Feedback Vertex Set in Mixed Graphs

Paul Bonsma[1] Daniel Lokshtanov[2]

Abstract. A mixed graph is a graph with both directed and undirected edges. We present an algorithm for deciding whether a given mixed graph on n vertices contains a feedback vertex set (FVS) of size at most k, in time $47.5^k \cdot k! \cdot O(n^4)$. This is the first fixed parameter tractable algorithm for FVS that applies to both directed and undirected graphs.

1 Introduction

For many algorithmic graph problems, the variant of the problem for directed graphs (*digraphs*) is strictly harder than the one for undirected graphs. In particular, replacing each edge of an undirected graph by two arcs going in opposite directions yields a reduction from undirected to directed graphs for most network design, routing, domination and independence problems including SHORTEST PATH, LONGEST PATH and DOMINATING SET.

The *Feedback Vertex Set* problem is an exception to this pattern. A *feedback vertex set (FVS)* of a (di)graph G is a vertex set $S \subseteq V(G)$ such that $G - S$ contains no cycles. In the *Feedback Vertex Set* (FVS) problem we are given a (di)graph G and an integer k and asked whether G has a feedback vertex set of size at most k. Indeed, if one replaces the edges of an undirected graph G by arcs in both directions, then every feedback vertex set of the resulting graph is a *vertex cover* of G and vice versa. Hence, this transformation can not be used to reduce FEEDBACK VERTEX SET in undirected graphs to the same problem in directed graphs, and other simple transformations do not seem possible either. Thus FVS problems on undirected and directed graphs are different problems; one is not a generalization of the other. This is reflected by the fact that the algorithms for the two problems differ significantly across algorithmic paradigms, be it approximation [2, 1, 11], exact exponential time algorithms [14, 15, 24] or parameterized algorithms [3, 8, 6, 7]. *In this paper we bridge the gap between the parameterized algorithms for* FEEDBACK VERTEX SET *by giving one algorithm that works for both*. More generally, we give the first algorithm for FVS in *mixed graphs*, which are graphs that may contain both edges and arcs. Cycles in mixed graphs are defined as expected: these may contain both edges and arcs, but all arcs should be in the same direction (see Section 2 for precise definitions).

For a mixed graph G on n vertices and an integer k, our algorithm decides in time $2^{O(k)} k!\, O(n^4)$ whether G contains a FVS S with $|S| \leq k$, and if so, returns one. Algorithms of this type are called *Fixed Parameter Tractable (FPT) algorithms*. In general,

---

[1] Humboldt-Universität zu Berlin, Institut für Informatik,Unter den Linden 6, 10099 Berlin, Germany, email: `bonsma@informatik.hu-berlin.de`. Supported by DFG project "Fast parametrized algorithms for directed graph problems"

[2] University of California, San Diego, Department of Computer Science and Engineering, 9500 Gilman Drive, La Jolla, CA 92093-0404, USA, email: `dlokshtanov@cs.ucsd.edu`.

the input for a *parameterized problem* consists of an instance X and integer parameter k. An algorithm for such a problem is an FPT algorithm if its time complexity is bounded by $f(k) \cdot O(|X|^c)$, where $|X|$ denotes the input size of X, $f(k)$ is an arbitrary computable function of k, and c is a constant independent of k. The function $f(k)$ is also called the *parameter function* of the complexity, or of the algorithm. Since the first systematic studies on FPT algorithms in the '90s (see e.g. [9]), this has become a very active area in algorithms. See [13, 21] for recent introductions to the area.

FEEDBACK VERTEX SET is one of the classical graph problems and it was one of the first problems to be identified as NP-hard [19]. The problem has found applications in many areas, see e.g. [12, 7] for references, with one of the main applications in *deadlock recovery* in databases and operating systems. Hence the problem has been extensively studied in algorithms [1, 2, 11, 14, 15, 24, 26]. The parameterized complexity of FEEDBACK VERTEX SET on undirected graphs was settled already in 1984 in a monograph by Melhorn [20]. Over the last two decades we have seen a string of improved algorithms [3, 9, 10, 22, 18, 23, 16, 8, 6] (in order of improving parameter function), and the current fastest FPT algorithm for the problem has running time $O\left(3.83^k kn^2\right)$ [5], where n denotes the number of vertices of the input graph. On the other hand, the parameterized complexity of FEEDBACK VERTEX SET on *directed* graphs was considered one of the most important open problems in Parameterized Complexity for nearly twenty years, until an FPT algorithm with running time $O\left(n^4 4^k k^3 k!\right)$ was given by Chen et al [7] in 2007. Interestingly, in [17], the permanent deadlock resolution problem as it appears in the development of distributed database systems, is reduced to feedback vertex set in mixed graphs. However, to the best of our knowledge, no algorithm for FVS on mixed graphs has previously been described.

We now give an overview of the paper. We start by giving precise definitions in Section 2. In Section 3 we give a sketch of the algorithm, and outline some the obstacles one needs to overcome in order to design an FPT algorithm for FVS in mixed graphs. Our algorithm has three main components: The frame of the algorithm is a standard iterative compression approach described in Section 3. The core of our algorithm consists of two parts: the first is a reduction from a variant of FVS to a multi-cut problem called SKEW SEPARATOR. This reduction, described in Section 4 is a non-trivial modification of the reduction employed for FVS in directed graphs by Chen et al [7]. Our reduction only works on pre-conditioned instances, we describe how to perform the necessary pre-conditioning in Section 5.

2 Preliminaries

We consider edge/arc labeled multi-graphs: formally, mixed graphs consist of a tuple $G = (V, E, A, \psi)$, where V is the vertex set, E is the edge set, and A is the arc set. The incidence function ψ maps edges $e \in E$ to sets $\{u, v\}$ with $u, v \in V$, also denoted as $uv = vu$. Arcs $a \in A$ are mapped by ψ to tuples (u, v) with $u, v \in V$. In the remainder, we will often denote mixed graphs simply by the tuple $G = (V, E, A)$, and denote $e = uv$ for edges $e \in E$ with $\psi(e) = \{u, v\}$, and $a = (u, v)$ for arcs $a \in A$ with $\psi(a) = (u, v)$. Mixed graphs with $A = \emptyset$ will also denoted by $G = (V, E)$. $V(G)$, $E(G)$ and $A(G)$ denote the vertex, edge and arc set respectively of the mixed graph G.

The operation of *contracting* an edge $e = uv$ into a new vertex w consists of the following operations: introduce a new vertex w, for every edge or arc with u or v as end vertex, replace this end vertex by w, and delete u and v. Note that edge identities are preserved: $\psi(e)$ may for instance change from $\{x, u\}$ to $\{x, w\}$, but e is still considered the same edge. Note also that contractions may introduce *parallel edges or arcs* (pairs of edges or arcs e and f with $\psi(e) = \psi(f)$), and *loops* (edges e with $\psi(e) = \{w, w\}$ or arcs a with $\psi(a) = (w, w)$).

For $G = (V, E, A)$ and $S \subseteq V$ or $S \subseteq E \cup A$, by $G[S]$ we denote the subgraph induced by S. In particular, $G[E]$ is obtained by deleting all arcs and resulting isolated vertices. Deletion of S is denoted by $G - S$. The *out-degree* $d^+(v)$ (*in-degree* $d^-(v)$) of a vertex $v \in V$ is the number of arcs $e \in A$ with $\psi(e) = (v, w)$ ($\psi(e) = (w, v)$) for some w. If an arc (v, w) ((w, v)) exists, w is called an *out-neighbor* (*in-neighbor*) of v. Similarly, the *edge degree* $d(v)$ is the number of incident edges, and if $vw \in E$ then w is an *edge neighbor* of v.

A *walk of length* l in a mixed graph $G = (V, E, A)$ is a sequence $v_0, e_1, v_1, e_2, \ldots, e_l, v_l$ such that for all $1 \leq i \leq l$, $e_i \in E \cup A$ and $e_i = v_{i-1}v_i$ or $e_i = (v_{i-1}, v_i)$. This is also called a (v_0, v_l)-*walk*. v_0, v_l are its *end vertices*, $v_1, \ldots, v_{l-1}$ its *internal vertices*. A walk is a *path* if all of its vertices are distinct. A walk $v_0, e_1, v_1, \ldots, v_l$ of length at least 1 is a *cycle* if the vertices $v_0, \ldots, v_{l-1}$ are distinct, $v_0 = v_l$, and all e_i are distinct. (Note that this last condition is only relevant for walks of length 2. Note also that if e is a loop on vertex u, then u, e, u is also considered a cycle.) We will usually denote walks, paths and cycles just by their vertex sequence $v_0, \ldots, v_l$. In addition, we will sometimes encode paths and cycles by their edge/arc set $E_P = \{e_1, \ldots, e_l\}$.

3 Outline of the algorithm

In this section we give an informal overview of our algorithm, the details are given in the following sections. Similar to many previous FVS algorithms [5–8, 16], we will employ the *iterative compression* technique introduced by Reed, Smith and Vetta [25]. Essentially, this means that we start with a trivial subgraph of G and increase it one vertex at a time until G is obtained, maintaining a FVS of size at most $k+1$ throughout the computation. Every time we add a vertex to the graph we perform a *compression* step. That is, given a graph G' with a FVS S of size $k+1$, the algorithm has to decide whether G' has a FVS S' of size k. If the algorithm concludes that G' has no FVS of size k, we can conclude that G does not either, since G' is a subgraph of G. In each compression step the algorithm loops over all 2^{k+1} possibilities for $S \cap S'$. For each choice of $S' \cap S$ we need to solve the following problem.

S-DISJOINT FVS:
INSTANCE: A mixed graph $G = (V, E, A)$ with a FVS S.
TASK: Find a FVS S' of G with $|S'| < |S|$ and $S' \cap S = \emptyset$, or report that this does not exist.

A FVS S' with $|S'| < |S|$ and $S' \cap S = \emptyset$ is called a *small S-disjoint FVS*. The application of iterative compression implies the following lemma.

Lemma 1 **($\star$).**[3] *Suppose S-*DISJOINT FVS *can be solved in time $O\left((k+1)!f(k)n^c\right)$, with $n = |V|$, $k = |S| - 1$ and $f(k)$ non-decreasing. Then FVS can be solved in time $O\left(k(k+1)!f(k)n^{c+1}\right)$.*

Chen et al [7] gave an algorithm for S-Disjoint FVS restricted to digraphs, which we will call *S-Disjoint Directed FVS*. In Section 4 we show that their algorithm can be extended in a non-trivial way to solve the following generalization of the problem to mixed graphs. Let G be an undirected graph with $S \subseteq V(G)$. A vertex set $S' \subseteq V(G)\backslash S$ is a *multiway cut* for S (in G) if there is no (u,v)-path in $G - S'$ for any two distinct $u, v \in S$.

FEEDBACK VERTEX SET / UNDIRECTED MULTIWAY CUT (FVS/UMC):
INSTANCE: A mixed graph $G = (V, E, A)$ with a FVS S, and integer k.
TASK: Find a FVS S' of G with $|S'| \leq k$ and $S' \cap S = \emptyset$, that is also a multiway cut for S in $G[E]$, or report that this does not exist.

A multiway cut S' for $G[E], S$ is also called an *undirected multiway cut (UMC)* for G, S. The remaining question is: how can the FPT algorithm for FVS/UMC be used to solve S-Disjoint FVS? Let G, S be an S-Disjoint FVS instance. Suppose there exists a small S-Disjoint FVS S' for the graph G. If we know which *undirected* paths between S-vertices do not contain any S'-vertices, then these can be contracted, and S' remains a FVS for the resulting graph G^*. In addition, this gives a new vertex set S^* consisting of the old S-vertices and the vertices introduced by the contractions. This then yields an instance G^*, S^* of FVS/UMC, for which S' is a solution. In Section 5 we prove this more formally. However, since we do not know S', it remains to find which undirected paths between S-vertices do not contain S'-vertices. One approach would be to try all possible combinations, but the problem is that the number of such paths may not be bounded by any function of $k = |S|-1$, see the example in Figure 1 (a). (More complex examples with many paths exist, where the solution S' is not immediately obvious.) The example in Figure 1 (a) contains many vertices of degree 2, which are simply reduced in nearly all fast undirected FVS algorithms [8, 26, 16, 5]. However in our case we can easily add arcs to the example to prevent the use of (known) reduction rules, see e.g. Figure 1 (b). Because there may be many such paths, and there are no easy ways to reduce these, we will guess which paths do not contain S'-vertices in two stages: this way we only have to consider $2^{O(k)}$ possibilities, which is shown in Section 5.

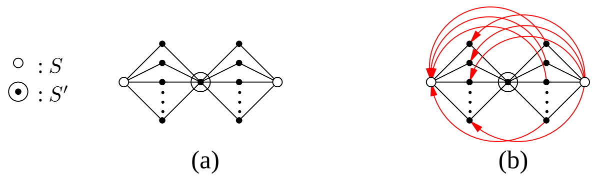


Fig. 1. Graphs with a FVS S and small S-disjoint FVS S', with many undirected S-paths.

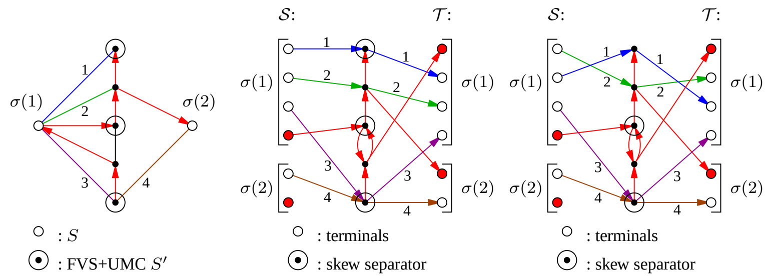
[3] The (full) proofs of claims marked with $\star$ can be found in the appendix.

4 An algorithm for FVS/UMC: reduction to Skew Separator

Let G be a digraph and $\mathcal{S} = s_1, \ldots, s_\ell$ and $\mathcal{T} = t_1, \ldots, t_\ell$ be mutually disjoint vertex sequences such that all $s_i \in V(G)$ have in-degree 0 and all $t_i \in V(G)$ have out-degree 0. A subset $C \subseteq V(G)$ disjoint from $\{s_1, \ldots, s_\ell, t_1, \ldots, t_\ell\}$ is called a *skew separator* if for all $i \geq j$, there is no (s_i, t_j)-path in $G - C$. The vertices in $\mathcal{S}$ will be called *out-terminals* and the vertices in $\mathcal{T}$ *in-terminals*. An FPT algorithm to solve the SKEW SEPARATOR problem defined below is given as a subroutine in the algorithm for DIRECTED FEEDBACK VERTEX SET by Chen et al [7].

SKEW SEPARATOR (SS):
INSTANCE: A digraph G, vertex sequences $\mathcal{S} = s_1, \ldots, s_\ell$ and $\mathcal{T} = t_1, \ldots, t_\ell$ where all $s_i \in V(G)$ have in-degree 0 and all $t_i \in V(G)$ have out-degree 0, and an integer k.
TASK: Find a skew separator C of size at most k, or report that this does not exist.

Theorem 1 (Chen et al [7]). *The Skew Separator problem on instances $G, \mathcal{S}, \mathcal{T}, k$ with $n = |V(G)|$ can be solved in time $4^k k \cdot O(n^3)$.*



(a) FVS/UMC instance G, S (b) correct transformation (c) incorrect transformation

Fig. 2. Correct and incorrect transformations from FVS/UMC to Skew Separator. In- and out-terminals are ordered from top to bottom, so ‘allowed paths’ go from top left to bottom right.

We will use this to give an algorithm for FVS/UMC, using a non-trivial extension of the way SS is used in [7] to give an algorithm for S-Disjoint Directed FVS. We will transform a FVS/UMC instance G, S to a SS instance G_{SS}, $\mathcal{S}$, $\mathcal{T}$, in such a way that S' is a FVS and UMC for G, S if and only if it is a skew separator for $G_{\text{SS}}, \mathcal{S}, \mathcal{T}$. Since every cycle in G contains at least one vertex from S, this can be done by replacing every S-vertex by a set of in- and out-terminals in G_{SS}. The following proposition shows how the order of these terminals should be chosen. A bijective function $\sigma : \{1, \ldots, \ell\} \to S$ is called a *numbering* of S. It is an *arc-compatible numbering* if there are no arcs $(\sigma(i), \sigma(j))$ in G with $i > j$.

Proposition 1 ($\star$). *Let $C \subseteq V \backslash S$ be a FVS and UMC for the graph $G = (V, E, A)$ and vertex set $S \subseteq V$. Then a numbering σ of S exists such that for all $1 \leq j < i \leq |S|$, there is no path from $\sigma(i)$ to $\sigma(j)$ in $G - C$.*

Since in our case edges are present, we cannot achieve the desired correspondence by introducing just one terminal pair for every S-vertex, as was done by Chen et al [7]. Instead, for every vertex $v \in S$, we introduce a single terminal pair for all arcs incident with v in G, and in addition, for every edge incident with v we introduce a terminal pair specifically for this edge. The transformation is illustrated in Figure 2 (a) and (b). Numbers and colors for edges show how edges in G correspond to arcs in G_{ss}. For every $v \in S$, the red terminal pair is used for all incident arcs. Observe that in this example, a set S' is a FVS and UMC in G, S if and only if it is a skew separator in $G_{\text{ss}}, \mathcal{S}, \mathcal{T}$. However, this correspondence does not hold for arbitrary orderings of the edges incident with a vertex $v \in S$, as is shown by the different order used in Figure 2 (c). The indicated skew separator of size 2 does not correspond to a FVS and UMC in G, S.

Construction: We now define the transformation in detail. Let G, S, k be an instance of FVS/UMC, with $|S| = \ell$. We define the relation $\prec$ on $V(G)\backslash S$ as follows: $u \prec v$ if and only if there is a (v,u)-path in $G - S$ but no (u,v)-path (and $u \neq v$). Observe that $\prec$ is transitive and antisymmetric, and therefore a partial order on $V(G)\backslash S$.

For any numbering σ of S, the graph $G_{\text{ss}}(G,\sigma)$ is obtained from G as follows: For every $i \in \{1,\dots,\ell\}$, we do the following: denote $v = \sigma(i)$. let $vw_1,\dots,vw_d$ be the edges incident with v, ordered such that if $w_x \prec w_y$ then $x < y$. Since $\prec$ is a partial order, such an ordering exists and is given by an arbitrary linear extension of $\prec$. Apply the following operations: **(1)** Add the vertices $s_i^1,\dots,s_i^{d+1}$ and $t_i^1,\dots,t_i^{d+1}$. **(2)** For every arc (v,u) with $u \notin S$, add an arc (s_i^{d+1},u). **(3)** For every arc (u,v) with $u \notin S$, add an arc (u,t_i^1). **(4)** For every edge vw_j, add arcs (s_i^j,w_j) and (w_j,t_i^{j+1}). **(5)** Delete v.

After this is done for every $v \in S$, replace all remaining edges xy with two arcs (x,y) and (y,x). This yields the digraph $G_{\text{ss}}(G,\sigma)$ and vertex sequences $\mathcal{S} = s_1^1,\dots, s_1^{d_1+1}, s_2^1,\dots,s_2^{d_2+1},\dots\dots, s_\ell^{d_\ell+1}$ and $\mathcal{T} = t_1^1,\dots,t_1^{d_1+1},t_2^1,\dots,t_2^{d_2+1},\dots\dots,t_\ell^{d_\ell+1}$, where $d_i = d(\sigma(i))$ is the edge degree of $\sigma(i)$. The integer k remains unchanged. $G_{\text{ss}}(G,\sigma),\mathcal{S},\mathcal{T},k$ is an instance for SS.

Lemma 2 **($\star$).** *Let S be a FVS for a mixed graph $G = (V,E,A)$, such that $G[S]$ contains no edges and G contains no cycles of length at most 2. Then $C \subseteq V(G)\backslash S$ is a FVS and UMC for G and S if and only if there exists an arc-compatible numbering σ of S such that C is a skew separator for $G_{\text{ss}}(G,\sigma),\mathcal{S},\mathcal{T}$, as constructed above.*

Proof sketch: Let C be a FVS and UMC for G, S. By Proposition 1, we can define a numbering σ of S such that for all $i > j$, there is no path from $\sigma(i)$ to $\sigma(j)$ in $G - C$. Therefore, σ is arc-compatible.

We now show that for this σ, C is a skew separator for $G_{\text{ss}}(G,\sigma),\mathcal{S},\mathcal{T}$. Let $G_{\text{ss}} = G_{\text{ss}}(G,\sigma)$. Suppose C is not a skew separator, so $G_{\text{ss}} - C$ contains a path $P = s_i^x, v_1, \dots, v_\ell, t_j^y$ with $i > j$, or with $i = j$ and $x \geq y$. Then $P' = \sigma(i), v_1, \dots, v_\ell, \sigma(j)$ is (the vertex sequence of) a walk in $G - C$; note that arcs of P may correspond to edges in P' but that the vertex sequence still constitutes a walk. If $i > j$, then all vertices of the walk P' are different and hence it is a $(\sigma(i),\sigma(j))$-path in $G - C$, contradicting the choice of σ. If $i = j$, then P' is a closed walk in $G - C$ of which all internal vertices are distinct. In all cases, it can be shown that P' is a cycle in $G - C$, which gives a contradiction (see appendix). In the case where P' has length 2 it follows from $x \geq y$ and the construction of G_{ss} that distinct e and f can be chosen to ensure that $P' = \sigma(i), e, v_1, f, \sigma(i)$ is a cycle. Thus, C is a skew separator for G_{ss}.

Let C be a skew separator for $G_{\text{ss}} = G_{\text{ss}}(G, \sigma)$, for some arc-compatible numbering σ of S. We prove that C is a FVS and UMC for G, S. Suppose $G[E] - C$ contains a (u, v)-path $P = u, v_1, \ldots, v_\ell, v$ with $u, v \in S$, and no internal vertices in S. Let $u = \sigma(i)$ and $v = \sigma(j)$. Since we assumed that $G[S]$ contains no edges, P has length at least 2. Since all edges not incident with S are replaced with arcs in both directions during the construction of G_{ss}, for some x, y this yields both a path $s_i^x, v_1, \ldots, v_\ell, t_j^{y+1}$ in $G_{\text{ss}} - C$ and a path $s_j^y, v_\ell, \ldots, v_1, t_i^{x+1}$ in $G_{\text{ss}} - C$. One of these paths contradicts that C is a skew separator. This shows that C is a multiway cut for $G[E]$ and S.

Next, suppose $G - C$ contains a cycle K. Since S is a FVS for G, K contains at least one vertex of S. If K contains at least two vertices of S, then K contains a path P from $\sigma(i)$ to $\sigma(j)$ for some $i > j$, with no internal vertices in S. Let $P = \sigma(i), v_1, \ldots, v_\ell, \sigma(j)$. P has length at least two, since σ is arc-compatible, and there are no edges in $G[S]$. Then $P' = s_i^x, v_1, \ldots, v_\ell, t_j^y$ is a path in $G_{\text{ss}} - C$ for some x, y, contradicting that C is a skew separator. So now we may suppose that K contains exactly one vertex of S, w.l.o.g. $K = \sigma(i), v_1, \ldots, v_\ell, \sigma(i)$. Every cycle in G has length at least 3, so $v_1 \neq v_\ell$. Using the relation $\prec$ that was used to construct G_{ss}, it can be shown that in every case K yields a path $P = s_i^x, v_1, \ldots, v_\ell, t_i^{y+1}$ in $G_{\text{ss}} - C$ for some $x > y$, or that K consists only of edges (see appendix). In the latter case, $P' = s_i^y, v_\ell, v_{\ell-1}, \ldots, v_1, t_i^{x+1}$ with $y > x$ is the path in $G_{\text{ss}} - C$ that contradicts that C is a skew separator. This concludes the proof that C is a FVS and UMC for G, S. $\square$

Lemma 2 yields a way to reduce FVS/UMC to the SS problem in the case that the input graph G does not contain any short cycles. To solve such an instance of FVS/UMC, we try all possible arc-compatible orderings σ of S (at most $\ell!$) and solve the instances of SS using Theorem 1. The FVS/UMC instance is a yes-instance if and only if at least one of the produced SS instances is. Using simple reduction rules one can reduce general instances of FVS/UMC to instances which do not contain short cycles. This reduction, together with Theorem 1 gives an FPT algorithm for FVS/UMC.

Theorem 2 **($\star$).** *FVS/UMC on instances G, S, k with $n = |V(G)|$, $k \geq 1$ and $\ell = |S|$ can be solved in time $O(n^3) \cdot \ell!\, 4^k k$.*

5 An algorithm for S-Disjoint FVS: contracting paths

In this section we give an FPT algorithm for S-Disjoint FVS, by reducing it to FVS/UMC. Throughout this section, let $G = (V, E, A)$ be a mixed graph, and S be a FVS for G. The main idea of our algorithm is to try out different guesses for a set of edges $F \subseteq E$ that is not hit by a possible S-disjoint FVS S', and contract F. If a solution S' exists and the appropriate set F that corresponds to S' is considered, then S' remains a FVS, but in addition becomes a UMC. So in the resulting graph, we have an algorithm for finding S'. We now make this precise with the following definition and propositions. Let G^* be the graph obtained from G by contracting a set of edges $F \subseteq E$. Let the set S^* consist of all vertices in G^* resulting from a contraction, and all remaining S-vertices (those that were not incident with an edge from F). Then we say that G^*, S^* *is the result of contracting* F *in* G, S. The short proof of Proposition 2 can be found in the appendix, while Proposition 3 follows easily from the definitions.

Proposition 2 **($\star$).** *Let S be a FVS in a mixed graph $G = (V, E, A)$. Let G^*, S^* be the result of contracting a set $F \subseteq E$ in G, S, where $G[F]$ is a forest. Then a set $S' \subseteq V(G)$ is an S-disjoint FVS for G that is not incident with edges from F if and only if it is an S^*-disjoint FVS in G^*.*

Proposition 3. *Let S be a FVS in a mixed graph $G = (V, E, A)$, and let S' be an S-disjoint FVS for G. Let $F \subseteq E$ be the set of all edges that lie on a path between two S-vertices in $G[E] - S'$. Let G^*, S^* be the result of contracting F in G, S. Then S' is a UMC for G^*, S^*.*

The previous two propositions show that it is safe to contract sets of edges that contain no cycles (no solutions are introduced), and when considering the appropriate set, a possible solution S' indeed becomes a FVS and UMC in the resulting graph. The remaining task is to find a way to consider only a limited number ($2^{O(k)}$) of possibilities for F, while ensuring that a correct choice is among them. To this end we introduce the following definitions and bounds. The definitions are illustrated in Figure 3.

A *branching vertex* for G, S is a vertex v such that there are at least three internally vertex disjoint paths from v to S in $G[E]$. By $\mathcal{B} = \mathcal{B}(G, S)$ we denote the set of branching vertices for G, S. A *connection path* is a path in $G[E]$ with both end vertices in S and $\mathcal{B}$, and no internal vertices in S and $\mathcal{B}$.

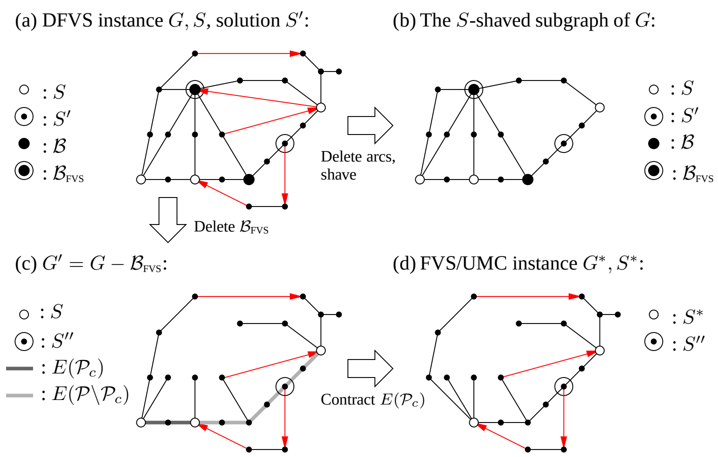


Fig. 3. The graphs and sets defined in Section 5.

Before we give a bound on the number of branching vertices and connection paths, we introduce a different viewpoint on these notions. Given a mixed graph $G = (V, E, A)$ and a FVS S, we construct the S*-shaved subgraph of* G by starting with $G[E]$, and iteratively deleting non-S-vertices that have degree 1, as long as possible. Hence the

Algorithm 1 An algorithm for S-Disjoint FVS

INPUT: A mixed graph $G = (V, E, A)$ with FVS S, and integer $k = |S| - 1$.
OUTPUT: a small S-disjoint FVS S' for G, S, or 'NO' if this does not exist.

```
1.  Compute the set B of branching vertices of G, S.
2.  if |B| > 3k then return 'NO'
3.  for all B_FVS ⊆ B with |B_FVS| ≤ k:
4.      k' := k − |B_FVS|.
5.      G' := G − B_FVS.
6.      Compute the set P of connection paths of G', S.
7.      if |P| > 3k + k' then continue
8.      for all P_c ⊆ P with |P| − |P_c| ≤ k':
9.          Let F = E(P_c).
            Let F* ⊆ F be the edges of components of G'[F] containing an S-vertex.
10.         if G'[F*] contains a cycle then continue
11.         Construct G*, S* by contracting F* in G', S.
12.         if G* contains no loops incident with S*-vertices and
                there is a FVS and UMC S'' for G*, S* with |S''| ≤ k', then
13.             return S' := S'' ∪ B_FVS
14. return 'NO'
```

S-shaved subgraph G_S of G is an undirected graph in which every non-S-vertex has degree at least 2. Considering the three internally vertex disjoint paths from a branching vertex $v \in \mathcal{B}$ to S, and using the fact that S-vertices are never deleted, we see that this process does not delete vertices from $\mathcal{B}$, and neither does it delete vertices from connection paths. Furthermore vertices in $\mathcal{B}$ still have degree at least 3 in G_S. In fact, it turns out that this is another way to characterize the branching vertices and connection paths of G, S:

Proposition 4 **($\star$).** *Let $\mathcal{B}$ be the set of branching vertices of a mixed graph $G = (V, E, A)$ with FVS S, and let G_S be its S-shaved graph. Then $\mathcal{B}$ is exactly the set of non-S-vertices in G_S that have degree at least 3.*

The graph G_S can be thought of as a forest whose all leaves are in S, but where a vertex in S can be simultaneously used as a leaf for multiple paths, or "branches", of the forest. With this viewpoint in mind one can use counting arguments that relate the number of leaves and vertices of degree at least 3 in forests to prove the following lemma.

Lemma 3 **($\star$).** *Let S be a FVS of a mixed graph $G = (V, E, A)$ with $k = |S| - 1$, and let S' be a small S-disjoint FVS for G, S. Then G has at most $3k$ branching vertices with respect to S, and G has at most $3k$ connection paths with no vertices in S'.*

We now present Algorithm 1, the algorithm for S-Disjoint FVS. Recall that the 'continue' statement continues with the next iteration of the smallest enclosing for- (or while-) loop, so it skips the remainder of the current iteration. Note that in Line 6, the set $\mathcal{P}$ of connection paths of $G' = G - \mathcal{B}_{\text{FVS}}$ is considered, not the connection paths of G. For a set of paths $\mathcal{P}_c \subseteq \mathcal{P}$, we denote by $E(\mathcal{P}_c)$ the set of all edges that occur in a path in $\mathcal{P}_c$. The following two lemmas prove the correctness of Algorithm 1.

Lemma 4. *Let S be a FVS of a mixed graph $G = (V, E, A)$ with $k = |S| - 1$. If Algorithm 1 returns a set $S' = S'' \cup \mathcal{B}_{\text{FVS}}$, then S' is a small S-disjoint FVS for G.*

Proof. Suppose a solution $S' = S'' \cup \mathcal{B}_{\text{FVS}}$ is returned in Line 13. Then S'' is a FVS and UMC for G^* and S^*, which are obtained from G', S by contracting the edge set $F^* \subseteq E(G')$. Since $G'[F^*]$ contains no cycles (otherwise the condition in Line 10 is satisfied), S'' is an S-disjoint FVS in G' (Proposition 2). Because $G' = G - \mathcal{B}_{\text{FVS}}$, $S'' \cup \mathcal{B}_{\text{FVS}}$ is then an S-disjoint FVS in G, of size at most $k' + |\mathcal{B}_{\text{FVS}}| = k$. □

Lemma 5. *Let S be a FVS of a mixed graph $G = (V, E, A)$ with $k = |S| - 1$. If there exists a small S-disjoint FVS S' for G, then a solution is returned by Algorithm 1.*

Proof. Let S' be a small S-disjoint FVS, and let $\mathcal{B} = \mathcal{B}(G, S)$. By Lemma 3, $|\mathcal{B}| \leq 3k$, so the algorithm does not terminate in Line 2. Now consider the iteration of the for-loop in Line 3 that considers $\mathcal{B}_{\text{FVS}} := \mathcal{B} \cap S'$, and thus the graph $G' = G - \mathcal{B}_{\text{FVS}}$ and parameter $k' = k - |\mathcal{B}_{\text{FVS}}|$. Let $S'' = S' \backslash \mathcal{B}_{\text{FVS}}$, which is an S-disjoint FVS for G' of size at most k'. So we may apply the propositions and lemmas from this section to G', S and S''.

Observe that after deleting a subset $\mathcal{B}_{\text{FVS}}$ of branching vertices of G, some other vertices may lose their branching vertex status, but no branching vertices are introduced. In other words, $\mathcal{B}(G', S) \subseteq \mathcal{B}(G, S) \backslash \mathcal{B}_{\text{FVS}}$. Therefore, $S'' \cap \mathcal{B}(G', S) = \emptyset$. From Proposition 4 and the fact that all connection paths of G' are part of the S-shaved subgraph of G', it follows that connection paths of G' share no internal vertices. Combining these two facts shows that at most $|S''| \leq k'$ connection paths of G' are incident with a vertex from S''. Lemma 3 shows that G' contains at most $3k$ connection paths not incident with S'', so there are at most $3k + k'$ connection paths in total. Therefore, in Line 7, the algorithm does not continue to the next iteration.

Now let $\mathcal{P}$ be the set of connection paths of G', S, and let $\mathcal{P}_c \subseteq \mathcal{P}$ be those connection paths that are not incident with an S''-vertex. Since we observed that $|\mathcal{P} \backslash \mathcal{P}_c| \leq k'$, we may consider the iteration of the for-loop in Line 8 that considers $\mathcal{P}_c$. Note that the set F^* constructed in Line 9 contains all edges of G' that lie on some undirected path P between two S-vertices in $G' - S''$, since every such path P consists of a sequence of connection paths. Since S'' is a FVS for G', every component of $G'[F^*]$ is a tree, so in Line 10 the algorithm does not continue to the next iteration. Let G^*, S^* be obtained by contracting F^* in G', S. By Proposition 2, S'' is an S^*-disjoint FVS in G^*. By Proposition 3, S'' is a UMC for G^*, S^*. Hence in Line 13, a solution will be returned. □

Proposition 5 **($\star$).** *For all constants $c > 2$, $\sum_{i=0}^{k} \binom{ck}{i} \in O\left(\left(\frac{c^c}{(c-1)^{c-1}}\right)^k\right)$.*

Theorem 3 **($\star$).** *On an instance $G = (V, E, A), S$ with $n = |V|$ and $k = |S| - 1$, Algorithm 1 correctly solves S-Disjoint FVS in time $O\left(k(k+1)!\, 47.5^k\, n^3\right)$.*

Lemmas 4 and 5 show that Algorithm 1 returns the correct answer, so it only remains to prove the complexity bound. A detailed analysis is given in the appendix, but here we argue that the complexity is bounded by $2^{O(k)}k! \cdot n^{O(1)}$: By Line 2, $|\mathcal{B}| \leq 3k$, so the number of iterations of the first for-loop is at most $\sum_{i=0}^{k} \binom{3k}{i} \in$

$O(6.75^k)$ (Proposition 5). For every such iteration, let $k' = k - |\mathcal{B}_{\text{FVS}}|$. By Line 7, $|\mathcal{P}| \leq 3k + k' \leq 4k$ holds whenever the second for-loop is entered, so this loop iterates at most $\sum_{i=0}^{k'} \binom{4k}{i} \in O(9.49^k)$ times (Proposition 5). At most once for every iteration, a FVS/UMC problem on the instance G^*, S^*, k' is solved, which can be done with parameter function $|S^*|! \cdot 4^{k'}k'$ (Theorem 2). By construction, every component of $G'[F^*]$ contains an S-vertex, so $|S^*| \leq |S|$, and therefore this contributes at most $(k+1)!4^k k$ to the parameter function. Hence the total parameter function is bounded by $O(6.75^k\ 9.49^k\ 4^k\ k\ (k+1)!) \subseteq O(256.5^k\ k!)$. The running time dependence on n is dominated by solving the FVS/UMC problem in time $O(n^3)$ (Theorem 2), and the construction of G^*, S^* which can also be shown to require $O(n^3)$ time. Combining Theorem 3 with Lemma 1 yields the main theorem of this paper.

Theorem 4. *In time $O\left((k+1)!k^2 47.5^k\ n^4\right)$, it can be decided whether a mixed graph $G = (V, E, A)$ with $|V| = n$ contains a FVS S with $|S| \leq k$.*

6 Discussion

Our research showed that for some problems, perhaps surprisingly, combining the undirected case with the directed case may provide a significant challenge. We therefore think that mixed graphs deserve more attention in the area of graph algorithms.

We remark that our algorithms can be used to decide whether a mixed graph G contains a set S of edges and arcs with $|S| \leq k$ such that $G - S$ is acylic (*Feedback Edge/Arc Set (FE/AS)*). For undirected graphs, this is a trivial problem. For directed graphs this can easily be reduced to directed FVS, by subdividing all arcs with a vertex and replacing all original vertices with $k+1$ copies (to ensure that they are not selected in a FVS of size at most k). For mixed graphs, this last transformation does not work. However we can extend our algorithms for a certain vertex weighted variant, which can then be used to solve FE/AS (See appendix D).

Our first question is whether the complexity of our algorithm can be improved, in particular whether the $k!$ factor can be removed. Not only does this factor asymptotically dominate the running time, but it also seems to be critical in practice: the 47.5^k factor is based on combining a number of upper bounds and it is unlikely that the worst case complexity bound actually applies to arbitrary instances.

Secondly, one may ask whether FVS in mixed graphs admits a polynomial kernelization (see e.g. [4, 26]). Both questions seem to be very challenging, in fact they remain unresolved even when restricted to planar digraphs (see [4]).

Acknowledgement The authors would like to thank Dániel Marx for suggesting that the FVS algorithm might be extended to solve Feedback Edge/Arc Set in mixed graphs.

References

1. V. Bafna, P. Berman, and T. Fujito, *A 2-approximation algorithm for the undirected feedback vertex set problem*, SIAM J. Discrete Math., 12 (1999), pp. 289–297.
2. A. Becker and D. Geiger, *Optimization of pearl's method of conditioning and greedy-like approximation algorithms for the vertex feedback set problem*, Artif. Intell., 83 (1996), pp. 167 – 188.

3. H. Bodlaender, *On disjoint cycles*, in WG'92, vol. 570 of LNCS, Springer, 1992, pp. 230–238.
4. H. L. Bodlaender, F. V. Fomin, D. Lokshtanov, E. Penninkx, S. Saurabh, and D. M. Thilikos, *(Meta) kernelization*, in FOCS'09, IEEE, 2009, pp. 629–638.
5. Y. Cao, J. Chen, and Y. Liu, *On feedback vertex set new measure and new structures*, in Algorithm Theory - SWAT 2010, vol. 6139 of LNCS, Springer, 2010, pp. 93–104.
6. J. Chen, F. Fomin, Y. Liu, S. Lu, and Y. Villanger, *Improved algorithms for feedback vertex set problems*, J. Comput. Syst. Sci., 74 (2008), pp. 1188 – 1198.
7. J. Chen, Y. Liu, S. Lu, B. O'sullivan, and I. Razgon, *A fixed-parameter algorithm for the directed feedback vertex set problem*, J. ACM, 55 (2008), pp. 1–19.
8. F. Dehne, M. Fellows, M. Langston, F. Rosamond, and K. Stevens, *An O*$(2^{O(k)}n^3)$ *FPT algorithm for the undirected feedback vertex set problem*, Theor. Comput. Syst., 41 (2007), pp. 479–492.
9. R. Downey and M. Fellows, *Fixed-parameter tractability and completeness I: Basic results*, SIAM J. Comput., 24 (1995), pp. 873–921.
10. R. G. Downey and M. R. Fellows, *Parameterized complexity*, Springer-Verlag, New York, 1999.
11. G. Even, J. Naor, B. Schieber, and M. Sudan, *Approximating minimum feedback sets and multicuts in directed graphs*, Algorithmica, 20 (1998), pp. 151–174.
12. P. Festa, P. M. Pardalos, and M. Resende, *Feedback set problems*, Kluwer Acad. Publ., Dordrecht, 1999, pp. 209–258.
13. J. Flum and M. Grohe, *Parameterized Complexity Theory*, Springer-Verlag, Berlin, 2006.
14. F. Fomin, S. Gaspers, and A. Pyatkin, *Finding a minimum feedback vertex set in time* $\mathcal{O}(1.7548^n)$, in IWPEC'06, vol. 4169 of LNCS, Springer, 2006, pp. 184–191.
15. F. Fomin and Y. Villanger, *Finding induced subgraphs via minimal triangulations*, in STACS'10, 2010, pp. 383–394.
16. J. Guo, J. Gramm, F. Hüffner, R. Niedermeier, and S. Wernicke, *Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization*, J. Comput. Syst. Sci., 72 (2006), pp. 1386 – 1396.
17. K. Jain, M. Hajiaghayi, and K. Talwar, *The generalized deadlock resolution problem*, in ICALP'05, vol. 3580 of LNCS, Springer, 2005, pp. 853–865.
18. I. Kanj, M. Pelsmajer, and M. Schaefer, *Parameterized algorithms for feedback vertex set*, in IWPEC'04, vol. 3162 of LNCS, Springer, 2004, pp. 235–247.
19. R. Karp, *Reducibility among combinatorial problems*, Plenum Press, New York, 1972, pp. 85–103.
20. K. Mehlhorn, *Data Structures and Algorithms 2: Graph Algorithms and NP-Completeness*, Springer, 1984.
21. R. Niedermeier, *Invitation to fixed-parameter algorithms*, Oxford University Press, Oxford, 2006.
22. V. Raman, S. Saurabh, and C. R. Subramanian, *Faster fixed parameter tractable algorithms for undirected feedback vertex set*, in ISAAC'02, vol. 2518 of LNCS, Springer, 2002, pp. 51–68.
23. ———, *Faster fixed parameter tractable algorithms for finding feedback vertex sets*, ACM Trans. Algorithms, 2 (2006), pp. 403–415.
24. I. Razgon, *Computing minimum directed feedback vertex set in O(1.9977*^{n}*)*, in ICTCS, 2007, pp. 70–81.
25. B. Reed, K. Smith, and A. Vetta, *Finding odd cycle transversals*, Oper. Res. Lett., 32 (2004), pp. 299–301.
26. S. Thomassé, *A quadratic kernel for feedback vertex set*, in SODA '09, SIAM, 2009, pp. 115–119.

A Proofs omitted from Section 3

Proof of Lemma 1: Suppose S-DISJOINT FVS can be solved in time $O\left((k+1)!f(k)n^c\right)$. We give an algorithm for FEEDBACK VERTEX SET. For a given mixed graph G let $V(G) = \{v_1, \ldots, v_n\}$. For every i let $V_i = \{v_1, \ldots, v_i\}$ and G_i be the subgraph of G induced by V_i. Clearly, for any FVS S of G, $S \cap V_i$ is a FVS of G_i. Thus, if G_i has no FVS of size at most k for some $i \leq n$ then neither does G. The algorithm proceeds as follows. It starts with $i = k$ and $S^* = V_k$ and maintains the invariant that S^* is a FVS of G_i of size at most k. Now $S = S^* \cup \{v_{i+1}\}$ is a FVS of G_{i+1} of size at most $k+1$. Suppose that there is a FVS S' of G_{i+1} of size at most k. Let $S_{\text{KEEP}} = S \cap S'$, $S_{\text{DEL}} = S \backslash S' = S \backslash S_{\text{KEEP}}$ and $S_{\text{NEW}} = S' \backslash S$. Then S_{NEW} is a small S_{DEL}-disjoint FVS of $G_{i+1} - S_{\text{KEEP}}$. For each possible choice of S_{KEEP} we run the algorithm for S_{DEL}-DISJOINT FVS. If this procedure produces a small S_{DEL}-disjoint FVS S_{NEW} for $G_{i+1} - S_{\text{KEEP}}$, for some choice of S_{KEEP}, then $S_{\text{NEW}} \cup S_{\text{KEEP}}$ is a FVS of size at most k for G_{i+1}. Then we proceed to the next iteration (with $S_{\text{NEW}} \cup S_{\text{KEEP}}$ in the role of S^*). If no FVS is found for any S_{KEEP}, then G_{i+1} and hence G has no FVS of size at most k.

Now we consider the complexity. We have to consider all possibilities for S_{KEEP}, and define $S_{\text{DEL}} = S \backslash S_{\text{KEEP}}$. If $|S_{\text{DEL}}| = j+1$, then by assumption, deciding whether there exists a small S_{DEL}-disjoint FVS of $G_{i+1} - S_{\text{KEEP}}$ takes time $O\left((j+1)!f(j)n^c\right)$. There are $\sum_{j=0}^{k} \binom{k+1}{j+1} = k+1+\sum_{j=1}^{k}\binom{k+1}{j+1}$ possibilities for S_{KEEP} to consider. This yields a complexity in the order of

$$\sum_{j=1}^{k} \binom{k+1}{j+1} (j+1)!\; f(j)\; n^c = (k+1)!\; n^c \sum_{j=1}^{k} \frac{f(j)}{(k-j)!} < (k+1)! k f(k)\; n^c.$$

We have to repeat this procedure for every G_i, which gives another factor n, which proves the stated complexity. $\square$

B Proofs omitted from Section 4

For proving Proposition 1 and Lemma 2 in detail, we need the following proposition.

Proposition 6. *Let G be an acyclic mixed graph. If G contains a (u,v)-path P_{uv} and a (v,u)-path P_{vu}, then P_{uv} is an undirected path.*

Proof. Suppose that P_{uv} contains at least one arc. Let $P_{uv} = v_0, e_1, v_1, e_2, \ldots, e_{l-1}, v_l$, with all $v_i \in V$ and all $e_i \in E \cup A$.

By induction one can show that if for some i, $e_1, \ldots, e_{i-1}$ are also part of the path P_{vu}, then these are all edges, and P_{vu} ends with the sub path $v_{i-1}, e_{i-1}, \ldots, v_1, e_1, v_0$. Therefore, since P_{uv} contains at least one arc, we can define i to be the smallest index such that e_i is not part of the path P_{vu}. Let j be the smallest index $j \geq i$ such that P_{vu} contains the vertex v_j (clearly such a j exists). Since P_{uv} is a path, and P_{vu} ends with the the sub path $v_{i-1}, e_{i-1}, \ldots, v_1, e_1, v_0$, it follows that v_j appears before v_{i-1} in the sequence P_{vu}. So we can consider the sub path of P_{uv} from v_{i-1} to v_j, and the sub path of P_{vu} from v_j to v_{i-1}. These paths only share the vertices v_{i-1} and v_j, so if one of them has length at least 2, combining them would yield a cycle in G. If both have length 1, then combining them yields the walk $v_{i-1}, e_i, v_i, f, v_{i-1}$, for some $f \in E \cup A$. By choice of e_i, we have $e_i \neq f$, so this is again a cycle in G, a contradiction. □

Proof of Proposition 1: Suppose that for some pair $u, v \in S$, $G - C$ contains both a (u, v)-path P_{uv} and a (v, u)-path P_{vu}. Then Proposition 6 shows that P_{uv} is undirected, which contradicts that C is a UMC for G. So we can define the following relation A_R on S: $(u, v) \in A_R$ if and only if a (u, v)-walk exists in $G - C$. By the above argument, the digraph (S, A_R) is acyclic (it is in fact a partial order), so a numbering σ of S exists with the desired properties: This is given by a topological ordering of the acyclic digraph (S, A_R) / a linear extension of the partial order (S, A_R). □

Detailed proof of Lemma 2: Let C be a FVS and UMC for G, S. By Proposition 1, we can define a numbering σ of S such that for all $i > j$, there is no path from $\sigma(i)$ to $\sigma(j)$ in $G - C$. Therefore, σ is arc-compatible.

We now show that for this σ, C is a skew separator for $G_{\text{ss}}(G, \sigma), \mathcal{S}, \mathcal{T}$. Let $G_{\text{ss}} = G_{\text{ss}}(G, \sigma)$. Suppose C is not a skew separator, so $G_{\text{ss}} - C$ contains a path $P = s_i^x, v_1, \ldots, v_\ell, t_j^y$ with $i > j$, or with $i = j$ and $x \geq y$. Then $P' = \sigma(i), v_1, \ldots, v_\ell, \sigma(j)$ is (the vertex sequence of) a walk in $G - C$; note that arcs of P may correspond to edges in P' but that the vertex sequence still constitutes a walk. If $i > j$, then all vertices of the walk P' are different and hence it is a $(\sigma(i), \sigma(j))$-path in $G - C$, contradicting the choice of σ. If $i = j$, then P' is a closed walk in $G - C$ of which all internal vertices are distinct. If P' has length at least 3, then all edges/arcs of P' are distinct, so it is a cycle, again a contradiction. If P' has length 1, there is a loop incident with $\sigma(i)$, contradicting the assumption that there are no cycles of length at most 2. Finally suppose the walk P' has length 2, so $P = s_i^x, v_1, t_i^y$ (here we denote P by its vertex sequence). Since $x \geq y$, by the construction of G_{ss} it follows that distinct arcs/edges e and f can be chosen in G such that $P' = \sigma(i), e, v_1, f, \sigma(i)$ is a cycle of length 2 in G, again a contradiction. Therefore, C is a skew separator for G_{ss}.

Let C be a skew separator for $G_{\text{ss}} = G_{\text{ss}}(G, \sigma)$, for some arc-compatible numbering σ of S. We prove that C is a FVS and UMC for G, S. Suppose $G[E] - C$ contains a (u, v)-path $P = u, v_1, \ldots, v_\ell, v$ with distinct $u, v \in S$, and no internal vertices in S. Let $u = \sigma(i)$ and $v = \sigma(j)$. Since we assumed that $G[S]$ contains no edges, P has length at least 2. Since all edges not incident with S are replaced with arcs in both directions during the construction of G_{ss}, for some x, y this yields both a path $s_i^x, v_1, \ldots, v_\ell, t_j^{y+1}$ in $G_{\text{ss}} - C$ and a path $s_j^y, v_\ell, \ldots, v_1, t_i^{x+1}$ in $G_{\text{ss}} - C$. One of these paths contradicts that C is a skew separator (depending on whether $i < j$ or $j < i$). This shows that C is a multiway cut for $G[E]$ and S.

Next, suppose $G - C$ contains a cycle K. Since S is a FVS for G, K contains at least one vertex of S. If K contains at least two vertices of S, then K contains a path P from $\sigma(i)$ to $\sigma(j)$ for some $i > j$, with no internal vertices in S. Let $P = \sigma(i), v_1, \ldots, v_\ell, \sigma(j)$. P has length at least two, since σ is arc-compatible, and there are no edges in $G[S]$. Then $P' = s_i^x, v_1, \ldots, v_\ell, t_j^y$ is a path in $G_{\text{ss}} - C$ for some x, y, contradicting that C is a skew separator.

So now we may suppose that K contains exactly one vertex of S, w.l.o.g. $K = \sigma(i), v_1, \ldots, v_\ell, \sigma(i)$. Every cycle in G has length at least 3, so $v_1 \neq v_\ell$. In the case that $(\sigma(i), v_1) \in A$, K yields a path $P = s_i^{d+1}, v_1, \ldots, v_\ell, t_i^y$ in $G_{\text{ss}} - C$ for some $y \leq d+1$, a contradiction (here $d = d(\sigma(i))$ is the edge degree of $\sigma(i)$). On the other hand, if $(v_\ell, \sigma(i)) \in A$, then K yields a path $P = s_i^x, v_1, \ldots, v_\ell, t_i^1$ in $G_{\text{ss}} - C$ for some $x \geq 1$, a contradiction. So finally suppose that $\sigma(i)v_1 \in E$ and $\sigma(i)v_\ell \in E$ are both edges. Then K gives a path $s_i^x, v_1, \ldots, v_\ell, t_i^{y+1}$ in $G_{\text{ss}} - C$ for some x, y. Since C is a skew separator, $x \leq y$. Since $v_1 \neq v_\ell$, $x < y$. Therefore $v_\ell \not\prec v_1$. The cycle K shows that there is a (v_1, v_ℓ)-path P in $G - S$. Then, by the definition of $\prec$, there must also be a (v_ℓ, v_1)-path in $G - S$. But this can only happen if P is an undirected path (Proposition 6). This shows that by reversing the cycle K, we again obtain a cycle $\sigma(i), v_\ell, v_{\ell-1}, \ldots, v_1, \sigma(i)$ in $G - C$, and therefore a path $s_i^y, v_\ell, v_{\ell-1}, \ldots, v_1, t_i^{x+1}$ in $G_{\text{ss}} - C$, a contradiction (since $x < y$). This concludes all cases, so C is a FVS for G. This concludes the proof that C is a FVS and UMC for G, S. □

Proof of Theorem 2: We may return 'NO' immediately if $G[S]$ contains edges, or if $G[S]$ contains cycles. The latter holds in particular if G contains loops. So suppose none of this holds. Then if G contains a cycle C of length 2, C must contain one S-vertex and one non-S-vertex v. Every FVS/UMC solution contains v, so we may reduce the instance by deleting v and decreasing k by one, to obtain an equivalent instance. Furthermore, if any vertex $u \notin S$ has an edge to at least two distinct vertices in S then any undirected multiway cut for S must contain u. Hence we may reduce the instance by deleting u and decreasing k by one. So we may now assume w.l.o.g. that G contains no cycles of length at most 2 and no two vertices in S have edges to the same vertex in $V(G) \setminus S$. To find a FVS and UMC, we try all arc-compatible numberings σ of S, and test whether $G_{\text{ss}}(G, \sigma)$ has a skew separator of size at most k. There are at most $l!$ such numberings. Return such a skew separator C if it is found for any arc-compatible numbering σ, or 'NO' otherwise. By Lemma 2, this correctly solves FVS/UMC. Note that $G_{\text{ss}}(G, \sigma)$ can be constructed in time $O(n^3)$. Since no two vertices in S have edges to the same vertex in $V(G) \setminus S$, $G_{\text{ss}}(G, \sigma)$ has at most $3n$ vertices. Thus, for every choice of σ, the complexity is bounded by $O(n^3) \cdot 4^k k$ (Theorem 1). □

C Proofs omitted from Section 5

Proof of Proposition 2: Let S' be an S-disjoint FVS for G that is not incident with edges from F. Then $S' \subseteq V(G^*) \backslash S^*$, and $G - S'$ is acyclic. Contracting an *edge* cannot introduce cycles, so $G^* - S'$ is acyclic. (Note that this property does not hold for contracting arcs.)

Now suppose that a set $S' \subseteq V(G)$ that is not incident with F is *not* an S-disjoint FVS in G, so $G - S'$ contains a cycle C. When contracting an edge of G, the remaining edges and arcs of C still form a cycle. Therefore, since C does not consist entirely of edges that are contracted ($G[F]$ contains no cycles), a cycle remains in $G^* - S'$, so S' is not an S^*-disjoint FVS in G^*. □

Proof of Proposition 4: It is sufficient to prove that for every vertex $v \in V(G_S) \backslash S$ with degree at least 3, v is a branching vertex. We can grow a path in G_S starting at v by starting at an incident edge, and adding edges until (i) a vertex of S is reached, or (ii) a vertex previously added to the path is reached. This can be done since non-S-vertices have degree at least 2. However case (ii) will not occur, since this would yield a cycle that does not contain an S-vertex, contradicting that S is a FVS for G_S. So this yields a path from v to S. We can grow three paths P_1, P_2 and P_3 leaving v this way, using three different incident edges. These paths cannot share internal vertices, since this would again yield a cycle in $G_S - S$. These paths are also part of G, so v is a branching vertex of G. □

Proof of Lemma 3 Let $\mathcal{B} = \mathcal{B}(G, S)$, and let G_S be the S-shaved subgraph of G. Let L be the set of non-S-vertices of degree 2 in G_S. By Proposition 4, $V(G_S)$ is the disjoint union of the sets S, $\mathcal{B}$ and L. Since S is also a FVS in the undirected subgraph G_S, $G_S - S$ is a forest. Therefore, it is possible to orient all edges of G_S such that *every non-S-vertex has exactly one in-neighbor*: For every tree in the forest $G_S - S$, choose a root vertex r adjacent to S, and orient all edges away from the root. Orient a single edge from S to r, and orient all other edges towards S-vertices. In the rest of the proof, out-degrees, denoted by $d^+(v)$, will refer to such an orientation of G_S. Denote $S'_L = L \cap S'$, $S'_{\mathcal{B}} = \mathcal{B} \cap S$, and $S'_T = S'_L \cup S'_{\mathcal{B}}$. (In the construction of G_S, some S' vertices may have been deleted so it may be that $S'_T \neq S'$.) Let $G'_S = G_S - S'_T$, which is an (oriented) forest. Denote $G'_S = (V'_S, E'_S)$. Since G'_S is a forest, we have

$$|E'_S| \leq |V'_S| - 1 = |S| + |\mathcal{B}| + |L| - |S'_T| - 1. \tag{1}$$

Note that the number of edges of G_S is at least $\sum_{v \in L \cup \mathcal{B}} d^+(v)$. Since every non-$S$-vertex has in-degree exactly 1, deleting such a vertex v removes at most $d^+(v) + 1$ edges. Therefore, after deleting S'_T from G_S the number of edges remaining is at least:

$$|E'_S| \geq \sum_{v \in \mathcal{B} \cup L} d^+(v) - \sum_{v \in S'_T} d^+(v) - |S'_T| = \sum_{v \in (\mathcal{B} \cup L) \setminus S'_T} d^+(v) - |S'_T| \geq$$

$$|L \setminus S'_L| + 2|\mathcal{B} \setminus S'_{\mathcal{B}}| - |S'_T| = |L| + 2|\mathcal{B}| - 2|S'_L| - 3|S'_{\mathcal{B}}|. \tag{2}$$

Combining the upper bound (1) for $|E'_S|$ with the lower bound (2) yields:

$$|\mathcal{B}| - |S'_L| - 2|S'_{\mathcal{B}}| \leq |S| - 1. \tag{3}$$

From Inequality (3) we immediately obtain $|\mathcal{B}| \leq 2|S'_T| + |S| - 1 \leq 3k$, proving the first statement. Secondly, from Inequality 3 we obtain that $|\mathcal{B}| - |S'_{\mathcal{B}}| \leq |S'_L| + |S'_{\mathcal{B}}| + |S| - 1 \leq 2k$. A connection path of G, S that is not incident with a vertex from S', is a path in G'_S with end vertices in $S \cup (\mathcal{B} \setminus S'_{\mathcal{B}})$, and no internal vertices in this set (Proposition 4). Since G'_S is a forest, there can be at most $|S| + |\mathcal{B} \setminus S'_{\mathcal{B}}| - 1 \leq 3k$ of those. This proves the second statement. □

Proposition 5 follows directly from the following two simple and often used bounds; proofs are included for completeness.

Proposition 7. *For all $c > 2$, $\sum_{i=0}^{k} \binom{ck}{i} < \frac{c-1}{c-2} \binom{ck}{k}$.*

Proof. For all $i \le k$,

$$\binom{ck}{i-1} \Big/ \binom{ck}{i} = \frac{i!(ck-i)!}{(i-1)!(ck-i+1)!} = \frac{i}{ck-i+1} \le \frac{i}{(c-1)i+1} < \frac{1}{c-1},$$

So we may write

$$\sum_{i=0}^{k} \binom{ck}{i} < \binom{ck}{k} \sum_{i=0}^{k} \left(\frac{1}{c-1}\right)^{k-i} < \frac{1}{1-\frac{1}{c-1}} \binom{ck}{k} = \frac{c-1}{c-2} \binom{ck}{k}.$$

□

Proposition 8. *For all constants $c > 1$, $\binom{ck}{k} \in O\left(\left(\frac{c^c}{(c-1)^{c-1}}\right)^k\right)$.*

Proof. By Stirling's approximation $n! \in \Theta\left(n^n e^{-n}\sqrt{n}\right)$,

$$\binom{ck}{k} \in O\left(\frac{(ck)^{ck} e^{-ck}\sqrt{ck}}{\big((c-1)k\big)^{(c-1)k} e^{-(c-1)k}\sqrt{(c-1)k}\; k^k e^{-k}\sqrt{k}}\right) \subset$$

$$O\left(\frac{(ck)^{ck}}{\big((c-1)k\big)^{(c-1)k} k^k}\right) = O\left(\left(\frac{c^c}{(c-1)^{c-1}}\right)^k\right).$$

□

Proof of Theorem 3: Lemmas 4 and 5 show that Algorithm 1 returns the correct answer, so it only remains to prove the complexity bound. First, consider the parameter function. By Line 2, $|\mathcal{B}| \le 3k$, so the number of iterations of the first for-loop is at most

$$\sum_{i=0}^{k} \binom{3k}{i}.$$

Here $i = |\mathcal{B}_{\text{FVS}}|$. Let $k' = k - i$. By Line 7, there are at most $3k + k'$ connection paths in G' whenever the second for-loop is entered, so there are at most

$$\sum_{j=0}^{k'} \binom{3k+k'}{j} = \sum_{j=0}^{k-i} \binom{4k-i}{j} < \frac{3}{2} \binom{4k-i}{k-i}$$

choices of $\mathcal{P}_c$ (Proposition 7). So we may bound the total number of iterations of the second for-loop by

$$\frac{3}{2} \sum_{i=0}^{k} \binom{3k}{i} \binom{4k-i}{k-i}.$$

At most once for every iteration, a FVS/UMC problem on the instance G^*, S^*, k' is solved, which can be done with parameter function $|S^*|! \cdot 4^{k'} k'$ (Theorem 2). In this case, by construction, every component of $G'[F^*]$ contains at least one S-vertex, so for every vertex added to S^* at least one is removed, and thus $|S^*| \leq |S| = k+1$. Therefore the parameter function of Algorithm 1 is bounded by a constant times

$$\sum_{i=0}^{k} \binom{3k}{i} \binom{4k-i}{k-i} (k+1)! 4^{k-i} \max\{1, k-i\} \leq$$

$$k(k+1)! \sum_{i=0}^{k} \frac{(3k)!}{i!(3k-i)!} \cdot \frac{(4k-i)!}{(k-i)!(3k)!} \cdot 4^{k-i} =$$

$$k(k+1)! \sum_{i=0}^{k} \binom{k}{i} \binom{4k-i}{k} 4^{k-i} <$$

$$k(k+1)! \binom{4k}{k} \sum_{i=0}^{k} \binom{k}{i} 4^{k-i} \in$$

$$O\left(k(k+1)!\, 9.5^k\, (1+4)^k\right) = O\left(k(k+1)!\, 47.5^k\right).$$

For the last line, we used Proposition 8, and $\frac{4^4}{3^3} = \frac{256}{27} < 9.5$.

Now we prove that the polynomial part of the complexity (the complexity for fixed k) can be bounded by $O(n^3)$, where $n = |V|$. Let $m = |E| + |A|$. Although we allow multi-graphs, w.l.o.g. we may assume $m \in O(n^2)$. Graphs are encoded with adjacency lists in such a way that edges can be deleted in constant time, vertices v can be deleted in time $O(d^T(v))$, and edges uv can be contracted in time $O(d^T(u) + d^T(v))$, where $d^T(v) = d(v) + d^+(v) + d^-(v)$ denotes the total number of arcs and edges incident with v. For most steps in the algorithm (that we did not already attribute to the parameter function) it can now be verified that they can be done in constant time or linear time $O(n+m) \subseteq O(n^2)$. (Lines 1, 5, 6, 9 and 10 require linear time.) In particular, using the alternative characterization of branching vertices from Proposition 4, and the observation that the S-shaved graph of G can be computed in linear time, it can be verified that the sets of branching vertices and connection paths can be computed in linear time in Lines 1 and 6. Only Line 12 and the contraction step in Line 11 need further consideration. If Line 11 is reached, then $G'[F^*]$ contains no cycles (Line 10), so $|F^*| \leq n-1$. Therefore, at most n edge contractions are done. Contracting a single edge (and updating S^*) requires at most time $O(m)$, which gives a complexity of $O(nm) \subseteq O(n^3)$ for Line 11. Evaluating in Line 12 whether a FVS and UMC exists takes time $O(n^3)$ as well for fixed k (Theorem 2). This proves that the total complexity is $O\left(k(k+1)!\, 47.5^k\, n^3\right)$. □

D Feedback Edge/Arc Set

We show that our algorithms can be extended to solve the *Feedback Edge/Arc Set* problem in mixed graphs. In this problem we need to decide whether there exists a set $S \subseteq E \cup A$ for a mixed graph $G = (V, E, A)$ such that $G - S$ is acylic, with $|S| \leq k$. Note that for undirected graphs, this problem can trivially be solved by counting the number of edges for each connected component of G: to make a component on n vertices with m edges acyclic, deleting $m - n + 1$ edges is necessary and sufficient. For directed graphs G, the problem is easily transformed to directed FVS as follows: for every arc (u, v), introduce a new vertex w and replace (u, v) by the arcs (u, w) and (w, v). Next, for every original vertex v (not introduced in the previous step), introduce k additional copies of v. That is, k vertices with the same set of in- and out-neighbors as v. It is easily seen that the resulting digraph G' has a FVS of size at most k if and only if G has a feedback arc set of size at most k. Continuing a familiar pattern, it again seems that for mixed graphs there is no similar trivial way to solve the problem: when edges are present, one cannot simply replace a single vertex by $k + 1$ vertices this way, without introducing new cycles.

We will remedy this by doing a similar replacement step in a later stage, where it is safe since no edges are present: when solving the skew separator problem. In fact, this will result in an FPT algorithm for the following more general problem. If w is a positive integer weight function on the vertices, edges and arcs of a mixed graph $G = (V, E, A)$, then for $S \subseteq V \cup E \cup A$, let $w(S)$ denote $\sum_{x \in S} w(x)$.

WEIGHTED FEEDBACK VERTEX/EDGE/ARC SET (WFVEAS):
INSTANCE: A mixed graph $G = (V, E, A)$, with positive integer vertex, edge and arc weights w, and integer k.
TASK: Find a set $S \subseteq V \cup E \cup A$ with $w(S) \leq k$ such that $G - S$ is acyclic, or report that this does not exist.

Note that the above problem generalizes Feedback Edge/Arc Set, since all vertex weights can be set to $k + 1$. We also generalize the Skew Separator problem introduced in Section 4 to a weighted variant.

WEIGHTED SKEW SEPARATOR (WSS):
INSTANCE: A digraph G with positive integer vertex weights w, vertex sequences $\mathcal{S} = s_1, \ldots, s_l$ and $\mathcal{T} = t_1, \ldots, t_l$ where all $s_i \in V(G)$ have in-degree 0 and all $t_i \in V(G)$ have out-degree 0, and an integer k.
TASK: Find a skew separator C with $w(C) \leq k$, or report that this does not exist.

Proposition 9. *The Weighted Skew Separator problem on instances $G, w, \mathcal{S}, \mathcal{T}, k$ with $n = |V(G)|$ can be solved in time $4^k \cdot O(k^4) \cdot O(n^3)$.*

Proof. Below, we will treat the sequences $\mathcal{S}$ and $\mathcal{T}$ as vertex sets. We transform the problem as follows. Without loss of generality, we may assume that there are no vertex weights higher than $k + 1$. For every vertex $v \in V(G) \backslash (\mathcal{S} \cup \mathcal{T})$, introduce a set V_v of $w(v)$ vertices. For a vertex $v \in \mathcal{S} \cup \mathcal{T}$, set $V_v = \{v\}$. For every arc (u, v), introduce arcs (x, y) for every $x \in V_x$ and $y \in V_y$. Finally, delete all vertices in $V(G) \backslash (\mathcal{S} \cup \mathcal{T})$. Denote the resulting graph by G'. It is easily seen that $S' \subseteq V(G)$ is a skew separator for G if and only if $S'' = \cup_{x \in S'} V_x$ is a skew separator for G', and that $|S''| = w(S')$. Furthermore, every minimum skew separator of G' is of this form. This shows how WSS can be reduced to SS. If $|V(G)| = n$, then $|V(G')| \leq (k + 1)n$. So when applying the algorithm from Theorem 1 for G', this yields an algorithm for solving WSS in time $4^k k \cdot O\big((kn)^3\big) = 4^k \cdot O(k^4) \cdot O(n^3)$. $\square$

We remark that a closer study of the proofs by Chen et al [7] shows that their skew separator algorithm can be modified for the weighted version of the problem as well, which would give a better complexity. This is however beyond the scope of the current paper.

Theorem 5. *In time $O\left((k+1)!k^5 47.5^k\ n^4\right)$, it can be decided whether a mixed graph $G = (V, E, A)$ with $|V| = n$ and integer vertex/edge/arc weights w admits a set $S \subseteq V \cup E \cup A$ such that $G - S$ is acyclic and $w(S) \leq k$.*

Proof. As a first step, we reduce the problem to Weighted FVS, where we search for a FVS S with $w(S) \leq k$. This is done by subdividing all edges and arcs: for every arc (u, v), introduce a new vertex x with weight $w(x) := w((u, v))$, arcs (u, x) and (x, v), and delete arc (u, v). Similarly, replace all edges uv by a vertex x with weight $w(x) := w(uv)$ and two edges ux and xv. Clearly, solving Weighted FVS on the resulting graph is equivalent to solving WFVEAS on the original graph.

For solving Weighted FVS, we can use the algorithms and transformations from Sections 4 and 5: note that since all weights are positive integers, for a set $S \subseteq V(G)$ with $w(S) \leq k$, it holds that $|S| \leq k$. Thus in particular the bounds from Lemma 3 again apply. Hence the entire algorithm and analysis can be applied as previously, with the only modification that only sets S with $w(S) \leq k$ are considered, instead of $|S| \leq k$.

For solving the Weighted Skew Separator problem in the end, we apply Proposition 9. This adds at most a factor $O(k^3)$ to the complexity, compared with Theorem 4. $\square$