Theory Comput Syst (2018) 62:63–92
DOI 10.1007/s00224-017-9805-6

A Randomized Polynomial Kernel for Subset Feedback Vertex Set

Eva-Maria C. Hols[1] · Stefan Kratsch[1]

Published online: 11 September 2017

Abstract The SUBSET FEEDBACK VERTEX SET problem generalizes the classical FEEDBACK VERTEX SET problem and asks, for a given undirected graph $G = (V, E)$, a set $S \subseteq V$, and an integer k, whether there exists a set X of at most k vertices such that no cycle in $G - X$ contains a vertex of S. It was independently shown by Cygan et al. (ICALP '11, SIDMA '13) and Kawarabayashi and Kobayashi (JCTB '12) that SUBSET FEEDBACK VERTEX SET is fixed-parameter tractable for parameter k. Cygan et al. asked whether the problem also admits a polynomial kernelization. We answer the question of Cygan et al. positively by giving a randomized polynomial kernelization for the equivalent version where S is a set of edges. In a first step we show that EDGE SUBSET FEEDBACK VERTEX SET has a randomized polynomial kernel parameterized by $|S| + k$ with $\mathcal{O}(|S|^2 k)$ vertices. For this we use the matroid-based tools of Kratsch and Wahlström (FOCS '12) that for example were used to obtain a polynomial kernel for s-MULTIWAY CUT. Next we present a preprocessing that reduces the given instance (G, S, k) to an equivalent instance (G', S', k') where the size of S' is bounded by $\mathcal{O}(k^4)$. These two results lead to a randomized polynomial kernel for SUBSET FEEDBACK VERTEX SET with $\mathcal{O}(k^9)$ vertices.

Keywords Parameterized complexity · Kernelization · Subset feedback vertex set

This article is part of the Topical Collection on *Theoretical Aspects of Computer Science*

A preliminary version of this work appeared in the proceedings of the 33th International Symposium on Theoretical Aspects of Computer Science (STACS 2016).

✉ Eva-Maria C. Hols
hols@cs.uni-bonn.de

Stefan Kratsch
kratsch@cs.uni-bonn.de

[1] Friedrich-Ebert-Allee, 144 D-53113 Bonn, Germany

1 Introduction

In the SUBSET FEEDBACK VERTEX SET (SUBSET FVS) problem we are given an undirected graph $G = (V, E)$, a set of vertices $S \subseteq V$, and an integer k, and have to determine whether there is a set X of at most k vertices that intersects all cycles that contain at least one vertex of S. Clearly, because we can choose $S = V$, this is a generalization of the well-studied FEEDBACK VERTEX SET (FVS) problem where, given G and k, we have to determine whether some set X of at most k vertices intersects *all cycles* in G. FEEDBACK VERTEX SET has been extensively studied in parameterized complexity: It is known to be fixed-parameter tractable (FPT) with parameter k, i.e., solvable in time $f(k) \cdot |V|^c$, and after a series of improvements the fastest known algorithms take deterministic time $\mathcal{O}^*(3.619^k)$ [11] and randomized time $\mathcal{O}^*(3^k)$ [2]. It is also known to admit a *polynomial kernelization* [1], i.e., there is an efficient algorithm that reduces any instance (G, k) of FVS to an equivalent instance of size polynomial in k; the best known kernelization creates an equivalent instance with $\mathcal{O}(k^2)$ vertices [21].

In 2011, Cygan et al. [3] and Kawarabayashi and Kobayashi [10] independently showed that SUBSET FVS is FPT. The algorithm of Cygan et al. runs in time $2^{\mathcal{O}(k \log k)} n^{\mathcal{O}(1)}$, while the one of Kawarabayashi and Kobayashi runs in time $\mathcal{O}(f(k) \cdot n^2 m)$. Wahlström [23] then gave the first single-exponential algorithm with running time $4^k \cdot n^{\mathcal{O}(1)}$; an algorithm with subexponential dependence on k is ruled out under the Exponential-Time Hypothesis (e.g., because SUBSET FVS generalizes VERTEX COVER). More recently, Lokshtanov et al. [13] gave algorithms with deterministic time $2^{\mathcal{O}(k \log k)} \cdot (n + m)$ and randomized time $\mathcal{O}(25.6^k \cdot (n + m))$.

Cygan et al. [3] ask whether the SUBSET FVS problem also admits a polynomial kernelization and suggest that the matroid-based tools of Kratsch and Wahlström [12] could be applicable. The latter work uses representative sets of independent sets in matroids to obtain, among others, polynomial kernels for s-MULTIWAY CUT and DELETABLE TERMINAL MULTIWAY CUT (DTMWC) with $\mathcal{O}(k^{s+1})$ and $\mathcal{O}(k^3)$ vertices, respectively. In MULTIWAY CUT we are given a graph $G = (V, E)$, a set $T \subseteq V$ of terminals, and an integer k and have to determine whether deletion of at most k non-terminal vertices separates all terminals. In s-MULTIWAY CUT the terminal set has size at most s, and in DTMWC we are also allowed to delete terminals (which is essentially the same as restricting terminals to be degree one).

Interestingly, Cygan et al. [3] claim a polynomial-time reduction from MULTIWAY CUT to SUBSET FVS that does not change the parameter value and, hence, implies that SUBSET FVS is at least as hard as MULTIWAY CUT regarding existence of polynomial kernels.[1] However, the construction given in [3, Section 5] works only as a reduction of DELETABLE TERMINAL MULTIWAY CUT to SUBSET FVS because it is not guaranteed that from a solution to the SUBSET FVS instance one obtains a solution with k deleted *non-terminal* vertices. Since there is presently no such reduction from MULTIWAY CUT to SUBSET FVS, a polynomial kernel for the latter does not imply one for the former. Thus, it makes sense to study kernels for SUBSET FVS before MULTIWAY CUT.

[1] The reduction builds upon a previous one by Even et al. [6] who reduce MULTIWAY CUT to WEIGHTED SUBSET FVS using a single vertex in S that has infinite weight.

Our work We apply the matroid-based tools of Kratsch and Wahlström [12] and develop a randomized polynomial kernelization that reduces instances (G, S, k) of SUBSET FVS to equivalent instances with at most $\mathcal{O}(k^9)$ vertices; this is our main result. Similarly to Cygan et al. [3] we also work on EDGE SUBSET FVS where S is a set of edges of G and X needs to intersect all cycles that contain at least one edge of S; since there are polynomial-time reductions between these two problems that do not change the solution size k, they are equivalent when it comes to kernelization complexity [3]. The result is obtained in two parts.

In the first part (Section 3) we establish a randomized polynomial kernelization for EDGE SUBSET FVS parameterized by $|S| + k$ that reduces to equivalent instances with at most $\mathcal{O}(|S|^2k)$ vertices. Note that nontrivial instances have $k < |S|$ since one could otherwise remove S by deleting one endpoint of each edge in S. Thus, parameterization by $|S|$ suffices, but $\mathcal{O}(|S|^2k)$ gives a tighter overall bound than $\mathcal{O}(|S|^3)$.

At high level, this part is similar to the polynomial kernel for DELETABLE TERMINAL MULTIWAY CUT. We show that certain solutions X, later called *dominant* solutions, allow particular path packings in the underlying graph G. For DELETABLE TERMINAL MULTIWAY CUT this is achieved by a fairly simple replacement argument for solutions X that are not sufficiently well connected to connected components of $G - X$. For EDGE SUBSET FVS the endpoints $T = V(S)$ of edges in S can be regarded as terminals, but this gives a different separation property: A solution X of DELETABLE TERMINAL MULTIWAY CUT generates many connected components, because every connected component in $G-X$ contains at most one terminal. Whereas a solution X of EDGE SUBSET FVS does not need to generate many connected components in $G - X$ since only S-cycles need to be prevented, and thus connected components of $G - X$ may contain many vertices of T. Rather, in $G - X$ there must be a tree-like (or forest-like) structure with components without S-edges playing the role of nodes and with edges given by S. Nevertheless, using the tree-like structure, a replacement argument can be found, implying that dominant solutions must create many components in $(G - X) - S$ containing vertices of T and be well connected to them. This allows to set up a gammoid on $G - S$ with sources T and apply, as in [12], a result of Lovász [14] (made algorithmic by Marx [16]) on representative sets in (linear) matroids that is then guaranteed to generate a superset of X. Randomization is only needed to generate a matrix representation for the gammoid.

In the second part (Section 4) we give a (deterministic) polynomial-time preprocessing that, given an instance (G, S, k) of EDGE SUBSET FVS, returns an equivalent instance (G', S', k') with $k' \leq k$ and $|S'| \in \mathcal{O}(k^4)$. Together with the randomized kernelization from the first part this implies the claimed randomized kernelization to $\mathcal{O}(k^9)$ vertices.

A reduction of the number of S-edges is also a crucial ingredient in the FPT algorithm for EDGE SUBSET FVS by Cygan et al. [3]. They achieve $|S| \in \mathcal{O}(k^3)$, but it is in a slightly more favorable setting: Using iterative compression, it suffices to solve the task of finding a solution X' of size k when given a solution X of size $k+1$. (This is well known in parameterized complexity, and we prefer not to repeat it here [19].) Considering some unknown solution X' of size k, one can guess the intersection D of X' with X, by trying all $\mathcal{O}(2^{k+1})$ possibilities. For the correct guess $D = X' \cap X$, the remaining problem is to find for $(G - D, S \setminus D, k - |D|)$ a solution Z' of size at most $k - |D|$ that is disjoint from $Z = X \setminus D$, since $Z' = X' \setminus D$ would be such a solution; here $S \setminus D$ denotes the set of edges in S with no endpoint in D. Cygan et al. make the nice observation that the guessing also allows to assume that there is no other solution X' with an even larger intersection with X.

In contrast, we cannot afford to run iterative compression for a kernelization to get a starting solution of size $k + 1$ and, as is common, we have to start with an approximate solution Z, which can be assumed to be of size at most $8k$ using an 8-approximation algorithm of Even et al. [7]. The idea of guessing the intersection of an optimal solution with Z is infeasible regarding both time and the number of created instances. Thus, while several structures like z-flowers or disjoint x, y-paths containing S-edges appear in both approaches, many things have to be handled differently. For example, having $k+2$ disjoint x, y-paths containing S-edges for $x, y \in Z$ implies that one of x and y must be in every solution of size k; Cygan et al. can stop here because the solution would not be disjoint from Z; we need to instead store the information about x and y to later detect S-edges that can be safely removed. Like Cygan et al., we also use Gallai's A-path Theorem but we avoid the 2-expansion lemma by using the properties of a blocking set of size at most $2k$ differently. (Such a blocking set can be found if certain flowers of order $k + 1$ do not exist, using Gallai's A-path Theorem.) Cygan et al. compute a blocking set B of size at most $3k$ to find an F-flower of order $|X|$ (with $F \subseteq V$ outer-abundant; see [3, Definition 3.4]) under the assumption that certain F-flowers of order $k+1$ do not exist and they show that there exists a solution that contains X (under the assumption that there exists a solution that is disjoint from F). We cannot assume that our solution is disjoint from F and we have to take another approach. Moreover, we observe that z-flowers can be found via matroid parity on an appropriate gammoid.[2]

2 Preliminaries

We use standard graph notation, mostly following Diestel [5]. All graphs are undirected and may contain multi-edges and loops; accordingly, they may contain cycles of length one and two (formed by loops and multi-edges, respectively). An edge $e \in E$ is called a *bridge* if $(V, E \setminus \{e\})$ has more connected components than G. For a set $X \subseteq V$, let $G[X]$ denote the subgraph of G induced by X and let $N_G(X)$ denote the neighborhood of X in G, i.e., $N_G(X) = \{v \in V \setminus X \mid \exists u \in X \colon \{u, v\} \in E\}$. Given two disjoint sets $X, Y \subseteq V$, by $E(X, Y)$ we denote the set of edges that have one endpoint in X and one endpoint in Y. For a set $E' \subseteq E$ of edges let $V(E')$ be the set of vertices that are incident with at least one edge in E'. For $X \subseteq V$ and $F \subseteq E$ we shorthand $G - X$ for $G[V \setminus X]$ and $G - F$ for $(V(G), E(G) \setminus F)$; if $X = \{x\}$ then we may also write $G - x$ instead of $G - \{x\}$. Note that the graph $(G - X) - F$ is the same graph as the graph $(G - F) - X$ and we will drop the parentheses.

[2] The latter is deterministic by applying a specialized matroid parity algorithm due to Tong et al. [22].

For $A \subseteq V$ a path with endpoints in A and internal vertices not in A is called an *A-path*. The following theorem about A-paths was already used by Cygan et al. [3] for SUBSET FVS and in the quadratic kernelization for FEEDBACK VERTEX SET by Thomassé [21].

Theorem 1 (Gallai [9]) *Let $A \subseteq V$ and $k \in \mathbb{N}$. If the maximum number of vertex-disjoint A-paths is strictly less than $k+1$, then there exists a set $B \subseteq V$ of at most $2k$ vertices that intersect every A-path.*

In particular it is possible to find either $(k+1)$-disjoint A-paths or a set B that intersects all A-paths in polynomial time. This follows from Schrijver's proof of Gallai's theorem [20].

Let (G, S, k) be an instance of the EDGE SUBSET FVS problem. We call a cycle C an *S-cycle*, if at least one edge of S is contained in C. Let x be a vertex of V. A set $\{C_1, C_2, \ldots, C_t\}$ of S-cycles that contain x is called an *x-flower* of order t, if the sets of vertices $C_i \setminus \{x\}$ are pairwise disjoint. Note that if there exists an x-flower of order at least $k+1$, then the vertex x must be in every solution for (G, S, k), if one exists. A set $B \subseteq V \setminus \{x\}$ of size t is called an *x-blocker* of size t, if each S-cycle through x also contains at least one vertex of the set B.

Parameterized complexity A *parameterized problem* is a language $\mathcal{Q} \subseteq \Sigma^* \times \mathbb{N}$, where Σ is any finite set. The second component of an instance (x, k) is called the *parameter*. We say that a parameterized problem $\mathcal{Q}$ is *fixed-parameter tractable* (FPT) if there exists a computable function $f\colon \mathbb{N} \to \mathbb{N}$ and an algorithm A that on input of $(x, k) \in \mathcal{Q} \times \Sigma^*$ takes time at most $f(k) \cdot |x|^{\mathcal{O}(1)}$ and correctly decides whether $(x, k) \in \mathcal{Q}$. A *kernelization* of a parameterized problem $\mathcal{Q}$ is an algorithm K that on input of $(x, k) \in \Sigma^* \times \mathbb{N}$ takes time polynomial in $|x| + k$ and returns an equivalent instance $(x', k') \in \Sigma^* \times k$ with $|x'| + k' \leq h(k)$, where h is a computable function. The function h is called the size of the kernel. We say that K is a *polynomial kernelization* if $h(k) \in \mathcal{O}(k^c)$ for some constant c. The polynomial kernelization obtained in this paper is randomized, which means that there is a small chance for the reduced instance to not be equivalent to the input. The error probability can be made exponentially small in the input size without increasing the size of the kernelization. Similarly to previous work [12], the only source for error is the need to compute a matrix representation for a particular matroid (preliminaries on matroids follow below).

Matroids, gammoids, and representative sets A matroid $M = (U, \mathcal{I})$ consists of a finite set U and a family $\mathcal{I}$ of subsets of U, called *independent sets*, fulfilling the following properties: (i) $\emptyset \in \mathcal{I}$; (ii) if $X \subseteq Y$ and $Y \in \mathcal{I}$ then also $X \in \mathcal{I}$; and (iii) if $X, Y \in \mathcal{I}$ with $|X| < |Y|$ then there exists $y \in Y \setminus X$ such that $X \cup \{y\} \in \mathcal{I}$. The *rank* of of a matroid M, denoted by $r(M)$, is the size of the largest independent set of the matroid M.

Let A be a matrix over an arbitrary field F. Let U be the set of columns of A and let $\mathcal{I}$ be the family of all sets $X \subseteq U$ of columns that are linearly independent over F. Then $M = (U, \mathcal{I})$ is a matroid, called the *linear matroid* or *vector matroid* of A, and we say that A *represents* M. We say that a matroid M is *representable* over a field F, if there exists a matrix A over field F that represents M. If $M = (U, \mathcal{I})$ is representable over some field, then it is also representable by an $r(M) \times |U|$ matrix; by Gaussian elimination we can always reduce a representing matrix for M to one with $r(M)$ many rows (cf. [16]). Let $M_1 = (U_1, \mathcal{I}_1)$ and $M_2 = (U_2, \mathcal{I}_2)$ be two matroids with $U_1 \cap U_2 = \emptyset$. The *direct sum* $M_1 \oplus M_2$ is a matroid over $U = U_1 \cup U_2$ with independent sets $\mathcal{I} = \{X \subseteq U \mid X \cap U_1 \in \mathcal{I}_1, X \cap U_2 \in \mathcal{I}_2\}$. If A_1 and A_2 represent the two matroids over the same field F, then matrix $A = \text{diag}(A_1, A_2) := \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}$ represents $M_1 \oplus M_2$.

Let $G = (V, E)$ be a graph that may have both directed and undirected edges and let $S \subseteq V$. A set $T \subseteq V$ is *linked* to S if there exist $|T|$ vertex-disjoint paths from S to T. Thus every vertex in T is endpoint of a different path from S. It holds that $M = (U, \mathcal{I})$, where $U \subseteq V$ and $\mathcal{I}$ contains all sets $T \subseteq U$ that are linked to S in G, is a matroid [18]. The matroid M is also called the *gammoid* on G with sources S and ground set U; if $U = V$ then M is also called a *strict gammoid*. Marx [16] gave a randomized (Monte Carlo) polynomial-time procedure for finding a matrix representation of a strict gammoid. The error probability can be made exponentially small in the size of the graph. (This is the only source of randomness and error in our kernelization.) A matrix representation for a gammoid for graph $G = (V, E)$ with ground set $U \subsetneq V$ and sources S can be obtained from the strict gammoid for G and S by simply deleting columns corresponding to elements of $V \setminus U$.

Let A, B be independent sets in a matroid. We say that A *extends* B if $A \cap B = \emptyset$ and $A \cup B$ is again an independent set. Note that from the independence of $A \cup B$ follows the independence of A and B due to the second matroid property.

Definition 1 Let $M = (U, \mathcal{I})$ be a matroid, let $\mathcal{A} \subseteq \mathcal{I}$, and let $q \in \mathbb{N}$. A set $\mathcal{A}' \subseteq \mathcal{A}$ is *q-representative* for $\mathcal{A}$ if for every independent set B of size at most q there is a set $A \in \mathcal{A}$ that extends B if and only if there is also a set $A' \in \mathcal{A}'$ that extends B.

Observe that if $\mathcal{A}'$ is q-representative for $\mathcal{A}$ and there exists a set $A \in \mathcal{A}$ that *uniquely extends* some given independent set I of size at most q, then this implies that $A \in \mathcal{A}'$.

The following theorem of Lovász [14] proves that for any linear matroid there exist small representative sets. It was made algorithmic by Marx [16] and, thus, permits to find representative sets in polynomial time when given a matrix representation of the matroid. A faster algorithm for this task was developed recently by Fomin et al. [8].

Lemma 1 (Lovász [14], Marx [16]) *Let M be a linear matroid of rank $q + p$, and let $\mathcal{T} = \{I_1, I_2, \ldots, I_t\}$ be a collection of independent sets, each of size p. If $|\mathcal{T}| > \binom{q+p}{p}$, then there is a set $I \in \mathcal{T}$ such that $\mathcal{T} \setminus \{I\}$ is q-representative for $\mathcal{T}$. Furthermore, given a representation A of M, we can find such a set I in $f(q, p) \cdot (\|A\| t)^{\mathcal{O}(1)}$ time, where $f(q, p)$ is a polynomial in $(p + q)^p$.*

Given a gammoid M we can compute in randomized polynomial-time a representation of the gammoid. Together with Theorem 1 it follows that given a gammoid M

and a collection $\mathcal{T} = \{I_1, \ldots, I_t\}$ of independent sets, each of constant size p, we can find in randomized polynomial time a set $\mathcal{T}' \subseteq \mathcal{T}$ of size at most $\binom{q+p}{p}$ that is q-representative for $\mathcal{T}$.

3 Randomized polynomial kernelization for parameter $|S| + k$

In this section we present a randomized polynomial kernelization for EDGE SUBSET FVS parameterized by $|S| + k$. Because deletion of one endpoint of each edge in S always constitutes a feasible solution, nontrivial instances have $|S| > k$. Thus, our kernelization also works for parameter $|S|$ alone. However, to achieve a better bound for EDGE SUBSET FVS parameterized by k only it is beneficial to give the kernel size in terms of $|S|$ and k rather than $|S|$ alone.

We use representative sets of independent sets of matroids to obtain a kernel of size $\mathcal{O}(|S|^2k)$. Our approach is similar to the kernelization of DELETABLE TERMINAL MULTIWAY CUT(k) [12]. As in that paper we construct path packings such that certain vertices can be shown to be in a representative set. Note that, unlike for multiway cut-type problems, a solution $X \subseteq V$ will not necessarily create many connected components. Rather, as used also in the FPT algorithm of Cygan et al. [3], it creates a particular tree-like structure in $G - X$. Nevertheless, endpoints of edges in S, denoted $T := V(S)$, will play the role of terminals that need to be separated in a certain way; hence a vertex x in T is called a *terminal*. We will focus on the graph $G - S$, i.e., with edges of S deleted, in which a solution X creates a grouping of (not deleted) terminals into connected components. The structure of these components will be crucial for a replacement argument (Lemma 3) that leads to the required path packing; this constitutes one of the key arguments for our result.

The kernelization consists of four steps. In the first step we show that if an instance is a yes-instance then there exists a solution X with a certain path packing from T to X. Then we define an appropriate gammoid to find in a next step a representative set of size $\mathcal{O}(|S|^2k)$ which is (essentially) a superset of X using Lemma 1. Finally we explain how to reduce the graph G, using the superset of the last step, to obtain an equivalent instance of EDGE SUBSET FVS.

Analyzing solutions. Let (G, S, k) be a yes-instance of EDGE SUBSET FVS($k + |S|$). We say that a solution X for (G, S, k) is *dominant*, if it has minimum size and contains a maximal number of vertices from T among solutions of minimum size. The vertices in $X \cap T$ correspond to endpoints of edges in S that we delete and the vertices in $X_0 = X \setminus T$ block all x-y paths with $\{x, y\} \in S_0 = \{e \in S \mid e \cap X = \emptyset\}$, except the one that consists of the edge $\{x, y\}$. We show that X is linked to T in a strong sense, with vertices of X_0 playing a special role.

Lemma 2 *Let X be a dominant solution for (G, S, k) and x any vertex in the set $X_0 = X \setminus T$. There exist $|X| + 2$ paths from T to X in $G - S$ that are vertex-disjoint except for three paths ending in vertex x. Moreover, the paths can be chosen in such a way that each connected component of $G - X - S$ is intersected by at most one path.*

We use Hall's Theorem and the lemma below to prove this. For this purpose we consider the two graphs $G - X$ and $G - X - S$ which simplify the analysis of a dominant solution. We call a connected component K of $G - X - S$ *interesting* if it contains a terminal, i.e., if $T \cap V(K) = (T \setminus X) \cap V(K) \neq \emptyset$, and we say that $x \in X_0$ *sees* a connected component K if x is adjacent to a vertex of K in G. We extend this definition by saying that $Y \subseteq X_0$ sees a component K if at least one vertex $y \in Y$ sees K.

Lemma 3 *Let X be a dominant solution and let $X_0 = X \setminus T$. Every nonempty set $Y \subseteq X_0$ sees at least $|Y| + 2$ interesting components of $G - X - S$.*

Proof Assume for contradiction that there exists a nonempty set $Y \subseteq X_0$ that sees at most $|Y| + 1$ interesting components of $G - X - S$. Let $\mathcal{C}$ denote the set of connected components of $G - X - S$, let $\mathcal{C}_i \subseteq \mathcal{C}$ denote the set of interesting components seen by Y, and let $\mathcal{C}_o \subseteq \mathcal{C}$ denote the other components seen by Y. We will show that there is an alternative solution $X' = (X \setminus Y) \cup Y'$ that is smaller than X or that contains more vertices of T, contradicting the choice of X as a dominant solution. To construct X' or, more precisely, to construct Y' we consider the graphs $G - X$ and $G - (X \setminus Y)$.

We study the structure of $G - (X \setminus Y)$ to find a set Y' that intersects all S-cycles in $G - (X \setminus Y)$. Accordingly, we are interested in the structure that is induced by the S-edges in $G - (X \setminus Y)$ and want to compare it to the structure that is induced by the S-edges in $G - X$.

To this end, we define, for any subgraph $G - Z$ of G with $Z \subseteq V$, the S-component graph H_Z that has a vertex for each connected component of $G - Z - S$ and for every S-edge e an edge between two (not necessary different) vertices which correspond to the connected components that contain the endpoints of e; note that H_Z can have parallel edges and loops. We say that $G - Z$ is an *S-forest* if the S-component graph H_Z is a forest. Observe that a set Z is a solution if and only if $G - Z$ is an S-forest. Note that vertices that correspond to components without terminals in $G - Z$ are isolated in H_Z because they are not incident with S-edges; e.g., this is true in H_X for non-interesting components of $G - X - S$.

Now, we compare the S-component graphs of $G - X$ and $G - (X \setminus Y)$ to construct an alternative solution X' (Fig. 1); let $\mathcal{C}'$ denote the set of connected components of

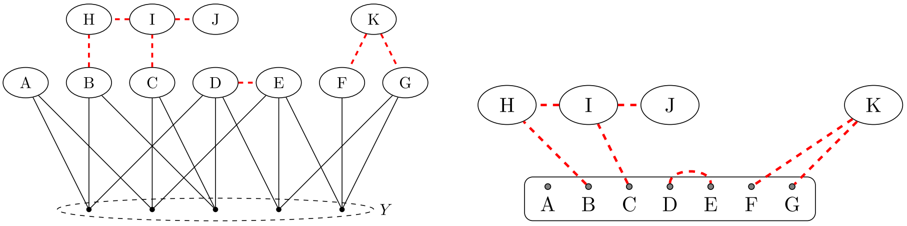

Fig. 1 The left picture shows the connected component C^+, where the ellipse-shaped bubbles represent the connected components of $C^+ - S - Y$. Here $\mathcal{C}_i^* = B, C, D, E, F, G$ and $CinC_0 \mid C \subseteq C^+ = A$. The right picture shows the graph F$'$. The dashed edges (in both pictures) belong to the edges in S

$G - (X \setminus Y) - S$. The connected components of $G - X - S$ that are seen by Y are not connected components in $G - (X \setminus Y)$, because we do not delete Y. Therefore, a connected component in $\mathcal{C}'$ either contains some vertices of Y and some components in $\mathcal{C}_i \cup \mathcal{C}_o$ or is equal to a component in $\mathcal{C} \setminus (\mathcal{C}_i \cup \mathcal{C}_o)$. Thus, the consequence of not deleting Y is that we merge some connected components in $\mathcal{C}_i \cup \mathcal{C}_o$. However, the S-component graphs H_X and $H_{X\setminus Y}$ have the same number of edges, because the sets X and $X \setminus Y$ intersect the same set of S-edges. This follows from the fact that $Y \subseteq X_0 = X \setminus T$ and hence that there are no additional vertices of T, i.e., $T \setminus X = T \setminus (X \setminus Y)$. In general, $G - (X \setminus Y)$ will not be an S-forest: The merging of vertices may lead to loops (from S-edges with both ends in the same component) and longer cycles in $H_{X\setminus Y}$.

We will show that deleting at most $|Y|$ edges of S, i.e., deleting a set Y' of at most $|Y|$ endpoints of S-edges, will suffice for $G - ((X \setminus Y) \cup Y')$ to be an S-forest, making $(X \setminus Y) \cup Y'$ a valid solution.

Let C^+ be an arbitrary connected component in $G - (X \setminus Y)$ whose corresponding connected component in $H_{X\setminus Y}$ is not cycle-free. Note that C^+ is a union of connected components in $\mathcal{C}'$ that are connected by S-edges. Therefore C^+ must contain connected components in $\mathcal{C}$ that are seen by Y. Let $\mathcal{C}_i^* = \{C \in \mathcal{C}_i \mid C \subseteq C^+\}$ be the set of interesting components seen by Y that are contained in C^+, let $a = |\mathcal{C}_i^*|$, and let $b = |\{C \in \mathcal{C}_o \mid C \subseteq C^+\}|$ be the number of non-interesting components seen by Y that are contained in C^+.

In $G - X$ the connected component C^+ may decompose into several separate connected components because we additionally delete the vertices of Y. Since Y sees only components in $\mathcal{C}_i \cup \mathcal{C}_o$ the set C^+ decomposes into at most $a + b$ separate components by deleting Y. Recall that components in $\mathcal{C}_o$ are isolated in $G - X$ and contain no vertices of T and, thus, they do not contribute any S-edges to C^+. It remains to consider the a components of $\mathcal{C}_i^*$.

Every connected component in $\mathcal{C}_i^*$ corresponds to a vertex in the forest H_X. Let F be the subforest of H_X that contains all connected components of H_X that contain at least one connected component of $\mathcal{C}_i^*$ as a vertex. Not deleting Y corresponds to merging a vertices in this forest into $d \geq 1$ new vertices; let F' be the connected subgraph in $H_{X\setminus Y}$ that results from F by this operation. If the subforest F consists of c vertices and, thus, at most $c - 1$ S-edges then we obtain $c - a + d$ vertices that are connected by at most $c - 1$ edges for the subforest F'. It therefore suffices to delete at most $(c - 1) - ((c - a + d) - 1) = a - d \leq a - 1$ S-edges, i.e., to delete one endpoint of each of at most $a - 1$ S-edges, to obtain a forest-structure in F'. (We cannot delete just *any* $a - 1$ edges but we can keep any $c - a + d - 1$ S-edges spanning the $c - a + d$ components and delete the at most $a - 1$ remaining S-edges.)

Overall, we get that a connected component C^+ in $G - (X \setminus Y)$ that fully contains a interesting components from $\mathcal{C}_i$ requires at most $a - 1$ vertex deletions of endpoints of S-edges to obtain an S-forest. Since Y sees at most $|Y| + 1$ such components, the worst case is achieved by a single component C^+ containing all $|Y| + 1$ interesting components in $\mathcal{C}_i$; this still costs at most $(|Y| + 1) - 1 = |Y|$ vertex deletions, as claimed.

Let Y' contain all the endpoints of S-edges that we delete to get an S-forest. We know that $|Y'| \leq |Y|$ and thus $|(X \setminus Y) \cup Y'| \leq |X|$. Moreover, by the initial considerations, we know that $X' = (X \setminus Y) \cup Y'$ is a feasible solution as $G - X'$ is an S-forest. If $|Y'| < |Y|$, including the case that $Y' = \emptyset$, then $|X'| < |X|$ as $Y \neq \emptyset$; this contradicts optimality of X (required for being a dominant solution). If $|Y'| = |Y|$ then $Y' \neq \emptyset$ and X' is an optimal solution that contains more vertices of $T \supseteq Y'$, contradicting the choice of X as a dominant solution. Thus, every nonempty set Y must see at least $|Y| + 2$ interesting connected components, as claimed. □

Now we are ready to give the proof of Lemma 2. The argument relies on Hall's Theorem and is similar to the one for DELETABLE TERMINAL MULTIWAY CUT [12].

Proof of Lemma 2 We know that every nonempty set $Y \subseteq X_0$ sees at least $|Y| + 2$ interesting components of $G - X - S$ by Lemma 3. To prove the existence of the required path packing we construct a bipartite graph where one side consists of the interesting components and the other side consists of the set X_0 and two copies x', x'' of the vertex $x \in X_0$. We connect $v \in X_0$ with an interesting component K if v sees K and we connect x' and x'' with the same interesting components as x. For this bipartite graph it holds that for all sets $Y \subseteq X_0 \cup \{x', x''\}$, the size of $N(Y)$ is at least $|Y|$: This holds trivially for $Y = \emptyset$; assume there exists a nonempty set $Y \subseteq X_0 \cup \{x', x''\}$ such that $|N(Y)| < |Y|$. But then we have $|N(Y \setminus \{x', x''\})| \leq |N(Y)| < |Y| \leq |Y \setminus \{x', x''\}| + 2$, which is a contradiction to Lemma 3.

Since Hall's condition is satisfied there exists a matching M that covers $X_0 \cup \{x', x''\}$. This matching gives rise to a path packing from T to X where exactly three paths end in x and no other vertices occur in more than one path: For each $v \in X \cap T$ pick the path of length zero that consists only of v. For each edge $\{K, v\}$ in the matching M, where $v \in X_0 \cup \{x', x''\}$, pick an arbitrary path from a terminal $t \in V(K) \cap (T \setminus X)$ to v that uses only vertices from $V(K) \cup \{v\}$. (For $v \in \{x', x''\}$ let the path end in x and use only vertices in $V(K) \cup \{x\}$.) Because K is an interesting component a terminal $t \in V(K) \cap (T \setminus X)$ must exist, and because K is a component of $G - X - S$ the path contains no other vertices of X. Similarly, the path cannot contain S-edges between vertices of K, and its final edge to v cannot be in S because $v \in X_0 = X \setminus T$, i.e., because v is not endpoint of any S-edge. Moreover, since each interesting component is matched to a single vertex $v \in X_0 \cup \{x', x''\}$, all the paths are vertex-disjoint except for the three paths that share their endpoint x. This path packing, including the trivial paths from $X \cap T$ to $X \cap T$, contains $|X| + 2$ paths from T to X in $G - S$ that are vertex-disjoint except for the three paths sharing endpoint x. By construction, there is at most one path to any vertex of X_0 starting in any interesting component K of $G - X - S$, because the components are used according to the matching M. All further paths are of length zero, consisting of only a vertex in $X \cap T$ and are, thus, not contained in components of $G - X - S$. □

Setting up the gammoid. The gammoid M that we use is the direct sum of two gammoids M_1 and M_2. To construct gammoid M_1 we define a graph $G_1 = (V_1, E_1)$ that is obtained from $G - S$ by adding two so called *sink-only copies* v' and v'' for every vertex $v \in V$. A sink-only copy of a vertex v is a vertex v' (or v'') that has a directed edge (u, v') for each edge $\{u, v\}$ in $G - S$; these were already used in previous work [12]. Note that adding sink-only copies of vertices does not affect the possible path packings to other vertices since they can only be endpoints of paths; however, they are convenient to capture multiple vertex-disjoint paths that, intuitively, end in the same vertex. The matroid M_1 is defined as the gammoid on G_1 with sources $T = V(S)$ and ground set $V_1 = \{v, v', v'' \mid v \in V\}$; note that the sink-only copies of vertices in T are not sources of M_1. The rank of matroid M_1 is $|T|$, because the set of all trivial paths is independent and at most $|T|$ vertices can be linked to T.

Matroid M_2 is the gammoid on the directed graph $G_2 = K_{k,n} = (S_2 \dot{\cup} \hat{V}, E_2)$ with sources S_2 and ground set $\hat{V} = \{\hat{v} \mid v \in V\}$; the edges in E_2 are directed from S_2 to $\hat{V}$. In other words, gammoid M_2 is a uniform matroid and a (deterministic) matrix representation could also be obtained by using a Vandermonde matrix. The rank of M_2 is $k = |S_2|$ because no more than $|S_2|$ vertices can be linked to S_2 and every set of at most k vertices of $\hat{V}$ is linked to S_2.

For the application of Lemma 1 we will use the matroid $M = M_1 \oplus M_2$, which has rank $|T| + k$. (Matroid M can also be seen as a gammoid on the graph $G_1 \dot{\cup} G_2$ with appropriate sources and ground set but we prefer the explicit direct sum and the implied block-diagonal representation obtained below.) Representations A_1 and A_2 for both M_1 and M_2 can be computed by a randomized polynomial-time algorithm with exponentially small error chance [16]; hence we get a representation for M by $\operatorname{diag}(A_1, A_2)$, i.e., the block-diagonal matrix with blocks A_1 and A_2. We may assume that A_1 has $|T|$ rows and A_2 has k rows since this could be achieved by Gaussian elimination (cf. [16]).

Applying the representative set lemma Let $\mathcal{T} := \{\{v', v'', \hat{v}\} \mid v \in V\}$. For clarity, by the above notation, this means that $v', v'' \in V_1$ and $\hat{v} \in \hat{V}$ for each $v \in V$. Using Lemma 1 we will prove that we can compute in randomized polynomial time a $(|T|+k-3)$-representative subset $\mathcal{T}'$ of $\mathcal{T}$ that contains for all $x \in X_0 = X \setminus T$ the set $\{x', x'', \hat{x}\}$, where X is any dominant solution for (G, S, k). Lemma 1 guarantees that $|\mathcal{T}'| \in \mathcal{O}((|T| + k)^3) = \mathcal{O}(|S|^3)$, since we can compute a matrix representation of M in randomized polynomial-time as described above. We will see later that we can find a $(|T| + k - 3)$-representative set of size $\mathcal{O}(|S|^2 k)$ by a careful look at the proof of Lemma 1, using the fact that the matroid M is the direct sum of two gammoids and that all sets $\{v', v'', \hat{v}\}$ in $\mathcal{T}$ have two elements from the first and one element from the second gammoid; a similar argument for getting a smaller representative set was already used by Kratsch and Wahlström [12].

To ensure that all sets $\{x', x'', \hat{x}\}$ with $x \in X_0$ are in $\mathcal{T}'$ we have to show that for each such set $\{x', x'', \hat{x}\}$ there exists an independent set I of size at most $|T| + k - 3$ such that $\{x', x'', \hat{x}\}$ uniquely extends I among triplets in $\mathcal{T}$. This directly implies that $\{x', x'', \hat{x}\}$ must be in every $(|T| + k - 3)$-representative set $\mathcal{T}'$ of $\mathcal{T}$.

Lemma 4 *Let X be a dominant solution for (G, S, k) and let $T = V(S)$. For all $x \in X_0 = X \setminus T$ there exists an independent set I of size at most $|T| + k - 3$ in M such that $\{x', x'', \hat{x}\}$ uniquely extends I.*

Proof Let x be an arbitrary vertex of X_0. In a first step we define an independent set I and show in a second step that $\{x', x'', \hat{x}\}$ uniquely extends I. Applying Lemma 2 implies the existence of a path packing $\mathcal{P}$ of $|X| + 2$ paths from T to X in $G - S$ that are vertex-disjoint except for three paths ending in x and such that each connected component of $G - X - S$ is intersected by at most one path of $\mathcal{P}$. This directly implies a path packing $\mathcal{P}_1$ in G_1 from T to $X \cup \{x', x''\}$ that is (fully) vertex-disjoint. We retain the property that at most one path intersects the vertex set of any component of $G - X - S$, but note that we do not get exactly the same property for $G_1 - X$ because of the still present sink-only copies of vertices in X. (The latter point will be no problem and should mainly explain why we need to talk about $G - X - S$ and not only G_1. Note that $G - S$ and G_1 by construction share the vertex set V to be able to refer to connected components of $G - X - S$ and the graph G_1 underlying the gammoid M_1.)

While we do not know the paths in $\mathcal{P}_1$ entirely, we know for sure that no vertex of $X \cup \{x', x''\}$ can be an internal vertex of any path in $\mathcal{P}_1$ because there is a path ending in each of those vertices. Similarly, we may assume that no vertex of T is internal to any path of $\mathcal{P}_1$: If not then any path $P \in \mathcal{P}_1$ with internal vertex from T can be shortened to start in that vertex; this argument cannot be repeated indefinitely (as the paths get shorter each time). There is still at most one path intersecting the vertex set of any component of $G - X - S$.

Now, define $T' \subseteq T$ as those vertices of T in which no path of $\mathcal{P}_1$ starts; there must be exactly $|T| - |\mathcal{P}| = |T| - (|X| + 2)$ of them since no vertex of T is internal. Moreover, for each component K of $G - X - S$, the set T' contains all but at most one vertex of $T \cap V(K)$: At most one path of $\mathcal{P}_1$ can start in $T \cap V(K)$ and no vertex can be internal. This will be important for proving the claim below.

Clearly, the set $T' \cup X \cup \{x', x''\}$ is independent in M_1 because an appropriate path packing $\mathcal{P}'$ can be obtained from $\mathcal{P}_1$ by adding length zero paths for each $v \in T'$. The set $\hat{X} = \{\hat{x} \mid x \in X\} \subseteq \hat{V}$ is clearly independent in M_2 since it has size at most k. Thus, the set $I' = T' \cup X \cup \{x', x''\} \cup \hat{X}$ is independent in $M = M_1 \oplus M_2$. Define I as $I' \setminus \{x', x'', \hat{x}\}$, i.e., $I = T' \cup X \cup (\hat{X} \setminus \{\hat{x}\})$. The size of I is at most

$$|T'| + |X| + (|\hat{X}| - 1) = |T| - (|X| + 2) + |X| + |X| - 1 = |T| + |X| - 3 \leq |T| + k - 3.$$

Clearly, $\{x', x'', \hat{x}\}$ extends I, as $I' = \{x', x'', \hat{x}\} \cup I$ is independent and both are disjoint by choice of I. We show that no other $\{v', v'', \hat{v}\} \in \mathcal{T}$ extends I.

Claim If $\{v', v'', \hat{v}\} \in \mathcal{T}$ extends I then $v = x$.

Proof Suppose that $\{v', v'', \hat{v}\}$ extends I. Clearly, this implies that $v \notin X \setminus \{x\}$ because otherwise $\{v', v'', \hat{v}\}$ would not be disjoint from $\hat{X} \setminus \{\hat{x}\} \subseteq I$. Thus, $v \in V \setminus (X \setminus \{x\})$.

Assume, for contradiction, that $v \in V \setminus X$, i.e., that $v \neq x$. We know that $\{v', v'', \hat{v}\} \cup I$ is independent in M, so $I_1 := I \cap V_1$ must be independent in M_1. Thus, there exists a collection $\mathcal{P}''$ of $|I_1|$ vertex-disjoint paths from T to I_1 in G_1. Because $X \subseteq I_1$, the paths, say $P_{v'}$ and $P_{v''}$, from T to $\{v', v''\}$ cannot have internal vertices from the set X. Furthermore, they cannot have other sink-only copies as internal vertices. Since $v \in V \setminus X$, this implies that $P_{v'}$ and $P_{v''}$ are entirely contained in some component K_1 of $G_1 - (X \cup \{x', x'' \mid x \in X\})$. (Component K_1 corresponds to a component K of $G - X - S$ but also has sink-only copies of each vertex.) Recall now that in T' we have all but at most one vertex of $T \cap V(K)$ for each connected component of $G - X - S$ and this is also true for $T \cap V(K_1)$ as $V(K_1) \cap V = V(K)$. Thus, in $\mathcal{P}''$ there is a path w of length zero for each vertex w in $T' \cap V(K_1)$, leaving at most one vertex of T to start paths to $\{v', v''\}$. This is a contradiction because $P_{v'}$ and $P_{v''}$ are entirely contained in K_1 and fully vertex-disjoint.

Thus, if $v \in V \setminus X$ then $\{v', v''\} \cup I_1$ is not independent in M_1 and, hence, $\{v', v'', \hat{v}\}$ does not extend I in M. Together with the first paragraph this implies that $v = x$, as claimed. □

The set I fulfills the required properties which completes the proof. □

We know now that for every vertex $x \in V \setminus T$ that is a vertex in a dominant solution the set $\{x', x'', \hat{x}\}$ is in every $(|T| + k - 3)$-representative set $\mathcal{T}'$. If we define $V(\mathcal{T}') = \{v \mid \{v', v'', \hat{v}\} \in \mathcal{T}'\}$ then this implies that $X_0 \subseteq V(\mathcal{T}')$ for each dominant solution X. Thus, every dominant solution X is contained in $V(\mathcal{T}') \cup T$.

Shrinking the input graph to $|V(\mathcal{T}') \cup T|$ vertices In the previous parts we have shown that if there exists a solution for (G, S, k), then there exists a solution that is completely contained in $W := V(\mathcal{T}') \cup T$. Using this we can make all vertices in $V \setminus W$ undeletable. We achieve this by applying a variant of the torso operation to the vertex set W in G. We construct a graph G' as follows: the graph G' has vertex set W and is derived from $G[W]$ by adding an edge between two vertices $u, v \in W$, if there exists a u, v-path in G with internal vertices from $V \setminus W$ and, if $\{u, v\}$ is either a non-edge in $G[W]$ or an S-edge. To clarify, this allows double edges in G' when one edge is contained in S, but we never create loops. Furthermore, all edges of S are preserved in G' because $T \subseteq W$.

Lemma 5 *(G', S, k) has a solution if and only if (G, S, k) has a solution.*

It follows from Lemma 5 that (G', S, k) is an equivalent instance and the graph of this instance contains at most $|W|$ vertices. This completes the kernelization. The correctness of Lemma 5 follows from the fact that the torso operation preserves the separators that are contained in W (cf. [17]). For completeness we give a short proof of the lemma.

Proof of Lemma 5 Let X be a solution for (G', S, k). We prove that X is also a solution for (G, S, k) by contradiction. Assume that X is not a solution for (G, S, k). Then there exists an S-cycle $C = v_1 v_2 \dots v_l$ in $G - X$. Note that $S \subseteq E(G')$, because $T = V(S) \subseteq W$ and therefore at least two vertices of C are contained in W. Now we modify C to obtain an S-cycle C' in G'. Let $v_i, v_j \in W \cap C$ two vertices of the cycle with $i < j$ such that $\{v_{i+1}, \dots, v_{j-1}\} \subseteq V \setminus W$. By definition there exists an edge

$\{v_i, v_j\}$ in G' and using these edges we obtain cycle C'. Note that C' contains no vertex of X and contains the same edges from S that C contains. Thus C is an S-cycle in $G' - X$ which contradicts the assumption that X is a solution of (G', S, k).

For the other direction we assume that (G, S, k) has a solution. Then there also exists a dominant solution X for (G, S, k) and we know that $X \subseteq W$. Again we prove that X is also a solution for (G', S, k) by contradiction. Assume that X is not a solution for (G', S, k). Then there exists a path P between the endpoints of an edge $e = \{x, y\} \in S$ in $G' - X$ that does not use the edge e. We modify P' to obtain a path P in G that does not contain the edge e. If P' uses an edge $\{u, v\}$ that is not contained in G, then there exists a u-v path in $V \setminus W$ connecting u and v. Crucially, $V \setminus W$ is disjoint from X so this replacement still yields a walk that avoids X. Overall we get a walk from x to y in G that does not contain e as an edge and that avoids X. This walk contains a path P from x to y and this path together with the edge e is an S-cycle in $G - X$ which is a contradiction to the assumption that X is a solution for the instance (G, S, k). □

So far we have a kernelization that creates an equivalent instance (G', S, k) such that G' has $|W|$ vertices. As mentioned above, Lemma 1 guarantees that $|W| \in \mathcal{O}(|S|^3)$ and this implies a polynomial kernel for EDGE SUBSET FVS parameterized by $|S|$. If we use the fact that the gammoid M is the direct sum of two gammoids M_1 and M_2, and that all sets $\{v', v'', \hat{v}\} \in \mathcal{T}$ contain exactly two elements of M_1 and one element of M_2, then we can prove that $|W| \in \mathcal{O}(|S|^2 k)$, which is an improvement for all nontrivial instances with parameter $k < |S|$.

Lemma 6 *Let $M = M_1 \oplus M_2$ be the gammoid of rank $|T| + k$ as defined above and $\mathcal{T} = \{I_1, I_2, \ldots, I_t\}$ be a subset of independent sets of M that we use for the kernelization. Let A be represented by* $\mathrm{diag}(A_1, A_2)$ *as above. If $|\mathcal{T}| > \binom{|T|}{2} \cdot \binom{k}{1}$, then there exists a set $I \in \mathcal{T}$ such that $\mathcal{T} \setminus \{I\}$ is* $(|T| + k - 3)$*-representative for $\mathcal{T}$.*

The proof of Lemma 6 is similar to Marx [16, Lemma 4.2]. We additionally use the fact that M is the direct sum of two gammoids to obtain that the vectors in the exterior algebra which represent the sets in $\mathcal{T}$ span a space of smaller dimension.

Proof of Lemma 6 Let U be the ground set of the matroid M which equals the set of columns of A. For each $e \in U$, let x_e be the corresponding $(|T| + k)$-dimensional column vector of A and let $w_i = \bigwedge_{e \in A_i} x_e$ be a vector in the exterior algebra of the linear space $F^{|T|+k}$. Every w_i is the wedge product of three vectors where exactly two are from $\binom{A_1}{0}$ and one from $\binom{0}{A_2}$. The two vectors corresponding to $\binom{A_1}{0}$ can only span a space of dimension $\binom{|T|}{2}$ and the vectors corresponding to $\binom{0}{A_2}$ can only span a space of dimension $\binom{k}{1}$. Thus, the w_i's span a space of dimension at most $\binom{|T|}{2} \cdot \binom{k}{1}$. If $|\mathcal{T}| > \binom{|T|}{2} \cdot \binom{k}{1}$, then the w_i's are not independent and there exists some vector w_l that can be expressed as a linear combination of the other vectors.

One can show analogously to Marx [16, Lemma 4.2] that $\mathcal{T} \setminus \{I_l\}$ is $(|T| + k - 3)$-representative for $\mathcal{T}$. We replicate this proof for convenience of the reader. Assume that there exists a set Y of size at most $|T|+k-3$ such that I_l extends Y and no other set I_i, $i \neq l$ extends Y. Let $y = \bigwedge_{e \in Y} x_e$. One property of the wedge product is that the product of some vectors in $F^{|T|+k}$ is zero if and only if they are not independent. Therefore it holds that $w_l \wedge y \neq 0$ and $w_i \wedge y = 0$ for every $i \neq l$. But w_l is a linear combination of other w_i's and by the multi-linearity of the wedge product we get that $w_l \wedge y \neq 0$ is a linear combination of the values $w_i \wedge y = 0$ for $i \neq l$, which is a contradiction. □

As mentioned above, Marx [16] showed that one can find in randomized polynomial-time a matrix with $r(M)$ rows that represents a given gammoid M. We can make this proof algorithmic in the same way Marx did [16, Lemma 4.2]. Combined with Lemma 6 it follows directly that we can find a $(|T|+k-3)$-representative subset $\mathcal{T}'$ of $|\mathcal{T}|$ whose size is at most $\binom{|T|}{2} \cdot \binom{k}{1} \in \mathcal{O}(|S|^2 k)$. This leads to a polynomial kernel with $\mathcal{O}(|S|^2 k)$ vertices for EDGE SUBSET FVS parameterized by $|S|$ and k.

4 Reducing the size of S

We have seen that EDGE SUBSET FVS parameterized by $|S|$ and k has a polynomial kernel. Now the goal is to reduce the size of the set S until $|S|$ is polynomially bounded in k. This will lead to a polynomial kernel of EDGE SUBSET FVS parameterized by k.

To begin, we do some initial modifications to ensure that we can always find a solution of size at most k that contains no vertex of the set $V(S)$, if one exists. For this we first delete all vertices $v \in V$ with the property that $e = \{v, v\} \in S$ is a loop in G; since the vertex v must be in any solution, we decrease the value k by one. Next we delete all remaining loops, because these loops are not in S and cannot be contained in any S-cycle. We also reduce the number of edges between two vertices $v, w \in V(G)$. If no edge that is incident with v and w is contained in the set S, then we delete all except one edge. On the other hand, if at least one edge between v and w is contained in S, then we delete all except two edges, so that one of these edges is contained in S and the other not. In the next step we add for every edge $e = \{v, w\} \in S$ two new vertices v_e, u_e to the graph, subdivide the edge e into three edges $\{v, v_e\}, \{v_e, w_e\}, \{w_e, w\}$, and edit S by replacing edge e by the edge $\{v_e, w_e\}$ in S. If a solution X of EDGE SUBSET FVS contains a vertex $x_e \in V(S)$, then we can instead add the vertex x to X and delete x_e from X, because every cycle that contains vertex x_e also contains vertex x; hence we can always find an optimal solution that is disjoint from $V(S)$. After these modifications, we obtain a graph where every endpoint of an S-edge has degree at most two and every vertex is endpoint of at most one S-edge. Note that these two properties are sufficient to ensure that we can always find a solution that is disjoint from $V(S)$. All our reduction rules, except Rule 1, will preserve these two properties, because we never add S-edges to the graph or increase the degree of a vertex; we only delete edges from the set S (not necessarily G) or vertices from G. Note that it is no problem that Rule 1 does not preserve this property, because this rule returns a trivially false instance and ends the kernelization.

Let (G, S, k) be an instance of EDGE SUBSET FVS, such that G is a graph with the above properties. Analogous to the paper of Cygan et al. [3] we consider a solution Z of the EDGE SUBSET FVS, with the difference that our solution is an 8-approximation of the problem, to reduce the size of S. Even et al. [7] show that there exists an 8-approximation algorithm for SUBSET FVS. Since there are polynomial-time target-value preserving reductions between SUBSET FVS and EDGE SUBSET FVS (cf. [3]), we can compute in polynomial time an 8-approximation for EDGE SUBSET FVS and we can assume that $Z \cap V(S) = \emptyset$. If $|Z| > 8k$, then we can stop immediately because no solution of size at most k can exist. On the other hand, if $|Z| \leq k$, then Z is a solution and we are done.

The set Z is a feasible solution to EDGE SUBSET FVS on $(G, S, |Z|)$. This implies that every edge $e \in S$ is a bridge in $G - Z$. In a next step we also remove all edges in S from $G - Z$. Every connected component in $G - Z - S$ contains no edge from S and, following Cygan et al. [3], we call such a component a *bubble*. We denote the set of bubbles by $\mathcal{D}_Z$ and define a graph $H_Z = (\mathcal{D}_Z, E_{\mathcal{D}_Z})$ whose vertices are bubbles and with bubbles I and J being adjacent, i.e., $\{I, J\} \in E_{\mathcal{D}_Z}$, if and only if the components I and J are connected by an edge from S. The graph H_Z is a forest, because Z is a solution for $(G, S, |Z|)$ and a cycle in H_Z would give rise to an S-cycle in $G - Z$. Similarly, no two bubbles can be connected by more than one edge of S. By V_I we denote the vertices that are contained in bubble I. Since $|E(V_I, V_J) \cap S| \leq 1$ for all $I, J \in \mathcal{D}_Z$ and equality holds if and only if $\{I, J\} \in E_{\mathcal{D}_Z}$, we can associate an edge $e = \{I, J\} \in E_{\mathcal{D}_Z}$ with the one edge $e_S = \{v_I, v_J\}$ in $E(V_I, V_J) \cap S$. If we add the vertex set Z and all edges $\{z, I\}$ with the property that $z \in Z, I \in \mathcal{D}_Z$ and $E(z, V_I) \neq \emptyset$ to the graph H_Z we obtain a graph H_Z^+ that contains S-cycles. Note that every S-cycle must contain a vertex of the set Z. We partition the set of bubbles according to the number of bubbles they are connected with.

Definition 2 A bubble $I \in \mathcal{D}_Z$ is called (i) solitary, if $\deg_{H_Z}(I) = 0$; (ii) leaf, if $\deg_{H_Z}(I) = 1$; and (iii) inner, if $\deg_{H_Z}(I) \geq 2$. By $\mathcal{D}_Z^s, \mathcal{D}_Z^l, \mathcal{D}_Z^i$ we denote the corresponding sets of of bubbles.

Let $X \subseteq V \setminus V(S)$ be a superset of Z. We define the graphs H_X, H_X^+ as well as the sets $\mathcal{D}_X$, $E_{\mathcal{D}_X}$ analogously to the graphs H_Z, H_Z^+ and the sets $\mathcal{D}_Z$, $E_{\mathcal{D}_Z}$. Observe that the number of edges in S is equal to the number of edges in the graph H_Z, because Z is a feasible solution to $(G, S, |Z|)$ that is disjoint from $V(S)$ and because two bubbles I, J in the forest H_Z are only connected when there exists an S-edge in $E(V_I, V_J)$. Since H_Z is a forest the number of edges in H_Z is bounded by the number of non-isolated vertices. Thus, the number of edges in S is at most $|\mathcal{D}_Z \setminus \mathcal{D}_Z^s|$.

So far our setup is essentially the same as the one used by Cygan et al. [3]. However, instead of an 8-approximate solution they use the framework of iterative compression, which provides a solution Z of size $k + 1$ and leaves them with the task of reducing the number of S-edges for the problem of finding a solution Z^* that is *disjoint* from Z. Moreover, it suffices for them to consider the case that every feasible solution (if one exists) is disjoint from Z. In this setting they are able to reduce to an equivalent instance (or find that some assumption was violated) with only $\mathcal{O}(k^3)$ edges in S.

Thus, while many relevant structures like z-flowers or parallel x-y paths containing S-edges are the same, many things have to be handled differently. In particular, if we find that at least one out of two vertices $x, y \in Z$ must be in the solution then we cannot stop (like Cygan et al. [3]) but need to continue and use this information in a more direct way. Cygan et al. [3] can stop whenever they find such a pair $\{x, y\}$, because they solve Disjoint Edge Subset FVS during the iterative compression step and are only interested in so called maximal instances where *every* solution of the instance is disjoint from Z.

During the reduction we detect certain pairs $\{x, y\}$ of different vertices with the property that each solution of size at most k must contain at least one of the vertices (if one exists). We store this fact as a *pair-constraint*. We keep and enforce this information in the final instance, unless we decide earlier to delete x or y. By $\mathcal{P}$ we denote the set of pair-constraints that we have found so far. We can interpret this set as a set of edges and by $V(\mathcal{P})$ we denote all vertices that are contained in a pair-constraint. Note that vertices from the set $V(S)$ are never contained in a pair-constraint from $\mathcal{P}$, because there always exists a solution that is disjoint from $V(S)$. We need the set $\mathcal{P}$ to detect edges in S that may be safely deleted. To this end, we generalize the edge subset fvs problem by adding a set of pair-constraints $\mathcal{P}$ to the input; we call this problem pair-constrained edge subset fvs.

pair-constrained edge subset feedback vertex set
Parameter: k
Input: An undirected graph G, a set $S \subseteq E$ of edges, a set $\mathcal{P}$ of pair-constraints with $V(\mathcal{P}) \cap V(S) = \emptyset$ and an integer k.
Question: Does there exist a set $X \subseteq V$ of size at most k such that $G-X$ contains no S-cycle and such that for each pair-constraint $\{x, y\} \in \mathcal{P}$ we have $x \in X$ or $y \in X$?

Clearly, instances (G, S, k) of edge subset fvs and $(G, S, \emptyset, k)$ of pair-constrained edge subset fvs are equivalent. Our goal is to reduce the size of S by detecting S-edges that we can delete from S without changing the outcome. This leads to the following definition:

Definition 3 Let $(G, S, \mathcal{P}, k)$ be an instance of pair-constrained edge subset fvs. We call an edge $e \in S$ *irrelevant*, if $X \subseteq V(G)$ is a solution for $(G, S, \mathcal{P}, k)$ if and only if X is a solution for $(G, S \setminus \{e\}, \mathcal{P}, k)$.

Note that if two different S-edges e and e' are irrelevant in $(G, S, \mathcal{P}, k)$, then e' is not necessarily irrelevant in $(G, S \setminus \{e\}, \mathcal{P}, k)$. In addition we do not expect to find all irrelevant edges or pair-constraints.

The reduction rules We now present our reduction rules. Throughout we assume that always the lowest numbered applicable rule is applied first. Correctness and efficiency of the overall reduction process will be proved later.

Let $(G, S, \mathcal{P} = \emptyset, k)$ be an instance for PAIR-CONSTRAINED EDGE SUBSET FVS and let Z be an 8-approximation of this problem with $k < |Z| \leq 8k$ that is disjoint from $V(S)$. In the following the graphs0 $G - Z$, $G - Z - S$, H_Z, and H_Z^+ are always defined with respect to the current instance $(G, S, \mathcal{P}, k)$ of PAIR-CONSTRAINED EDGE SUBSET FVS. Note that $Z \subseteq V$ and we delete a vertex from Z if we delete the corresponding vertex in V. Furthermore, we delete a pair-constraint $\{x, y\}$ from $\mathcal{P}$ if we delete x or y from G.

Rule 1 If $k < 0$, or if $k = 0$ and there exists an S-cycle, then we reduce $(G, S, \mathcal{P}, k)$ to some trivial false instance, i.e. $G' := (\{x\}, \{e = \{x, x\}\})$, $S' := \{e\}$, $\mathcal{P}' = \emptyset$ and $k' := 0$.

Rule 2 Delete all bridges and all connected components in G not containing any edge from S.

Rule 3 If there exists an edge $e \in S$ such that e is a bridge in $(V, E \setminus (S \setminus \{e\}))$, then we reduce to $S' = S \setminus \{e\}$.

Note that, if Rule 3 deletes an edge e from S then this edge is still contained in G. The Rules 2 and 3 ensure that each bubble $I \in \mathcal{D}_Z$ is adjacent to a vertex in Z in the graph H_Z^+, i.e. for all $I \in \mathcal{D}_Z$ we have $E_{H_Z^+}(I, Z) \neq \emptyset$: Since Rule 2 is not applicable every bubble $I \in \mathcal{D}_Z$ must be adjacent to a bubble $J \in \mathcal{D}_Z \setminus I$, or a vertex in Z; otherwise $G[V_I]$ would be a connected component of G that does not contain any edge from S (V_I was deleted in Rule 2). From Rule 3 follows that a bubble $I \in \mathcal{D}_Z$ must be adjacent to a vertex in Z; otherwise every edge $e \in E_G(V_I, N(V_I)) \cap S$ would be a bridge in $(V, E \setminus (S \setminus \{e\}))$.

Rule 4 If there exists a vertex v in the set $V(\mathcal{P})$ that is contained in at least $k + 1$ pair-constraints of $\mathcal{P}$, then we reduce to $G' = G - v$ and $k' = k - 1$.

Rule 5 If $|\mathcal{P}| > k^2$ (and Rule 4 is not applicable), then we reduce $(G, S, \mathcal{P}, k)$ to some trivial false instance.

Rule 6 If there exists a z-flower of order $k + 1$ in G for a vertex $z \in Z$, then we reduce to $G' := G - z$ and $k' := k - 1$.

For the next rules we need a maximal matching M in H_Z that covers all inner bubbles $\mathcal{D}_Z^i$ in H_Z. We will show later (Lemma 17) that such a matching exists. Note that two adjacent leaf bubbles I_1, I_2 are not adjacent to an inner bubble and form a K_2 in H_Z, hence the edge $\{I_1, I_2\} \in E_{\mathcal{D}_Z}$ is contained in every maximal matching in H_Z. We use this matching to detect pair-constraints in Z. To this end we introduce the following definition: Let $e = \{I, J\}$ be an edge in the matching

M. We say e *sees* the pair $\{x, y\}$ of different vertices $x, y \in Z$ respectively the vertex $x \in Z$, if $\{I, x\}, \{J, y\} \in E(H_Z^+)$ or $\{I, y\}, \{J, x\} \in E(H_Z^+)$ respectively $\{I, x\}, \{J, x\} \in E(H_Z^+)$.

Rule 7 If at least $(k + 2)$ edges in M see a pair $\{x, y\}$ of different vertices in Z, then we add $\{x, y\}$ to the set of pair-constraints $\mathcal{P}$.

Rule 8 If there exists an edge $e \in M$ such that e sees no single vertex $z \in Z$ and for every pair $\{x, y\}$ seen by e the pair $\{x, y\}$ is a pair-constraint in $\mathcal{P}$, then we remove e_S from S and e from M. (Recall: If $e = \{I, J\} \in M \subseteq E(H_Z)$, then e_S is the unique edge in $E(V_I, V_J) \cap S$.)

The matching M is always recomputed if, through application of rules, it does no longer cover every inner bubble or is not maximal when testing whether Rules 7 or 8 apply (i.e., if the preceding rules do not apply). If M does cover all inner bubbles but neither Rule 7 nor 8 applies then, as we will prove later, this implies $|M| \in \mathcal{O}(k^3)$ and, hence, that there are at most $2|M| \in \mathcal{O}(k^3)$ inner bubbles.

Let $L = \mathcal{D}_Z^l \setminus V(M)$ be the set of leaf bubbles that are not covered by M. Because the matching M covers at least all inner bubbles of H_Z, we know that the number of non-isolated vertices in the forest H_Z is at most $2|M| + |L|$. Since the number of edges in a forest is bounded by the number of non-isolated vertices and $|S| = |E_{H_Z}|$, we get $|S| \leq 2|M| + |L|$. Therefore we have to find a reduction rule that reduces the number of leaf bubbles in L. Every leaf bubble in L is adjacent to an inner bubble in H_Z, because M covers all leaf bubbles that are not adjacent to an inner bubble. To bound the number of leaf bubbles in L we define for each $z \in Z$ a graph G_z with the help of the following two sets. The first one, $L_z = N_{H_Z^+}(z) \cap L$, is the set of all unmatched leaf bubbles I that are adjacent to z in H_Z^+. The other $V_z^i = \{v \in V \mid \exists J \in N_{H_Z}(L_z)\colon v \in V_J\}$ consists of all vertices that are contained in an inner bubble that is adjacent to a leaf bubble in L_z (Fig. 2).

$$\begin{aligned} V(G_z) &= \{z\} \cup L_z \cup V_z^i \\ E(G_z) &= E_{H_Z^+}(z, L_z) \cup (E(G[V_z^i]) \setminus S) \\ &\qquad \cup \{\{I, w\} \mid I \in L_z, w \in V_z^i \text{ and } \exists v \in V_I : \{v, w\} \in S\} \end{aligned}$$

In the graph G_z each leaf bubble $I \in L_z$ is a single vertex. We are not interested in the internal structure of leaf bubbles in L_z, whereas we are interested in the structure of the inner bubbles that are adjacent to the leaf bubbles in L_z. Thus we add the connected component that corresponds to an inner bubble which is adjacent to a bubble in L_z to G_z. In order to apply the concept of flowers and blocking sets in G_z, we need to define which edges of G_z are S-edges: An edge $e \in E(G_z)$ is an S-edge if one endpoint of the edge e is contained in L_z and the other is contained in V_z^i. Note that e is an edge in G_z, because there exists an S-edge $e' = \{v, w\}$ in G with $v \in V_I$ and $w \in V_z^i$.

Lemma 7 *If there exists no z-flower of order $k+1$ in G_z for a vertex $z \in Z$, then we can find a z-blocker $B_z \subseteq V_z^i \setminus V(S)$ of size at most $2k$ in G_z.*

The lemma follows from Theorem 1 and the preprocessing as well as the construction of G_z.

Proof of Lemma 7 The maximum number of vertex-disjoint L_z-paths in $G_z - z$ is at most k, otherwise the L_z-paths together with vertex z would correspond to a z-flower of order $k+1$ in G_z; this contradicts the assumption. From Theorem 1 it follows that there exists a set $B_z \subseteq V(G_z - z) = L_z \cup V_z^i$ of size at most $2k$ intersecting every L_z-path. Since every S-cycle through z in G_z must contain an L_z-path, B_z is a z-blocker of size at most $2k$ in G_z.

It remains to show that there exists a z-blocker $B_z \subseteq V_z^i \setminus V(S)$. First we assume that there exists a vertex $I \in B_z \cap L_z$. From the construction of G_z it follows that every leaf bubble has degree one in $G_z - z$. Thus instead of I we can choose the unique vertex in $N_{G_z}(I) \cap V_z^i$ for the z-blocker B_z to obtain that $B_z \subseteq V_z^i$.

In the next step we take care that B_z is also disjoint from $V(S)$. Assume that B_z contains a vertex $v_e \in V(S) \cap V_z^i$. From the preprocessing it follows that we can add $v \in V_z^i \setminus V(S)$ to B_z and delete v_e from B_z, because every cycle that contains v_e also contains v.

Note that we delete at least as many vertices from B_z as we add to B_z, hence B_z is still of size at most $2k$. □

Since no previous rule is applicable and a z-flower of order $k+1$ in G_z gives rise to a z-flower of order $k+1$ in G, we find a z-blocker of size at most $2k$ for every vertex $z \in Z$. Let $B = \bigcup_{z \in Z} B_z$ be the union of all z-blockers B_z of size at most $2k$. The set L is the union of all sets L_z with $z \in Z$, because every leaf bubble is adjacent to a vertex in Z due to Rule 2, hence $L = \bigcup_{z \in Z} L_z$.

The following lemma provides three nice properties of the graph $H_{Z \cup B} = (\mathcal{D}_{Z \cup B}, E_{\mathcal{D}_{Z \cup B}})$ which helps us to bound the number of leaf bubbles in $L \subseteq \mathcal{D}_Z^l$. To memorize: The set $\mathcal{D}_{Z \cup B}$ is the set of bubbles in $G - (Z \cup B) - S$ and two bubbles I, J are adjacent in $H_{Z \cup B}$ if and only if $E(V_I, V_J) \cap S \neq \emptyset$.

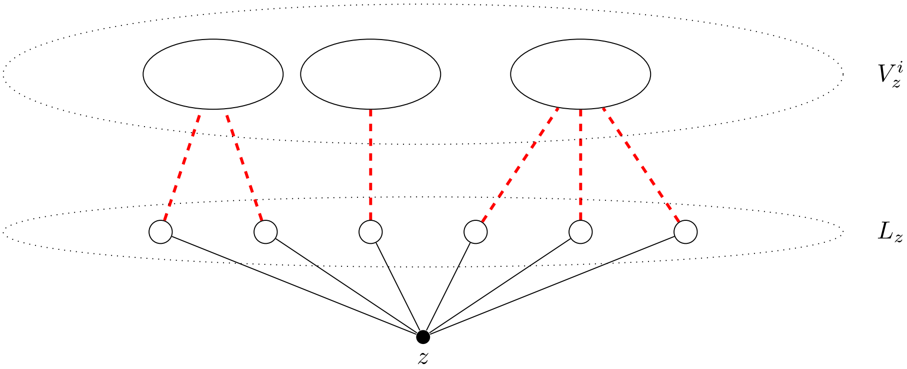

Fig. 2 Graph G_z, where the dashed edges belong to the set S

Lemma 8 *The graph $H_{Z\cup B}$ has the following properties:*

1. *For each bubble $I \in \mathcal{D}_{Z\cup B}$ there exists a bubble $J \in \mathcal{D}_Z$, such that $V_I \subseteq V_J$.*
2. *For each leaf bubble $J \in \mathcal{D}_Z$ there exists a leaf bubble $I \in \mathcal{D}_{Z\cup B}$, such that $V_I = V_J$.*
3. *Let $I, J \in L$ and $K \in \mathcal{D}^i_{Z\cup B}$, such that $\{I, K\}, \{J, K\} \in E_{\mathcal{D}_{Z\cup B}}$. Then for all $z \in Z$ it holds that $z \notin N_G(V_I)$ or $z \notin N_G(V_J)$.*

Proof Property 1 holds because the set B only splits bubbles of $G - Z - S$ further (because we are now looking at deleting $Z \cup B$ from $G - S$ instead of deleting only Z) and thus does not merge any two bubbles. Property 2 follows from the fact that the set B is disjoint from the set of leaf bubbles. Next we show Property 3 by contradiction. We assume that some vertex $z \in Z$ is contained in $N_G(V_I)$ and in $N_G(V_J)$. Then I and J are both vertices of the graph G_z and hence both are contained in the set L_z. The consequence is that there exists an L_z path from bubble I over bubble K to bubble J in $H_{Z\cup B}$ which can be extended to a L_z path in G_z not containing any vertex in B; this contradicts the fact that $B_z \subseteq B$ blocks all L_z-paths in the graph G_z. □

From Lemma 8 it follows that $L \subseteq \mathcal{D}^l_{Z\cup B}$; thus we can use $H_{Z\cup B}$ to bound the number of leaf bubbles in L. Let $\boldsymbol{I} = \{J \in \mathcal{D}^i_{Z\cup B} \mid E(L, J) \neq \emptyset\}$ be the set of inner bubbles in $H_{Z\cup B}$ that are adjacent to a leaf bubble in L. Clearly the number of edges between $\boldsymbol{I}$ and L in $H_{Z\cup B}$ equals the number $|L|$. Instead of again using a matching to reduce this number we consider more carefully the properties of these edges. For this we define the property of seeing a pair in a slightly different way. Let $e = \{I, J\}$ be an edge with $I \in \boldsymbol{I}$ and $J \in L$. We say that $e = \{I, J\}$ with $I \in \boldsymbol{I}$ and $J \in L$ sees the pair $\{x, y\}$ of different vertices $x \in Z \cup B$ and $y \in Z$, if $\{I, x\}, \{J, y\} \in E(H^+_{Z\cup B})$. Observe that a bubble in L is never adjacent to a vertex in B in the graph $H_{Z\cup B}$, because $B \subseteq \bigcup_{z\in Z} V^i_z \setminus V(S)$.

Rule 9 If at least $(k + 2)$ edges $\{I_1, J_1\}, \{I_2, J_2\}, \ldots \{I_l, J_l\}$ with $l \geq k + 2$, $I_i \in \boldsymbol{I}$ and $J_i \in L$ for $1 \leq i \leq l$ see a pair $\{x, y\}$ of different vertices, such that $x \in Z \cup B$ is adjacent to I_i, $y \in Z$ is adjacent to J_i for all $i \in \{1, 2, \ldots, l\}$, then we add $\{x, y\}$ to the set of pair-constraints $\mathcal{P}$.

At first sight Rule 7 and 9 may seem somewhat similar, but on closer inspection one can observe a decisive difference. Obviously, both Rules use the fact that if there are $k + 2$ disjoint S-edges seeing a pair $\{x, y\}$ of different vertices, then either x or y must be in a solution of size at most k; hence it is safe to add the pair $\{x, y\}$ to the set of pair-constraints. But, we need different arguments to show that the $k + 2$ S-edges are disjoint; it is clear in Rule 7, because M is a matching. This is not the case in Rule 9, where we have to use a different argument which follows from Lemma 8 (see Lemma 9).

The difference of these two rules is that in Rule 9 we consider only edges between the two disjoint sets $\boldsymbol{I}$ and L. For this reason we can require in Rule 9 that every endpoint of the at least $k+2$ edges that is contained in $\boldsymbol{I}$ is adjacent to the vertex x and every endpoint that is contained in L is adjacent to vertex y. This is not possible in Rule 7, because an edge in the matching M can be between any type of non-isolated bubbles.

Rule 10 If there exists an edge $e = \{I, J\}$ with $I \in \boldsymbol{I}$ and $J \in L$ such that e sees no single vertex $z \in Z$ and for every pair $\{x, y\}$, with $x \in Z \cup B$, $y \in Z$, seen by e the pair $\{x, y\}$ is a pair-constraint in $\mathcal{P}$, then we remove e_S from S, delete J from L and replace I by $I \cup J$ in $\boldsymbol{I}$. (Recall: e_S is the unique edge in $E(V_I \cap V_J) \cap S$.)

Note that bubbles in L are never adjacent to vertices in B, hence the vertex z resp. the vertex x cannot be contained in set B.

If we delete an edge $e_S \in E(V_I \cap V_J) \cap S$ from S by applying Rule 10, then the consequence is that bubbles I and J are now merged into a single bubble. Observe that it is sufficient to continue with Rule 9, because M is still a matching that covers all inner bubbles in the current graph H_Z and B still has the properties of Lemma 8 with respect to the current graph $H_{Z \cup B}$. That the edge set M is still a matching in H_Z holds because we never delete an edge in M or an endpoint of an edge in M; we only merge an endpoint of an edge in M with an unmatched leaf bubble in L. The first two properties of Lemma 8 obviously hold with respect to the current graph H_Z. That Property 3 also holds follows from the fact that the leaf bubbles that are still in L are the same as before and adjacent to the same inner bubbles as before.

The reduction rules are safe First we show that our reduction rules are safe, i.e. that there exists a solution for $(G, S, \mathcal{P}, k)$ if and only if there exists a solution for $(G', S', \mathcal{P}', k')$. Note that Rules 1, 2, and 6 are obviously safe and Rule 3 is safe because for every S-cycle through an edge $e \in S$ that is a bridge in $(V, E \setminus (S \setminus \{e\}))$ there is another S-edge e' on the cycle. Let us consider the set $\mathcal{P}$ of pair-constraints to see that Rules 4 and 5 are safe. The set $\mathcal{P}$ naturally leads to the graph $P = (V(\mathcal{P}), \mathcal{P})$ and has the property that we have to pick at least on vertex of each pair-constraint for a solution for $(G, S, \mathcal{P}, k)$. Hence any solution for $(G, S, \mathcal{P}, k)$ must contain a vertex cover of P. Thus, Rules 4 and 5 are direct analogs of classical reduction rules for the VERTEX COVER problem, and hence safe. To show that the other rules are safe, we first show a technical Lemma about a property of edges in $H_{Z \cup B}$.

Lemma 9 *If two different edges $\{I_1, J_1\}$ and $\{I_2, J_2\}$ in $H_{Z \cup B}$ with $I_1, I_2 \in \boldsymbol{I}$, $J_1, J_2 \in L$ see a vertex $z \in Z$, respectively a pair $\{x, y\}$ with $x \in Z \cup B$ and $y \in Z$ such that $\{x, I_1\}, \{x, I_2\}, \{y, J_1\}, \{y, J_2\} \in E(H^+_{Z \cup B})$, then it holds that they are disjoint, i.e. that $I_1 \neq I_2$ and $J_1 \neq J_2$.*

Proof Note that if $J_1 = J_2$, then it holds that $I_1 = I_2$, because every leaf bubble in L sees only one other bubble. Thus, to finish the proof it suffices to show that $I_1 \neq I_2$.

Assume for contradiction that $I_1 = I_2$. This implies that J_1 and J_2 are leaf bubbles in L which are adjacent to the same inner bubble $I = I_1 = I_2$ in $H_{Z \cup B}$. For J_1 and

J_2 it must hold that $z \in N_G(V_{J_i})$ respectively $y \in N_G(V_{J_i})$ for $i = 1, 2$. But this is a contradiction to Property 3 of Lemma 8. □

To show that Rules 7 and 9 are safe, we have to prove that we only add a pair $\{x, y\}$ of vertices to the set $\mathcal{P}$ of pair-constraints if either x or y must be in each solution of size at most k. The $(k+2)$ edges that see a pair $\{x, y\}$ are pairwise disjoint, because M is a matching respectively Lemma 9 holds. Hence we have at least $(k+2)$ internally disjoint x-y paths in H_Z^+ respectively $H_{Z\cup B}^+$ which we can extend to at least $(k+2)$ internally disjoint x-y paths in G and each of these paths contains an S-edge. This is the reason why at least one of x and y must be in any solution (otherwise we have to delete at least $k+1$ vertices, one of each path) and it is safe to add $\{x, y\}$ to $\mathcal{P}$ as a pair-constraint.

It remains to show that Rules 8 and 10 are safe. For this we prove that the edges that we delete in these rules are irrelevant. First we prove the following lemma.

Lemma 10 *Let $Y \subseteq V \setminus V(S)$ be a superset of Z, hence $G - Y$ contains no S-cycle. If $e = \{I, J\} \in H_Y$ sees no single vertex $y \in Y$ and for every pair $\{x, y\}$ with $x, y \in Y$ seen by e the pair $\{x, y\}$ is a pair-constraint in $\mathcal{P}$, then the unique edge e_S in $E(V_I, V_J) \cap S$ is irrelevant for the instance $(G, S, \mathcal{P}, k)$.*

Proof Let $e = \{I, J\} \in H_Y$ be an edge with the properties of the lemma and let $e_S = \{v_I, v_J\}$ be the single edge in $E(V_I, V_J) \cap S$. To show that e_S is irrelevant for instance $(G, S, \mathcal{P}, k)$ we have to show that $X \subseteq V(G)$ is a solution for $(G, S, \mathcal{P}, k)$ if and only if X is a solution for $(G, S \setminus \{e_S\}, \mathcal{P}, k)$. Since every solution X for $(G, S, \mathcal{P}, k)$ is also a solution for $(G, S \setminus \{e_S\}, \mathcal{P}, k)$, we only have to show the other direction.

Let X be a solution for $(G, S \setminus \{e_S\}, \mathcal{P}, k)$. We assume for contradiction that there exists an S-cycle C in $G - X$. This S-cycle C can only contain the S-edge e_S; otherwise C would be an $(S \setminus \{e_S\})$-cycle which contradicts the fact that X is a solution for $(G, S \setminus \{e_S\}, \mathcal{P}, k)$.

Claim If an S-cycle C in G only contains the S-edge e_S, then there exists either a vertex $y \in Y$ such that e sees the single vertex y and y is contained in cycle C or two different vertices $x, y \in Y$ such that e sees the pair $\{x, y\}$ and cycle C contains x and y.

Proof Let C be an S-cycle with the properties of the claim. Thus C must exit bubble I and bubble J by edges that end in Y, because this is the only way to obtain a path from v_I to v_J that uses no edge from S. If these two edges share their endpoint y in Y, then e sees the single vertex y and y is contained in C. On the other hand if these two edges have different endpoints x, y in Y, then e sees the pair $\{x, y\}$ and the vertices x, y are contained in C. □

Based on the claim, it follows that edge $e = \{I, J\}$ must see a single vertex $y \in Y$ that is contained in C or a pair $\{x, y\}$ with $x, y \in Y$ such that x and y are contained in

C. However, edge e sees no single vertex and every pair $\{x, y\}$ that is seen by e must be contained in a pair-constraint (this follows from the properties of edge e according to the requirements of the lemma).

Hence the edge e sees only pairs $\{x, y\}$ with $x, y \in Y$ that are contained in the set $\mathcal{P}$. Let $\{x, y\}$ be the pair that is seen by e such that x, y are vertices of cycle C (using the claim). But at least one vertex of the pair $\{x, y\}$ must be in the solution X for $(G, S \setminus \{e_S\}, \mathcal{P}, k)$. Since e sees only pairs that are contained in the set $\mathcal{P}$ of pair-constraints, the cycle C is no cycle in $G - X$. □

From Lemma 10 follows that we only delete an edge e_S in Rule 8 and 10 when e_S is irrelevant for instance $(G, S, \mathcal{P}, k)$; this holds because $Y = Z$ respectively $Y = Z \cup B$ is a superset of Z.

Applying the Rules. First we show that if none of the rules can be applied, then the size of S is bounded by $\mathcal{O}(k^4)$. For this we prove two lemmas. One bounds the size of M which helps us to bound the number of inner bubbles and the other bounds the number of leaf bubbles in L.

Lemma 11 *If the matching M covers all inner bubbles in H_Z and we cannot apply Rules 1 through 8, then the size of M is at most $\mathcal{O}(k^3)$.*

Proof Each edge in M sees either a pair of vertices in Z that do not form a pair-constraint or a single vertex in Z; otherwise we could apply Rule 2, Rule 3 or Rule 8. Recall that Rule 2 and Rule 3 ensure that each bubble is adjacent to a vertex in Z and that Rule 8 deletes an edge e from the matching M (resp. we delete the edge in S, that corresponds to the matching edge, from the set S) when the edge e sees neither a single vertex nor a pair $\{x, y\}$ that is not contained in a pair constraint. The number of pairs in Z is at most $\binom{|Z|}{2} \leq |Z|^2$. Therefore the number of pairs in Z that are not in the set $\mathcal{P}$ of pair-constraints is at most $|Z|^2$. Because we cannot apply Rule 7, at most $(k + 1)$ edges in M see any pair that is not in the set of pair-constraints. Thus at most $(k + 1)|Z|^2$ edges of M can see a pair of vertices in Z that is not in $\mathcal{P}$. The number of edges in M that see a single vertex in Z is at most $k|Z|$; otherwise we can apply Rule 6, because at least one single vertex z in Z is seen by at least $k + 1$ edges from M and these edges together with z are a z-flower of order $k+1$ in H_Z^+ which we can expand to a z-flower of order $k + 1$ in G. Since we cannot apply Rules 6, 7 or 8, this leads to at most $(k + 1)|Z|^2 + k|Z| \in \mathcal{O}(k^3)$ edges in M, because $|Z| \leq 8k$. □

From the lemma it follows that the number of inner bubbles in H_Z is at most $2|M| \in \mathcal{O}(k^3)$.

Lemma 12 *If we cannot apply Rules 1 through 10 then the size of L is bounded by $\mathcal{O}(k^4)$.*

Proof We claim that the number of edges between bubbles in $\boldsymbol{I}$ and bubbles in L is at most $(k+1)|Z|(|B|+|Z|)+k|Z|$, if no rule is applicable. This implies that there are at most $\mathcal{O}(k^4)$ leaf bubbles in L, because $|Z| \leq 8k$ and $|B| \leq 2k|Z|$.

By Rule 10, each edge between a bubble I in $\boldsymbol{I}$ and an unmatched leaf bubble J in L sees a pair $\{x, y\}$, that is not contained in $\mathcal{P}$, meaning that $\{x, I\}, \{y, J\} \in E(H^+_{Z\cup B})$ for $x \in Z \cup B$, $y \in Z$, or sees a single vertex z in Z. The number of pairs in $Z \times (Z \cup B)$ is at most $|Z|(|Z|+|B|)$.

Rule 9 adds $\{x, y\}$ to $\mathcal{P}$ if at least $(k+2)$ edges $\{I_1, J_1\}, \{I_2, J_2\}, \ldots, \{I_l, J_l\}$ with $l \geq k+2$, $I_i \in \boldsymbol{I}$ and $J_i \in L$ for $1 \leq i \leq l$, see the pair $\{x, y\}$ such that $x \in Z \cup B$ is adjacent to I_i and $y \in Z$ is adjacent to J_i for $1 \leq i \leq l$. This bounds the number of edges between vertices in $\boldsymbol{I}$ and L which see a pair, whose vertices are not a pair in the set $\mathcal{P}$ of pair-constraints, by $(k+1)|Z|(|Z|+|B|)$. The number of edges between vertices in $\boldsymbol{I}$ and L that see a certain vertex $z \in Z$ is at most k, otherwise the at least $k+1$ edges between $\boldsymbol{I}$ and L that see vertex z together with vertex z form a z-flower of order $k+1$ in $H^+_{Z\cup B}$, because Lemma 9 ensures that the edges are disjoint. But then we can apply Rule 6 and delete vertex z. Hence at most $k|Z|$ edges between vertices $\boldsymbol{I}$ and L can see a vertex in Z. This leads to at most $(k+1)|Z|(|B|+|Z|)+k|Z|$ edges between vertices in $\boldsymbol{I}$ and L, because we cannot apply Rules 6, 9 or 10; this implies that $|L| \in \mathcal{O}(k^4)$, because $|Z| \leq 8k$ and $|B| \leq 2k|Z| \leq 16k^2$. □

If we combine these two results, we know that $|\mathcal{D}^i_Z| + |\mathcal{D}^l_Z| \in \mathcal{O}(k^4)$. As mentioned above this is an upper bound for the number of edges in S, because H_Z is a forest, because there is at most one edge of S between any two bubbles, and because $V(S) \cap Z = \emptyset$.

Finally we have to prove that we can perform the reduction in polynomial time. First we prove that each rule is applied a polynomial number of times and second that every single rule application can be performed in polynomial time.

Lemma 13 *Each reduction rule is applied at most a number of times that is polynomially bounded in the input size.*

Proof Note that we reduce in each rule, except Rules 7 and 9, the size of at least one of the sets V, E, S, the value k or decide that no solution of size at most k exists. Furthermore, we never increase the size of such sets or the parameter k. In Rules 7 and 9 we add pair-constraints to $\mathcal{P}$, but if $\mathcal{P}$ contains more than k^2 pair-constraints, we either find a vertex $z \in V(\mathcal{P})$ that we delete in Rule 4 and reduce k by one or we decide in Rule 5 that no solution of size at most k exists. This bounds the number of pair-constraints that we add to $\mathcal{P}$ during the reduction by k^3 because we can decrease k at most k times. Thus, each rule is applied at most a number of times that is polynomial in the size of the input. □

Next we show that each single rule application can be performed in polynomial time. It is obvious that we can apply Rules 1 through 5 in polynomial time. The following lemma addresses Rule 6 by solving a matroid parity problem on an appropriate gammoid.

Lemma 14 *Let $G = (V, E)$, $z \in V$, and $S \subseteq E$. A z-flower of maximum order, i.e., a maximum number of S-cycles that intersect only in z, can be found in (deterministic) polynomial time.*

Proof For simplicity, we assume that there are no edges of S incident with z and that no two edges of S are incident with the same vertex of G. If this is not the case, then we subdivide every S-edge $e = \{v, w\}$ into three edges $\{v, v_e\}$, $\{v_e, v_w\}$, $\{v_w, w\}$. In the set S we replace edge e by the edge $\{v_e, w_e\}$. In the resulting graph no S-edge is incident with an original vertex in the graph or to an other S-edge. Subdividing the S-edges like this does not change the maximum order of z-flowers. Furthermore, the set of z-flowers does not change and it is easy to transfer a z-flower in the original graph to one in the new graph and vice versa.

Let $\{C_1, \ldots, C_t\}$ be a z-flower of order t. Each C_i gives rise to a path P_i between two different neighbors u and v of z; all these paths are fully vertex-disjoint. By our above assumption, there are no S-edges incident with z, hence, each P_i must contain two consecutive vertices, say s_i and t_i, with $\{s_i, t_i\} \in S$. In this way, each path P_i can be split into two paths, $P_{i,s}$ and $P_{i,t}$, from $N(v)$ to $\{s_i, t_i\}$; all these $2t$ paths are pairwise vertex-disjoint and do not contain the vertex z. Thus, from any z-flower of order t we get $2t$ vertex-disjoint paths in $G - z$ from $N(z)$ to $T \subseteq V(S)$, i.e., endpoints of S-edges, such that T can be partitioned into t two-sets of vertices that are also edges in S. In the language of gammoids this means that T is an independent set in the gammoid on graph $G - z$, with sources $N(z)$, and ground set $V(S)$.

Conversely, any independent set T in the mentioned gammoid implies the existence of $|T|$ vertex-disjoint paths in $G - z$ from $N(z)$ to T. If, as above, T can be partitioned into edges of S then this gives rise to a z-flower of order $t = |T|/2$: Clearly, $|T|$ must be even to allow for the partition into sets of size two. Moreover, the paths are vertex-disjoint and, thus, two paths from $N(z)$ ending in $\{s_i, t_i\} \in S$ can be combined, using that $\{s_i, t_i\}$ must be an edge of G into a single path, say P_i, from $N(z)$ to $N(z)$ that contains at least one edge of S. Note that, because s_i and t_i are ends of two paths in the packing they cannot occur in any other paths, so this combination still yields vertex-disjoint paths in $G - z$. Finally, adding the vertex z, the paths $P_1, \ldots, P_t$ can be combined into t S-cycles that intersect only in z.

Thus, the task of finding a z-flower of maximum order reduces to that of solving a matroid parity problem on a gammoid: The underlying graph is $G - z$, the source set is $N_G(z)$, the ground set is $V(S)$, and the pairs are given by S. Recall that pairs in S are vertex-disjoint. Using the algorithm due to Lovász [15], one may find a maximum independent set composed of pairs in S in polynomial time, when provided with a matrix representation for the gammoid. A small caveat would be that one would need a randomized algorithm for finding said representation. Conveniently, specialized deterministic algorithms exist for subclasses of linear matroids; we can use a deterministic algorithm due to Tong et al. [22] that solves the problem by reduction to weighted matching on graphs. (Note that given a maximum independent set T composed of pairs, the cycles of the z-flower can be found by computing vertex-disjoint paths from $N(z)$ to T in $G - z$ via a vertex-capacitated flow computation.) □

It remains to show that we can apply Rules 7 through 10 in polynomial time.

Lemma 15 *We can apply Rule 7 and 8 in polynomial time.*

Proof First of all we store for each edge $e = \{I, J\} \in M$ all vertices $z \in Z$ seen by edge e and all pairs $\{x, y\}$ with $x, y \in Z$ seen by edge e. For each edge we need at most $\mathcal{O}(|Z|^2)$ time; we only have to test for each vertex $z \in Z$ respectively each pair $\{x, y\}$ with $x, y \in Z$ whether $\{I, z\}, \{J, z\} \in E(H_Z)$ respectively $\{I, x\}, \{J, y\} \in E(H_Z)$ or $\{I, y\}, \{J, x\} \in E(H_Z)$. Next we count how many edges see a pair $\{x, y\}$ with $x, y \in Z$ and denote this value by $c_{\{x,y\}}$. It takes at most $\mathcal{O}(|E||Z|^2)$ time to compute all values; we only have to count for how many edges we store a certain pair. If a counter $c_{\{x,y\}}$ has value at least $k + 2$, then we add the pair $\{x, y\}$ to the set $\mathcal{P}$ of pair-constraints. We can check this for all counters in $\mathcal{O}(|Z|^2)$ time. The above computation corresponds to the computation we need for Rule 7. To apply Rule 8 we only have to look at all vertices and pairs that we stored for an edge $e \in M$. If we have stored no single vertex and only pairs that are pair-constraints in $\mathcal{P}$, then e fulfills the conditions of an edge that we delete in Rule 8. To check this for one edge takes at most $\mathcal{O}(|Z|^2)$ time. □

We prove that we can apply Rule 9 and 10 in polynomial time similar to how we prove that we can apply Rule 7 and 8 in polynomial time. We only have to remember which endpoint is adjacent to which vertex in a pair.

Lemma 16 *We can apply Rule 9 and 10 in polynomial time.*

Proof First of all we store for each edge $e = \{I, J\}$ with $I \in \boldsymbol{I}$, $J \in L$ all vertices $z \in Z$ seen by edge e and all pairs (x, y) with $x \in Z \cup B$ adjacent to I, $y \in Z$ adjacent to J such that e sees the pair $\{x, y\}$. For each edge $e = \{I, J\}$ with $I \in \boldsymbol{I}$, $J \in L$ we need at most $\mathcal{O}(|Z \cup B||Z|)$ time; we only have to test for each vertex $z \in Z$ respectively each pair (x, y) with $x \in Z \cup B$, $y \in Z$ whether $\{I, z\}, \{J, z\} \in E(H_{Z\cup B})$ respectively $\{I, x\}, \{J, y\} \in E(H_{Z\cup B})$. Next we count for how many edges we stored the pair (x, y) with $x \in Z \cup B$, $y \in Z$ and denote this value by $c_{(x,y)}$. It takes at most $\mathcal{O}(|E||Z \cup B||Z|)$ time to compute all values; we only have to count for how many edges we store a certain pair. If a counter $c_{(x,y)}$ has value at least $k + 2$, then we add the pair $\{x, y\}$ to the set $\mathcal{P}$ of pair-constraints. We can check this for all counters in $\mathcal{O}(|Z \cup B||Z|)$ time. The above computation corresponds to the computation we need for Rule 9, because we only store the pair (x, y) for an edge if the edge sees the pair $\{x, y\}$. To apply Rule 10 we only have to look at all vertices and pairs that we stored for an edge e between bubbles in $\boldsymbol{I}$ and bubbles in L. If we have stored no single vertex and only pairs (x, y) such that $\{x, y\}$ is a pair-constraints in $\mathcal{P}$, then e fulfills the conditions of an edge that we delete in Rule 10. To check this for one edge takes at most $\mathcal{O}(|Z \cup B||Z|)$ time. □

Finally, we show that we can compute the matching M and the set B in polynomial time.

Lemma 17 *We can compute a maximal matching M in H_Z that covers all inner bubbles in polynomial time.*

Proof The graph H_Z is a forest where all inner bubbles have degree greater than one. Thus, it is enough to show that every forest F has a maximal matching that covers all vertices of degree greater that one. We will prove this by induction on the number of edges.

If the forest F has no edges, then $M = \emptyset$ is a valid solution that covers all vertices of degree greater that one. Otherwise, since F is a forest with at least one edge, there exists at least one vertex v of degree one. By induction, $F' = F - v$ has a maximal matching M' that covers all vertices of degree greater than one. Now, if M' covers the unique neighbor u of v, then $M = M'$ is a matching that covers all vertices of degree greater that one in the forest F. Otherwise, the matching $M' = M \cup \{\{u, v\}\}$ is a matching that covers all vertices of degree greater that one in the forest F.

This argument can be easily converted into a recursive algorithm for computing the desired matching in polynomial time. □

It remains to show that we can find B in polynomial time. From Schrijver's proof of Theorem 1 [20] and the proof of Lemma 7 it follows that we can find in polynomial time either a z-flower of order $(k + 1)$ or a z-blocker of size at most $2k$ in G_z. Since there exists no z-flower in G_z when we compute B, we compute for every z exactly once the set B_z and since B is simply the union of all z-blockers we can compute B in polynomial time.

Finding an equivalent instance for Edge Subset Feedback Vertex Set Up to now we can only bound the number of edges in S for the PAIR-CONSTRAINED EDGE SUBSET FVS problem. As mentioned above the instance $(G, S, \mathcal{P} = \emptyset, k)$ for PAIR-CONSTRAINED EDGE SUBSET FVS is equivalent to the instance (G, S, k) of EDGE SUBSET FVS. Therefore we only have to show that we can find in polynomial time an instance of EDGE SUBSET FVS that is equivalent to the instance $(G, S, \mathcal{P}, k)$ of PAIR-CONSTRAINED EDGE SUBSET FVS and has at most $\mathcal{O}(k^4)$ S-edges. Let $\{x, y\} \in \mathcal{P}$ be a pair-constraint. If there are two edges between x and y of which at least one is contained in S, then x or y must be in any solution, because xy is an S-cycle. For this reason, the instance $(G', S' = S \cup \mathcal{P}, k)$ of EDGE SUBSET FVS is equivalent to the instance $(G, S, \mathcal{P}, k)$ of PAIR-CONSTRAINED EDGE SUBSET FVS, where G' is created from G by adding one edge $\{x, y\}$ between every two vertices x and y with $\{x, y\} \in \mathcal{P}$ when $\{x, y\} \notin E$ and by adding an edge $\{x, y\}$ between x and y that is also contained in S'; hence there are two edges between x and y with $\{x, y\} \in \mathcal{P}$ in graph G' and we add exactly one edge between x and y to S'. Because we cannot apply Rule 4 or 5 to $(G, S, \mathcal{P}, k)$, we know that $|\mathcal{P}| \leq k^2$. This leads to a bound of $|S| + |\mathcal{P}| \in \mathcal{O}(k^4)$ edges in S' for the EDGE SUBSET FVS problem after the reduction.

Finally, we combine the results of Sections 3 and 4 to obtain a polynomial kernel for EDGE SUBSET FVS parameterized by k. Let us first make some comments about the reduction of the size of S and the kernelization: For the reduction of the size of S we use the fact that we can always find a solution that is disjoint from $T = V(S)$.

This only holds because we modified the graph accordingly. But since this is a correct reduction it holds that an input instance (G, S, k) of EDGE SUBSET FVS has a solution if and only if the output instance (G', S', k') of the reduction in Section 4 has a solution. Thus it is no problem that we consider dominant solutions for the kernelization in Section 3 and that the kernelization only guarantees the preservation of dominant solutions. Every instance (G', S', k') has a dominant solution of size at most k' when a solution of size at most k' exists; remember that X is a dominant solution for (G', S', k') if it has minimum size and contains a maximal number of vertices from T' among solutions of minimum size. Hence if (G', S', k') has a solution then it has a dominant solution X. Let (G'', S', k') be the output instance of the kernelization in Section 3. From Lemma 5 it follows that (G', S', k') has a solution if and only if (G'', S', k') has a solution.

Summarized, the reduction of the number of edges in S to $\mathcal{O}(k^4)$ edges together with the kernelization to $\mathcal{O}(|S|^2k)$ vertices for EDGE SUBSET FVS parameterized by $|S|$ and k, results in a kernelized instance with $\mathcal{O}(k^9)$ vertices for EDGE SUBSET FVS parameterized by k.

5 Conclusions

We have shown that the SUBSET FVS problem has a randomized polynomial kernelization using the matroid-based tools of Kratsch and Wahlström [12], positively answering the question of Cygan et al. [3]. As in previous work [12] the error-probability can be made exponentially small without increasing the kernel size. Nevertheless, it would of course be very interesting whether the use of randomization and/or matroids can be avoided. Furthermore, there is quite a gap between $\mathcal{O}(k^9)$ vertices and a lower bound of *size* $\mathcal{O}(k^{2-\varepsilon})$ that is inherited from VERTEX COVER [4], conditioned on non-collapse of the polynomial hierarchy.

Other open problems regarding existence of polynomial kernels, possibly amenable to the matroid tools, are MULTIWAY CUT and DIRECTED FEEDBACK VERTEX SET (DFVS). There is also a directed version of SUBSET FVS, called DIRECTED SUBSET FEEDBACK VERTEX SET, but it generalizes DFVS, whose kernel status has remained open for quite some time now.

Acknowledgments We would like to thank the anonymous reviewers for several useful comments on the presentation of our results.

References

1. Burrage, K., Estivill-Castro, V., Fellows, M.R., Langston, M.A., Mac, S., Rosamond, F.A.: The undirected feedback vertex set problem has a poly(k) kernel. In: IWPEC 2006, LNCS, vol. 4169, pp. 192–202. Springer (2006). https://doi.org/10.1007/11847250_18
2. Cygan, M., Nederlof, J., Pilipczuk, M., Pilipczuk, M., Van Rooij, J.M.M., Wojtaszczyk, J.O.: Solving connectivity problems parameterized by treewidth in single exponential time. In: FOCS 2011, pp. 150–159. IEEE Computer Society (2011). https://doi.org/10.1109/FOCS.2011.23

3. Cygan, M., Pilipczuk, M., Pilipczuk, M., Wojtaszczyk, J.O.: Subset feedback vertex set is fixed-parameter tractable. SIAM. J. Discrete Math. **27**(1), 290–309 (2013). https://doi.org/10.1137/110843071
4. Dell, H., van Melkebeek, D.: Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. J. ACM **61**(4), 23:1–23:27 (2014). https://doi.org/10.1145/2629620
5. Diestel, R.: Graph theory (graduate texts in mathematics) (2005)
6. Even, G., Naor, J., Schieber, B., Zosin, L.: Approximating minimum subset feedback sets in undirected graphs with applications. SIAM J. Discrete Math. **13**(2), 255–267 (2000). https://doi.org/10.1137/S0895480195291874
7. Even, G., Naor, J., Zosin, L.: An 8-approximation algorithm for the subset feedback vertex set problem. SIAM J. Comput. **30**(4), 1231–1252 (2000). https://doi.org/10.1137/S0097539798340047
8. Fomin, F.V., Lokshtanov, D., Saurabh, S.: Efficient computation of representative sets with applications in parameterized and exact algorithms. In: SODA 2014, pp. 142–151. SIAM (2014). https://doi.org/10.1137/1.9781611973402.10
9. Gallai, T.: Maximum-minimum sätze und verallgemeinerte faktoren von graphen. Acta Math. Hung. **12**(1-2), 131–173 (1961)
10. Kawarabayashi, K., Kobayashi, Y.: Fixed-parameter tractability for the subset feedback set problem and the S-cycle packing problem. J. Comb. Theory, Ser. B **102**(4), 1020–1034 (2012). https://doi.org/10.1016/j.jctb.2011.12.001
11. Kociumaka, T., Pilipczuk, M.: Faster deterministic feedback vertex set. Inf. Process. Lett. **114**(10), 556–560 (2014). https://doi.org/10.1016/j.ipl.2014.05.001
12. Kratsch, S., Wahlström, M.: Representative sets and irrelevant vertices: New tools for kernelization. In: FOCS 2012, pp. 450–459. IEEE Computer Society (2012). https://doi.org/10.1109/FOCS.2012.46
13. Lokshtanov, D., Ramanujan, M.S., Saurabh, S.: Linear time parameterized algorithms for subset feedback vertex set. In: ICALP 2015, LNCS, vol. 9134, pp. 935–946. Springer (2015). https://doi.org/10.1007/978-3-662-47672-7_76
14. Lovász, L.: Flats in matroids and geometric graphs. In: Combinatorial surveys (Proc. Sixth British Combinatorial Conf., Royal Holloway Coll., Egham, 1977), pp. 45–86. Academic Press, London (1977)
15. Lovász, L.: Matroid matching and some applications. J. Comb. Theory, Ser. B **28**(2), 208–236 (1980). https://doi.org/10.1016/0095-8956(80)90066-0
16. Marx, D.: A parameterized view on matroid optimization problems. Theor. Comput. Sci. **410**(44), 4471–4479 (2009). https://doi.org/10.1016/j.tcs.2009.07.027
17. Marx, D., O'Sullivan, B., Razgon, I.: Finding small separators in linear time via treewidth reduction. ACM Trans. Algorithm. **9**(4), 30 (2013). https://doi.org/10.1145/2500119
18. Perfect, H.: Applications of Menger's graph theorem. J. Math. Anal. Appl. **22**, 96–111 (1968)
19. Reed, B.A., Smith, K., Vetta, A.: Finding odd cycle transversals. Oper. Res. Lett. **32**(4), 299–301 (2004). https://doi.org/10.1016/j.orl.2003.10.009
20. Schrijver, A.: A short proof of Mader's sigma-paths theorem. J. Comb. Theory, Ser. B **82**(2), 319–321 (2001). https://doi.org/10.1006/jctb.2000.2029
21. Thomassé, S.: A $4k^2$ kernel for feedback vertex set. ACM Trans. Algorithm. **6**(2), 32:1–32:8 (2010). https://doi.org/10.1145/1721837.1721848
22. Tong, P., Lawler, E.L., Vazirani, V.V.: Solving the weighted parity problem for gammoids by reduction to graphic matching. Computer Science Division University of California, USA (1982)
23. Wahlström, M.: Half-integrality, lp-branching and FPT algorithms. In: SODA 2014, pp. 1762–1781. SIAM (2014). https://doi.org/10.1137/1.9781611973402.128