Close Relatives of Feedback Vertex Set Without Single-Exponential Algorithms Parameterized by Treewidth

Benjamin Bergougnoux
Department of Informatics, University of Bergen, Norway
benjamin.bergougnoux@uib.no

Édouard Bonnet
Université Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668, France
edouard.bonnet@ens-lyon.fr

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

O-joung Kwon
Department of Mathematics, Incheon National University, South Korea
Discrete Mathematics Group, Institute for Basic Science (IBS), Daejeon, South Korea
ojoungkwon@gmail.com

Abstract

The Cut & Count technique and the rank-based approach have lead to single-exponential FPT algorithms parameterized by treewidth, that is, running in time $2^{\mathcal{O}(\mathsf{tw})}n^{\mathcal{O}(1)}$, for Feedback Vertex Set and connected versions of the classical graph problems (such as Vertex Cover and Dominating Set). We show that Subset Feedback Vertex Set, Subset Odd Cycle Transversal, Restricted Edge-Subset Feedback Edge Set, Node Multiway Cut, and Multiway Cut are unlikely to have such running times. More precisely, we match algorithms running in time $2^{\mathcal{O}(\mathsf{tw} \log \mathsf{tw})}n^{\mathcal{O}(1)}$ with tight lower bounds under the Exponential Time Hypothesis, ruling out $2^{o(\mathsf{tw} \log \mathsf{tw})}n^{\mathcal{O}(1)}$, where n is the number of vertices and tw is the treewidth of the input graph. Our algorithms extend to the weighted case, while our lower bounds also hold for the larger parameter pathwidth and do not require weights. We also show that, in contrast to Odd Cycle Transversal, there is no $2^{o(\mathsf{tw} \log \mathsf{tw})}n^{\mathcal{O}(1)}$-time algorithm for Even Cycle Transversal.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Fixed parameter tractability

Keywords and phrases Subset Feedback Vertex Set, Multiway Cut, Parameterized Algorithms, Treewidth, Graph Modification, Vertex Deletion Problems

Digital Object Identifier 10.4230/LIPIcs.IPEC.2020.3

Related Version A full version of the paper is available at `https://arxiv.org/abs/2007.14179`.

Acknowledgements This work was initiated while the authors attended the "2019 IBS Summer research program on Algorithms and Complexity in Discrete Structures", hosted by the IBS discrete mathematics group.

1 Introduction

Courcelle's Theorem [8] states that any problem definable in MSO_2 logic can be solved in linear time on graphs of bounded treewidth. However, the algorithms obtained through Courcelle's meta-theorem have a huge dependency on treewidth. For certain problems, efforts have been spent to find the "smallest" function f for which we can obtain an algorithm that, given a graph with treewidth tw, has running time $f(\mathsf{tw})n^{\mathcal{O}(1)}$. For Feedback Vertex

Set, standard dynamic programming techniques can be used to obtain an algorithm running in $2^{\mathcal{O}(\mathsf{tw}\log \mathsf{tw})}n^{\mathcal{O}(1)}$ time, and for a while many believed this to be, in a sense, best possible. However, this changed in 2011 when Cygan et al. [11] developed the Cut&Count technique, by which they obtained a *single-exponential* $3^{\mathsf{tw}}n^{\mathcal{O}(1)}$-time randomized algorithm. Following this, Bodlaender et al. [3] showed there is a deterministic $2^{\mathcal{O}(\mathsf{tw})}n^{\mathcal{O}(1)}$-time algorithm, using a rank-based approach and the concept of representative sets. Moreover, also in 2011, Lokshtanov et al. [17] developed a framework yielding $2^{\Omega(\mathsf{tw}\log \mathsf{tw})}n^{\mathcal{O}(1)}$-time lower bounds under the Exponential Time Hypothesis (ETH) [16]. Recall that the ETH asserts that there is a real number $\delta > 0$ such that 3-SAT cannot be solved in time $2^{\delta n}$ on n-variable formulas. Lokshtanov et al.'s paper prompted several authors to investigate the exact time-dependency on treewidth for a variety of graph modification problems.

For a *vertex-deletion problem*, the task is to delete at most k vertices so that the resulting graph is in some target class. Feedback Vertex Set can be viewed as a vertex-deletion problem where the graphs in the target class consist of blocks with at most two vertices (a *block* is a maximal subgraph H such that H has no cut vertices). Bonnet et al. [6] considered the class of problems, generalizing Feedback Vertex Set, where the target graphs consist of blocks each of which has a bounded number of vertices, and is in some fixed hereditary, polynomial-time recognizable class $\mathcal{P}$. They showed that such a problem is solvable in time $2^{\mathcal{O}(\mathsf{tw})}n^{\mathcal{O}(1)}$ precisely when each graph in $\mathcal{P}$ is chordal (when $\mathcal{P}$ does not satisfy this condition, an algorithm with running time $2^{o(\mathsf{tw}\log \mathsf{tw})}n^{\mathcal{O}(1)}$ would refute the ETH). Baste et al. [2] studied another generalization of Feedback Vertex Set: the vertex-deletion problem where the target graphs are those having no minor isomorphic to a fixed graph H. They showed a single-exponential parameterized algorithm in treewidth is possible precisely when H is a minor of the banner (the cycle on four vertices with a degree-1 vertex attached to it), but H is not P_5 (the path graph on five vertices), assuming the ETH holds.

Slightly superexponential parameterized algorithms, running in time $2^{\mathcal{O}(\mathsf{tw}\log \mathsf{tw})}n^{\mathcal{O}(1)}$, are by no means a formality for problems that are FPT in treewidth. For instance, Pilipczuk [19] showed that deciding if a graph has a transversal of size at most k hitting all cycles of length exactly ℓ (or length at most ℓ) for a fixed value ℓ cannot be solved in time $2^{o(\mathsf{tw}^2)}n^{\mathcal{O}(1)}$, unless the ETH fails. This lower bound matches a dynamic-programming based algorithm running in time $2^{\mathcal{O}(\mathsf{tw}^2)}n^{\mathcal{O}(1)}$. Cygan et al. [9] investigated the more general problem of hitting all subgraphs H of a given graph G, for a fixed pattern graph H, again parameterized by treewidth. For various H, they found algorithms running in time $2^{\mathcal{O}(\mathsf{tw}^{u(H)})}n^{\mathcal{O}(1)}$, and proved ETH lower bounds in $2^{\Omega(\mathsf{tw}^{\ell(H)})}n^{\mathcal{O}(1)}$, for values $1 < \ell(H) \leqslant u(H)$ depending on H. Another recent example is provided by Sau and Uéverton [20] who prove similar results for the analogous problem where "subgraphs" is replaced by "induced subgraphs". Finally, for the vertex-deletion problem where the target class is a proper minor-closed class given by the non-empty list of forbidden minors, it is still open if the double-exponential dependence on treewidth is asymptotically best possible [1].

Sometimes, only a seemingly slight generalization of Feedback Vertex Set can result in problems with no single-exponential algorithm parameterized by treewidth. Bonamy et al. [5] showed that Directed Feedback Vertex Set can be solved in time $2^{\mathcal{O}(\mathsf{tw}\log \mathsf{tw})}n^{\mathcal{O}(1)}$ but not faster under the ETH, where tw is the treewidth of the underlying undirected graph. In this paper, we consider another collection of problems that generalize Feedback Vertex Set, and that do not have single-exponential algorithms parameterized by treewidth. An equivalent formulation of FVS is to find a transversal of *all* cycles in a given graph. We consider problems where the goal is to find a transversal of *some subset* of the cycles of a given graph. If this subset of cycles is the set of cycles intersecting some fixed set of vertices S, we obtain the following problem:

Subset Feedback Vertex Set (Subset FVS) **Parameter:** $\mathsf{tw}(G)$
Input: A graph G, a subset of vertices $S \subseteq V(G)$, and an integer k.
Question: Is there a set of at most k vertices hitting all the cycles containing a vertex in S?

If we further restrict this set of cycles to those that are odd, we obtain the next problem:

Subset Odd Cycle Transversal (Subset OCT) **Parameter:** $\mathsf{tw}(G)$
Input: A graph G, a subset of vertices $S \subseteq V(G)$, and an integer k.
Question: Is there a set of at most k vertices hitting all the odd cycles going through a vertex in S?

Both of these problems are NP-complete. By setting $S = V(G)$, one sees that the latter problem generalises Odd Cycle Transversal, for which Fiorini et al. [14] presented a $2^{\mathcal{O}(\mathsf{tw})}n^{\mathcal{O}(1)}$-time algorithm.

Alternatively, one can require a transversal of even cycles. We first consider the problem of finding a transversal of *all* even cycles since, to the best of our knowledge, the fine-grained complexity of this problem parameterized by treewidth has not previously been studied.

Even Cycle Transversal (ECT) **Parameter:** $\mathsf{tw}(G)$
Input: A graph G and an integer k.
Question: Is there a set of at most k vertices hitting all the even cycles of G?

We now move to edge variants of FVS. Note that Feedback Edge Set, where the goal is to find a set of edges of weight at most k that hits the cycles, can be solved in linear time, since it is equivalent to finding a maximum-weight spanning forest. Xiao and Nagamochi showed that the subset variants Vertex-Subset Feedback Edge Set and Edge-Subset Feedback Edge Set, where the deletion set only needs to hit cycles containing a vertex or an edge (respectively) of a given set S, can also be solved in linear time [21]. On the other hand, the latter problem becomes NP-complete when the deletion set cannot intersect S. This problem is known as Restricted Edge-Subset Feedback Edge Set.

Restricted Edge-Subset Feedback Edge Set (RESFES) **Parameter:** $\mathsf{tw}(G)$
Input: A graph G, a subset of edges $S \subseteq E(G)$, and an integer k.
Question: Is there a set of at most k edges of $E(G) \setminus S$ whose removal yields a graph without any cycle containing at least one edge of S?

The final two NP-complete problems we consider are closely related to Subset Feedback Vertex Set and Restricted Edge-Subset Feedback Edge Set. They are well-known problems with an abundant literature of approximation and parameterized algorithms.

Node Multiway Cut **Parameter:** $\mathsf{tw}(G)$
Input: A graph G, a subset of vertices $T \subseteq V(G)$, called *terminals*, and an integer k.
Question: Is there a set of at most k vertices of $V(G) \setminus T$ hitting every path between a pair of terminals?

Multiway Cut **Parameter:** $\mathsf{tw}(G)$
Input: A graph G, a subset of vertices $T \subseteq V(G)$, called *terminals*, and an integer k.
Question: Is there a set of at most k edges hitting every path between a pair of terminals?

The look-alike problem MULTICUT, where the task is to separate each pair of terminals in a given set of pairs (rather than all the pairs in a given set), is NP-complete on trees [15]. Therefore a parameterization by treewidth cannot help here. In the language of parameterized complexity, MULTICUT parameterized by treewidth is paraNP-complete.

1.1 Our contribution

With the exception of EVEN CYCLE TRANSVERSAL, for which we provide only a lower bound, we show that all the problems formally defined so far admit a slightly superexponential parameterized algorithm, and that this running time cannot be improved, unless the ETH fails. We leave as an open problem the existence of a slightly superexponential algorithm for (SUBSET) EVEN CYCLE TRANSVERSAL parameterized by treewidth. We note that Deng et al. [12] have already shown that MULTIWAY CUT can be solved in time $2^{\mathcal{O}(\mathsf{tw} \log \mathsf{tw})} n^{\mathcal{O}(1)}$. Our algorithms work for treewidth and weights, while our lower bounds hold for the larger parameter pathwidth and do not require weights.

On the algorithmic side we show the following:

▶ **Theorem 1.** *The following problems can be solved in time $2^{\mathcal{O}(\mathsf{tw} \log \mathsf{tw})} n^{\mathcal{O}(1)}$ on n-vertex graphs with treewidth* tw:
- *SUBSET FEEDBACK VERTEX SET,*
- *SUBSET ODD CYCLE TRANSVERSAL,*
- *RESTRICTED EDGE-SUBSET FEEDBACK EDGE SET, and*
- *NODE MULTIWAY CUT.*

We provide algorithms having the claimed running time for the weighted versions of each of the four problems in Theorem 1. In these weighted versions, the input graph is given with a weight function w on the vertices when the problem is to find a set of vertices, or on the edges when the problem is to find a set of edges. Furthermore, in the weighted versions, the problem asks for a solution of weight at most k.

On the complexity side, the main conceptual contribution of the paper is to show that problems seemingly quite close to FEEDBACK VERTEX SET do not admit a single-exponential algorithm parameterized by treewidth, under the ETH.

▶ **Theorem 2.** *Unless the ETH fails, the following problems cannot be solved in time $2^{o(\mathsf{pw} \log \mathsf{pw})} n^{\mathcal{O}(1)}$ on n-vertex graphs with pathwidth* pw:
- *SUBSET FEEDBACK VERTEX SET,*
- *SUBSET ODD CYCLE TRANSVERSAL,*
- *EVEN CYCLE TRANSVERSAL,*
- *RESTRICTED EDGE-SUBSET FEEDBACK EDGE SET,*
- *NODE MULTIWAY CUT, and*
- *MULTIWAY CUT.*

For the last two problems, our reductions build instances where the number of terminals $|T|$ is $\Theta(\mathsf{pw})$. Thus we also rule out a running time of $|T|^{o(\mathsf{pw})}$. All the reductions are from $k \times k$-(PERMUTATION) INDEPENDENT SET/CLIQUE following a strategy suggested by Lokshtanov et al. [18] (see for instance, [2, 5–7, 13]). These problems cannot be solved in time $2^{o(k \log k)}$, unless the ETH fails.

$k \times k$-INDEPENDENT SET	**Parameter:** k
Input: A graph H with vertex set $V(H) = [k]^2$ for some integer k.	
Question: An independent set of size k hitting each column exactly once.	

$k \times k$-Permutation Independent Set	**Parameter:** k
Input: A graph H with vertex set $V(H) = [k]^2$ for some integer k.	
Question: An independent set of size k hitting each column and each row exactly once.	

A *row* is a set of vertices of the form $\{(i, 1), (i, 2), \ldots, (i, k)\} \subset V(H)$ for some $i \in [k]$, while a column is a set $\{(1, j), (2, j), \ldots, (k, j)\} \subset V(H)$ for some $j \in [k]$. The problem $k \times k$-(Permutation) Clique is defined analogously, where the solution is required to be a clique rather than an independent set.[1]

Roadmap for the lower bounds. To prove Theorem 2, we start by designing a gadget specification for generic vertex-deletion problems. We show that any such problem, allowing for gadgets respecting the specification, has the lower bound given in Theorem 2. This is achieved by a meta-reduction from $k \times k$-Permutation Independent Set. We give gadgets for Subset FVS, Subset OCT, and ECT that comply with the specification. We thus obtain the first three items of the theorem in a unified way, with simple and reusable gadgets. This mini-framework may in principle be useful for other vertex-deletion problems.

In order to show a stronger lower bound for Node Multiway Cut, with the number of terminals in $\Theta(k)$, we depart from the previous specification slightly, although we still use some shared notation and arguments to bound the pathwidth, where convenient. This reduction is from $k \times k$-Independent Set.

Finally, the reduction to Multiway Cut is more intricate. For this problem it is surprisingly challenging to discourage the undesirable solutions "cutting close" to every terminal but one, where the deletion set yields a very large connected component for one terminal, and small components for the rest of the terminals. In particular, the trick used for the Node Multiway Cut lower bound cannot be replicated. We overcome this issue by designing a somewhat counter-intuitive edge gadget which encourages the retention of as many pairs of endpoints linked to two (distinct) terminals as possible. This uses the simple fact that, in a Δ-regular graph, a clique of size k minimizes the number of edges covered by k vertices: $\Delta k - \binom{k}{2}$ vs Δk for an independent set of size k. We then reduce from $k \times k$-Permutation Clique. We discuss why getting the same lower bound for a regular variant of $k \times k$-Permutation Clique is technical, and bypass that difficulty by encoding a *degree-equalizer* gadget directly in the Multiway Cut instance. As a side note, we nevertheless prove that a semi-regular variant of $k \times k$-Clique also has the slightly superexponential lower bound. This proof uses a constructive version of the Hajnal-Szemerédi theorem on equitable colorings.

Roadmap for the algorithms. To prove Theorem 1, we first present a $2^{\mathcal{O}(\mathsf{tw} \log \mathsf{tw})} n^3$-time algorithm for the weighted variant of Subset OCT. With a few modifications, it can solve the weighted variant of Subset FVS. We obtain algorithms for the other problems in Theorem 1 by reducing these problems to the weighted variant of Subset FVS.

Let us explain our approach for Subset OCT on a graph G with $S \subseteq V(G)$. We solve Subset OCT indirectly by finding a set $X \subseteq V(G)$ of maximum weight that induces a graph with no odd cycles traversing S (we call such a graph S-bipartite). We prove that a graph has no odd cycle traversing S if and only if for each block C, either C is bipartite or C has no vertex in S. From this characterization, we prove that it is enough to store $2^{\mathcal{O}(\mathsf{tw} \log \mathsf{tw})}$ partial solutions at each bag B of a tree decomposition.

[1] Observe that we switch the columns and the rows compared to the original definition of $k \times k$-Clique [18]. While this is of course equivalent, it will make the representation of some gadgets slightly more conducive to the page layout.

Let B be a bag of the tree decomposition of G and G_B be the graph induced by the vertices in B and its descendant bags in the tree decomposition. A partial solution of G_B is a set $X \subseteq V(G_B)$ that induces an S-bipartite graph. We design an equivalence relation $\equiv_B$ on the partial solutions of G_B such that for every $X \equiv_B Y$ and $W \subseteq V(G) \setminus V(G_B)$, $G[X \cup W]$ is S-bipartite if and only if $G[Y \cup W]$ is S-bipartite. Consequently, it is enough to keep a partial solution of maximum weight for each equivalence class of $\equiv_B$. Intuitively, the equivalence relation $\equiv_B$ is based on the information: (1) how the blocks of $G[X]$ intersecting B are connected, (2) whether important blocks (that have the possibility to create an S-traversing odd cycle later) contain a vertex of S, and (3) the parity of the paths between the vertices in B. Since $\equiv_B$ has $2^{\mathcal{O}(\mathsf{tw}\log\mathsf{tw})}$ equivalence classes, we deduce from this equivalence relation a $2^{\mathcal{O}(\mathsf{tw}\log\mathsf{tw})}n^3$-time algorithm with standard dynamic-programming techniques. The polynomial factor n^3 appears because we can test $X \equiv_B Y$ in time $\mathcal{O}(n^2)$.

For the weighted variant of Subset FVS, we can use the same equivalence relation without (3). We reduce the weighted variant of Node Multiway Cut to Subset FVS by adding a vertex v of infinite weight adjacent to the set of terminals, setting $S = \{v\}$, and also giving infinite weights to the terminals. Furthermore, we reduce the weighted variant of Restricted Edge-Subset Feedback Edge Set to the weighted variant of Subset FVS by subdividing each edge, setting S as the set of subdivided vertices corresponding to the given subset of edges, and giving infinite weights to the original vertices and the vertices in S. These two reductions show that both problems admit $2^{\mathcal{O}(\mathsf{tw}\log\mathsf{tw})}n^3$-time algorithms.

Organization. The rest of the paper is organized as follows. In Section 3 we prove all the ETH lower bounds of Theorem 2. More precisely, in Section 3.1 we introduce a gadget specification for a generic vertex-deletion problem, and we show the slightly superexponential lower bound for any problem complying with the gadget specification. In Section 3.2 we design gadgets for Subset FVS, Subset OCT, ECT, and thus obtain the first three items of Theorem 2. In Sections 3.3 and 3.4 we present specific reductions for Node Multiway Cut and Multiway Cut, respectively. In Section 4 we prove that the weighted variants of Subset OCT, Subset FVS, Restricted Edge-Subset Feedback Edge Set, and Node Multiway Cut admit $2^{\mathcal{O}(\mathsf{tw}\log\mathsf{tw})}n^3$-time algorithms. The statements marked with a $\star$ have their proof deferred to the full version.

2 Preliminaries

Our graph-theoretic terminology is standard; any terminology undefined here is deferred to the long version. A set $X \subseteq V(G)$ is a *clique* if G has an edge between every pair of vertices in X. A graph with vertex set $X \cup Y$ that has an edge between every vertex $x \in X$ and $y \in Y$ is called a *biclique*, and is denoted $K_{|X|,|Y|}$. For $u, v \in V(G)$, we say that u and v are *twins* if $N(u) = N(v)$. If u and v are adjacent, then we say that u and v are *true* twins; whereas when u and v are non-adjacent twins, we say that u and v are *false* twins.

A vertex v of G is a *cut vertex* if the deletion of v from G increases the number of connected components. We say G is *2-connected* if it is connected and has no cut vertices. Note that every connected graph on at most two vertices is 2-connected. A *block* of G is a maximal 2-connected subgraph of G.

A *tree decomposition* of a graph G is a pair $(T, \mathcal{B})$ consisting of a tree T and a family $\mathcal{B} = \{B_t\}_{t\in V(T)}$ of sets $B_t \subseteq V(G)$, called *bags*, satisfying the following three conditions:

1. $V(G) = \bigcup_{t\in V(T)} B_t$,
2. for every edge uv of G, there exists a node t of T such that $u, v \in B_t$, and
3. for $t_1, t_2, t_3 \in V(T)$, $B_{t_1} \cap B_{t_3} \subseteq B_{t_2}$ whenever t_2 is on the path from t_1 to t_3 in T.

The *width* of a tree decomposition $(T, \mathcal{B})$ is $\max\{|B_t| - 1 : t \in V(T)\}$. The *treewidth* of G is the minimum width over all tree decompositions of G. A *path decomposition* is a tree decomposition $(P, \mathcal{B})$ where P is a path. The *pathwidth* of G is the minimum width over all path decompositions of G. We denote a path decomposition $(P, \mathcal{B})$ as $(B_{v_1}, \dots, B_{v_t})$, where P is a path $v_1 v_2 \cdots v_t$.

3 Superexponential lower bounds parameterized by treewidth

Our reductions for Subset FVS, Subset OCT, and ECT, in Section 3.2, will have the same skeleton. In order to avoid repeating the same arguments, we show in Section 3.1 the lower bound of Theorem 2 for a meta-problem. We prove the lower bound for Node Multiway Cut in Section 3.3, and the lower bounds for Multiway Cut and Restricted Edge-Subset Feedback Edge Set in Section 3.4.

3.1 Lower bound for a generic vertex-deletion problem

The scope of application of Theorem 2 is any *hereditary* vertex-deletion problem Π; that is, if $G - X$ satisfies a problem instance $P(\Pi)$, then $G - X'$ also satifies $P(\Pi)$ for every $X' \supseteq X$. The main part of the input is a graph G and a non-negative integer k'. In addition, we allow any sort of labelings of G, be it subsets of vertices $S_1, S_2, \ldots \subseteq V(G)$, of edges $E_1, E_2, \ldots, \subseteq E(G)$, pairs of vertices $P_1, P_2, \ldots \subseteq \binom{V(G)}{2}$, etc. The goal is to find a subset $X \subseteq V(G)$ of k' vertices such that a property $P(\Pi)$, dependent on Π, is satisfied on $G - X$ with its induced labeling. A subset of vertices $A \subseteq V(G)$ is a Π-*obstruction* if $G[A]$ does not satisfy $P(\Pi)$. A set $X \subseteq V(G)$ is Π-*legal* if $G - X$ satisfies $P(\Pi)$ (in particular, solutions are Π-legal sets of size k'). As $P(\Pi)$ is assumed hereditary, a Π-legal set intersects every Π-*obstruction*. Finally a Π-*legal* s-*deletion within* Y is a set $X \subseteq Y$ of size at most s such that $G[Y \setminus X]$ satisfies $P(\Pi)$.

Common base

The meta-result of Theorem 3 concerns hereditary vertex-deletion problems admitting four types of gadgets. These gadgets, which will eventually depend on Π, are attached to a common problem-independent base. We first describe the common base. $H_\bullet$ is a set of $2k^2$ vertices, for some implicit positive integer k. We denote these vertices by $v_\bullet(i, j, z)$ for each $i \in [k]$, $j \in [k]$, and $z \in [2]$. We imagine the vertices of $H_\bullet$ being displayed in a k-by-k grid with $v_\bullet(i, j, 1)$ and $v_\bullet(i, j, 2)$ side by side in the i-th row and j-th column.

The *base* consists of copies of $H_\bullet$ that we denote by $H_1, H_2, \ldots$ and typically index by p. The vertices of H_p are denoted by $v_p(i, j, z)$. The vertices $v_p(i, j, 1)$ and $v_p(i, j, 2)$ are said to be *homologous*. We set $C_{p,j} := \bigcup_{i \in [k], z \in [2]} \{v_p(i, j, z)\}$ and refer to it as the j-*th column* of H_p. Similarly $R_{p,i} := \bigcup_{j \in [k], z \in [2]} \{v_p(i, j, z)\}$ is called the i-*th row* of H_p. We can attach to the base a list of gadgets as detailed now. The vertices added to the base are called *additional* or *new*.

Column selector gadget

A k-*column selector* gadget has the following specification. Its vertex set is a single column $C_{p,j}$ plus $\mathcal{O}(k)$ additional vertices $\mathcal{C}_{\text{sel}}(p, j)$. The only restriction on the edge set of the gadget is that homologous vertices should remain non-adjacent. Other than that, any edge can be added within $C_{p,j}$. However the open neighborhood of $\mathcal{C}_{\text{sel}}(p, j)$ has to be contained in $C_{p,j}$.

A problem Π *admits a column selector gadget* if, for every positive integer k, one can build in time $k^{\mathcal{O}(1)}$ a k-column selector such that the only Π-legal $(2k-2)$-deletions within $C_{p,j} \cup \mathcal{C}_{\text{sel}}(p,j)$ are one of the k sets: $C_{p,j} \setminus \{v_p(1,j,1), v_p(1,j,2)\}, C_{p,j} \setminus \{v_p(2,j,1), v_p(2,j,2)\}, \ldots,$ $C_{p,j} \setminus \{v_p(k,j,1), v_p(k,j,2)\}$.

Row selector gadget

In order to keep small balanced separators, our *k-row selector* gadget is quite different from the k-column selector. Its vertex set is a single row $R_{p,i}$ plus $\mathcal{O}(1)$ additional vertices $\mathcal{R}_{\text{sel}}(p,i)$. Furthermore *no* edge can be added within $R_{p,i}$. Again the open neighborhood of $\mathcal{R}_{\text{sel}}(p,i)$ has to be contained in $R_{p,i}$.

A problem Π *admits a row selector gadget* if, for every positive integer k, one can build in time $k^{\mathcal{O}(1)}$ a k-row selector such that, for every $j \neq j' \in [k]$, $\mathcal{R}_{\text{sel}}(p,i) \cup \{v_p(i,j,1), v_p(i,j,2),$ $v_p(i,j',1), v_p(i,j',2)\}$ is a Π-obstruction.

Edge gadget

The vertex set of an *edge gadget* is of the form $\{v_p(i,j,1), v_p(i,j,2), v_p(i',j',1), v_p(i',j',2)\} \cup \mathcal{E}_p(i,j,i',j')$ where $i \neq i' \in [k]$, $j \neq j' \in [k]$, and $\mathcal{E}_p(i,j,i',j')$ is a set of $\mathcal{O}(k)$ vertices[2]. There is no restriction on the edge set. As usual the open neighborhood of $\mathcal{E}_p(i,j,i',j')$ has to be contained in $\{v_p(i,j,1), v_p(i,j,2), v_p(i',j',1), v_p(i',j',2)\}$.

A problem Π *admits an edge gadget* if one can build in time $k^{\mathcal{O}(1)}$ an edge gadget such that $\mathcal{E}_p(i,j,i',j') \cup \{v_p(i,j,1), v_p(i,j,2), v_p(i',j',1), v_p(i',j',2)\}$ is a Π-obstruction.

Propagation gadget

The vertex set of a *propagation gadget* is of the form $H_p \cup H_{p+1} \cup \mathcal{P}_p$ where $\mathcal{P}_p$ is a set of $k^{\mathcal{O}(1)}$ vertices. There is a subset $\mathcal{P}'_p \subseteq \mathcal{P}_p$ of size $\mathcal{O}(k)$ such that each vertex of $\mathcal{P}_p \setminus \mathcal{P}'_p$ has at most one neighbor in $H_p \cup H_{p+1}$ and the rest of its neighborhood in $\mathcal{P}'_p$. This fairly technical condition aims to give some extra flexibility while keeping sufficiently small separators between H_p and H_{p+1}. In particular, if $\mathcal{P}_p$ is itself of size $\mathcal{O}(k)$, then the condition is trivially met with $\mathcal{P}'_p = \mathcal{P}_p$. The propagation gadget has no edge with both endpoints in $H_p \cup H_{p+1}$. Everything else is permitted, but the open neighborhood of $\mathcal{P}_p$ has to be contained in $H_p \cup H_{p+1}$.

A problem Π *admits a propagation gadget* if one can build in time $k^{\mathcal{O}(1)}$ a propagation gadget such that for every $i, j \neq j' \in [k]$, $\mathcal{P}_p \cup \{v_p(i,j,1), v_p(i,j,2), v_{p+1}(i,j',1), v_{p+1}(i,j',2)\}$ is a Π-obstruction.

Intended-solution property

A hereditary vertex-deletion problem Π and a description of the four above gadgets for Π have the *intended-solution property* if the following holds. On any graph G built by adding to the base $H_1 \cup \ldots \cup H_p \cup \ldots H_m$ at most one edge gadget in each H_p, one propagation gadget between *consecutive* pairs H_p and H_{p+1}, and some column and row selector gadgets, every deletion set $\bigcup_{p \in [m], i \in [k], j \in [k] \setminus \{j_i\}, z \in [2]} \{v_p(i,j,z)\}$ (with $\{j_1, j_2, \ldots, j_k\} = [k]$) intersecting every edge gadget is Π-legal.

We can now state the lower bound for the generic hereditary vertex-deletion problems.

---

[2] $\mathcal{O}(1)$ vertices will actually suffice for all the gadgets of Section 3.2.

▶ **Theorem 3.** *Unless the ETH fails, every vertex-deletion problem* Π *admitting a column selector, a row selector, an edge, and a propagation gadget, satisfying the intended-solution property, cannot be solved in time* $2^{o(\mathsf{pw}\log\mathsf{pw})}n^{\mathcal{O}(1)}$ *on* n*-vertex graphs with pathwidth* pw.

Proof. From any instance H of $k \times k$-Permutation Independent Set, we build an equivalent Π-instance $(G, k' = k^{\mathcal{O}(1)})$ of size $k^{\mathcal{O}(1)}$ with pathwidth in $\mathcal{O}(k)$. Since under the ETH there is no algorithm solving $k \times k$-Permutation Independent Set in time $2^{o(k\log k)}k^{\mathcal{O}(1)}$, we derive the claimed lower bound.

Construction. We number the edges in $E(H)$ as $e_1, \ldots, e_m$. We start with a base consisting of m copies of $H_\bullet$, labelled H_p for $p \in [m]$ (see description of the common base). The vertices $v_p(i,j,1)$ and $v_p(i,j,2)$ encode the vertex $(i,j) \in V(H)$; recall that we call such a pair *homologous*. We attach to each column $C_{p,j}$, for $p \in [m]$ and $j \in [k]$, a column selector gadget (for Π), with additional vertices $\mathcal{C}_{\text{sel}}(p,j)$. For each pair $p \in [m], i \in [k]$, we add a row selector gadget to $R_{p,i}$, with additional vertices $\mathcal{R}_{\text{sel}}(p,i)$.

For each edge $e_p = (i_p, j_p)(i'_p, j'_p) \in E(H)$ $(p \in [m])$, we attach an edge gadget, with additional vertices $\mathcal{E}_p(i_p, j_p, i'_p, j'_p)$, to $\{v_p(i_p,j_p,1), v_p(i_p,j_p,2), v_p(i'_p,j'_p,1), v_p(i'_p,j'_p,2)\}$. For each $p \in [m-1]$, we add a propagation gadget between H_p and H_{p+1}, with additional vertices $\mathcal{P}_p$. This finishes the construction of G. We set $k' := 2(k-1)km$.

Correctness. We first assume that there is a solution I to $k \times k$-Permutation Independent Set. That is, I is an independent set of H with exactly one vertex per column and per row. Say the vertices of I are $(1,j_1), (2,j_2), \ldots (k,j_k)$ with $\{j_1, j_2, \ldots, j_k\} = [k]$. Then

$$X := \bigcup_{p \in [m]} H_p \setminus \cup_{i \in [k]}\{v_p(i,j_i,1), v_p(i,j_i,2)\}$$

is a solution to Π. Indeed it is Π-legal since it intersects every edge gadget (if not, the edge gadget would be between two vertices of I, a contradiction) and Π satisfies the intended-solution property, by assumption. Furthermore $|X| = 2mk(k-1) = k'$.

We now assume that the Π-instance (G,k') admits a solution (of size k'), say X. The graph G has km disjoint Π-obstructions $C_{p,j} \cup \mathcal{C}_{\text{sel}}(p,j)$. For each of these sets, at least $s := 2(k-1)$ vertices must be deleted, by the specification of the column sector gadget. Since globally only $k' = kms$ vertices can be deleted, X intersects each $C_{p,j} \cup \mathcal{C}_{\text{sel}}(p,j)$ at a set $C_{p,j} \setminus \{v_p(i_{j,p},j,1), v_p(i_{j,p},j,2)\}$ for some $i_{j,p} \in [k]$. Moreover, the k row selector gadgets attached to each H_p enforce that $\{i_{1,p}, i_{2,p}, \ldots, i_{k,p}\} = [k]$, and the propagation gadget $\mathcal{P}_p$ enforces that $i_{j,p} = i_{j,p+1}$ for every $j \in [k]$. This implies that $i_{j,1} = i_{j,2} = \ldots = i_{j,m}$ for every $j \in [k]$, and we simply denote this common value by i_j. We claim that $\{(i_1,1), (i_2,2), \ldots, (i_k,k)\}$ is a solution to the $k \times k$-Permutation Independent Set instance. We have already argued that $\{i_1, i_2, \ldots, i_k\} = [k]$. Finally there cannot be an edge $e_p = (i_j,j)(i_{j'},j') \in E(H)$ since then the Π-obstruction $\mathcal{E}_p(i_j,j,i_{j'},j') \cup \{v_p(i_j,j,1), v_p(i_j,j,2), v_p(i_{j'},j',1), v_p(i_{j'},j',2)\}$ would be disjoint from X.

Pathwidth in $\mathcal{O}(k)$. Let $\mathcal{P}'_p$ be the $\mathcal{O}(k)$ vertices of $\mathcal{P}_p$ with strictly more than one neighbor in $H_p \cup H_{p+1}$. For every $p \in [m-1]$, we set $Y_p := \mathcal{P}'_p \cup \mathcal{E}_p(i_p,j_p,i'_p,j'_p) \cup C_{p,j_p} \cup \mathcal{C}_{\text{sel}}(p,j_p) \cup C_{p,j'_p} \cup \mathcal{C}_{\text{sel}}(p,j'_p) \cup \bigcup_{i \in [k]} \mathcal{R}_{\text{sel}}(p,i)$, and we observe that $|Y_p| = \mathcal{O}(k)$ (this is where it is important that each $\mathcal{R}_{\text{sel}}(p,i)$ has constant size). For each $p \in [m]$ and $j \in [k-2]$, let $Z_{p,j}$ be $C_{p,j^*} \cup C_{\text{sel}}(p,j^*)$ where j^* is the j-th index, by increasing value, in $[k] \setminus \{j_p, j'_p\}$. Again we notice that $|Z_{p,j}| = \mathcal{O}(k)$.

Here is a path-decomposition of G of width $\mathcal{O}(k)$ in case every $\mathcal{P}_p \setminus \mathcal{P}'_p$ is empty: $Y_1, Y_1 \cup Z_{1,1}, Y_1 \cup Z_{1,2}, \ldots, Y_1 \cup Z_{1,k-2}, Y_1 \cup Y_2, Y_1 \cup Y_2 \cup Z_{2,1}, Y_1 \cup Y_2 \cup Z_{2,2}, \ldots, Y_1 \cup Y_2 \cup Z_{2,k-2}, Y_2 \cup Y_3, \ldots, Y_{p-2} \cup Y_{p-1}, Y_{p-2} \cup Y_{p-1} \cup Z_{p-1,1}, Y_{p-2} \cup Y_{p-1} \cup Z_{p-1,2}, \ldots, Y_{p-2} \cup Y_{p-1} \cup$

$Z_{p-1,k-2}, Y_{p-1}, Y_{p-1} \cup Z_{p,1}, Y_{p-1} \cup Z_{p,2}, \ldots, Y_{p-1} \cup Z_{p,k-2}$. Indeed the maximum bag size is $\mathcal{O}(k)$ and each edge of G appears in at least one bag. Two crucial properties used in this path-decomposition are that (1) the removal of $\mathcal{P}'_p \cup \mathcal{P}'_{p+1}$, so in particular of $Y_p \cup Y_{p+1}$, disconnects H_{p+1} from the rest of G, and (2) there is no edge between $Z_{p,j}$ and $Z_{p,j'}$ for $j \neq j' \in [k-2]$ and $p \in [m]$.

In the general case, a path-decomposition of width $\mathcal{O}(k)$ for G is obtained from the previous decomposition by observing the following rule. Each time a vertex of H_p appears in a bag for the first time, we introduce and immediately remove each of its neighbors in $\mathcal{P}_p \setminus \mathcal{P}'_p$ one after the other. ◀

3.2 Designing ad hoc gadgets

We now build specific gadgets for Subset Feedback Vertex Set, Subset Odd Cycle Transversal, and Even Cycle Transversal. For these problems, we always use S to denote the prescribed subset of vertices through which no cycle, no odd cycle, or no even cycle should go, respectively.

3.2.1 Column selector gadgets

We begin with the column selector gadget $\mathcal{G}_1(\mathcal{C})$ used for Subset FVS and Subset OCT, followed by the gadget $\mathcal{G}_2(\mathcal{C})$ used for ECT. The column selector gadget $\mathcal{G}_1(\mathcal{C})$ attached to a column $C_{p,j}$ is defined as follows. It comprises $3k$ additional vertices. These $3k$ vertices are all added to S, and they form an independent set. Each of the first k vertices, $d_{p,j}(1,1), \ldots, d_{p,j}(k,1) \in S$, are adjacent to all vertices in $\bigcup_{i \in [k]} \{v_p(i,j,1)\}$, so these vertices induce a biclique. The next k vertices, $d_{p,j}(1,2), \ldots, d_{p,j}(k,2) \in S$, also twins, are adjacent to all vertices in $\bigcup_{i \in [k]} \{v_p(i,j,2)\}$. We add $d_{p,j}(1), \ldots, d_{p,j}(i), \ldots, d_{p,j}(k)$ and, for each $i \in [k]$, we link $d_{p,j}(i)$ to all the vertices in $\{v_p(i,j,1)\} \cup \bigcup_{i' \in [k] \setminus \{i\}} \{v_p(i',j,2)\}$. Finally we make every distinct pair $v_p(i,j,z), v_p(i',j,z')$ adjacent, except if $i = i'$.

We obtain the column selector gadget $\mathcal{G}_2(\mathcal{C})$ from $\mathcal{G}_1(\mathcal{C})$ by adding, for each $z \in [2]$, a vertex $d_{p,j}(k+1,z)$ adjacent to all vertices in $\bigcup_{i \in [k]} \{v_p(i,j,z)\}$, and by subdividing each edge $d_{p,j}(i)v_p(i,j,1)$ once.

▶ **Lemma 4.** *$\mathcal{G}_1(\mathcal{C})$ is a column selector gadget for Subset Feedback Vertex Set and Subset Odd Cycle Transversal, and $\mathcal{G}_2(\mathcal{C})$ is a column selector gadget for Even Cycle Transversal.*

Proof. The gadgets $\mathcal{G}_1(\mathcal{C})$ and $\mathcal{G}_2(\mathcal{C})$ add $3k$ and $4k+2$, respectively, new vertices, thus $\mathcal{O}(k)$. Their edge set respects the specification of the column selector.

We first show that the only Π-legal $(2k-2)$-deletions within $\mathcal{G}_1(\mathcal{C})$ are the sets $C_{p,j} \setminus \{v_p(i,j,1), v_p(i,j,2)\}$ (for $i \in [k]$), for $\Pi \in \{$Subset FVS, Subset OCT$\}$. For every $p \in [m]$, $j \in [k]$, and $z \in [2]$, the biclique $K_{k,k}$ between $\bigcup_{i \in [k]} \{v_p(i,j,z)\}$ and $\bigcup_{i \in [k]} \{d_{p,j}(i,z)\} \subseteq S$ forces the removal of all but at most one vertex of $\bigcup_{i \in [k]} \{v_p(i,j,z)\}$, or all the vertices in $\bigcup_{i \in [k]} \{d_{p,j}(i,z)\}$. Indeed, recall that the former set is a clique, while the latter set is an independent set and is contained in the prescribed set S. Hence keeping at least one vertex in $\bigcup_{i \in [k]} \{d_{p,j}(i,z)\}$ and at least two in $\bigcup_{i \in [k]} \{v_p(i,j,z)\}$ results in an odd cycle (a triangle) going through at least one vertex of S. Thus the only Π-legal $(2k-2)$-deletions within $\mathcal{G}_1(\mathcal{C})$ have to remove exactly $k-1$ vertices in $\bigcup_{i \in [k]} \{v_p(i,j,1)\}$ and exactly $k-1$ vertices in $\bigcup_{i \in [k]} \{v_p(i,j,2)\}$. Let Y denote such a deletion set, and observe that $Y \cap S = \emptyset$. We further claim that if $v_p(i,j,1)$ is not in Y, then $v_p(i,j,2)$ is also not in Y. Assume, for the sake of contradiction, that $v_p(i,j,1)$ and $v_p(i',j,2)$ are two (adjacent) vertices, not in Y, with $i \neq i'$. Then $d_{p,j}(i) \in S$ forms a surviving triangle with $v_p(i,j,1)$ and $v_p(i',j,2)$. Thus $Y = C_{p,j} \setminus \{v_p(i,j,1), v_p(i,j,2)\}$ for some $i \in [k]$.

This finishes the proof that $\mathcal{G}_1(\mathcal{C})$ is a column selector gadget for Subset FVS and Subset OCT. We now adapt the arguments for $\mathcal{G}_2(\mathcal{C})$ and Π = ECT. Now the biclique $K_{k,k+1}$ between $\bigcup_{i \in [k]} \{v_p(i,j,z)\}$ and $\bigcup_{i \in [k+1]} \{d_{p,j}(i,z)\} \subseteq S$ forces the removal of all but at most one vertex of $\bigcup_{i \in [k]} \{v_p(i,j,z)\}$, or all but at most one vertex of $\bigcup_{i \in [k+1]} \{d_{p,j}(i,z)\}$, otherwise there would be a surviving even cycle C_4. Since only $k-1$ vertices can be removed from each Π-obstruction $\bigcup_{i \in [k]} \{v_p(i,j,z)\} \cup \bigcup_{i \in [k+1]} \{d_{p,j}(i,z)\} \subseteq S$ (with $z \in [2]$), the only Π-legal $(2k-2)$-deletions within $\mathcal{G}_2(\mathcal{C})$ remove all but one vertex in $\bigcup_{i \in [k]} \{v_p(i,j,1)\}$ and in $\bigcup_{i \in [k]} \{v_p(i,j,2)\}$. The end of the proof is similar to the previous paragraph since the triangle $d_{p,j}(i)v_p(i,j,1)v_p(i',j,2)$ is now a C_4 (recall that we subdivided the edge $d_{p,j}(i)v_p(i,j,1)$ once). ◀

3.2.2 Row selector gadgets

The row selector $\mathcal{G}_1(\mathcal{R})$, attached to $R_{p,i}$, consists of two additional vertices $r_1(p,i), r'_1(p,i) \in S$ made adjacent to every vertex in $\bigcup_{j \in [k]} \{v_p(i,j,1)\}$. The row selector $\mathcal{G}_2(\mathcal{R})$ consists of three additional vertices $r_2(p,i), r'_2(p,i), r''_2(p,i)$, each adjacent to all vertices in $\bigcup_{j \in [k]} \{v_p(i,j,1)\}$. We put only $r'_2(p,i)$ in S, and we add an edge between $r_2(p,i)$ and $r''_2(p,i)$.

▶ **Lemma 5** (⋆). *$\mathcal{G}_1(\mathcal{R})$ is a row selector gadget for Subset Feedback Vertex Set and Even Cycle Transversal, and $\mathcal{G}_2(\mathcal{R})$ is a row selector gadget for Subset Odd Cycle Transversal.*

Crucially for the intended-solution property, the odd cycle $r_2(p,i)v_p(i,j,1)r''_2(p,i)$ does not contain any vertex of S.

3.2.3 Edge gadgets

Let $\mathcal{G}_1(\mathcal{E})$ be the following edge gadget, that we present for $e_p = (i,j)(i',j')$. We add an edge between $v_p(i,j,1)$ and $v_p(i',j',1)$. We add a vertex s_p adjacent to both $v_p(i,j,1)$ and $v_p(i',j',1)$. We add s_p to the set $S \subseteq V(G)$. The edge gadget $\mathcal{G}_2(\mathcal{E})$ is obtained from $\mathcal{G}_1(\mathcal{E})$ by subdividing the edge $s_p v_p(i',j',1)$ once.

▶ **Lemma 6** (⋆). *$\mathcal{G}_1(\mathcal{E})$ is an edge gadget for Subset Feedback Vertex Set and Subset Odd Cycle Transversal, and $\mathcal{G}_2(\mathcal{E})$ is an edge gadget for Even Cycle Transversal.*

3.2.4 Propagation gadgets

We present $\mathcal{G}_1(\mathcal{P})$, a propagation gadget inserted between H_p and H_{p+1}. We first add an independent set of $2k$ vertices. Among them, the k vertices $r_{p,1}, \dots, r_{p,k}$ represent the row indices in H_p and H_{p+1}, while the k other vertices $c_{p,1}, \dots, c_{p,k}$ represent the column indices. We link $r_{p,i}$ to all the vertices in $\bigcup_{j \in [k]} \{v_p(i,j,2)\} \cup \bigcup_{j \in [k]} \{v_{p+1}(i,j,1)\}$. Similarly, we link $c_{p,j}$ to all the vertices in $\bigcup_{i \in [k]} \{v_p(i,j,2)\} \cup \bigcup_{i \in [k]} \{v_{p+1}(i,j,1)\}$. Finally, we add a vertex $c_p \in S$ adjacent to all the vertices $c_{p,1}, \dots, c_{p,k}$.

The gadget $\mathcal{G}_2(\mathcal{P})$ is defined similarly, except that we subdivide the edge $r_{p,i}v_p(i,j,2)$ once, for each $i,j \in [k]$. Finally the gadget $\mathcal{G}_3(\mathcal{P})$ adds to $\mathcal{G}_2(\mathcal{P})$, a vertex $c'_{p,j}$, for each $j \in [k]$. The vertex $c'_{p,j}$ is linked to $c_{p,j}$ and to c_p.

▶ **Lemma 7** (⋆). *$\mathcal{G}_1(\mathcal{P})$ is a column selector gadget for* Subset Feedback Vertex Set*, $\mathcal{G}_2(\mathcal{P})$ is a column selector gadget for* Subset Odd Cycle Transversal*, and $\mathcal{G}_3(\mathcal{P})$ is a column selector gadget for* Even Cycle Transversal*.*

3.2.5 Wrap-up

We can now use the above gadgets to establish the following.

▶ **Theorem 8.** *Unless the ETH fails, the following problems cannot be solved in time $2^{o(\mathsf{pw} \log \mathsf{pw})} n^{\mathcal{O}(1)}$ on n-vertex graphs with pathwidth pw:*

- *Subset Feedback Vertex Set,*
- *Subset Odd Cycle Transversal, and*
- *Even Cycle Transversal.*

Proof. We need to check that these problems satisfy the preconditions of Theorem 3. Sections 3.2.1 to 3.2.4 and Lemmas 4 to 7 show how to build the four types of gadgets. Which problem uses which version of the gadget is summarized in Table 1. See Figure 1 for a schematic representation of the construction for Subset FVS.

Table 1 The different gadgets used for the different problems.

	column selector	row selector	edge gadget	propagation gadget
Subset FVS	$\mathcal{G}_1(\mathcal{C})$	$\mathcal{G}_1(\mathcal{R})$	$\mathcal{G}_1(\mathcal{E})$	$\mathcal{G}_1(\mathcal{P})$
Subset OCT	$\mathcal{G}_1(\mathcal{C})$	$\mathcal{G}_2(\mathcal{R})$	$\mathcal{G}_1(\mathcal{E})$	$\mathcal{G}_2(\mathcal{P})$
ECT	$\mathcal{G}_2(\mathcal{C})$	$\mathcal{G}_1(\mathcal{R})$	$\mathcal{G}_2(\mathcal{E})$	$\mathcal{G}_3(\mathcal{P})$

Finally we have to check that the problems have the intended-solution property. We shall prove that every set $X := \bigcup_{p \in [m], i \in [k], z \in [2]} \{v_p(i, j_i, z)\}$, with $\{j_1, \ldots, j_k\} = [k]$ and intersecting all the edge gadgets is Π-legal in any graph G obtained by attaching to the base the four types of gadgets with respect to their specification of Section 3.1. The set X is a solution to $\Pi \in \{$Subset FVS, Subset OCT, ECT$\}$, if and only if no 2-connected component (i.e., a block of size at least 3) of $G - X$ is a Π-obstruction. Indeed no cycle can go through a cut-vertex.

We first note that there is no 2-connected component within $\mathcal{G}_1(\mathcal{C})$, $\mathcal{G}_2(\mathcal{C})$, $\mathcal{G}_1(\mathcal{R})$, $\mathcal{G}_1(\mathcal{E})$, $\mathcal{G}_2(\mathcal{E})$ restricted to $G - X$. For the latter two gadgets, this is because, by assumption, X intersects every edge gadget. In a gadget $\mathcal{G}_2(\mathcal{R})$ restricted to $G - X$, there is one 2-connected component, namely a triangle; but none of its vertices belongs to S.

We now observe that every vertex c_p is a cut-vertex in $\mathcal{G}_1(\mathcal{P})$, $\mathcal{G}_2(\mathcal{P})$, and $\mathcal{G}_3(\mathcal{P})$ restricted to $G - X$. So the remaining 2-connected components of $G - X$ are induced cycles C_4 of the form $r_{p,i} v_p(i, j, 2) c_{p,j} v_{p+1}(i, j, 1)$ when $\mathcal{G}_1(\mathcal{P})$ is used, or induced C_5 when $\mathcal{G}_2(\mathcal{P})$ is used, or triangle and induced cycle C_5 when $\mathcal{G}_3(\mathcal{P})$ is used. In the first two cases, none of the vertices of the cycles belongs to S. In the third case, no cycle is even. This establishes that Subset FVS, Subset OCT, and ECT with their respective combination of gadgets have the intended-solution property. ◀

3.3 Lower bound for Node Multiway cut

For Node Multiway Cut we will also start from the base $\bigcup_{p \in [m]} H_p$ but we will deviate from the gadget specification of Section 3.1. We will "communalize" the selector, edge, and propagation gadgets. That way, we are able to show the claimed lower bound even when the

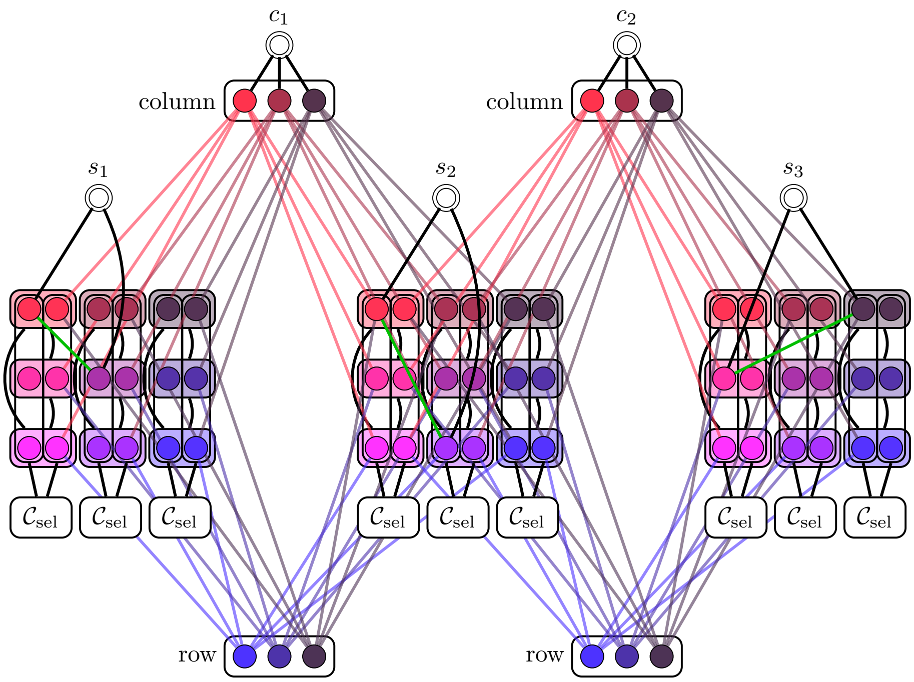

Figure 1 Example of the overall picture for Subset Feedback Vertex Set. The first three edges (in green) in the reduction from $k \times k$-Permutation Independent Set, with $k = 3$, to Subset FVS. The doubly-circled vertices are vertices in S. The column selector gadget $\mathcal{C}_{\text{sel}}$, of size $\mathcal{O}(k)$, forces that only one pair of homologous vertices is retained in each column. We did *not* represent the row selector gadget.

number of terminals is linearly tied to the pathwidth. This is unlike our constructions for Subset FVS and Subset OCT in Theorem 8 where the size of the prescribed subsets S is significantly larger than the pathwidth.

▶ **Theorem 9** (⋆). *Unless the ETH fails, Node Multiway Cut cannot be solved in time $2^{o(p \log p)} n^{\mathcal{O}(1)}$ on n-vertex graphs where $p = \mathsf{pw} + |T|$ is the sum of the pathwidth of the input graph and the number of terminals.*

3.4 Lower bound for Multiway Cut

To obtain the lower bound for Multiway Cut, we reduce from $k \times k$-Permutation Clique.

▶ **Theorem 10** (⋆). *Unless the ETH fails, Multiway Cut cannot be solved in time $2^{o(p \log p)} n^{\mathcal{O}(1)}$ on n-vertex graphs where $p = \mathsf{pw} + |T|$ is the sum of the pathwidth of the input graph and the number of terminals.*

By a simple reduction from Multiway Cut to Restricted Edge-Subset Feedback Edge Set, we obtain the following as a corollary.

▶ **Theorem 11.** *Unless the ETH fails, Restricted Edge-Subset Feedback Edge Set cannot be solved in time $2^{o(p \log p)} n^{\mathcal{O}(1)}$ on n-vertex graphs where $p = \mathsf{pw} + |S|$ is the sum of the pathwidth of the input graph and the number of undeletable (terminal) edges.*

It is not difficult to adapt the construction of Theorem 10 for the directed variant of Multiway Cut.

▶ **Theorem 12.** *Unless the ETH fails,* Directed Multiway Cut *cannot be solved in time* $2^{o(\mathsf{pw}\log\mathsf{pw})}n^{\mathcal{O}(1)}$ *on* n*-vertex directed graphs whose underlying undirected graph has pathwidth* pw*.*

4 Slightly superexponential algorithms

In this section, we present $2^{\mathcal{O}(\mathsf{tw}\log\mathsf{tw})}n^3$-time algorithms for the weighted variants of the considered problems with the exception of ECT. We first present in Theorem 15 a $2^{\mathcal{O}(\mathsf{tw}\log\mathsf{tw})}n^3$-time algorithm for Subset OCT. Then, we show that with simple modifications this algorithm can solve Subset FVS. We deduce the algorithms for the other problems by reducing these problems to the weighted variant of Subset FVS.

Let us focus on the Subset OCT problem. For a graph G and a vertex set S of G, we say that G is S*-bipartite* if it has no odd cycle containing a vertex of S. Solving Subset OCT is equivalent to find an S-bipartite induced subgraph of maximum size. The following characterization of S-bipartite graphs will be useful.

▶ **Lemma 13** ($\star$)**.** *A graph* G *is* S*-bipartite if and only if for every block* B *of* G*, either* B *has no vertex of* S*, or it is bipartite.*

One can easily modify the proof of one direction of Lemma 13 to prove the following fact.

▶ **Fact 14.** *If a graph* G *is 2-connected and not bipartite, then there exists an odd path and an even path between every pair of vertices.*

A tree decomposition $(T, \mathcal{B} = \{B_t\}_{t\in V(T)})$ is a *nice tree decomposition* with root node $r \in V(T)$ if T is a rooted tree with root node r, and every node t of T is one of the following:

1. a *leaf node*: t is a leaf of T and $B_t = \emptyset$;
2. an *introduce node*: t has exactly one child t' and $B_t = B_{t'} \cup \{v\}$ for some $v \in V(G) \setminus B_{t'}$;
3. a *forget node*: t has exactly one child t' and $B_t = B_{t'} \setminus \{v\}$ for some $v \in B_{t'}$; or
4. a *join node*: t has exactly two children t_1 and t_2, and $B_t = B_{t_1} = B_{t_2}$.

▶ **Theorem 15.** *(Weighted) Subset Odd Cycle Transversal can be solved in time* $2^{\mathcal{O}(\mathsf{tw}\log\mathsf{tw})}n^3$ *on* n*-vertex graphs with treewidth* tw*.*

Proof. In the following, we fix a graph G, $S \subseteq V(G)$, and a weight function $w : V(G) \to \mathbb{R}$. Using Bodlaender et al.'s fpt approximation algorithm [4] and an algorithm of constructing a nice tree-decomposition (folklore; see Lemma 7.4 in [10]), we can obtain a nice tree decomposition of G of width at most $5\mathsf{tw} + 4$ in time $\mathcal{O}(c^{\mathsf{tw}} \cdot n)$ for some constant c. Let $(T, \{B_t\}_{t\in V(T)})$ be the resulting nice tree decomposition. For each node t of T, let G_t be the subgraph of G induced by the union of all bags $B_{t'}$ where t' is a descendant of t.

Let t be a node of T. A *partial solution of* G_t is a subset $X \subseteq V(G_t)$ such that $G[X]$ is S-bipartite. We are going to introduce an equivalence relation $\equiv_t$ between partial solutions in order to obtain the property that if $X \equiv_t Y$, then for every $W \subseteq V(\overline{G_t})$, $G[X \cup W]$ is S-bipartite if and only if $G[Y \cup W]$ is S-bipartite.

Let $X \subseteq V(G)$ (not necessarily contained in G_t). We denote by $\mathrm{Inc}(X)$ the block-cut tree of $G[X]$, that is the bipartite graph whose vertices are the blocks and the cut vertices of $G[X]$ and where a block B is adjacent to a cut vertex v if $v \in V(B)$. Observe that $\mathrm{Inc}(X)$ is by definition a forest.

We say that a vertex v of $\mathrm{Inc}(X)$ is *active* (with respect to t) if:

- v is a cut vertex of $G[X]$ in B_t,
- v is a block of $G[X]$ that contains at least two vertices in B_t, or
- v is a block of $G[X]$ that contains exactly one vertex in B_t that is not a cut vertex.

Note that every vertex in B_t is an active cut vertex or it is in an active block of $G[X]$.

We construct the auxiliary graphs $\text{Aux}_p(X,t)$ and $\text{Aux}(X,t)$ from $\text{Inc}(X)$ as follows:

1. We remove recursively the leaves and the isolated vertices that are inactive. Let $\text{Aux}_p(X,t)$ be the resulting graph (p for "prototype").
2. For every maximal path P of $\text{Aux}_p(X,t)$ between u and v and with inactive internal vertices of degree 2, we remove the internal vertices of P and we add an edge between u and v (shrinking degree 2 nodes that are inactive).

Figure 2 illustrates the constructions of $\text{Aux}_p(X,t)$ and $\text{Aux}(X,t)$. Observe that Operation 1 removes the inactive blocks of $G[X]$ that contain one vertex in B_t. Thus, every block in $\text{Aux}_p(X,t)$ that contains vertices in B_t is active. By construction, $\text{Aux}(X,t)$ is a forest whose vertices are the active vertices of $\text{Inc}(X)$ and the inactive vertices that have degree at least 3 in $\text{Aux}_p(X,t)$. Importantly, the algorithm uses the graphs $\text{Aux}(X,t)$ for $X \subseteq V(G_t)$ and in the proof we will use $\text{Aux}(X,t)$ and $\text{Aux}_p(X,t)$ for $X \subseteq V(G_t)$ or $X \subseteq B_t \cup V(\overline{G_t})$.

By Step 2, any edge uv of $\text{Aux}(X,t)$ corresponds to an alternating sequence P of cut vertices and blocks $A_1, A_2, \dots, A_x$ that forms a path from $u = A_1$ to $v = A_x$ in $\text{Inc}(X)$. We define the graph M_{uv} as the union of the blocks in P. Note that one of A_1 and A_2 is a cut vertex and one of A_{x-1} and A_x is a cut vertex. We say that these cut vertices are the endpoints of M_{uv}.

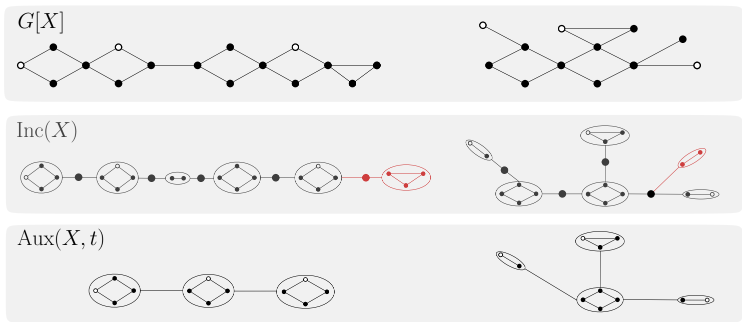

Figure 2 Example of graphs $\text{Inc}(X)$ and $\text{Aux}(X,t)$ constructed from a graph $G[X]$. The vertices in B_t are white filled. The red vertices and edges in $\text{Inc}(X)$ are those we remove to obtain $\text{Aux}_p(X,t)$.

Let X and Y be two partial solutions of G_t. We say that $X \equiv_t Y$ if $X \cap B_t = Y \cap B_t$, and there is an isomorphism φ from $\text{Aux}(X,t)$ to $\text{Aux}(Y,t)$ such that:

1. For every vertex v in $\text{Aux}(X,t)$, v is active if and only if $\varphi(v)$ is active.
2. For every vertex v in $\text{Aux}(X,t)$, v is a block if and only if $\varphi(v)$ is a block.
3. For every active cut vertex v in $\text{Aux}(X,t)$, we have $\varphi(v) = v$.
4. For every active block B in $\text{Aux}(X,t)$:
   a. $V(B) \cap B_t = V(\varphi(B)) \cap B_t$,
   b. $V(B) \cap S \neq \emptyset$ if and only if $V(\varphi(B)) \cap S \neq \emptyset$, and
   c. B is bipartite if and only if $\varphi(B)$ is bipartite.
5. For every edge uv in $\text{Aux}(X,t)$:
   a. M_{uv} is bipartite if and only if $M_{\varphi(u)\varphi(v)}$ is bipartite, and
   b. $V(M_{uv}) \cap S \neq \emptyset$ if and only if $V(M_{\varphi(u)\varphi(v)}) \cap S \neq \emptyset$.
6. For every pair (u,v) of vertices in $B_t \cap X$ and every path P_X between u and v in $G[X]$, there exists a path P_Y in $G[Y]$ between u and v with the same parity as P_X.

▷ Claim 16 ($\star$). For every node t of T, $\equiv_t$ has $2^{\mathcal{O}(\mathsf{tw}\log\mathsf{tw})}$ equivalence classes.

▷ Claim 17 (⋆). Let t be a node of T and X, Y be two partial solutions associated with t. If $X \equiv_t Y$, then, for every $Z \subseteq V(\overline{G_t})$, the graph $G[X \cup Z]$ is S-bipartite if and only if $G[Y \cup Z]$ is S-bipartite.

We are now ready to describe our algorithm. For each node t of T and $I \subseteq B_t$, let $\mathcal{P}[t, I]$ be the set of all partial solutions X of G_t where $X \cap B_t = I$. A reduced set $\mathcal{R}[t, I]$ is a subset of $\mathcal{P}[t, I]$ satisfying that

- for every partial solution $X \in \mathcal{P}[t, I]$, there exists $X' \in \mathcal{R}[t, I]$ where $X \equiv_t X'$ and $w(X') \geq w(X)$, and
- no two partial solutions in $\mathcal{R}[t, I]$ are equivalent.

We will recursively compute a reduced set $\mathcal{R}[t, I]$ for every node t of T and $I \subseteq B_t$. Claim 16 guarantees that $|\bigcup_{I \subseteq B_t} \mathcal{R}[t, I]| = 2^{\mathcal{O}(\mathsf{tw} \log \mathsf{tw})}$.

We describe how to compute a reduced set $\mathcal{R}[t, I]$ depending on the type of the node t. We fix a node t and $I \subseteq B_t$. For each leaf node t and $I = \emptyset$, we assign $\mathcal{R}[t, I] := \emptyset$. For $\mathcal{A} \subseteq 2^{V(G_t)}$, we define $\mathsf{reduce}_t(\mathcal{A})$ as the operation which removes the elements of $\mathcal{A}$ that does not induce S-bipartite graph and then returns a set that contains, for each equivalence class $\mathcal{C}$ of $\equiv_t$ over $\mathcal{A}$, a partial solution of $\mathcal{C}$ of maximum weight.

1) t is an introduce node with child t' and $B_t \setminus B_{t'} = \{v\}$:

If $v \notin I$, then it is easy to see that $\mathcal{R}[t', I]$ is a reduced set of $\mathcal{P}[t, I] = \mathcal{P}[t', I]$. In this case, we take $\mathcal{R}[t, I] = \mathcal{R}[t', I]$. Assume now that $v \in I$. We set $\mathcal{R}[t, I] = \mathsf{reduce}_t(\mathcal{A})$ with $\mathcal{A}$ the set that contains $X \cup \{v\}$ for every $X \in \mathcal{R}[t', I \setminus \{v\}]$.

2) t is a forget node with child t' and $B_{t'} \setminus B_t = \{v\}$:

We simply set $\mathcal{R}[t, I] = \mathsf{reduce}_t(\mathcal{R}[t', I] \cup \mathcal{R}[t', I \cup \{v\}])$.

3) t is a join node with two children t_1 and t_2:

We set $\mathcal{R}[t, I] = \mathsf{reduce}_t(\mathcal{A})$ where $\mathcal{A}$ is the set that contains $X_1 \cup X_2$ for every $X_1 \in \mathcal{R}[t_1, I]$ and $X_2 \in \mathcal{R}[t_2, I]$.

We defer the proof of the correctness and the runtime to the long version. ◀

▶ **Theorem 18** (⋆). *Subset Feedback Vertex Set, Restricted Edge-Subset Feedback Edge Set, and Node Multiway Cut, and their weighted variants can be solved in time $2^{\mathcal{O}(\mathsf{tw} \log \mathsf{tw})} n^3$ on n-vertex graph with treewidth* tw.

References

1 Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Hitting minors on bounded treewidth graphs. *CoRR*, abs/1704.07284, 2017. arXiv:1704.07284.

2 Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. A complexity dichotomy for hitting connected minors on bounded treewidth graphs: the chair and the banner draw the boundary. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 951–970. SIAM, 2020. doi:10.1137/1.9781611975994.57.

3 Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Inf. Comput.*, 243:86–111, 2015. doi:10.1016/j.ic.2014.12.008.

4 Hans L. Bodlaender, Pål Grønås Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lokshtanov, and Michal Pilipczuk. A $c^k n$ 5-approximation algorithm for treewidth. *SIAM J. Comput.*, 45(2):317–378, 2016. doi:10.1137/130947374.

5 Marthe Bonamy, Lukasz Kowalik, Jesper Nederlof, Michal Pilipczuk, Arkadiusz Socala, and Marcin Wrochna. On directed feedback vertex set parameterized by treewidth. In *Graph-Theoretic Concepts in Computer Science - 44th International Workshop, WG 2018, Cottbus, Germany, June 27-29, 2018, Proceedings*, pages 65–78, 2018. doi:10.1007/978-3-030-00256-5_6.

6 Édouard Bonnet, Nick Brettell, O-joung Kwon, and Dániel Marx. Generalized feedback vertex set problems on bounded-treewidth graphs: chordality is the key to single-exponential parameterized algorithms. *Algorithmica*, 81(10):3890–3935, 2019. doi:10.1007/s00453-019-00579-4.

7 Hajo Broersma, Petr A. Golovach, and Viresh Patel. Tight complexity bounds for FPT subgraph problems parameterized by the clique-width. *Theor. Comput. Sci.*, 485:69–84, 2013. doi:10.1016/j.tcs.2013.03.008.

8 Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

9 Marek Cygan, Fedor V. Fomin, Alexander Golovnev, Alexander S. Kulikov, Ivan Mihajlin, Jakub Pachocki, and Arkadiusz Socala. Tight lower bounds on graph embedding problems. *J. ACM*, 64(3):18:1–18:22, 2017. doi:10.1145/3051094.

10 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

11 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 150–159, 2011. doi:10.1109/FOCS.2011.23.

12 Xiaojie Deng, Bingkai Lin, and Chihao Zhang. Multi-multiway cut problem on graphs of bounded branch width. In *Frontiers in Algorithmics* and *Algorithmic Aspects in Information and Management, Third Joint International Conference, FAW-AAIM 2013, Dalian, China, June 26-28, 2013. Proceedings*, pages 315–324, 2013. doi:10.1007/978-3-642-38756-2_32.

13 Pål Grønås Drange, Markus S. Dregi, and Pim van 't Hof. On the computational complexity of vertex integrity and component order connectivity. *Algorithmica*, 76(4):1181–1202, 2016. doi:10.1007/s00453-016-0127-x.

14 Samuel Fiorini, Nadia Hardy, Bruce A. Reed, and Adrian Vetta. Planar graph bipartization in linear time. *Discret. Appl. Math.*, 156(7):1175–1180, 2008. doi:10.1016/j.dam.2007.08.013.

15 Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18(1):3–20, 1997. doi:10.1007/BF02523685.

16 Russell Impagliazzo and Ramamohan Paturi. On the Complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

17 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Slightly superexponential parameterized problems. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 760–776, 2011. doi:10.1137/1.9781611973082.60.

18 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Slightly superexponential parameterized problems. *SIAM J. Comput.*, 47(3):675–702, 2018. doi:10.1137/16M1104834.

19 Michal Pilipczuk. Problems parameterized by treewidth tractable in single exponential time: A logical approach. In *Mathematical Foundations of Computer Science 2011 - 36th International Symposium, MFCS 2011, Warsaw, Poland, August 22-26, 2011. Proceedings*, pages 520–531, 2011. doi:10.1007/978-3-642-22993-0_47.

20 Ignasi Sau and Uéverton S. Souza. Hitting forbidden induced subgraphs on bounded treewidth graphs. *CoRR*, abs/2004.08324, 2020. arXiv:2004.08324.

21 Mingyu Xiao and Hiroshi Nagamochi. An FPT algorithm for edge subset feedback edge set. *Inf. Process. Lett.*, 112(1-2):5–9, 2012. doi:10.1016/j.ipl.2011.10.007.