Close relatives of Feedback Vertex Set without single-exponential algorithms parameterized by treewidth

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

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

Nick Brettell
Department of Computer Science, Durham University, Durham, UK
nbrettell@gmail.com

O-joung Kwon
Department of Mathematics, Incheon National University, Incheon, 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 (ETH), 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 under the ETH.

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

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

Many NP-hard graph problems admit polynomial-time algorithms on graphs with bounded *treewidth*, a measure of how well a graph accommodates a decomposition into a tree-like structure. In fact, Courcelle's Theorem [9] states that any problem definable in MSO_2 logic can be solved in linear time on graphs of bounded treewidth. To obtain a more fine-grained perspective on the dependence on treewidth for certain problems, it is useful to study the parameterized complexity with respect to treewidth. In particular, we can ask: what is 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. 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. The same year, Pilipczuk [25] exhibited a logic fragment whose model checking admits a single-exponential algorithm parameterized by the treewidth of the input graph, thereby providing a scaled-down but more fine-grained version of Courcelle's theorem. Moreover, also in 2011, Lokshtanov et al. [21] developed a framework yielding $2^{\Omega(\mathsf{tw}\log\mathsf{tw})}n^{\mathcal{O}(1)}$-time lower bounds under the Exponential Time Hypothesis (ETH). Let us 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 [17]. 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 are those consisting 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.

So-called *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 [25] 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. [10] 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 [26] 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 given proper minor-closed class (given by the 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. 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 those that intersect 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 containing a vertex in S?

Both of these problems are NP-complete. By setting $S = V(G)$, one sees that the latter problem generalizes Odd Cycle Transversal, for which Fiorini et al. [15] 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 note that parameterizations by solution size have been studied for these three problems [12, 19, 23, 24, 27].

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 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 need only hit cycles containing a vertex or an edge (respectively) of a given set S, can also be solved in linear time [28]. 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, respectively (see the remark in Section 1.1). They are well-established 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 [16]. Therefore the 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. [13] 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 solution is a set of vertices, or on the edges when the solution is 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. [22] (see for instance, [2, 5–7, 14]). 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.

A remark on parameter-preserving reductions between the problems. There is an easy reduction from Node Multiway Cut to Weighted Subset Feedback Vertex Set (WSFVS, for short). It consists of adding a vertex v of "infinite" weight adjacent to all the terminals of the Multiway Cut instance, which also all get "infinite" weight. The set S of the WSFVS instance is $\{v\}$. The same process yields a reduction from Multiway Cut to Restricted Edge-Subset Feedback Edge Set, where now the set S of the Restricted

[1] Observe that we switch the columns and the rows compared to the original definition of $k \times k$-Clique [22]. While this is of course equivalent, it will make the representation of some gadgets slightly more conducive to the page format.

Edge-Subset Feedback Edge Set instance contains all the edges incident to v (recall that these edges are thus undeletable).

From the latter reduction, we can immediately derive the lower bound for Restricted Edge-Subset Feedback Edge Set from the lower bound for Multiway Cut (see Theorem 13). However, the hardness result for Node Multiway Cut (see Theorem 11) does not imply anything for Subset Feedback Vertex Set. Indeed, to encode the "infinite" weight that makes v and its neighbors undeletable, one would have to duplicate these vertices many times. This would result in a large biclique, or at the very least a large biclique minor, and would thereby make the pathwidth or treewidth large. Therefore Theorem 10 is necessary and cannot be obtained by a simple modification of Theorem 11. Finally, we observe that the straightforward reduction from Multiway Cut to Node Multiway Cut requires vertex weights, or blows up the treewidth. So again Theorem 11 cannot be derived from Theorem 12.

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, this algorithm 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.

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 operations.

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 as explained in the previous subsection: 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 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.

1.2 Organization of the paper

The rest of the paper is organized as follows. In Section 2 we give the required graph-theoretic definitions and notation. 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.

2 Preliminaries

We assume all graphs have no loops or parallel edges. Let G be a graph. We denote the vertex set and the edge set of G by $V(G)$ and $E(G)$, respectively. For a vertex v in G, we use $G - v$ to denote the *deletion* of v from G, that is, the graph obtained by removing v and its incident edges. For $X \subseteq V(G)$, we denote by $G - X$ the graph obtained by removing all vertices in X and their incident edges. For $X \subseteq V(G)$, we denote by $G[X]$ the subgraph induced by the vertex set X. A subgraph H of G is an *induced subgraph* of G if $H = G[X]$ for some vertex subset X of G. For two graphs G_1 and G_2, $G_1 \cup G_2$ is the graph with the vertex set $V(G_1) \cup V(G_2)$ and the edge set $E(G_1) \cup E(G_2)$, and $G_1 \cap G_2$ is the graph with the vertex set $V(G_1) \cap V(G_2)$ and the edge set $E(G_1) \cap E(G_2)$. 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 a vertex v in G, we denote by $N_G(v)$ the set of neighbors of v in G, and $N_G[v] := N_G(v) \cup \{v\}$. For $X \subseteq V(G)$, we let $N_G(X) := (\bigcup_{v \in X} N_G(v)) \setminus X$, and say $N_G(X)$ is the *(open) neighborhood* of X. For $u, v \in V(G)$, we say that u and v are *twins* if $N(u) = N(v)$. If $N[u] = N[v]$, then we also 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.

Let G be a graph. A *walk* in G is a sequence of vertices where every consecutive pair of a vertices is an edge of G. The first and last vertices in a walk are called *end-vertices*. A walk is *closed* if its two end-vertices are the same. Given two walks $W_1 = (v_1, \ldots, v_t)$ and $W_2 = (v_t, v_{t+1}, \ldots, v_k)$ whose internal vertices are pairwise distinct, we denote by $W_1 \cdot W_2$ the walk $(v_1, \ldots, v_t, v_{t+1}, \ldots, v_k)$. We say that a walk is odd (resp. even) if the number of edges used by the the walk is odd (resp. even). Given $S \subseteq V(G)$, we say that a walk is *S-traversing* if it contains at least one vertex in S. For a graph H and subgraph B of H, we say that a walk W in H is a B-walk if the endpoints of W are in B and the internal vertices of W are not in B. A *path* of a graph is a walk where each vertex is used at most once. A *cycle* of a graph is a closed walk where each vertex, except the end-vertices, is used at most once.

2.1 Treewidth

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 $(B_{v_1}, \ldots, B_{v_t})$, where P is a path $v_1 v_2 \cdots v_t$.

To design a dynamic programming algorithm, we use a convenient form of a tree decomposition known as a nice tree decomposition. A tree T is said to be *rooted* if it has a specified node called the *root*. Let T be a rooted tree with root node r. A node t of T is called a *leaf* node if it has degree one and it is not the root. For two nodes t_1 and t_2 of T, t_1 is a *descendant* of t_2 if the unique path from t_1 to r contains t_2. If a node t_1 is a descendant of a node t_2 and $t_1 t_2 \in E(T)$, then t_1 is called a *child* of t_2.

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 $B_t = B_{t_1} = B_{t_2}$.

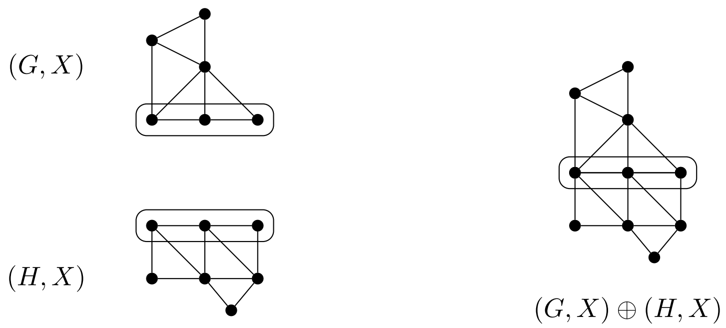

Figure 1 An example of the sum $(G, X) \oplus (H, X)$.

▶ **Theorem 3** (Bodlaender et al. [4]). *Given an n-vertex graph G and a positive integer k, one can either output a tree decomposition of G with width at most $5k + 4$, or correctly answer that the treewidth of G is larger than k, in time $2^{\mathcal{O}(k)}n$.*

▶ **Lemma 4** (folklore; see Lemma 7.4 in [11]). *Given a tree decomposition of an n-vertex graph G of width w, one can construct a nice tree decomposition $(T, \mathcal{B})$ of width w with $|V(T)| = \mathcal{O}(wn)$ in time $\mathcal{O}(k^2 \cdot \max(|V(T)|, |V(G)|))$.*

2.2 Boundaried graphs

For a graph G and $X \subseteq V(G)$, the pair (G, X) is called a *boundaried graph*. Two boundaried graphs (G, X) and (H, X) are said to be *compatible* if $V(G - X) \cap V(H - X) = \emptyset$ and $G[X] = H[X]$. For two compatible boundaried graphs (G, X) and (H, X), the *sum* of two graphs is the graph obtained from the disjoint union of G and H by identifying each vertex of X in G with the same vertex in H and removing an edge from multiple edges that appear in X. We denote the resulting graph by $(G, X) \oplus (H, X)$. See Figure 1 for an example.

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 core 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 s such that $G[Y \setminus X]$ satisfies $P(\Pi)$.

Common base

The meta-result of Theorem 5 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 $H_1, H_2, \dots$ 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: $C_{p,j} \setminus \{v_p(1,j,1), v_p(1,j,2)\}, C_{p,j} \setminus \{v_p(2,j,1), v_p(2,j,2)\}, \dots, C_{p,j} \setminus \{v_p(k,j,1), v_p(k,j,2)\}$.

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 $\{j_1, j_2, \dots, j_k\} = [k]$) 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 5.** *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. The first k vertices, $d_{p,j}(1,1),\dots,d_{p,j}(k,1) \in S$, are complete to the vertices in $\bigcup_{i\in[k]}\{v_p(i,j,1)\}$. The next k vertices, $d_{p,j}(1,2),\dots,d_{p,j}(k,2) \in S$, also twins, are complete to the vertices in $\bigcup_{i\in[k]}\{v_p(i,j,2)\}$. We add $d_{p,j}(1),\dots,d_{p,j}(i),\dots,$ $d_{p,j}(k)$ 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 $i = i'$. See Figure 2 for an illustration.

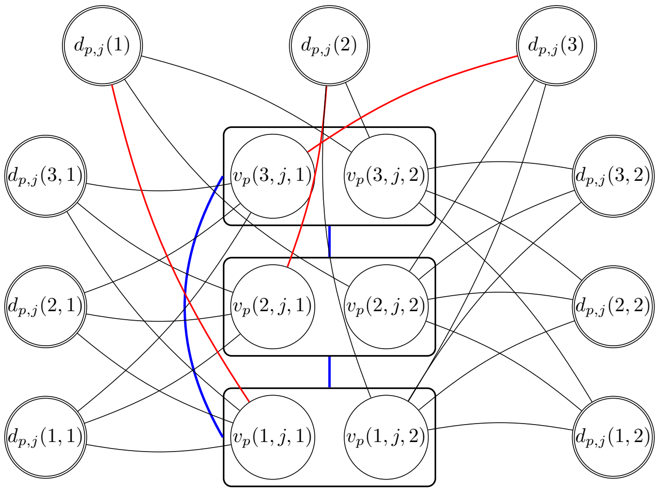

Figure 2 The column selector gadget $\mathcal{G}_1(\mathcal{C})$. Doubly-circled vertices are in S. Blue edges linking boxes denote bicliques between the two surrounded vertex sets. The gadget $\mathcal{G}_2(\mathcal{C})$ is obtained by subdividing each red edge once, and adding a false twin to $d_{p,j}(k,1)$ (or equivalently, any $d_{p,j}(i,1)$) and a false twin to $d_{p,j}(k,2)$.

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 $d_{p,j}(k+1,z)$ complete to $\bigcup_{i\in[k]}\{v_p(i,j,z)\}$, and by subdividing each edge $d_{p,j}(i)v_p(i,j,1)$ once.

▶ **Lemma 6.** *$\mathcal{G}_1(\mathcal{C})$ 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 $\Pi =$ 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 $r_2(p,i), r'_2(p,i), r''_2(p,i)$ complete to $\bigcup_{j \in [k]} \{v_p(i,j,1)\}$. 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 7.** *$\mathcal{G}_1(\mathcal{R})$ 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*.*

Proof. The gadgets $\mathcal{G}_1(\mathcal{R})$ and $\mathcal{G}_2(\mathcal{R})$ add 2 and 3 new vertices, respectively, thus $\mathcal{O}(1)$. Their edge set respects the specification of the row selector.

The set $\{r_1(p,i), r'_1(p,i), v_p(i,j,1), v_p(i,j',1)\}$ is a Π-obstruction, for every pair $j \neq j' \in [k]$, for every problem $\Pi \in \{$Subset FVS, ECT$\}$. Indeed it induces an even cycle (a C_4) and, in the case of Subset FVS, we note that this cycle goes through two vertices of S. The set $\{r_2(p,i), r'_2(p,i), r''_2(p,i), v_p(i,j,1), v_p(i,j',1)\}$ is a Π-obstruction, for every pair $j \neq j' \in [k]$, for $\Pi =$ Subset OCT. Indeed it contains an odd cycle $r_2(p,i) v_p(i,j,1) r'_2(p,i) v_p(i,j',1) r''_2(p,i)$ going through $r'_2(p,i) \in S$. ◀

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 $e_p v_p(i',j',1)$ once.

▶ **Lemma 8.** *$\mathcal{G}_1(\mathcal{E})$ 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*.*

Proof. Both gadgets introduce a constant number of additional vertices (1 and 2, respectively, so $\mathcal{O}(k)$), and their edge set respects the specification. The gadget $\mathcal{G}_1(\mathcal{E})$ is an odd cycle (a triangle) with a vertex in S, hence an obstruction for Subset Feedback Vertex Set and Subset Odd Cycle Transversal. The gadget $\mathcal{G}_2(\mathcal{E})$ is an even cycle (a C_4), hence an obstruction 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 $r_{p,1}, \ldots, r_{p,k}$ represent the row indices in H_p and H_{p+1}, while the k other vertices $c_{p,1}, \ldots, c_{p,k}$ 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 $c_{p,1}, \ldots, c_{p,k}$.

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 9.** *$\mathcal{G}_1(\mathcal{P})$ 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*.*

Proof. Let $\mathcal{P}_p^1 := \{r_{p,1}, \dots, r_{p,k}, c_{p,1}, \dots, c_{p,k}, c_p\}$. The gadget $\mathcal{G}_1(\mathcal{P})$ adds to the base the set $\mathcal{P}_p^1$ of size $2k+1$, thus $\mathcal{O}(k)$. Hence it trivially satisfies the technical condition of the propagation gadget. The gadget $\mathcal{G}_2(\mathcal{P})$ adds a further k^2 vertices, stemming from the subdivision of the edges $r_{p,i}v_p(i,j,2)$. These vertices have exactly one neighbor in $H_p \cup H_{p+1}$ and the rest of their neighbors in $\mathcal{P}_p^1$, so satisfy the specification. For the same reason, $\mathcal{G}_3(\mathcal{P})$ also satisfies the specification. We denote by $\mathcal{P}_p^2$ the set of $2k+1+k^2$ vertices consisting of $\mathcal{P}_p^1$ plus the subdivision vertices, and $\mathcal{P}_p^3$ the set of $3k+1+k^2$ vertices added in $\mathcal{G}_3(\mathcal{P})$. The edge sets of $\mathcal{G}_1(\mathcal{P}), \mathcal{G}_2(\mathcal{P}), \mathcal{G}_3(\mathcal{P})$ respect the specification of the propagation selector.

For every $i, j \neq j' \in [k]$, $\mathcal{P}_p^1 \cup \{v_p(i,j,2), v_{p+1}(i,j',1)\}$ is a Π-obstruction for $\Pi =$ Subset FVS. Indeed $r_{p,i}v_p(i,j,2)c_{p,j}c_jc_{p,j'}v_{p+1}(i,j',1)$ is a cycle (a C_6) going through $c_j \in S$. Similarly $\mathcal{P}_p^2 \cup \{v_p(i,j,2), v_{p+1}(i,j',1)\}$ is a Π-obstruction for $\Pi =$ Subset OCT, the same cycle being now of odd length (a C_7), due to the subdivision of $r_{p,i}v_p(i,j,2)$. Finally $\mathcal{P}_p^3 \cup \{v_p(i,j,2), v_{p+1}(i,j',1)\}$ is a Π-obstruction for $\Pi =$ ECT since $r_{p,i}w_p(i,j,2)v_p(i,j,2)c_{p,j}c'_{p,j}c_jc_{p,j'}v_{p+1}(i,j',1)$ is an even cycle (a C_8), where $w_p(i,j,2)$ is the subdivided vertex stemming from the edge $r_{p,i}v_p(i,j,2)$. ◀

3.2.5 Wrap-up

▶ **Theorem 10.** *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 5. Sections 3.2.1 to 3.2.4 and Lemmas 6 to 9 show how to build the four types of gadgets. Which problem uses which version of the gadget is summarized in Table 1. See Figure 3 for a schematic representation of the construction for Subset FVS.

	column selector	row selector	edge gadget	propagation gadget
Subset Feedback Vertex Set	$\mathcal{G}_1(\mathcal{C})$	$\mathcal{G}_1(\mathcal{R})$	$\mathcal{G}_1(\mathcal{E})$	$\mathcal{G}_1(\mathcal{P})$
Subset Odd Cycle Transversal	$\mathcal{G}_1(\mathcal{C})$	$\mathcal{G}_2(\mathcal{R})$	$\mathcal{G}_1(\mathcal{E})$	$\mathcal{G}_2(\mathcal{P})$
Even Cycle Transversal	$\mathcal{G}_2(\mathcal{C})$	$\mathcal{G}_1(\mathcal{R})$	$\mathcal{G}_2(\mathcal{E})$	$\mathcal{G}_3(\mathcal{P})$

Table 1 The different gadgets used for the different problems.

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 $\{j_1, \dots, j_k\} = [k]$ 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. ◀

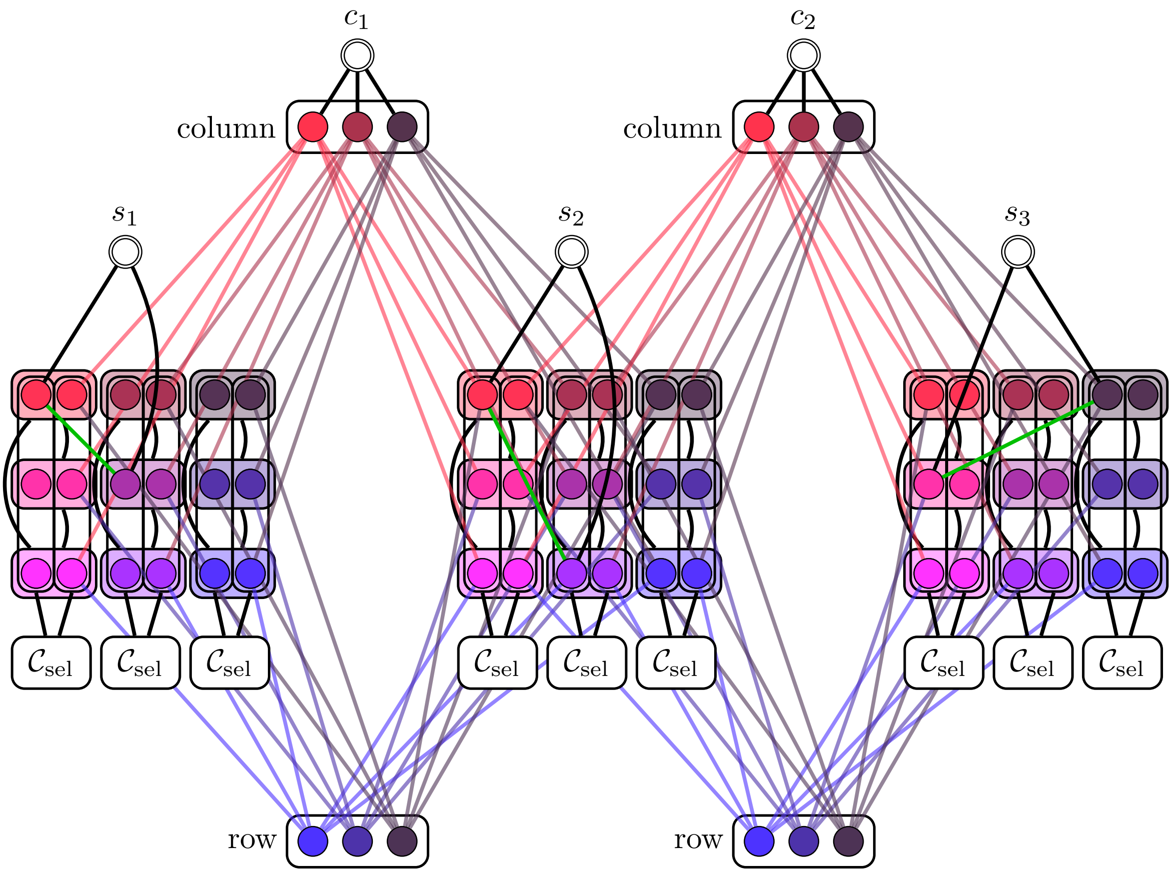

Figure 3 Example of the overall picture for Subset Feedback Vertex Set. The first three edges (in green) in the reduction from $k \times k$-Permutation Independent Set, with $k = 3$, to Subset FVS. The doubly-circled vertices are vertices in S. The column selector gadget $\mathcal{C}_{\text{sel}}$, of size $\mathcal{O}(k)$, forces that only one pair of homologous vertices is retained in each column (see Figure 2). We did *not* represent the row selector gadget.

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 number of terminals is linearly tied to the pathwidth.

▶ **Theorem 11.** *Unless the ETH fails,* Node Multiway Cut *cannot be solved in time* $2^{o(\mathsf{pw} \log \mathsf{pw})} n^{\mathcal{O}(1)}$ *on* n*-vertex graphs with pathwidth* pw.

Proof. We now reduce from $k \times k$-Independent Set. Again let H be an m-edge $k \times k$-Independent Set instance. We build an equivalent Node Multiway Cut instance $(G, T, k' := 2(k-1)km)$, with $|T| = k + 2$, by adding only $2k+2$ new vertices to the base $\bigcup_{p \in [m]} H_p$. We link every non-homologous pair of vertices within each column $C_{p,j}$ (for $p \in [m], j \in [k]$). We add two terminals $t, t' \in T$. For every edge $e_p = (i_p, j_p)(i'_p, j'_p) \in E(H)$, we make $v_p(i_p, j_p, 2)$ and $v_p(i'_p, j'_p, 2)$ adjacent. We also link t to $v_p(i_p, j_p, 2)$, and t' to $v_p(i'_p, j'_p, 2)$.

We add k terminals $r_1, \ldots, r_k \in T$. We link every vertex on an i-th row ($R_{p,i}$) to r_i, except if the vertex is already adjacent to t or t'. This exception concerns the vertices $v_p(i_p, j_p, 2)$ and $v_p(i'_p, j'_p, 2)$. Finally we add k (non-terminal) vertices $c_1, \ldots, c_k$. For each $p \in [m], j \in [k], i \in [k]$, we add an edge between $v_p(i, j, 1)$ and c_i. This finishes the construction of G. The set of terminals is $T := \{t, t', r_1, \ldots, r_k\}$. We ask for a deletion set of size $k' := 2(k-1)km$. The pathwidth of G is $\mathcal{O}(k)$, since it is obtained by adding $2k+2$ vertices ($\{t, t', r_1, \ldots, r_k\}$) to a graph satisfying the gadget specification of Section 3.1 (with "empty" row selector and propagation gadgets).

We now show the correctness of this reduction. Assume that the graph H admits an independent set $I := \{(i_1, 1), (i_2, 2), \ldots, (i_k, k)\}$. We claim that $X := \bigcup_{p \in [m], j \in [k], z \in [2]} H_m \setminus \{v_p(i_j, j, z)\}$ is a solution to the Node Multiway Cut instance. We first observe that the connected component of $G - X$ containing t (and similarly t') does not contain any other terminal. Indeed, since I is an independent set, at most one of $v_p(i_p, j_p, 2)$ and $v_p(i'_p, j'_p, 2)$ is preserved in $G - X$ when $(i_p, j_p)(i'_p, j'_p)$ is an edge of H. Hence each vertex $v_p(i_p, j_p, 2)$ or $v_p(i'_p, j'_p, 2)$ that exists in $G - X$ has degree 1: it is adjacent only to t or t'. So the connected component in $G - X$ of t (resp. t') is a star centered at t (resp. t') whose leaves are all in $\bigcup_{p \in [m]} H_p$, hence are non-terminals. We now observe that there is no path between r_i and $r_{i'}$ (with $i \neq i'$) in $G - X$. Such a path would have to go through a vertex c_j. Indeed, no edge within a column $C_{p,j}$ is preserved in $G - X$ (nor the edges $v_p(i_p, j_p, 2)v_p(i'_p, j'_p, 2)$), so there is no other way to go from one row to another. But each vertex c_j is adjacent to a single row in $G - X$, since we kept only one pair $v_p(i, j, 1), v_p(i, j, 2)$ per column $C_{p,j}$, and we made the same choice in every H_p.

Let us now assume that X is a solution to the Node Multiway Cut instance (G, T, k'). A first observation is that no edge within a column $C_{p,j}$ can be present in $G - X$, otherwise there is a 3-edge path between a pair of terminals in $\{r_1, \ldots, r_k, t, t'\}$, since every edge within $C_{p,j}$ is between non-homologous vertices, and every vertex in $C_{p,j}$ is adjacent to a terminal. This implies that for each $p \in [m]$ and $j \in [k]$, we have $\{v_p(i_{p,j}, j, 1), v_p(i_{p,j}, j, 2)\} \subseteq C_{p,j} \setminus X$ for some $i_{p,j} \in [k]$. In fact, since at least $2(k-1)k$ vertices of H_p must be removed for each $p \in [m]$, and the solution X has size at most $2(k-1)km$, we have $C_{p,j} \setminus X = \{v_p(i_{p,j}, j, 1), v_p(i_{p,j}, j, 2)\}$. In particular, $X \subseteq \bigcup_{p \in [m]} H_p$, so $c_i \notin X$ for each $i \in [k]$. We now show that the $i_{p,j}$'s coincide for each $p \in [m]$. Assume for the sake of contradiction that $v_p(i, j, 1)$ and $v_{p'}(i', j, 1)$ are both present in $G - X$ with $p \neq p'$ and $i \neq i'$. Then $r_i v_p(i, j, 1) c_j v_{p'}(i', j, 1) r_{i'}$ is a path in $G - X$, a contradiction. Therefore $i_{1,j} = \ldots = i_{m,j}$. Let i_j denote this common value. We claim that $\{(i_1, 1), \ldots, (i_k, k)\}$ is an independent set in H. Suppose there is an edge $(i_j, j)(i_{j'}, j') \in E(H)$ for distinct $j, j' \in [k]$. Then there is a path $t v_p(i_j, j, 2) v_p(i_{j'}, j', 2) t'$ in G for some $p \in [m]$, between the terminals t and t', a contradiction. ◀

3.4 Lower bound for Multiway Cut

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

However, we note that reducing from Semi-Regular $k \times k$-Permutation Clique, where all the vertices of a column have the same degree towards another column, and there is no edge with both endpoints in the same row, would make the construction cleaner. So the first reflex is to try and show the same $2^{o(k \log k)}$ lower bound for this variant. Ensuring the semi-regularity condition can be done rather smoothly; it requires revisiting the grouping technique from, say, 3-Coloring, and using known results on equitable colorings. An interested reader can find a complete proof in the appendix. Nonetheless, getting rid of the "horizontal" edges (with both endpoints in the same row) in order to obtain an instance $k \times k$-Permutation Clique, while preserving the semi-regularity, is unnecessarily complex. In particular, the reduction from $k \times k$-Clique to $k \times k$-Permutation Clique presented in the seminal paper [22] does not preserve semi-regularity. To prove the next theorem, we will instead directly reduce from $k \times k$-Permutation Clique and "regularize" the degree by some ad hoc gadgetry.

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

Proof. We reduce from an instance H of $k \times k$-Permutation Clique, so we may assume that there is no edge of H with both endpoints in the same row. Let μ be the number of edges of H, and let Δ be the maximum degree of vertices of H. We associate each $v \in V(H)$ to the non-negative integer $\delta(v) := \Delta - d_H(v)$, where $d_H(v)$ is the degree of v. It is useful to consider the graph H' obtained from H by attaching $\delta(v)$ pendant leaves to each $v \in V(H)$, where each vertex in $V(H)$ has degree Δ in H'. We set $m := k^2 \Delta$, and observe that $m \geqslant \mu$ corresponds to the number of edges in H'.

We build an equivalent Multiway Cut instance (G, T, k'), with $|T| = k + 1$, by adding a polynomial number of vertices to the base $\bigcup_{p \in [\mu + k^2]} H_p$. We do not need the vertices $v_p(i, j, 2)$, so we rename every $v_p(i, j, 1)$ into simply $v_p(i, j)$. Now $R_{p,i}$ is the set $\{v_p(i, 1), \ldots, v_p(i, k)\}$ and $C_{p,j}$ is $\{v_p(1, j), \ldots, v_p(k, j)\}$.

We encode weighted edges in the following way. When we say that we add an edge of weight $w \in \mathbb{N}$ between two vertices u, v, we mean that we add w "parallel" 2-edge paths between u and v. None of the introduced vertices are terminals, so the instance behaves equivalently as with the weighted edge. Thus, for the sake of simplicity, we will treat these parallel paths as a weighted edge. If we require an edge to be "undeletable", we give it weight $k' + 1$, just above the total budget. All the weights of the construction are encoded with polynomially many vertices and unit-weight edges. Therefore the lower bound does apply to the unweighted version of Multiway Cut.

We set $h := 12m - k\Delta - \binom{k}{2}$ and $k' := (h+1)(k-1)k(\mu+k^2)+h$. We add k terminals $r_1,\ldots,r_k \in T$, and we link every vertex in the i-th row (that is, in a set $R_{p,i}$) to r_i by an edge of weight $h+1$. We add k non-terminals $c_1,\ldots,c_k$ and for each $p \in [\mu+k^2], j \in [k], i \in [k]$, we add an edge of weight $k'+1$ (an undeletable edge) between $v_p(i,j)$ and c_i. For every $e_p = (i_p,j_p)(i'_p,j'_p) \in E(H)$ with $p \in [\mu]$, we add the following edge gadget between $v_p(i_p,j_p)$ and $v_p(i'_p,j'_p)$. We first build a 5-vertex path where the edge weights are, from one endpoint to the other, $5,3,3,5$. We denote by z_p the central vertex, and we link the second vertex, x_p, and the fourth vertex, y_p, by an edge of weight 3. We link the first vertex (one endpoint) of the gadget to $v_p(i_p,j_p)$ and to r_{i_p} by edges of weight 3, and the last vertex (the other endpoint) to $v_p(i'_p,j'_p)$ and to $r_{i'_p}$ by edges of weight 3. Finally we link z_p to an additional terminal t (common to every $p \in [\mu]$) by an edge of weight $k'+1$. See Figure 4 for an illustration of the edge gadget and how it is attached to the terminals. So far we

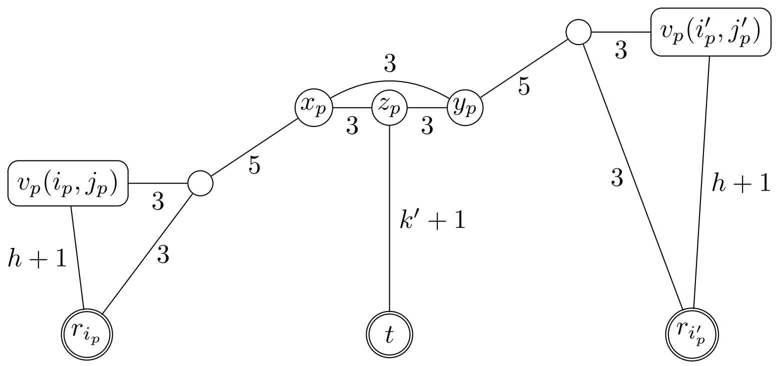

Figure 4 The edge gadget for Multiway Cut.

have added edge gadgets to the first μ copies $H_1,\ldots,H_\mu$. We now describe what we (potentially) add to the last k^2 copies $H_{\mu+1},\ldots,H_{\mu+k^2}$. We put an arbitrary total order over $V(H)$, say, the natural $\leqslant$ where (i,j) is interpreted as $n(i,j) = i + (j-1)k$. We attach to $v_{\mu+n(i,j)}(i,j)$, r_i, and t the following simple gadget, called a *degree-equalizer*, which can be seen as a degenerate case of an edge gadget with multiplicity $\delta((i,j))$ (henceforth we simply write $\delta(i,j)$ for the sake of legibility). We add a vertex $w_{\mu+n(i,j)}(i,j)$, and link it to t by an edge of weight $11\delta(i,j)$, and to $v_{\mu+n(i,j)}(i,j)$ and r_i by edges of weight $6\delta(i,j)$ each. This finishes the construction of $(G, T := \{r_1,\ldots,r_k,t\}, k' := (h+1)(k-1)k(\mu+k^2)+h)$.

The pathwidth of G is $\mathcal{O}(k)$ following the arguments for the Node Multiway Cut construction.

We now show the correctness of the reduction. Assume that there is a clique $C := \{(i_1,1),\ldots,(i_k,k)\}$ in H, with $\{i_1,\ldots,i_k\} = [k]$. We build the following edge deletion-set X for the Multiway Cut instance. We start by including in X all the edges of weight $h+1$ between r_i and $v_p(i,j)$ $(p \in [\mu+k^2])$ such that $(i,j) \notin C$. This represents $k(k-1)(\mu+k^2)$ weighted edges, and $(h+1)k(k-1)(\mu+k^2)$ unit-weight edges.

We distinguish three cases for the edge gadget of every $e_p = (i_p,j_p)(i'_p,j'_p)$ $(p \in [\mu])$. If $\{(i_p,j_p),(i'_p,j'_p)\} \cap C = \emptyset$ (i.e., e_p has no endpoint in C), we add to X the four weight-3 edges incident to $v_p(i_p,j_p)$, r_{i_p}, $v_p(i'_p,j'_p)$, and $r_{i'_p}$; a total of 12 edges. If $|\{(i_p,j_p),(i'_p,j'_p)\} \cap C| = 1$, say, without loss of generality, that $(i_p,j_p) \in C$, then we add the weight-5 edge incident to x_p and the two weight-3 edges incident to $v_p(i'_p,j'_p)$ and $r_{i'_p}$. This consists of 11 edges in total. In the symmetric case $(i'_p,j'_p) \in C$, we would remove the weight-5 edge incident to y_p and the two weight-3 edges incident to $v_p(i_p,j_p)$ and r_{i_p}. Finally if $|\{(i_p,j_p),(i'_p,j'_p)\} \cap C| = 2$, we add the 9 edges of the weighted triangle $x_py_pz_p$ to X.

For every degree-equalizer gadget attached to $H_{\mu+n(i,j)}$, we add to X the weight-$11\delta(i,j)$ edge incident to t if $(i,j) \in C$, and the two weight-$6\delta(i,j)$ edges incident to $w_{\mu+n(i,j)}(i,j)$ if $(i,j) \notin C$. Note that these numbers of edges correspond to what we would remove in $\delta(i,j)$ copies of an edge gadget where the other endpoint is not in C. This finishes the construction of X.

There are $\binom{k}{2}$ edges of H' with both endpoints in C, there are $k\Delta - 2\binom{k}{2}$ edges with exactly one endpoint in C, and $m - k\Delta + \binom{k}{2}$ edges with no endpoint in C. So there are $9\binom{k}{2} + 11(k\Delta - 2\binom{k}{2}) + 12(m - k\Delta + \binom{k}{2}) = 12m - k\Delta - \binom{k}{2} = h$ edges added to X from edge and degree-equalizer gadgets. Thus X has size $(h+1)k(k-1)(\mu+k^2)+h = k'$ as imposed. Let G' be the graph $(V(G), E(G) \setminus X)$. We show that every connected component of G' contains at most one terminal. Observe that in $G' - \{t\}$, each vertex z_p is in a connected component contained in the edge gadget of e_p (and, in particular, not containing a terminal). Since t is only adjacent (by weighted edges) to the vertices z_p and $w_{\mu+n(i,j)}(i,j)$, it follows that the connected component in G' containing t has no other terminals. Note furthermore that the removal of the edges in X disconnects every pair $v_p(i_p,j_p), v_p(i'_p,j'_p)$ in the edge gadget of $e_p = (i_p,j_p)(i'_p,j'_p)$ for $p \in [\mu]$. Thus the vertices reachable from r_i in G' are $\{c_j\} \cup \bigcup_{p \in [m]} C_{p,j}$, such that j is unique integer of $[k]$ with $i_j = i$, as well as some non-terminal vertices in some edge and degree-equalizer gadgets. In particular there is no path between r_i and $r_{i'}$, with $i \neq i'$, in G'. Thus X is a solution.

Let us now assume that the Multiway Cut instance (G,T,k') has a solution X, and let G' be $(V(G), E(G) \setminus X)$. A first observation is that there is a path in G' between any pair of vertices in the j-th column, say $v_p(i,j)$ and $v_{p'}(i',j)$, since there are undeletable edges between c_j and each vertex $v_p(i,j)$. Thus there is a component of G' containing $\bigcup_{p \in [\mu+k^2]} C_{p,j}$, for each $j \in [k]$, and this component contains at most one terminal. With a budget of $(h+1)k(k-1)(\mu+k^2)+h$, one can remove at most $k(k-1)(\mu+k^2)$ edges of weight $h+1$. Since no two edges $r_i v_p(i,j)$ and $r_{i'} v_p(i',j)$ can remain in G', for distinct $i,i' \in [k]$, $j \in [k]$, and $p \in [\mu+k^2]$, at least $k(k-1)$ edges of weight $h+1$ incident to a vertex in H_p must be in X, for each $p \in [\mu+k^2]$, for a total of at least $k(k-1)(\mu+k^2)$ edges of weight $h+1$. Now the only possibility is that, for each $j \in [k]$, there exists an $i_j \in [k]$ such that X contains all the edges of weight $h+1$ from $\bigcup_{p \in [\mu+k^2]} C_{p,j}$ to $\{r_1,\dots,r_k\}$ except those incident to r_{i_j}. We set $C := \{(i_1,1),\dots,(i_k,k)\}$, and we will now show that C is a clique in H. In particular $\{i_1,\dots,i_k\} = [k]$ since there is no edge of H with endpoints in the same row.

First we consider an edge $e_p = (i_p,j_p)(i'_p,j'_p) \in E(H)$ such that $\{(i_p,j_p),(i'_p,j'_p)\} \cap C = \emptyset$. Note that, in this case, $v_p(i_p,j_p)$ (resp. $v_p(i'_p,j'_p)$) is, in G', in the connected component of $r_{i_{j_p}} \neq r_{i_p}$ (resp. $r_{i_{j'_p}} \neq r_{i'_p}$). We then need to separate the seven pairs: $(r_{i_p}, v_p(i_p,j_p))$, $(t, v_p(i_p,j_p))$, (r_{i_p},t), $(r_{i'_p}, v_p(i'_p,j'_p))$, $(t, v_p(i'_p,j'_p))$, $(r_{i'_p},t)$, and $(r_{i_p}, r_{i'_p})$. This requires 12 edge deletions.

We now consider an edge $e_p = (i_p,j_p)(i'_p,j'_p) \in E(H)$ such that $|\{(i_p,j_p),(i'_p,j'_p)\} \cap C| = 1$. We assume that $(i_p,j_p) \in C$ (the other case is symmetric). In this case, $v_p(i'_p,j'_p)$ is, in G', in the connected component of $r_{i_{j'_p}} \neq r_{i'_p}$. We then need to separate the six pairs: $(t, v_p(i_p,j_p))$, (r_{i_p},t), $(r_{i'_p}, v_p(i'_p,j'_p))$, $(t, v_p(i'_p,j'_p))$, $(r_{i'_p},t)$, and $(r_{i_p}, r_{i'_p})$. This requires 11 edge deletions: the weight-5 edge incident to x_p and the two weight-3 edges incident to $v_p(i'_p,j'_p)$ and to $r_{i'_p}$.

Finally let us assume that $e_p = (i_p,j_p)(i'_p,j'_p) \in E(H)$ is such that $|\{(i_p,j_p),(i'_p,j'_p)\} \cap C| = 1$. Here we need to separate the five pairs: $(t, v_p(i_p,j_p))$, (r_{i_p},t), $(t, v_p(i'_p,j'_p))$, $(r_{i'_p},t)$, and $(r_{i_p}, r_{i'_p})$. This requires 9 edge deletions: the three weight-3 edges in the triangle $x_p y_p z_p$.

We now turn to the degree-equalizer gadgets. If $(i,j) \notin C$, then we need to separate the three pairs $(r_i, v_{\mu+n(i,j)}(i,j))$, $(t, v_{\mu+n(i,j)}(i,j))$, and (r_i,t). This requires $12\delta(i,j)$ edge deletions (the weighted edges $r_i w_{\mu+n(i,j)}(i,j)$ and $v_{\mu+n(i,j)}(i,j) w_{\mu+n(i,j)}(i,j)$). If on the contrary $(i,j) \in C$, we only need to separate the two pairs $(t, v_{\mu+n(i,j)}(i,j))$ and (r_i,t). This requires $11\delta(i,j)$ deletions (the weighted edge $t w_{\mu+n(i,j)}(i,j)$).

We denote by s the number of edges in $H[C]$. Since the edge and degree-equalizer gadgets are pairwise edge-disjoint, what we have shown implies that X contains at least $9s + 11(k\Delta - 2s) + 12(m - k\Delta + s) = 12m - k\Delta - s$ edges in the edge gadgets. As X is of size at most k', we have that s has to be equal to $\binom{k}{2}$. This implies that C is a clique. ◀

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

▶ **Theorem 13.** *Unless the ETH fails,* Restricted Edge-Subset Feedback Edge Set *cannot be solved in time* $2^{o(\mathsf{pw} \log \mathsf{pw})} n^{\mathcal{O}(1)}$ *on n-vertex graphs with pathwidth* pw.

It is not difficult to adapt the construction of Theorem 12 for the directed variant of Multiway Cut.

▶ **Theorem 14.** *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 oriented graphs whose underlying 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 17 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 15.** *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.*

Proof. ($\Rightarrow$) Assume toward a contradiction that G is S-bipartite and that a block B of G contains a vertex $s \in S$ and B is not bipartite. Because B is not bipartite, there exists an odd cycle C in B. Since G is S-bipartite by assumption, C does not contain s.

Since B is 2-connected and has at least 3 vertices, there exist two paths P_{sc} and $P_{c's}$ between s and two distinct vertices c, c' of C such that the internal vertices of P_{sc} and $P_{c's}$ and the vertices of C are pairwise distinct. Let $P_{cc'}$ and $\widehat{P}_{cc'}$ be the two paths between c and c' in C. The concatenations $C_1 = P_{sc} \cdot P_{cc'} \cdot P_{c's}$ and $C_2 = P_{sc} \cdot \widehat{P}_{cc'} \cdot P_{c's}$ are two S-traversing cycles. Since C is an odd cycle, the parity of $P_{cc'}$ and $\widehat{P}_{cc'}$ are not the same. Hence, one of the two cycles C_1 and C_2 is an odd S-traversing cycle. This yields a contradiction.

($\Leftarrow$) Assume that G is not S-bipartite. Then, G contains an odd S-traversing cycle C. This cycle is contained in a block B of G. Thus G has a block that is not bipartite and that contains at least one vertex in S. ◀

One can easily modify the proof of the first direction of Lemma 15 to prove the following fact.

▷ Fact 16. 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.

▶ **Theorem 17.** *(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 Theorem 3 and Lemma 4, we obtain a nice tree decomposition of G of width at most $5w + 4$ in time $\mathcal{O}(c^w \cdot n)$ 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. In the following, we introduce a notion of auxiliary graph in order to design an equivalence relation $\equiv_t$ between partial solutions such that $X \equiv_t Y$ if, 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 $G[X]$. Intuitively, the auxiliary graph associated with a partial solution X needs to encode how the active blocks of $\mathrm{Inc}(X)$ are connected together.

We construct the auxiliary graphs $\mathrm{Aux}_p(X, t)$ and $\mathrm{Aux}(X, t)$ from $\mathrm{Inc}(X)$ with the following operations.

1. We remove recursively the leaves and the isolated vertices that are inactive. Let $\mathrm{Aux}_p(X, t)$ be the resulting graph (p for prototype).
2. For every maximal path P of $\mathrm{Aux}_p(X, t)$ 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 5 illustrates the constructions of $\mathrm{Aux}_p(X,t)$ and $\mathrm{Aux}(X,t)$. Observe that Operation 1 removes the inactive blocks of $G[X]$ that contain one vertex in B_t. Thus, every block in $\mathrm{Aux}_p(X,t)$ that contains vertices in B_t are active. By construction, $\mathrm{Aux}(X,t)$ is a forest whose vertices are the active vertices of $\mathrm{Inc}(X)$ and the inactive vertices that have degree at least 3 in $\mathrm{Aux}_p(X,t)$. An important remark is that the algorithm uses the graphs $\mathrm{Aux}(X,t)$ for $X \subseteq V(G_t)$ and in the proof we will use $\mathrm{Aux}(X,t)$ and $\mathrm{Aux}_p(X,t)$ for $X \subseteq V(G_t)$ or $X \subseteq B_t \cup V(\overline{G_t})$.

By Step 2, any edge uv of $\mathrm{Aux}(X,t)$ corresponds to an alternating sequence P of cut vertices and blocks $A_1, A_2, \ldots, A_x$ so that it forms a path from $u = A_1$ to $v = A_x$ in $\mathrm{Inc}(X)$. 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}.

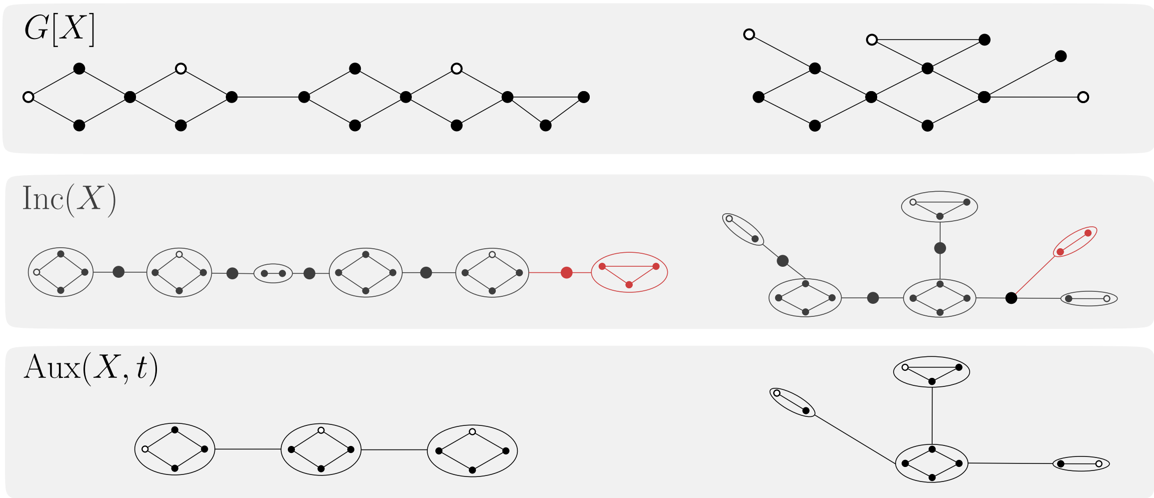

Figure 5 Example of graphs $\mathrm{Inc}(X)$ and $\mathrm{Aux}(X,t)$ constructed from a graph $G[X]$. The vertices in B_t are white filled. The red vertices and edges in $\mathrm{Inc}(X)$ are those we remove to obtained $\mathrm{Aux}_p(X,t)$.

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 $\mathrm{Aux}(X,t)$ to $\mathrm{Aux}(Y,t)$ such that the following conditions are satisfied.

1. For every vertex v in $\mathrm{Aux}(X,t)$, v is active if and only if $\varphi(v)$ is active.
2. For every vertex v in $\mathrm{Aux}(X,t)$, v is a block if and only if $\varphi(v)$ is a block.
3. For every active cut vertex v in $\mathrm{Aux}(X,t)$, we have $\varphi(v) = v$.
4. For every active block B in $\mathrm{Aux}(X,t)$:
   a. $V(B) \cap B_t = V(\varphi(B)) \cap B_t$,
   b. $V(B) \cap S \neq \emptyset$ if and only if $V(\varphi(B)) \cap S \neq \emptyset$,
   c. B is bipartite if and only if $\varphi(B)$ is bipartite.
5. For every edge uv in $\mathrm{Aux}(X,t)$:
   a. M_{uv} is bipartite if and only if $M_{\varphi(u)\varphi(v)}$ is bipartite,
   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 18. For every node t of T, the equivalence relation $\equiv_t$ has $2^{\mathcal{O}(\mathsf{tw}\log\mathsf{tw})}$ equivalence classes.

Proof. Let t be a node and X be a partial solution of G_t. Let $k = |B_t|$. In the following, we will upper bound the number of possibilities for the conditions in the definition of $\equiv_t$. Notice that there are at most 2^k possibilities for $X \cap B_t$.

Now, observe that the number of active blocks of $G[X]$ is at most k. Note that if an active block contains one vertex of B_t, then it is not a cut vertex of $G[X]$, and if an active block intersects at least two vertices of B_t, then it contains either two cut vertices of $G[X]$ contained in B_t, or it contains at least one vertex of B_t that is not a cut vertex of $G[X]$. We consider $\mathrm{Inc}(X)$ as a rooted forest (where each tree has a root), and give an injection ϕ from the set of active blocks to B_t as follows. For each active block B,

- if it contains a vertex in B_t that is not a cut vertex of $G[X]$, then choose such a vertex v and set $\phi(B) = v$, and
- if all vertices of $B_t \cap V(B)$ are cut vertices of $G[X]$, then $|B_t \cap V(B)| \geq 2$ and we choose one vertex $v \in B_t \cap V(B)$ that is a child of B in $\text{Inc}(X)$, and set $\phi(B) = v$.

Clearly, ϕ is an injection, and it shows that the number of active blocks of $G[X]$ is at most k. We deduce that $\text{Aux}(X,t)$ contains at most $2k$ active vertices as there are at most k active blocks and at most k active cut vertices.

By construction, all the vertices of degree at most 2 in $\text{Aux}(X,t)$ are active vertices of $\text{Inc}(X)$. In particular, the leaves of $\text{Aux}(X,t)$ are active vertices. The leaves connected to vertices of degree at least 3 induce an independent set in $\text{Aux}(X,t)$. We can easily show from this fact that there are at most k leaves connected to vertices of degree 3 (at most the number of part in a partition of k elements). Since $\text{Aux}(X,t)$ is a forest, the number of vertices of degree at least 3 is at most k. We deduce that $\text{Aux}(X,t)$ has at most $2k$ active vertices and k inactive vertices. Hence, there are at most $(2k+1)(k+1)$ possibilities for Condition 1 as it is at most the number of ways of choosing two numbers one in $[0,2k]$ and one in $[0,k]$. By Cayley's formula [8], the number of forests on $3k$ labeled vertices is $(3k+1)^{3k-1}$. Thus, there are $2^{\mathcal{O}(\mathsf{tw}\log\mathsf{tw})}$ non-isomorphic graphs in $\{\text{Aux}(W,t) \mid W \text{ is a partial solution of } G_t\}$.

For Condition 2, there are 2 possibilities for each vertex in $\text{Aux}(X,t)$: either it is a block or a cut vertex. Thus, there are at most 2^{3k} possibilities for this condition.

We claim that there are $2^{\mathcal{O}(k\log k)}$ possibilities for Conditions 3 and 4.a. Let $v_1,\dots,v_d$ be the cut vertices of $G[X]$ in B_t and $X_1,\dots,X_\ell$ be the intersections between B_t and the vertex sets of the active blocks of $G[X]$. Note that for every distinct X_i and X_j, $|X_i \cap X_j| \leq 1$. Moreover, since they came from $\text{Inc}(X)$, there is no cyclic structure; that is, $X_{i_1} - v_{i_2} - X_{i_3} \cdots - v_{i_{\alpha-1}} - X_{i_\alpha}$ where $X_{i_1} = X_{i_\alpha}$ and each v_{i_j} only belongs to X_{i_j} and $X_{i_{j+1}}$. This means that the number of possibilities for $v_1,\dots,v_d$ and $X_1,\dots,X_\ell$ is the same as the number of ways of partitioning a set of k vertices into blocks and cut vertices, as isolated vertices are single blocks.

We claim that the number of ways of partitioning a set of k vertices into blocks and cut vertices is $2^{\mathcal{O}(k\log k)}$. Let T be a set of k vertices. First take a partition $\mathcal{P}$ of T. There are at most $2^{k\log k}$ possibilities for $\mathcal{P}$. Choose among the singletons of $\mathcal{P}$ the cut vertices. There are at most 2^k possibilities. The other parts of $\mathcal{P}$ indicate the vertex set of blocks after removing cut vertices. We add k new dummy parts to $\mathcal{P}$ representing the possible blocks that may contain only cut vertices. Now, we have at most $2k$ parts in $\mathcal{P}$. Observe that any forest between the parts of $\mathcal{P}$ that represent the cut vertices and those that represent the blocks induces one way of decomposing the k vertices into blocks and cut vertices. A dummy part adjacent to the singletons containing the vertices $w_1, w_2, \dots, w_\ell$ indicate that $\{w_1, w_2, \dots, w_\ell\}$ forms a block. By Cayley's formula [8], the number of forests on r labeled vertices is $(r+1)^{r-1}$. So, there are at most $(2k+1)^{2k-1}$ ways. Hence, the number of ways of partitioning a set of k vertices into blocks and cut vertices is at most $2^{\mathcal{O}(k\log k)}$.

For Conditions 4.b and 4.c, there are 3 possibilities for each active block of $\text{Aux}(X,t)$. Indeed, since $G[X]$ is S-bipartite and by Lemma 15, if a block is not bipartite, then it cannot contain vertices in S. Thus, there are at most 3^k possibilities for Condition 4.b and 4.c.

For Condition 5, there are 6 possibilities for each edge uv of $\text{Aux}(X,t)$: 3 possibilities for the parity of paths between the endpoints of M_{uv} and two for the existence of a vertex in S in M_{uv}. Since $\text{Aux}(X,t)$ is a forest with at most $3k$ vertices, we have at most 6^{3k-1} possibilities for Condition 5.

It remains to upper bound the number of possibilities for Condition 6 on the parities of the paths between the vertices in B_t. Let u and v be two vertices in $X \cap B_t$. If there is no path between u and v, then we can see this in $\text{Aux}(X,t)$ as it implies that there is no path between the active vertices associated with u and v.

Assume that u and v are connected in $G[X]$. Let P_{uv} be a path between u and v in $G[X]$. Let B_u be the block of $G[X]$ that contains u and its neighbor in P_{uv}. Similarly, let B_v be the block that contains v and its neighbor P_{uv}. Observe that we can have $B_u = B_v$ if and only if u and v are in the same block. By construction, there exists a unique path P between B_u and B_v in $\text{Aux}(X,t)$ (this path can have length 0 if $B_u = B_v$). If there exists a block B in P that is not bipartite, then by Fact 16, we deduce that there exist an odd path and an even path between u and v. If such a non-bipartite block B exists, then either $B = B_u = B_v$ or there exists an edge uv used by P such that M_{uv} contains B. If $B = B_u = B_v$, then B is an active block of $\text{Inc}(X)$ since it contains at least two vertices in B_t. In this case, Condition 4.b stores the information that B is not bipartite. Otherwise, if there is an edge uv used by P such that M_{uv} contains B, then M_{uv} is not bipartite and Condition 5 stores this information.

Suppose now that every block in P is bipartite. Let H be the subgraph that is the union of the bipartite blocks in $G[X]$, and let (X_1, X_2) be a bipartition of H. By assumption the union of the blocks in P is a subgraph of H. Thus, every path between u and v has the same parity and this only depends on whether u and v belong to the same part of $(X_1 \cap B_t, X_2 \cap B_t)$. We deduce that the number of possibilities for Condition 6 are at most 2^k.

For each condition on $\equiv_t$, we proved that the number of possibilities are $2^{\mathcal{O}(k)}$ or $2^{\mathcal{O}(k \log k)}$. Since $k = B_t \leq 5\mathsf{tw} + 4$, we conclude that $\equiv_t$ has at most $2^{\mathcal{O}(\mathsf{tw} \log \mathsf{tw})}$ equivalence classes. ◀

▷ Claim 19. 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.

Proof. Assume that $X \equiv_t Y$ and let $Z \subseteq V(\overline{G_t})$ such that $G[Y \cup Z]$ is S-bipartite. Let φ be the isomorphism from $\text{Aux}(X, t)$ to $\text{Aux}(Y, t)$ given by $\equiv_t$. We show that $G[X \cup Z]$ is S-bipartite. This will prove the claim because $\equiv_t$ is an equivalence relation. To prove that $G[X \cup Z]$ is S-bipartite, by Claim 15, it is sufficient to prove that every block B of $G[X \cup Z]$ is S-bipartite.

Let $G_X := G[X], G_Y := G[Y]$, and $G_Z := G[Z \cup (B_t \cap X)]$. We observe that $G[X \cup Z] := (G_X, (B_t \cap X)) \oplus (G_Z, (B_t \cap X))$ and $G[Y \cup Z] := (G_Y, (B_t \cap X)) \oplus (G_Z, (B_t \cap X))$.

Let B be a block of $G[X \cup Z]$. Observe that if B is a block of $G[X]$, then it is S-bipartite because X is a partial solution. Moreover, if B is a block of $G[Z]$, then it is S-bipartite because $G[Y \cup Z]$ is S-bipartite.

In the following, we assume that B contains vertices from X and Z. Consequently, B has at least 2 vertices in B_t. Observe that for every block B' of G_X or G_Z, either $|V(B') \cap V(B)| \leq 1$, or B' is fully contained in B. Thus, all the blocks of G_X or G_Z contained in B are in $\text{Aux}_p(X, t)$ or $\text{Aux}_p(Z \cup (X \cap B_t), t)$.

We will take a corresponding 2-connected subgraph in $G[Y \cup Z]$. Let $\mathcal{B}_X$ be the set that contains the blocks and the cut vertices of $\text{Aux}(X, t)$ contained in B. We take the subset Y_B of Y that contains (1) $\varphi(v)$ for every cut vertex v in $\mathcal{B}_Y$, (2) $V(\varphi(B'))$ for every blocks B' in $\mathcal{B}_X$ and (3) $V(M_{\varphi(u)\varphi(v)})$ for every edge uv of $\text{Aux}(X, t)$ with $u, v \in \mathcal{B}_X$. Let $F := G[Y_B \cup (V(B) \cap Z)]$. Since all the blocks in $\text{Aux}(X, t)$ with vertices in S are active, Condition 4.a of $\equiv_t$ guarantees that $V(B) \cap B_t = V(F) \cap B_t$.

We claim that F is a 2-connected induced subgraph of $G[Y \cup Z]$ such that

- F contains a vertex of S if and only if B contains a vertex of S,
- F is bipartite if and only if B is bipartite.

By Lemma 15, this will imply that B is S-bipartite, because F is a subgraph of the S-bipartite graph $G[Y \cup Z]$. Let $F_Y := F \cap G_Y$ and $F_Z := F \cap G_Z$.

(1) (F is 2-connected.) It is not difficult to see that F is connected from the construction of Y_B and because $Y \equiv_t X$ implies that (1) $V(B') \cap B_t = V(\varphi(B')) \cap B_t$ for every active block of $G[X]$ and (2) two blocks $B', \widehat{B}$ of $\text{Aux}(X, t)$ are connected if and only if $\varphi(B')$ and $\varphi(\widehat{B})$ are connected in $\text{Aux}(Y, t)$.

Assume towards a contradiction that F has a cut vertex c. Let a_1, a_2 be the neighbors of c that are contained in distinct components of $F - c$. For each $i \in \{1, 2\}$, let U_i' be a block of G_Y or G_Z containing $a_i c$. Note that U_1' and U_2' are fully contained in F, because every block of G_Y or G_Z containing two vertices of F is contained in F. Therefore, U_i' appears in $\text{Aux}_p(Y, t)$ if it is a block of G_Y and it appears in $\text{Aux}_p(V(G_Z), t)$ otherwise.

Now, we choose U_1 and U_2 in $\text{Aux}(Y, t)$ and $\text{Aux}(V(G_Z), t)$ related to U_1' and U_2', respectively.

- (Case 1. U_1', U_2' are in the same part of $\text{Aux}_p(Y, t)$ or $\text{Aux}_p(V(G_Z), t)$.)
  Let us assume that both U_1' and U_2' are in $\text{Aux}_p(Y, t)$. A similar argument holds for the other case. As c is the intersection of U_1' and U_2' which are blocks of G_Y, c is a cut vertex of G_Y, and it appears in $\text{Aux}_p(Y, t)$. Following the path of $\text{Aux}_p(Y, t)$ with direction from c to U_i', we choose the first vertex U_i in $\text{Aux}(Y, t)$.
- (Case 2. U_1', U_2' are not in the same part of $\text{Aux}_p(Y, t)$ or $\text{Aux}_p(V(G_Z), t)$.)
  Without loss of generality, we assume that U_1' is in $\text{Aux}_p(Y, t)$ and U_2' is in $\text{Aux}_p(V(G_Z), t)$. We explain how to choose U_1. The symmetric argument is applied to U_2. In this case, U_1' and U_2' share c, and therefore, either c is an active cut vertex in G_Y, or U_1' is an active block of G_Y. In the former case, following the path of $\text{Aux}_p(Y, t)$ with direction from c to U_1', we choose the first vertex U_1 in $\text{Aux}(Y, t)$. In the latter case, we set $U_1 := U_1'$.

For each $i \in \{1, 2\}$, if U_i is block in $\text{Aux}(Y, t)$, let $V_i = \varphi^{-1}(U_i)$, otherwise, if U_i is a block in $\text{Aux}(V(G_Z), t)$, let $V_i = U_i$. Because of the construction, V_1 and V_2 are blocks contained in B. We choose a vertex c_X in B corresponding to c in F. If $c \in B_t \cup Z$, then we set $c_X = c$. Otherwise, $c \in Y \setminus B_t$ and c must be a cut vertex in

G_Y and either (A) c is a vertex of $\mathrm{Aux}(Y,t)$ with neighbors U_1 and U_2 or (B) U_1U_2 is an edge of $\mathrm{Aux}(Y,t)$ and c is a vertex of $M_{U_1U_2}$. If (A) holds, then we set $c_X = \varphi^{-1}(c)$ and if (B) holds, then we take c_X a cut vertex in $M_{\varphi^{-1}(U_1)\varphi^{-1}(U_2)}$ (every M_{uv} admits at least one cut vertex by definition).

Since B is 2-connected, there is a path $p_1p_2\cdots p_m$ from $V_1 - c_X$ to $V_2 - c_X$ in $B - c_X$. This provides a sequence $B_1, B_2, \ldots, B_{m'}$ of blocks that appear in $\mathrm{Aux}(X,t)$ or $\mathrm{Aux}(V(G_Z),t)$ such that $B_1 = V_1$ and $B_{m'} = V_2$ and for every $i \leq m'-1$, either B_iB_{i+1} is an edge in $\mathrm{Aux}(X,t)$ or in $\mathrm{Aux}(V(G_Z),t)$, or B_i and B_{i+1} are contained in distinct parts of G_X and G_Z and they share a vertex in B_t.

For every $i \leq m'$, let $\widehat{B}_i$ be B_i if B_i is a block in $\mathrm{Aux}(V(G_Z),t)$ or $\varphi(B_i)$ if B_i is a block in $\mathrm{Aux}(X,t)$. Let $i \leq m'-1$. If B_i and B_{i+1} is an edge of $\mathrm{Aux}(X,t)$, then $\varphi(B_i)\varphi(B_{i+1}) = \widehat{B}_i\widehat{B}_{i+1}$ is an edge in $\mathrm{Aux}(Y,t)$. Now, suppose that B_i and B_{i+1} are contained in distinct parts of G_X and G_Z and assume w.l.o.g. that U_i is a block in G_Y. By Condition 4.a in the definition of $\equiv_t$, we have $V(B_i) \cap B_t = V(\widehat{B}_i) \cap B_t$. We deduce that $\widehat{B}_i$ and $B_{i+1} = \widehat{B}_{i+1}$ share a vertex in B_t. From the sequence $\widehat{B}_1, \ldots, \widehat{B}_{m'}$, we conclude that there exists a path in F between $U_1 - c$ and $U_2 - c$ in $F - c$. This contradicts the assumption that c is a cut vertex.

(2) (F contains a vertex of S if and only if B contains a vertex of S.) Observe that for each block U of $\mathrm{Aux}(Y,t)$, U contains a vertex of S if and only if $\varphi(U)$ contains a vertex of S, and for every edge uv of $\mathrm{Aux}(Y,t)$, M_{uv} has a vertex of S if and only if $M_{\varphi(u)\varphi(v)}$ has a vertex of S. Since $F \cap G_Z = B \cap G_Z$, we obtain the result.

(3) (F is bipartite if and only if B is bipartite.) Suppose that B is bipartite. We take the bipartition (L,R) of B. As a connected bipartite graph has a unique bipartition, this is unique up to changing L and R.

As $F \cap G_Z = B \cap G_Z$, this gives a bipartition (L',R') of $F \cap G_Z$. Let $u,v \in V(F) \cap B_t$ be vertices that are contained in the same connected component of $F \cap G_Y$. Assume u,v are contained in the same part of L' and R'. Since u,v are also contained in the same connected component of $B \cap G_X$ and they are in the same part of L and R, all the paths from u to v in $B \cap G_X$ have even length. Note that the blocks containing edges of the path from u to v are all contained in B, and those blocks appear in $\mathrm{Aux}_p(X,t)$. As B is bipartite, each of these blocks is bipartite.

Since $\mathrm{Aux}(Y,t)$ is isomorphic to $\mathrm{Aux}(X,t)$, there is a corresponding sequence of blocks whose last blocks contain u and v, respectively, and all these blocks are bipartite. By the last condition of the equivalence relation $\equiv_t$, there is an even path from u to v in G_Y. This shows that a bipartition of $F \cap G_Y$ is compatible with the bipartition (L',R') of $F \cap G_Z$. Thus, F is 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 18 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, and prove the correctness and the running time of each procedure. 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 $X \in \mathcal{R}[t', I \setminus \{v\}]$.

We claim that $\mathcal{R}[t,I]$ is a reduced set of $\mathcal{P}[t,I]$. Let $X \in \mathcal{P}[t,I]$. As $v \in I$, we have that $X \setminus \{v\} \in \mathcal{P}[t', I \setminus \{v\}]$. Since $\mathcal{R}[t', I \setminus \{v\}]$ is a reduced set of $\mathcal{P}[t', I \setminus \{v\}]$, there exists $X' \in \mathcal{P}[t', I \setminus \{v\}]$ such that $X' \equiv_{t'} X \setminus \{v\}$ and $w(X') \geq w(X \setminus \{v\})$. By the construction of $\mathcal{R}[t,I]$, we added $\widehat{X}$ to $\mathcal{R}[t,I]$ where $\widehat{X} \equiv_t X' \cup \{v\}$ and $w(\widehat{X}) \geq w(X' \cup \{v\})$. It is not difficult to check that $\widehat{X} \equiv_t X$ by considering G_t and the sum $(G_{t'}, B_t \setminus \{v\}) \oplus (G[B_t], B_t \setminus \{v\})$ and applying Claim 21.

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

We set $\mathcal{R}[t, I] = \mathsf{reduce}_t(\mathcal{R}[t', I] \cup \mathcal{R}[t', I \cup \{v\}])$. We easily deduce that $\mathcal{R}[t, I]$ is a reduced set of $\mathcal{P}[t, I]$ from the fact that, by definition, $\mathcal{P}[t, I] = \mathcal{P}[t', I] \cup \mathcal{P}[t', I \cup \{v\}]$.

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 claim that $\mathcal{R}[t, I]$ is a reduced set of $\mathcal{P}[t, I]$. Let $X \in \mathcal{P}[t, I]$. For each $i \in \{1, 2\}$, let $X_i := X \cap V(G_{t_i})$. It is not difficult to see that $X_i \in \mathcal{P}[t_i, I]$, as it is a partial solution of G_{t_i}.

Since $\mathcal{R}[t_i, I]$ is a reduced set of $\mathcal{P}[t_i, I]$, there exists $X_i' \in \mathcal{R}[t_i, I]$ such that $X_i' \equiv_t X_i$ and $w(X_i') \geq w(X_i)$. By construction, $X_1' \cup X_2' \in \mathcal{A}$, and there exists $X' \in \mathcal{R}[t, I]$ such that $X' \equiv_t X_1' \cup X_2'$ and $w(X') \geq w(X_1' \cup X_2')$. It is not difficult to check that $X' \equiv_t X$ by considering G_t and the sum $(G_{t_1}, B_t) \oplus (G_{t_2}, B_{t_2})$ and applying Claim 21 twice. As $w(X_1' \cup X_2') \geq w(X_1 \cup X_2) = w(X)$, we conclude that $\mathcal{R}[t, I]$ is a reduced set.

It remains to prove the correctness and the running time of our algorithm. We can assume w.l.o.g. that the bag B_r associated with the root r of T is empty. By definition $\mathcal{P}[t, \emptyset]$ contains an optimal solution. Since $\mathcal{R}[t, \emptyset]$ is a reduced set, we conclude that $\mathcal{R}[t, \emptyset]$ also contains an optimal solution.

For the running time, we use the fact that we can compute $\mathsf{reduce}_t(\mathcal{A})$ in time $|\mathcal{A}| 2^{\mathcal{O}(\mathsf{tw} \log \mathsf{tw})} n^2$. This follows from the upper bound of Claim 18 on the number of equivalence classes of $\equiv_t$ and the fact that for every partial solutions X, Y of G_t, we can decide whether $X \equiv_t Y$ in time $\mathcal{O}(n^2)$.

Let t be a node of T that is not a leaf. Observe that we set $\mathcal{R}[t, I] = \mathsf{reduce}_t(\mathcal{A})$ where $\mathcal{A}$ is some sets that depends on the type of the node t. Let k be maximum size of a reduced set $\mathcal{R}[t', I']$ for $t' \in V(T)$ and $I' \subseteq B_{t'}$. By construction, if t is an introduce node, then the size of $\mathcal{A}$ is at most k. If t is a forget node, then the size of $\mathcal{A}$ is at most $2k$. And if t is a join node, then the size of $\mathcal{A}$ is at most k^2. From Claim 18, we have $k \leq 2^{\mathcal{O}(\mathsf{tw} \log \mathsf{tw})}$. We deduce that, for every node t and $I \subseteq B_t$, we can compute $\mathcal{R}[t, I]$ in time $2^{\mathcal{O}(\mathsf{tw} \log \mathsf{tw})} \cdot n^2$. Since T has at most $\mathcal{O}(n \cdot \mathsf{tw}^2)$ nodes and there are at most $2^{\mathcal{O}(\mathsf{tw})}$ possibilities for I, the running time of our algorithm is $2^{\mathcal{O}(\mathsf{tw} \log \mathsf{tw})} n^3$. ◀

The dependency n^3 on the input size n in Theorem 17 was obtained because we keep the partial solutions themselves, and have to check their equivalences. We believe that with a careful argument by keeping only auxiliary graphs $\mathrm{Aux}(X, t)$, we can reduce the dependence of the input size; however, for simplicity, we present the running time with n^3 factor.

Now, we solve the other problems.

▶ **Theorem 20.** Subset Feedback Vertex Set, Restricted Edge-Subset Feedback Edge Set, 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 graphs with treewidth* tw.

Proof. It is easy to check that a graph has no S-traversing cycle if and only if every block of size at least 3 has no vertex of S. So, for Subset FVS, we can simply adapt the algorithm for Subset OCT, with ignoring necessary things for checking bipartiteness. A partial solution at a node t, is a subset Y of $V(G_t)$ such that $G_t[Y]$ is a graph having no S-traversing cycles.

We use the same auxiliary graph $\mathrm{Aux}(Y, t)$, and use the equivalence relation obtained by removing the conditions for bipartiteness in the equivalence relation for Subset OCT. In more detail, we say that two partial solutions are equivalent for Subset FVS if $X \cap B_t = Y \cap B_t$, and there is an isomorphism φ from $\mathrm{Aux}(X, t)$ to $\mathrm{Aux}(Y, t)$ such that the following conditions are satisfied.

- For every vertex v in $\mathrm{Aux}(X, t)$, v is active if and only if $\varphi(v)$ is active.
- For every vertex v in $\mathrm{Aux}(X, t)$, v is a block if and only if $\varphi(v)$ is a block.
- For every active cut vertex v in $\mathrm{Aux}(X, t)$, we have $\varphi(v) = v$.
- For every active block B in $\mathrm{Aux}(X, t)$:
  - $V(B) \cap B_t = V(\varphi(B)) \cap B_t$,
  - $V(B) \cap S \neq \emptyset$ if and only if $V(\varphi(B)) \cap S \neq \emptyset$,
- For every edge uv in $\mathrm{Aux}(X, t)$:
  - $V(M_{uv}) \cap S \neq \emptyset$ if and only if $V(M_{\varphi(u)\varphi(v)}) \cap S \neq \emptyset$.

It is straightforward to adapt an algorithm for Subset OCT to Subset FVS with this equivalence relation. We conclude that Subset FVS admits a $2^{\mathcal{O}(\mathsf{tw}\log\mathsf{tw})}n^3$ time algorithm.

For the weighted variant of Node Multiway Cut, we use the reduction to Subset FVS which is described in the introduction. Given an instance (G, S, k) of Weighted Node Multiway Cut with weight function $w : V(G) \to \mathbb{R}$, we construct an equivalent instance (G', S', k) of Weighted Subset FVS with $\mathsf{tw}(G') \leq \mathsf{tw}(G) + 1$ as follows. We add a new vertex v' to G adjacent to all the vertices in S. Let G' be the resulting graph and $S' = \{v'\}$. Since we only add a vertex, we have $\mathsf{tw}(G') \leq \mathsf{tw}(G) + 1$. For every vertex v in G, we set the weight of v to be "infinite" if $v \in S \cup \{v'\}$ and $w(v)$ otherwise. Consequently, a solution cannot contain vertices in $S \cup \{v'\}$. One easily proves that (G, S, k) is a yes-instance of Weighted Node Multiway Cut if and only if (G', S', k) is a yes-instance of Weighted Subset FVS. Hence, Node Multiway Cut and its weighted variant admit $2^{\mathcal{O}(\mathsf{tw}\log\mathsf{tw})}n^3$ time algorithms.

It remains to prove that the theorem holds for Weighted Restricted Edge-Subset Feedback Edge Set (WSFES for short). For doing so, we use a reduction to Weighted Subset FVS. Let (G, S, k) be a instance of WSFES with weight function $w : E(G) \to \mathbb{R}$. We will construct a instance (G', S', k) of Weighted Subset FVS where $\mathsf{tw}(G') = \mathsf{tw}(G)$.

Let G' be the graph obtained by subdividing the edges of G. Since subdivisions do not increase the treewidth, we have $\mathsf{tw}(G') = \mathsf{tw}(G)$. For every edge e of G, we call v_e the vertex in G' created from the subdivision of e. Let S' be the set $\{v_e \mid e \in S\}$. We give "infinite" weights on the vertices of G' that belong to $V(G) \cup S'$. Moreover, for every edge $e \in E(G) \setminus S$, we give to v_e the same weight as e. Consequently, a solution cannot contain vertices in $V(G) \cup S'$. It is easy to prove that (G, S, k) is a yes-instance of WSFES if and only (G', S', k) is a yes instance of Weighted Subset FVS. We conclude that WSFES and its weighted variant admit $2^{\mathcal{O}(\mathsf{tw}\log\mathsf{tw})}n^3$ time algorithms. ◀

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 Arthur Cayley. A theorem on trees. *Quart. J. Math.*, 23:376–378, 1889.

9 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.

10 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.

11 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.

12 Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. Subset feedback vertex set is fixed-parameter tractable. *SIAM J. Discret. Math.*, 27(1):290–309, 2013. doi:10.1137/110843071.

13 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.

14 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.

15 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.

16 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.

17 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.

18 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which Problems Have Strongly Exponential Complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

19 Naonori Kakimura and Ken-ichi Kawarabayashi. Fixed-parameter tractability for subset feedback set problems with parity constraints. *Theor. Comput. Sci.*, 576:61–76, 2015. doi:10.1016/j.tcs.2015.02.004.

20 Hal A. Kierstead and Alexandr V. Kostochka. A short proof of the Hajnal-Szemerédi Theorem on equitable colouring. *Comb. Probab. Comput.*, 17(2):265–270, 2008. doi:10.1017/S0963548307008619.

21 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.

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

23 Daniel Lokshtanov, Pranabendu Misra, M. S. Ramanujan, and Saket Saurabh. Hitting selected (odd) cycles. *SIAM J. Discret. Math.*, 31(3):1581–1615, 2017. doi:10.1137/15M1041213.

24 Pranabendu Misra, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Parameterized algorithms for even cycle transversal. In Martin Charles Golumbic, Michal Stern, Avivit Levy, and Gila Morgenstern, editors, *Graph-Theoretic Concepts in Computer Science - 38th International Workshop, WG 2012, Jerusalem, Israel, June 26-28, 2012, Revised Selcted Papers*, volume 7551 of *Lecture Notes in Computer Science*, pages 172–183. Springer, 2012. doi:10.1007/978-3-642-34611-8_19.

25 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.

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

27 Magnus Wahlström. Half-integrality, LP-branching and FPT algorithms. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1762–1781. SIAM, 2014. doi:10.1137/1.9781611973402.128.

28 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.

A Semi-regular instances of $k \times k$-Clique

Here we show the claimed ETH lower bound for Semi-Regular $k \times k$-Clique, namely that it is not easier than $k \times k$-Clique. We insist that we do not need this result in the paper, but we believe that it may be helpful in a different context. We recall that by *semi-regular*, we mean that every vertex of a given column has the same degree towards another fixed column.

The known parameterized reduction from k-Clique on general graphs to k-Clique on regular graphs will not work here, since it would increase the parameter polynomially (which we cannot afford). We take a couple of steps back and show how to partition the vertex set $V(G)$ of a hard instance of Bounded-Degree 3-Coloring in such a way that the seminal reduction from 3-Coloring to $k \times k$-Clique only produces semi-regular instances. Let us recall that the latter reduction builds one vertex per 3-coloring of a part of the partition, and links by an edge every pair of consistent partial colorings (i.e., the union of the colorings is proper in the graph induced by the two parts). If the number of parts k is chosen so that $3^{|V(G)|/k} \approx k$, we obtain a "square" k-by-k instance of Clique. Now we want to ensure, in addition, that each 3-coloring of a part P has the same number of consistent 3-colorings of another part P'.

It is a folklore consequence of the Sparsification Lemma [18] and known reductions that 3-coloring a bounded-degree n-vertex graph cannot be done in $2^{o(n)}$. For instance Cygan et al. show the following.

▶ **Theorem 21** (Lemma 1 in [10]). *Unless the ETH fails,* 3-Coloring *on n-vertex graphs of maximum degree 4 cannot be solved in $2^{o(n)}$.*

Ultimately this result is a reduction from 3-SAT. The 3-Coloring instances produced by that reduction serve as our starting point.

▶ **Theorem 22.** *Unless the ETH fails,* Semi-Regular $k \times k$-Clique *cannot be solved in time $2^{o(k \log k)}$.*

Proof. Let G be a hard instance of 3-Coloring with maximum degree 4, produced by the reduction of Theorem 21. G^2, the square of G, (with an edge between two vertices at distance at most 2) has degree 16. By Hajnal-Szemerédi Theorem, G^2 admits an equitable coloring using 17 colors. This equitable coloring can further be found in polynomial time, by Kierstead and Kostochka [20]. We refine this 17 classes arbitrarily into k classes of equal size, such that $\lceil k \log_3 k \rceil = |V(G)|$. Let us call $\mathcal{P}$ the obtained partition of $V(G)$. We perform the reduction of 3-Coloring to $k \times k$-Clique with this equipartition $\mathcal{P}$. The resulting instance is semi-regular. By design, for every pair of parts $P \neq P' \in \mathcal{P}$, $G[P \cup P']$ consists of some isolated edges between P and P', and isolated vertices. Indeed an edge within P, or within P', or a degree-2 vertex would all contradict the coloring of G^2 (with 17 colors). Thus any 3-coloring of P is consistent with the same number of 3-colorings of P'. This shared number is 3 times the number of isolated vertices of $G[P \cup P']$ in P' times twice the number of edges in $G[P \cup P']$. ◀