contains an edge uv. But then $\{u_1, v_1, t\}$ induces a triangle of G, since $V(D_j)$ is complete to $V(D_i)$, so $G[B_T]$ contains a contradictory odd T-cycle. $\diamond$

Claim 3. Suppose $B_T \cap T \neq \emptyset$ and there exist distinct $i, j \in \{1, \ldots, \ell\}$ such that $B_T \cap V(D_i) \neq \emptyset$ and $B_T \cap V(D_j) \neq \emptyset$. Either

- *$B_T \cap V(D_i)$ and $B_T \cap V(D_j)$ are independent sets of G, or*
- *$|B_T \cap V(D_i)| = 1$ and $B_T \cap V(D_i) \cap T = \emptyset$, and $B_T \cap V(D_j)$ is a T-independent set, up to swapping i and j.*

We prove Claim 3 as follows. Suppose B_T meets D_i and D_j, but $G[B_T \cap V(D_j)]$ contains an edge u_1v_1. Then, by Claim 2, $B_T \cap V(D_i)$ is disjoint from T. But $B_T \cap T \neq \emptyset$, so $B_T \cap V(D_j)$ contains some $t \in T$. Again by Claim 2, $B_T \cap V(D_i)$ is independent. It remains to show that $B_T \cap V(D_j)$ is a T-independent set and that $|B_T \cap V(D_j)| = 1$. Suppose $B_T \cap V(D_j)$ contains an edge tw_1, where $t \in T$. Then for any vertex $w_2 \in B_T \cap V(D_i)$, we have that $\{t, w_1, w_2\}$ induces a triangle, so $G[B_T]$ has a contradictory odd T-cycle. We deduce that each vertex of T in $B_T \cap V(D_j)$ is isolated in $G[B_T \cap V(D_j)]$. Now suppose there exist distinct $w_2, w_2' \in B_T \cap V(D_i)$. Then $tw_2u_1v_1w_2't$ is an odd T-cycle, a contradiction. So $|B_T \cap V(D_i)| = 1$. $\diamond$

We now describe the polynomial-time algorithm. Our strategy is to compute, in polynomial time, a collection of $\mathcal{O}(n^2)$ sets B_T such that $G[B_T]$ is T-bipartite, where a maximum-weight B_T is guaranteed to be in this collection. It then suffices to output a set from this collection of maximum weight.

First, we compute the co-components $D_1, D_2, \ldots, D_\ell$ of G. For each of the $\ell = \mathcal{O}(n)$ co-components D_i, we can recognise in polynomial time if D_i is $3P_1$-free or P_3-free (by brute force checking all vertex triples in D_i). Now, for each co-component D_i, we solve Weighted Subset Odd Cycle Transversal for D_i. Note that we can do this in polynomial time by Theorem 3 if D is $3P_1$-free and by Theorem 4 if D_i is P_3-free.

Now we consider each pair $\{D_1, D_2\}$ of distinct co-components. Note there are $\mathcal{O}(n^2)$ pairs to consider. For each pair we will compute three sets B_T such that $G[B_T]$ is T-bipartite.

1. We compute a maximum-weight independent set I_1 of D_1, and a maximum-weight independent set I_2 of D_2, where the weightings are inherited from the weighting w of G. Set $B_T = I_1 \cup I_2$. Note that $G[I_1 \cup I_2]$ is a complete bipartite graph, so it is certainly T-bipartite. We can compute these independent sets in polynomial time when restricted to $3P_1$-free or P_3-free graphs (for example, see [19]).
2. We select a maximum-weight vertex v_1 from $V(D_1) \setminus T$, and compute a maximum-weight T-independent set I_2 of D_2. When D_2 is $3P_1$-free, we can solve this in polynomial time by Lemma 9. On the other hand, when D_2 is P_3-free, we solve the complementary problem, in polynomial time, by Lemma 10. Set $B_T = I_2 \cup \{v_1\}$. Note that $G[B_T]$ is T-bipartite, since every vertex of T has degree 1 in $G[B_T]$ (its only neighbour is v_1).
3. This case is the symmetric counterpart to the previous: choose a maximum-weight vertex v_2 of $V(D_2) \setminus T$, compute a maximum-weight T-independent set I_1 of D_1, and set $B_T = I_1 \cup \{v_2\}$.

Finally, take the maximum-weight B_T among the (at most) $3\binom{\ell}{2}+\ell+1$ possibilities described, where the final possibility is that $B_T = V(G) \setminus T$.

To prove correctness of this algorithm, suppose B_T is a maximum-weight set such that $G[B_T]$ is T-bipartite. If $B_T \subseteq V(G) \setminus T$, then certainly the algorithm will either output $V(G) \setminus T$ or another solution with weight equal to $w(B_T)$. So we may assume that $B_T \cap T \neq \emptyset$. Now, by Claim 1, B_T meets one or two co-components of G. If it meets exactly one co-component D, then B_T is a maximum-weight set such that $D[B_T]$ is T-bipartite, which will be found by the algorithm in the first phase. If it meets two co-components D_1 and D_2, then the correctness of the algorithm follows from Claim 3. This concludes the proof. □

6 The Proof of Theorem 2

We need one new hardness result. The result is an analogue of Papadopoulos and Tzimas's hardness result for Weighted Subset Feedback Vertex Set on $5P_1$-free graphs [28, Theorem 2]. The proof is omitted as it is essentially identical, as all the relevant T-cycles in the constructed Weighted Subset Feedback Vertex Set instance are odd.

Theorem 6. Weighted Subset Odd Cycle Transversal *is* NP*-complete for* $5P_1$*-free graphs.*

We are now ready to prove Theorem 2.

Theorem 2 (restated). *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.*

Proof. We first recall the result of [8] that Odd Cycle Transversal, that is, Weighted Subset Odd Cycle Transversal where $T = \emptyset$ and $w \equiv 1$, is NP-complete on H-free graphs if H has a cycle or a claw. In the remaining case H is a linear forest. If H contains an induced $2P_2$, then we use a result of [6], which states that Subset Odd Cycle Transversal is NP-complete for split graphs, or equivalently, $(C_4, C_5, 2P_2)$-free graphs. If H contains an induced $5P_1$, then we use Theorem 6. In the other cases, we use Theorems 3, 5 and 4, respectively. □

7 Conclusions

We determined the complexity of Weighted Subset Odd Cycle Transversal on H-free graphs except when $H \in \{2P_1 + P_3,\ P_1 + P_4,\ 2P_1 + P_4\}$. In particular, our results demonstrate that the classifications of Weighed Subset Odd Cycle Transversal and Subset Odd Cycle Transversal do not coincide for H-free graphs, just as was known already for Weighted Subset Feedback Vertex Set and Subset Feedback Vertex Set [28]. In addition, our results demonstrate that so far, the complexities of Weighted Subset Feedback Vertex Set and Weighted Subset Odd Cycle Transversal do coincide on H-free graphs.

We believe that the case $H = 2P_1 + P_3$ is polynomial-time solvable for both problems using the methodology of our framework and our algorithms for $H = P_1 + P_3$ as a subroutine. We leave this for future research. The other two cases are open even for Odd Cycle Transversal. 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.

We note finally that that there are other similar transversal problems that have been studied, but their complexity classifications on H-free graphs have not been settled: (Subset) Even Cycle Transversal [18, 21, 24], for example. Versions of the transversal problems that we have considered that have the additional constraint that the transversal must induce either a connected graph or an independent set have also been studied for H-free graphs [4, 8, 13, 17]. An interesting direction for further research is to consider the subset variant of these problems, and, more generally, to understand the relationships amongst the computational complexities of all these problems.

Acknowledgments. We thank two anonymous reviewers for helpful comments.

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. Algorithmica, to appear (2022)
3. Bodlaender, H.L., Möhring, R.H.: The pathwidth and treewidth of cographs. SIAM Journal on Discrete Mathematics **6**, 181–188 (1993)
4. Bonamy, M., Dabrowski, K.K., Feghali, C., Johnson, M., Paulusma, D.: Independent feedback vertex set for P_5-free graphs. Algorithmica **81**, 1342–1369 (2019)
5. Brandstädt, A., Le, V.B., Spinrad, J.P.: Graph Classes: A Survey, SIAM Monographs on Discrete Mathematics and Applications, vol. 3. SIAM (1999)
6. Brettell, N., Johnson, M., Paesani, G., Paulusma, D.: Computing subset transversals in H-free graphs. Theoretical Computer Science **902**, 76–92 (2022)
7. Brettell, N., Johnson, M., Paulusma, D.: Computing weighted subset transversals in H-free graphs. Proc. WADS 2021, LNCS **12808**, 229–242 (2021)
8. 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)
9. Corneil, D.G., Lerchs, H., Burlingham, L.S.: Complement reducible graphs. Discrete Applied Mathematics **3**, 163–174 (1981)
10. Corneil, D.G., Perl, Y., Stewart, L.K.: A linear recognition algorithm for cographs. SIAM Journal on Computing **14**, 926–934 (1985)
11. Courcelle, B., Makowsky, J.A., Rotics, U.: Linear time solvable optimization problems on graphs of bounded clique-width. Theory of Computing Systems **33**, 125–150 (2000)
12. 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)
13. Dabrowski, K.K., Johnson, M., Paesani, G., Paulusma, D., Zamaraev, V.: 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)
14. Fomin, F.V., Heggernes, P., Kratsch, D., Papadopoulos, C., Villanger, Y.: Enumerating minimal subset feedback vertex sets. Algorithmica **69**, 216–231 (2014)
15. Gartland, P., Lokshtanov, D., Pilipczuk, M., Pilipczuk, M., Rzążewski, P.: Finding large induced sparse subgraphs in $c_{>t}$-free graphs in quasipolynomial time. Proc. STOC 2021 pp. 330–341 (2021)
16. Grzesik, A., Klimošová, T., Pilipczuk, M., Pilipczuk, M.: Polynomial-time algorithm for maximum weight independent set on P_6-free graphs. ACM Transactions on Algorithms **18**, 1–57 (2022)
17. Johnson, M., Paesani, G., Paulusma, D.: Connected Vertex Cover for $(sP_1 + P_5)$-free graphs. Algorithmica **82**, 20–40 (2020)
18. Kakimura, N., Kawarabayashi, K., Kobayashi, Y.: Erdős-Pósa property and its algorithmic applications: parity constraints, subset feedback set, and subset packing. Proc. SODA 2012 pp. 1726–1736 (2012)
19. Lokshtanov, D., Vatshelle, M., Villanger, Y.: Independent set in P_5-free graphs in polynomial time. Proc. SODA 2014 pp. 570–581 (2014)
20. Lozin, V., Malyshev, D., Mosca, R., Zamaraev, V.: Independent domination versus weighted independent domination. Information Processing Letters **156**, 105914 (2020)

21. Misra, P., Raman, V., Ramanujan, M.S., Saurabh, S.: Parameterized algorithms for even cycle transversal. Proc. WG 2012 pp. 172–183 (2012)
22. Munaro, A.: On line graphs of subcubic triangle-free graphs. Discrete Mathematics **340**, 1210–1226 (2017)
23. Olariu, S.: Paw-free graphs. Information Processing Letters **28**(1), 53–54 (1988)
24. Paesani, G., Paulusma, D., Rzążewski, P.: Feedback Vertex Set and Even Cycle Transversal for H-free graphs: finding large block graphs. Proc. MFCS 2021, LIPIcs **202**, 1–14 (2021)
25. Paesani, G., Paulusma, D., Rzazewski, P.: Feedback Vertex Set and Even Cycle Transversal for H-free graphs: Finding large block graphs (journal version of [24]). CoRR **abs/2105.02736** (2021)
26. Paesani, G., Paulusma, D., Rzążewski, P.: Classifying Subset Feedback Vertex set for H-free graphs. CoRR **abs/2201.00430** (2022)
27. 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)
28. Papadopoulos, C., Tzimas, S.: Subset feedback vertex set on graphs of bounded independent set size. Theoretical Computer Science **814**, 177–188 (2020)
29. Poljak, S.: A note on stable sets and colorings of graphs. Commentationes Mathematicae Universitatis Carolinae **15**, 307–309 (1974)