

# Towards a Polynomial Kernel for Directed Feedback Vertex Set\*

Benjamin Bergougnoux<sup>1</sup>, Eduard Eiben<sup>2</sup>, Robert Ganian<sup>3</sup>,  
Sebastian Ordyniak<sup>4</sup>, and M. S. Ramanujan<sup>5</sup>

1 Université Clermont Auvergne, LIMOS, CNRS, Aubière, France  
[benjamin.bergougnoux@uca.fr](mailto:benjamin.bergougnoux@uca.fr)

2 Algorithms and Complexity Group, TU Wien, Vienna, Austria  
[eiben@ac.tuwien.ac.at](mailto:eiben@ac.tuwien.ac.at)

3 Algorithms and Complexity Group, TU Wien, Vienna, Austria  
[ganian@ac.tuwien.ac.at](mailto:ganian@ac.tuwien.ac.at)

4 Algorithms and Complexity Group, TU Wien, Vienna, Austria  
[ordyniak@ac.tuwien.ac.at](mailto:ordyniak@ac.tuwien.ac.at)

5 Algorithms and Complexity Group, TU Wien, Vienna, Austria  
[ramanujan@ac.tuwien.ac.at](mailto:ramanujan@ac.tuwien.ac.at)

---

## Abstract

In the DIRECTED FEEDBACK VERTEX SET (DFVS) problem, the input is a directed graph  $D$  and an integer  $k$ . The objective is to determine whether there exists a set of at most  $k$  vertices intersecting every directed cycle of  $D$ . DFVS was shown to be fixed-parameter tractable when parameterized by solution size by Chen, Liu, Lu, O’Sullivan and Razgon [JACM 2008]; since then, the existence of a polynomial kernel for this problem has become one of the largest open problems in the area of parameterized algorithmics.

In this paper, we study DFVS parameterized by the feedback vertex set number of the underlying *undirected graph*. We provide two main contributions: a polynomial kernel for this problem on general instances, and a linear kernel for the case where the input digraph is embeddable on a surface of bounded genus.

**1998 ACM Subject Classification** F.2 Analysis of Algorithms and Problem Complexity

**Keywords and phrases** parameterized algorithms, kernelization, (directed) feedback vertex set

**Digital Object Identifier** 10.4230/LIPIcs.MFCS.2017.36

## 1 Introduction

FEEDBACK SET problems are fundamental combinatorial optimization problems. Typically, in these problems, we are given a graph  $G$  (directed or undirected) and a positive integer  $k$ , and the objective is to select at most  $k$  vertices, edges or arcs to hit all cycles of the input graph. FEEDBACK SET problems are among Karp’s 21 NP-complete problems [27] and have been a topic of active research from algorithmic [1, 2, 3, 6, 7, 8, 9, 10, 13, 14, 20, 23, 28, 25, 30, 32, 37] as well as structural points of view [19, 26, 29, 31, 33, 34, 35]. In particular, such problems constitute one of the most important topics of research in parameterized algorithms [6, 8, 9, 10, 13, 14, 28, 25, 30, 32, 37], spearheading the development of several

---

\* Supported by the French Agency for Research under the GraphEN project (ANR-15-CE-0009) and by the Austrian Science Fund (FWF), projects P26696 and W1255-N23. Robert Ganian is also affiliated with FI MU, Brno, Czech Republic.



new techniques. In this paper, we study the DFVS problem, where the objective is to find a set of  $k$  vertices that intersects all directed cycles in a given digraph.

For over a decade resolving the fixed-parameter tractability of DFVS (whether there is an algorithm running in time  $f(k) \cdot n^{\mathcal{O}(1)}$  for some computable function  $f$ ) was considered the most important open problem in parameterized complexity. In fact, this problem was posed as an open problem in the first few papers on fixed-parameter tractability (FPT) [16, 17]. The problem can be formally stated as follows.

**DIRECTED FEEDBACK VERTEX SET (DFVS)**

*Instance:* A digraph  $D$  and a positive integer  $k$ .

*Parameter:*  $k$

*Question:* Does there exist a vertex subset of size at most  $k$  that intersects every cycle in  $D$ ?

DFVS was shown to be fixed-parameter tractable in a breakthrough paper by Chen, Liu, Lu, O’Sullivan and Razgon [9] in 2008.

One of the most natural follow-up questions once a problem has been classified as fixed-parameter tractable is ‘does it admit a polynomial kernel?’. A *polynomial kernel* is essentially a polynomial-time preprocessing algorithm that transforms the given instance of the problem into an equivalent one whose size is bounded polynomially in the parameter. Following the resolution of the fixed-parameter tractability of DFVS, the question of whether the problem admits a polynomial kernel was raised and has since become one of the main open problems in the area of parameterized complexity.

The most frequently used way of parameterizing problems, like in the case of DFVS, is taking the solution size as the parameter. An alternate method of parameterization is choosing a parameter that never exceeds the solution size but can potentially be much smaller. A classical example of this approach is parameterizing the VERTEX COVER problem by the feedback vertex set (FVS) number. Clearly, the number of vertices required to make a graph acyclic never exceeds the number of vertices needed to make the graph edgeless. On the other hand, the feedback vertex set number could be arbitrarily smaller than the size of the smallest vertex cover. This problem was first studied by Bodlaender and Jansen [24], who showed that VERTEX COVER parameterized by the feedback vertex number has a polynomial kernel. This was later extended by Cygan et al. [12] who systematically studied several generalizations of this problem and obtained several positive as well as negative results with respect to the existence of polynomial kernels.

While this kind of an alternate parameterization is interesting for problems which are known to have polynomial kernels when parameterized by the solution size, it is the *exact opposite* approach which is useful when dealing with problems for which this question has been answered negatively or remains open. This paper deals with such a parameterization for DFVS as an intermediate step towards answering the main question, that of a polynomial kernel for DFVS. To this end, we chose a natural parameter which is never less than the solution size, in particular the feedback vertex set number of the undirected graph underlying the given digraph. The problem we are interested in can formally be stated as follows.

**DIRECTED FEEDBACK VERTEX SET PARAMETERIZED BY FVS (DFVS[FVS])**

*Instance:* A digraph  $D$ , an integer  $p$ , and a set  $F$  such that  $F$  is an UFVS of  $D$ .

*Parameter:*  $|F|$

*Question:* Does there exist a vertex subset of size at most  $p$  that intersects every cycle in  $D$ ?

Since every UFVS (undirected feedback vertex set) of  $D$  is also a DFVS (directed feedback vertex set) of  $D$ , we may assume without loss of generality that  $p \leq |F|$  for every instance of DFVS[FVS]. Furthermore, we use  $k$  to denote the size of  $F$ . Our first result is a polynomial kernel for DFVS[FVS], formally stated below.

► **Theorem 1.** *There is a kernel with  $\mathcal{O}(k^4)$  vertices for DFVS[FVS].*

The overall approach for proving Theorem 1 is inspired by the result of Bodlaender and Dijk [5] on kernelizing the undirected feedback vertex set problem. However, several obstacles needed to be overcome for the techniques to be applicable in the directed setting.

Interestingly, the existence of a polynomial kernel for DFVS parameterized by the solution size remains open even in the restricted setting of planar graphs. While our Theorem 1 naturally also provides a polynomial kernel for DFVS[FVS] on planar graphs, as our second main contribution we show that one can in fact obtain a significantly stronger result not only on planar graphs, but on all graphs embeddable on orientable surfaces.

► **Theorem 2.** *There is a kernel with  $\mathcal{O}(k)$  vertices for DFVS[FVS] when the input digraph is embeddable on a surface of constant genus.*

We note that the existence of such a linear kernel can also be obtained from the Meta-Kernelization framework [4] by observing that DFVS[FVS] has finite integer index. However, the framework is non-constructive; unlike our Theorem 2, it does not provide a concrete kernelization algorithm for the problem.

*Some proofs have been omitted due to space limitation.*

## 2 Preliminaries

**Graphs and Digraphs.** We consider undirected and directed graphs that may contain self-loops and multiple edges and mostly use standard notation that can be found, for instance, in the textbook by Diestel [15]. For an undirected or directed graph  $D$  we denote by  $V(D)$  its vertex set and by  $E(D)$  its edgeset (or arcsset). For a set of vertices  $A \subseteq V(D)$ , we denote by  $D \setminus A$  the (di-)graph obtained from  $D$  after deleting all vertices in  $A$  as well as all arcs/edges incident to some vertex in  $V$ . Moreover,  $D[A]$  denotes the graph  $D \setminus (V(D) \setminus A)$ . We say that a vertex  $u$  is a *neighbor* of a vertex  $v$  in  $D$  if  $\{u, v\} \in E(D)$  (in the case that  $D$  is undirected) or at least one of  $(u, v) \in E(D)$  or  $(v, u) \in E(D)$  holds (in the case that  $D$  is directed). We denote by  $N_D(v)$  (or by  $N(D)$  if  $D$  is clear from the context) the set of all neighbors of  $v$  in  $D$  and refer to  $|N_D(v)|$  as the *degree* of  $v$  in  $D$  (if  $D$  is undirected) or as the *total degree* of  $v$  in  $D$  (if  $D$  is directed). For a set of vertices  $A$ , we denote by  $N_D(A)$  the set  $(\bigcup_{a \in A} N_D(a)) \setminus A$ .

In addition to the above, we also use standard notions such as *in-neighbors* and *out-neighbors*, *paths* and *directed paths* as well as *endpoints* and *internal vertices* of such paths, *contractions*, minors and others. We denote by  $N_D^-(u)$  and  $N_D^+(u)$  the set of all in-neighbors and out-neighbors of  $u$  in  $D$ , respectively; once again, we drop the subscript  $D$  if it can be inferred from the context and for a vertex set  $A \subseteq V(D)$  we write  $N_D^-(A)$  and  $N_D^+(A)$  to denote the sets  $(\bigcup_{a \in A} N_D^-(a)) \setminus A$  and  $(\bigcup_{a \in A} N_D^+(a)) \setminus A$ , respectively. We denote by  $\overline{D}$  the undirected graph obtained from  $D$  after replacing every arc  $(u, v) \in E(D)$  with an edge  $\{u, v\}$ ; if there are arcs in both directions between  $u$  and  $v$ , then we say that there is a *bidirectional arc* between  $u$  and  $v$  and  $\overline{D}$  will contain two parallel  $u$ - $v$  edges.

► **Proposition 3** ([22]). The orientable and nonorientable genus, denoted by  $\gamma$  and  $\tilde{\gamma}$ , of complete bipartite graphs is given by the following formulae:

$$\gamma(K_{m,n}) = \left\lceil \frac{(m-2)(n-2)}{4} \right\rceil, m, n \geq 2; \quad \tilde{\gamma}(K_{m,n}) = \left\lceil \frac{(m-2)(n-2)}{2} \right\rceil, m, n \geq 2.$$

► **Corollary 4.** If  $G$  is a graph such that  $\gamma(G) \leq g$  and  $\tilde{\gamma}(G) \leq h$  for some constants  $g$  and  $h$ , then  $G$  does not contain  $K_{3,4g+3}$  nor  $K_{3,2h+3}$  as a minor.

For a more detailed treatment of topological graph theory the reader is referred to [22].

**Parameterized Algorithms and Kernelization.** For a detailed illustration of the following facts the reader is referred to [11, 18]. A *parameterized problem* is a language  $\Pi \subseteq \Sigma^* \times \mathbb{N}$ , where  $\Sigma$  is a finite alphabet; the second component  $k$  of instances  $(I, k) \in \Sigma^* \times \mathbb{N}$  is called the *parameter*. A parameterized problem  $\Pi$  is *fixed-parameter tractable* if it admits a *fixed-parameter algorithm*, which decides instances  $(I, k)$  of  $\Pi$  in time  $f(k) \cdot |I|^{\mathcal{O}(1)}$  for some computable function  $f$ .

A *kernelization* for a parameterized problem  $\Pi$  is a polynomial-time algorithm that given any instance  $(I, k)$  returns an instance  $(I', k')$  such that  $(I, k) \in \Pi$  if and only if  $(I', k') \in \Pi$  and such that  $|I'| + k' \leq f(k)$  for some computable function  $f$ . The function  $f$  is called the *size* of the kernelization, and we have a polynomial kernelization if  $f(k)$  is polynomially bounded in  $k$ . It is known that a parameterized problem is fixed-parameter tractable if and only if it is decidable and has a kernelization. However, the kernels implied by this fact are usually of superpolynomial size.

A *reduction rule* is an algorithm that takes as input an instance  $I = (D, p, F)$  of DFVS[FVS] and outputs an instance  $I' = (D', p', F')$  of the same problem. We say that the reduction rule is *sound* if  $I$  is a *yes*-instance if and only if  $I'$  is a *yes*-instance. In order to describe our kernelization algorithm, we present a series of reduction rules. We prove the soundness of each reduction rule immediately after presenting its description, unless the soundness is obvious. The reduction rules we present will be executed in the order in which they appear. That is, if at any point we may apply Reduction Rule  $i$  as well as Reduction Rule  $j$  where  $i < j$ , we will execute Reduction Rule  $i$ .

### 3 A Polynomial Kernel for DFVS[FVS]

Note that every FVS of  $\overline{D}$  is also a DFVS of  $D$ . Given an instance  $D$  of DFVS, our kernelization algorithm for DFVS parameterized by FVS first computes a 2-approximate FVS  $S$  of  $\overline{D}$  (using for instance the algorithm given in [1]) and then uses  $S$  to reduce the instance in polynomial-time into an equivalent instance with at most  $\mathcal{O}(|S|^4)$  vertices.

Hence in the following we will assume that  $D$  is a directed graph and  $S$  is a FVS of  $\overline{D}$  of size  $k$ . Our first two reduction rules are sound because (a) neither sinks nor sources can appear on a directed cycle and (b) if a vertex  $v$  has exactly one in-neighbor  $u$  in  $D$  then every directed cycle containing  $v$  has to use the arc  $(u, v)$  (a symmetric statement holds for vertices with exactly one out-neighbor).

► **Reduction Rule 5.** Delete all sources and sinks from  $D$ .

► **Reduction Rule 6.** Let  $l$  be an arbitrary vertex in  $D$ .

- If  $N^+(l) = \{p\}$ , then we contract the arc  $(l, p)$  into a new vertex  $l^*$ .
- If  $N^-(l) = \{p\}$ , then we contract the arc  $(p, l)$  into a new vertex  $l^*$ .

After the exhaustive application of these two rules, we may assume without loss of generality that the digraph  $D$  has no sinks or sources, and that every vertex has at least 2 in-neighbors and at least 2 out-neighbors. We now state one of our main reduction rules.

► **Reduction Rule 7.** Let  $u$  and  $v$  be two (not necessarily distinct) vertices in  $S$  such that there are at least  $k + 1$  internally vertex-disjoint directed  $u$ - $v$  paths in  $D$ . Then,

- if  $u \neq v$  and  $(u, v) \notin E(D)$ , we add an arc from  $u$  to  $v$  to  $D$ , or
- if  $u = v$ , we remove  $u$  from  $D$  and decrease the parameter  $k$  by one.

**Proof of soundness.** Let  $u, v \in S$  be as above and let  $D'$  be the digraph obtained from  $D$  after applying the reduction rule. If  $u = v$  then clearly every DFVS for  $D$  of size at most  $k$  contains  $u$ , which shows the soundness of the reduction rule.

If on the other hand  $u \neq v$ , we will show that a set  $S' \subseteq V(D) = V(D')$  of size at most  $k$  is a DFVS for  $D$  if and only if it is also a DFVS for  $D'$ . The backward direction is trivial because  $D$  is a subgraph of  $D'$ . For the forward direction let  $S'$  be a DFVS for  $D$  of size at most  $k$  and assume for a contradiction that  $S'$  is not a DFVS for  $D'$ . Then  $D' \setminus S'$  contains a directed cycle  $C$  that contains the arc  $(u, v)$ . Hence  $D' \setminus S'$  and thus also  $D \setminus S'$  contains a directed  $v$ - $u$  path  $P$ . Moreover, since  $S'$  has size at most  $k$  and there are at least  $k + 1$  vertex-disjoint directed  $u$ - $v$  paths in  $D$ , we conclude that there is a  $u$ - $v$  path  $P'$  in  $D \setminus S'$ . But then  $P \cup P'$  must contain a directed cycle, which is also a directed cycle in  $D \setminus S'$ , a contradiction to our assumption that  $S'$  is a DFVS of  $D$ . ◀

We will use Reduction Rule 7 to reduce the number of vertices in  $D \setminus S$  that ‘directly contribute’ to (pairs of) vertices in  $S$ . We formalize this idea in the following definition.

► **Definition 8.** Let  $(u, v)$  be an ordered pair of vertices in  $S$ . If  $u \neq v$ , then we refer to  $(u, v)$  as a *potential arc* in  $D[S]$  and if additionally  $(u, v) \notin E(D)$  then we refer to  $(u, v)$  as a *non-arc*. If on the other hand  $u = v$ , then we refer to  $(u, v)$  as a *self-loop*. We say that a vertex  $v \in V(D) \setminus S$  *contributes* to a potential arc or self-loop  $(u, w)$ , if  $(u, v) \in E(D)$  and  $(v, w) \in E(D)$ .

After the exhaustive application of Reduction Rule 7, we have the following structural observation regarding the input.

► **Observation 9.** For every  $u \in S$  there are at most  $k$  internally vertex-disjoint  $u$ - $u$  paths in  $D$ ; moreover for every two distinct vertices  $u$  and  $v$  in  $S$  with  $(u, v) \notin E(D)$ , there are at most  $k$  vertex disjoint  $u$ - $v$  paths in  $D$ . As a result, for every non-arc or self-loop  $(u, v)$ , there are at most  $k$  vertices that contribute to  $(u, v)$ .

Since  $S$  has at most  $k$  vertices and at most  $k(k-1)$  ordered pairs of vertices, Observation 9 implies the following.

► **Observation 10.** There are at most  $k^2(k-1)$  vertices in  $D \setminus S$  that contribute to some non-arc of  $D[S]$ . Moreover, there are at most  $k^2$  vertices in  $D \setminus S$  that contribute to some self-loop of  $D[S]$ .

Our next aim is to bound the number of vertices in  $A = D \setminus S$  in terms of  $k$ . Towards achieving this we will distinguish these vertices in terms of their degree in  $D \setminus S$ . We therefore denote by  $A_0, A_1, A_2$ , and  $A_{\geq 3}$  the sets of all vertices in  $A$  that have total degree 0, 1, 2, and at least 3, respectively, in  $D \setminus S$ .

### 3.1 Bounding $A_0$ , $A_1$ and $A_{\geq 3}$ .

Note that Observation 10 already provides a bound for the number of vertices in  $A_0$  that contribute to some self-loop of  $D[S]$ . Hence, in order to bound  $A_0$ , it is sufficient to provide a bound for the remaining vertices, in the following denoted by  $A'_0$ , in  $A_0$ . In the following let  $v$  be a vertex in  $A'_0$ . Because of Rule 5  $v$  must have at least one in-neighbor and one out-neighbor in  $S$ . Consequently,  $v$  contributes to at least one potential arc of  $D[S]$ .

► **Reduction Rule 11.** If  $v$  does not contribute to a non-arc of  $D[S]$ , then we remove  $v$  from  $D$ .

**Proof of soundness.** Let  $v$  be as above and let  $D'$  be the directed graph obtained from  $D$  after deleting  $v$ . We show that a set  $S' \subseteq V(D)$  is a DFVS for  $D$  if and only if  $S'$  is a DFVS for  $D'$ . The forward direction of this claim is trivial because  $D'$  is a subgraph of  $D$ . Towards showing the backward direction let  $S'$  be a DFVS for  $D'$  and assume for a contradiction that  $S'$  is not a DFVS for  $D$ . Then there must exist a directed cycle  $C$  in  $D \setminus S'$  that contains  $v$  as well as two arcs  $(s, v)$  and  $(v, s')$  for some  $s, s' \in S$ . Because  $v$  does not contribute to a self-loop of  $D[S]$ , we have that  $s \neq s'$ . Because  $v$  does not contribute to a non-arc of  $D[S]$ , it follows that  $(s, s') \in E(D)$ . Hence the arc  $(s, s')$  together with the directed path from  $s'$  to  $s$  contained in  $C$  forms a directed cycle in  $D' \setminus S'$ , a contradiction to our assumption that  $S'$  is a DFVS for  $D'$ . ◀

After the exhaustive application of the above rule, we obtain that  $v$  contributes to some non-arc of  $D[S]$  and hence together with Observation 10, we obtain the following.

► **Observation 12.** There are at most  $k^2(k-1)$  vertices in  $A'_0$ .

**Bounding  $A_1$ .** We now present the reduction rules we use to bound the size of  $A_1$ , i.e., the number of leaves in  $A$ . Again it is sufficient to bound the number of vertices in  $A_1$  that do not contribute to some self-loop of  $D[S]$ , in the following denoted by  $A'_1$ . Namely, we will introduce reduction rule that ensure that every vertex in  $A'_1$  contributes to at least one non-arc in  $D[S]$ . Together with Observation 10 this then bounds the size of  $A'_1$ . Recall that at this point every vertex in  $D$  has at least two in-neighbors and at least two out-neighbors and since moreover every vertex in  $A'_1$  does not contribute to a self-loop, we obtain that every vertex in  $A'_1$  has at least one in-neighbor and at least one out-neighbor in  $S$  that are distinct. Hence we obtain:

► **Observation 13.** Every vertex in  $A'_1$  has at least one in-neighbor and at least one out-neighbor in  $S$  and hence every vertex in  $A'_1$  contributes to a potential arc of  $D[S]$ .

The next reduction rule reduces leaves that do not contribute to a non-arc of  $D[S]$ .

► **Reduction Rule 14.** If  $l \in A'_1$  does not contribute to a non-arc of  $D[S]$ , then:

- if  $l$  is a source in  $D \setminus S$ , then we delete all arcs from  $l$  to vertices in  $S$ ,
- if  $l$  is a sink in  $D \setminus S$ , then we delete all arcs from vertices in  $S$  to  $l$ .

Note that after application of Rule 14,  $l$  will only have either in-neighbors or out-neighbors in  $S$  and can hence be reduced further using Rule 6. Consequently, after the exhaustive application of the above rules, we conclude that every vertex in  $A'_1$  contributes to at least one non-arc of  $D[S]$ . Due to Observation 10 we conclude that there are at most  $k^2(k-1)$  vertices in  $A'_1$ . Finally, since  $\overline{D \setminus S}$  is a forest and the number of vertices of degree at least 3 in a forest is at most equal to the number of leaves minus two, we get the following.

► **Observation 15.** There are at most  $k^3 - 2$  vertices in  $A_{\geq 3}$ .

Note that at this point, we have bounded the size of the sets  $A_0, A_1$  and  $A_{\geq 3}$  and the only set that remains is  $A_2$ .

### 3.2 Bounding $A_2$

Our next aim is to bound the number of vertices in  $A_2$ .

► **Definition 16.** Let  $v$  be a vertex in  $A_2$ . We say that  $v$  is a *sink-vertex* or a *source-vertex* if the two arcs of  $D \setminus S$  incident on it are both incoming arcs or outgoing arcs respectively. Otherwise we say that  $v$  is a *balanced-vertex*.

Note that due to Reduction Rule 6, there are no balanced vertices in  $D \setminus S$  which have *no* neighbors in  $S$ . This is because otherwise, we would have already contracted one of the two arcs incident to  $v$  in  $D \setminus S$ .

Therefore, at this point, we infer the following.

► **Observation 17.** Every vertex in  $A_2$  has at least one neighbor in  $S$ .

► **Definition 18.** Let  $P = (v_1, \dots, v_r)$  be a directed path of maximum length in  $D \setminus S$  whose internal vertices are in  $A_2$ . Then we say that  $P$  is a *path segment* in  $D \setminus S$ . We say that  $P$  is an *outer path segment* if at least one of its endpoints is not in  $A_2$ , otherwise we say that  $P$  is an *inner path segment*.

Note that path segments are by definition *directed paths*. Our strategy now is to obtain a bound on the total number of path segments and then proceed to bound the length of each path segment. We first bound the number of outer path segments in  $D \setminus S$  as follows. Let  $G$  be the undirected graph obtained from  $\overline{D} \setminus S$  after contracting all edges which are incident to at least one vertex of degree 2. Then the number of outer path segments in  $D \setminus S$  is equal to two times the number of edges of  $G$ . Because  $G$  is a forest without degree two vertices it holds that the number of edges of  $G$  is equal to the number of leaves plus the number of non-leaves in  $G$  minus one. Hence the number of outer path segments is at most  $2(|A_1 \cup A_{\geq 3}| - 1)$ , which together with the already obtained bound on these sets and Observation 15 allows us to infer the following.

► **Observation 19.** The number of outer path segments in  $D \setminus S$  is at most  $4k^2(k-1) + 2k^2 - 6$ .

In order to bound the number of inner path segments, we need to introduce a new reduction rule. We begin by defining the notion of a path segment ‘contributing’ to a potential arc.

► **Definition 20.** We say that a path segment  $P = (v_1, \dots, v_r)$  *contributes* to a potential arc  $(s, s')$  of  $D[S]$  if there are  $i$  and  $j$  with  $1 \leq i \leq j \leq r$  such that  $(s, v_i) \in E(D)$  and  $(v_j, s') \in E(D)$ . Moreover, we say that  $P$  contributes to a self-loop if there are  $i$  and  $j$  with  $1 \leq i \leq j \leq r$  such that  $(s, v_i) \in E(D)$  and  $(v_j, s) \in E(D)$  for some  $s \in S$ .

► **Reduction Rule 21.** If an inner path segment does not contribute to a non-arc or to a self-loop of  $D[S]$ , then we remove all internal vertices of  $P$ .

After the exhaustive application of the above rule, we obtain:

► **Observation 22.** Every inner path segment contributes to at least one non-arc or self-loop of  $D[S]$ .

Because every pair of inner path segments that contribute to some non-arc or self-loop  $(s, s')$  of  $D[S]$  increase the number of disjoint paths between  $s$  and  $s'$  in  $D$  by at least one, Observation 9 implies that for every non-arc or self-loop  $(s, s')$  of  $D[S]$  there are at most  $2k$  inner path segments that contribute to  $(s, s')$ . Finally, because  $S$  has at most  $k$  vertices and at most  $k(k-1)$  ordered pairs of vertices, we conclude that there are at most  $2k^2(k-1) + 2k^2$  inner path segments in  $D \setminus S$ . Having obtained a bound on the number of inner path segments too, we conclude the following.

► **Observation 23.** The number of path segments in  $D \setminus S$  is at most  $6k^2(k-1) + 4k^2 - 6$ .

Our next aim is to provide a bound on the overall length of path segments and use it to bound the size of  $A_2$ . Towards this aim we introduce reduction rules that allow us to bound the in-degree and the out-degree w.r.t.  $S$  of any vertex occurring internally in path segments.

► **Definition 24.** Let  $s \in S$  and let  $P = (v_1, \dots, v_r)$  be an induced directed path in  $D \setminus S$ , whose internal vertices are in  $A_2$  and that satisfies:

- $(s, v_1) \in E(D)$  and  $(s, v_r) \in E(D)$  and  $v_1$  is a balanced vertex in  $A_2$ ,
- for every  $i$  with  $1 < i < r$ , it holds that  $(s, v_i) \notin E(D)$ .

If  $P$  satisfies the above properties we call  $P$  an *out-segment* for  $s$ . We say that  $P$  *contributes* to a potential arc or self-loop  $(s, s')$  in  $D[S]$  if there is an index  $i$  with  $1 \leq i < r$  such that  $(v_i, s') \in E(D)$  for some  $s' \in S$ .

We now introduce a reduction rule that allows us to preprocess and reduce certain out-segments.

► **Reduction Rule 25.** Let  $s \in S$  and let  $P = (v_1, \dots, v_r)$  be an out-segment for  $s$ . If  $P$  does not contribute to any non-arc or self-loop of  $D[S]$ , then we remove the arc  $(s, v_1)$ .

After the exhaustive application of the above rule, we obtain the following.

► **Observation 26.** For each  $s \in S$ , there are at most  $k^2$  out-segments for  $s$ .

**Proof.** Since Rule 25 does not apply, every out-segment for  $s$  contributes to at least one non-arc or self-loop of  $D[S]$ . Furthermore, every out-segment for  $s$  that contributes to some non-arc or self-loop  $(s, s')$  of  $D[S]$  increases the number of internally vertex-disjoint paths  $s$ - $s'$  paths in  $D$  by one. Observation 9 implies that for every non-arc or self-loop  $(s, s')$  of  $D[S]$ , there are at most  $k$  out-segments for  $s$  in  $D \setminus (S \cup B)$  that contribute to  $(s, s')$ .

Finally, because every vertex  $s$  is contained in at most a single self-loop and in at most  $k-1$  non-arcs of  $D[S]$ , we infer that there are at most  $k(k-1) + k$  out-segments for  $s$ . This completes the proof of the observation. ◀

We are now ready to bound the size of the set  $A_2$ .

► **Observation 27.** The number of vertices in  $A_2$  is at most  $12k^4 - 2k^3 - 12k$ .

**Proof.** We begin by arguing that for every  $s \in S$ ,  $s$  has at most  $2k \cdot (6k^2(k-1) + 4k^2 - 6 + k(k-1) + k)$  neighbors in  $A_2$ . Since the number of out-neighbors of  $s$  in  $A_2$  is at most the number of path segments (bounded by Observation 23) plus the number of out-segments for  $s$  (bounded by Observation 26), we obtain an upper bound of  $6k^2(k-1) + 4k^2 - 6 + k^2$  on the size of the out-neighborhood (and by symmetry, the in-neighborhood) of  $s$  in  $A_2$ .

Consequently, the total number of neighbors of vertices in  $S$  to vertices in  $A_2$  and thus (because of Observation 17) the total number of vertices in  $A_2$  is at most  $2k \cdot (6k^2(k-1) + 4k^2 - 6 + k^2)$ . Finally, since the size of  $S$  is bounded by  $k$ , the observation follows. ◀



We are now ready to prove a bound on the size of the kernel. Recall that thus far we have obtained the following bounds:

- there are at most  $k^2$  vertices in  $D \setminus S$  contributing to a self-loop in  $D[S]$  (Observation 10),
- $|A'_0| \leq k^2(k-1)$  (Observation 12),
- $|A'_1| \leq k^2(k-1)$  (see paragraph before Observation 15),
- $|A_2| \leq 12k^4 - 2k^3 - 12k$  (Observation 27),
- $|A_{\geq 3}| \leq k^3 - 2$  (Observation 15),

It follows that the total number of vertices in the reduced graph is at most  $k^2 + |A'_0 \cup A'_1 \cup A_2 \cup A_{\geq 3} \cup S|$  which is at most  $k^2 + 2k^2(k-1) + k^3 - 2 + 12k^4 - 2k^3 - 12k + k$ , thus proving Theorem 1. This completes the description of our kernel for general instances of the problem; we now proceed to the linear kernel on graphs of bounded genus.

#### 4 A Linear Kernel for DFVS[FVS] on Bounded Genus graphs

Throughout this section we will use  $D$  to denote a directed graph of genus at most some fixed bound  $g$ . We let  $S$  be a feedback vertex set of  $\overline{D}$  and let  $c$  be a constant such that  $\overline{D}$  is a  $K_{3,c}$ -minor-free graph, where  $c$  depends only on  $g$  as per Corollary 4. We begin with the following lemma, which follows directly from [21, Lemma 4.3].

► **Lemma 28.** *Let  $G = (X, Y, E)$  be a bipartite graph and  $c$  a constant such that  $G$  is  $K_{3,c}$ -minor-free. Then,*

- *there are  $\mathcal{O}(|X|)$  subsets  $X' \subseteq X$  such that  $X' = N(u)$  for some  $u \in Y$  and*
- *for any subset  $X' \subseteq X$  such that  $|X'| \geq 3$ , the set  $Y' = \{y \in Y : N(y) \supseteq X'\}$  has size at most  $c - 1$ .*

We mainly use this lemma to bound the number of connected components of  $\overline{D} \setminus S$ , it gives directly a bound on the number of connected components with at least 3 neighbors in  $S$ , for the connected components with at most 2 neighbors in  $S$ , we need to introduce some new reduction rules. We will need a few additional notions to provide a concise presentation of the results in this section. A digraph  $H$  is called a *road* iff  $\overline{H}$  is a path; the first and last vertex on a road are called its *endpoints*, and all other vertices on a road are called *internal* vertices. Moreover, for a directed graph  $G$  consider a connected component of  $\overline{G}$  with vertex set  $A$  such that  $G[A]$  is acyclic. Then  $G[A]$  (and, equivalently, the set  $A$ ) is called an *acyclic component* of  $G$ . Observe that there is a one-to-one correspondence between connected components of  $\overline{D} \setminus S$  and acyclic components of  $D \setminus S$ .

For each distinct  $x, y \in S$ , we denote by  $\mathcal{C}_{x,y}$  the set of all acyclic components  $C$  of  $D \setminus S$  with  $N(C) = \{x, y\}$ . Finally, we use  $\mathcal{C}_{x,y}^{\rightarrow}$  to denote the subset of  $\mathcal{C}_{x,y}$  of components  $C$  with the property that  $D[C \cup \{x, y\}]$ :

- contains a directed path from  $x$  to  $y$ , but
- contains neither an  $x$ - $x$  directed path nor a  $y$ - $y$  directed path intersecting  $C$ .

Observe that any road within  $D$  that is disjoint from a feedback vertex set of  $\overline{D}$  may contain exactly one arc between two adjacent vertices. Furthermore, in any instance where Reduction Rule 5 is not applicable, the input digraph  $D$  itself does not contain an acyclic component. That is, every connected component of  $\overline{D}$  contains a directed cycle.

► **Observation 29.** If Reduction Rule 5 is not applicable and  $C$  is an acyclic component of  $D \setminus S$ , then there is a directed  $N(C)$ - $N(C)$  path in  $D[C \cup N(C)]$  containing at least one vertex of  $C$ .

Crucially, Observation 29 implies that for each component  $C$  in  $\mathcal{C}_{x,y}$ , either  $D[C \cup \{x\}]$  (or  $D[C \cup \{y\}]$  by symmetry) contains a cycle, or  $C \in \mathcal{C}_{x,y}^{\rightarrow} \cup \mathcal{C}_{y,x}^{\rightarrow}$ .

► **Reduction Rule 30.** If  $C$  is an acyclic component of  $D \setminus S$  where  $N_{\overline{D}}(C) = \{x\}$  for some  $x \in S$  and  $C \cup \{x\}$  contains a cycle, then we remove  $D[C \cup \{x\}]$  from  $D$  and reduce  $k$  by 1.

The soundness of the above rule follows from the fact that any DFVS which does not contain  $x$  must necessarily intersect  $C$ , and hence there also exists a DFVS of at most the same size which contains  $x$  but does not intersect  $C$ .

By expanding the above argument, we observe that if there exists a DFVS  $T$  containing at least two vertices from  $\mathcal{C}_{x,y} \cup \{x, y\}$ , then the set  $T' = (T \setminus \mathcal{C}_{x,y}) \cup \{x, y\}$  is also clearly a solution of at most the same size as  $T$ . Hence every minimum DFVS contains at most two vertices from  $\mathcal{C}_{x,y} \cup \{x, y\}$ . Consequently, if we have a minimum DFVS  $T$  and  $C_1, C_2, C_3 \in \mathcal{C}_{x,y}$ , then at least one of  $C_1, C_2$  or  $C_3$  has an empty intersection with  $T$ . The soundness of the following three Reduction Rules follows.

► **Reduction Rule 31.** If  $\mathcal{C}_{x,y}$  contains at least 3 acyclic components  $C_1, C_2, C_3$  such that  $D[C_1 \cup x]$ ,  $D[C_2 \cup x]$  and  $D[C_3 \cup x]$  each contains a cycle, we remove  $x$  and decrease  $k$  by 1.

► **Reduction Rule 32.** If  $\mathcal{C}_{x,y}^{\rightarrow} \cap \mathcal{C}_{y,x}^{\rightarrow}$  contains at least 4 components, then we remove all components of  $\mathcal{C}_{x,y}^{\rightarrow} \cup \mathcal{C}_{y,x}^{\rightarrow}$  from  $D$  and add the arcs  $(x, y)$  and  $(y, x)$  to  $D$ .

► **Reduction Rule 33.** If  $\mathcal{C}_{x,y}^{\rightarrow}$  contains at least 3 components and  $\mathcal{C}_{x,y}^{\rightarrow} \setminus \mathcal{C}_{y,x}^{\rightarrow}$  is not empty, then we remove all components of  $\mathcal{C}_{x,y}^{\rightarrow} \setminus \mathcal{C}_{y,x}^{\rightarrow}$  from  $D$  and add the arc  $(x, y)$  to  $D$ .

► **Lemma 34.** *After applying Reduction Rules 5 to 6 and Reduction Rules 30 to 33, the resulting digraph is also  $K_{3,c}$ -minor-free.*

**Proof.** Since Reduction Rule 5, 30 and 31 only remove vertices, it clearly does not affect the fact that  $D$  is  $K_{3,c}$ -minor-free. Furthermore, the operations in Reduction Rule 6, 32 and 33 result in a graph  $D'$  such that  $\overline{D'}$  is a minor of  $\overline{D}$ . Hence, the graph resulting from these reduction rules is also  $K_{3,c}$ -minor-free. ◀

We now argue the main structural consequence of applying these reduction rules.

► **Lemma 35.** *Suppose that Reduction Rule 5 and Reduction Rules 30 to 33 do not apply. Then  $D \setminus S$  has  $\mathcal{O}(|S|)$  acyclic components.*

► **Lemma 36.** *Let  $C$  be a connected component of  $\overline{D} \setminus S$ , and  $\ell$  be the number of neighbors of  $C$  in  $S$ . If Reduction Rules 5 and 6 are not applicable, then  $\overline{D}[C]$  has  $\mathcal{O}(\ell)$  leaves.*

**Proof.** Recall that when the first two reduction rules do not apply, every vertex in  $D$  is incident to at least 4 arcs and thus every leaf of  $C$  is incident to at least 3 arcs with endpoints in  $S$ . However, since there can be bidirectional arcs between  $C$  and  $S$ , every leaf of  $C$  has at least 2 neighbors in  $S$ . From Lemma 28 it follows that there are only  $\mathcal{O}(\ell)$  vertices in  $C$  with at least 3 neighbors in  $S$ . Hence it suffices to obtain an  $\mathcal{O}(\ell)$  bound on the leaves of  $C$  with exactly two neighbors in  $S$ .

Recalling Lemma 28, we observe that each of the leaves of  $C$  has one of  $\mathcal{O}(\ell)$  possible neighborhoods in  $S$ . Let us fix two distinct vertices  $x$  and  $y$  in  $N(C)$ . We will show that there are at most  $c - 1$  leaves of  $C$  with both  $x$  and  $y$  as neighbors. Suppose for a contradiction that there is a set  $L$  of at least  $c$  leaves of  $\overline{D}[C]$  which are adjacent to  $x$  and  $y$ . Since  $\overline{D}[C]$  is a tree and  $L$  is a subset of its leaves, the graph  $\overline{D}[C \setminus L]$  is also a tree. If we contract  $\overline{D}[C \setminus L]$  into a single vertex, say  $z$ , then the subgraph induced on the vertices in  $L \cup \{x, y, z\}$  would be isomorphic to  $K_{3,c}$ , contradicting the fact that  $\overline{D}$  is  $K_{3,c}$ -minor-free. We conclude that  $C$  can have at most  $c - 1$  leaves with neighbors  $x$  and  $y$ , completing the proof. ◀

► **Lemma 37.** *If none of Reduction Rules 5, 6, 30-33 apply, then  $\overline{D} \setminus S$  has at most  $\mathcal{O}(|S|)$  vertices of degree at least 3.*

**Proof.** Let  $\mathcal{C}$  be the set of all components of  $D \setminus S$ . Since none of the aforementioned reduction rules apply, we can invoke Lemma 35 and Lemma 36. That is, we conclude that there are only  $\mathcal{O}(|S|)$  components in  $\mathcal{C}$  and that the number of leaves in a component  $C \in \mathcal{C}$  is  $\mathcal{O}(|N(C)|)$ .

Since the number of leaves in a tree gives an upper bound on the number of vertices of degree at least 3, this implies that the number of vertices of degree at least 3 in  $\overline{D}[C]$  is also bounded by  $\mathcal{O}(|N(C)|)$ . Thus it suffices to show that  $\sum_{C \in \mathcal{C}} |N(C)| = \mathcal{O}(|S|)$ . However,  $\sum_{C \in \mathcal{C}} |N(C)|$  is the same as the number of edges in the graph  $G$  which we obtain from  $\overline{D}$  by contracting each component of  $\mathcal{C}$  to a single vertex and removing all edges between vertices in  $S$ . Since  $G$  is clearly a minor of  $\overline{D}$ , it is also  $K_{3,c}$ -minor-free and hence  $|E(G)|$  is at most  $\mathcal{O}(|V(G)|) = \mathcal{O}(|S| + |\mathcal{C}|) = \mathcal{O}(|S|)$ , which concludes the proof. ◀

The main consequence of the above lemma is that we can now add all the vertices of degree at least 3 to the set  $S$  in order to get a set  $S'$  which is also a feedback vertex set of  $\overline{D}$  of size  $\mathcal{O}(|S|) = \mathcal{O}(k)$ . At the same time, the graph  $\overline{D} \setminus S'$  is significantly more structured: every connected component of this graph is in fact a path and this will play a crucial role in the rest of this section. Since  $|S'| \in \mathcal{O}(|S|)$ , it suffices to obtain a reduced instance of size linear in  $|S'|$ , and so for ease of presentation we will hereinafter set  $S := S'$ .

► **Reduction Rule 38.** If none of the Reduction Rules 5-6, 30-33 apply, then we add all vertices of total degree three in  $D \setminus S$  to  $S$ .

Observe that after applying Reduction Rule 38,  $D \setminus S$  is a set of roads. Furthermore, once we ensure that  $D \setminus S$  is a set of roads, none of the reduction rules in this section will ever create a new degree 3 vertex in  $D \setminus S$ . Hence we can exhaustively apply all the reduction rules in this section once again to ensure that the number of roads in  $D \setminus S$  is  $\mathcal{O}(|S|)$ . In the rest of the section, we present reduction rules to handle the roads in  $D \setminus S$ .

## 4.1 Dealing with roads

Our first step will be to transform our instance so that all roads in  $D \setminus S$  are even more structured with respect to their adjacencies with  $S$ .

► **Definition 39.** A road  $P$  in  $D \setminus S$  is *nice* if  $|N(P') \cap S| \leq 2$ , where  $P'$  are the internal vertices of  $P$ .

In other words, nice roads are roads whose internal vertices are *all* adjacent to at most two specific vertices from  $S$  (other than the endpoints of the road); observe that this is equivalent to requiring that  $|N(P')| \leq 4$ , where  $P'$  is the set of internal vertices of the nice road  $P$ .

In order to achieve this transformation, we will iteratively construct an auxiliary vertex set  $Q$  to store certain vertices that form separators between nice road segments in  $D - S$ . In the course of this procedure, we will also construct an injective mapping  $\delta$  from  $Q$  to the connected components of  $D \setminus (S \cup Q)$ . We initialize by setting  $\delta = Q = \emptyset$ .

► **Reduction Rule 40.** Let  $A$  be a connected component which is a road in  $D \setminus (S \cup Q)$  that is not nice. Moreover, let  $A'$  be a maximal nice subroad of  $A$  which contains a leaf in  $\overline{D} \setminus (S \cup Q)$  and let  $a'$  be the unique neighbor of  $A'$  in  $A$ . Then add  $a'$  to  $Q$  and add  $a' \mapsto A'$  to  $\delta$ .

For each vertex  $q \in Q$ , let  $R_q = \{q\} \cup \delta(q)$ . Observe that  $R_q$  is a road which contains at least 3 neighbors in  $S$ . Furthermore, for any  $q, q' \in Q$  our construction of  $\delta$  ensures that  $R_q$  and  $R_{q'}$  are vertex-disjoint since  $\delta(q)$  is a nice road in  $D \setminus (S \cup Q)$ , for all  $q \in Q$ .

► **Lemma 41.** *After the exhaustive application of Reduction Rule 40, we have  $|Q| = \mathcal{O}(|S|)$ .*

The next rule is only applied *once* after the exhaustive application of Reduction Rule 40. Note that it does not increase the parameter by more than a linear factor due to Lemma 41.

► **Reduction Rule 42.** Set  $S := S \cup Q$ .

Observe that after the exhaustive application of Reduction Rule 40 and the application of Reduction Rule 42, each road in  $D - S$  is nice. Furthermore, the number of roads in  $D - S$  is still linear in  $S$ , since removing  $|Q|$  vertices from a set of roads only increases the number of roads in the set by at most  $|Q|$ . Our next task is to deal with nice roads, but we first state a useful observation about general roads.

► **Observation 43.** Let  $P$  be a road in  $D \setminus S$  and let  $P'$  be the internal vertices of  $P$ . For any DFVS  $T$  of  $D$ , the set  $(T \setminus P') \cup N(P')$  is also a DFVS of  $D$ . In particular, every minimum DFVS contains at most  $|N(P')|$  vertices of  $P \cup N(P')$ .

► **Reduction Rule 44.** Let  $P$  be a nice road in  $D \setminus S$ ,  $P'$  internal vertices of  $P$ , and  $x$  a vertex in  $N(P') \setminus V(P)$ . If  $D[P' \cup \{x\}]$  contains at least  $|N(P')|$  directed cycles intersecting only in  $x$ , then we remove  $x$  from  $D$  and set  $k = k - 1$ .

For internal vertices of a road  $P$ , we define an equivalence relation  $\sim_P$  such as  $a \sim_P b$  if and only if  $N^+(a) \setminus V(P) = N^+(b) \setminus V(P)$  and  $N^-(a) \setminus V(P) = N^-(b) \setminus V(P)$  (i.e.,  $a$  and  $b$  have same out- and in- neighborhoods outside of  $P$ ). We are now ready to state our final reduction rule, which will later allow us to bound the length of each nice road by a constant.

► **Reduction Rule 45.** Let  $P$  be a nice road in  $D \setminus S$ , and let  $P'$  be internal vertices of  $P$  with  $\ell = |N(P')|$ . If  $P'$  contains a directed subpath  $Q = (q_1, \dots, q_{\ell+2})$ , such that  $q_i \sim_P q_j$  for all  $1 \leq i, j \leq \ell + 1$ , then we remove  $q_\ell$  from  $D$  and add the arc  $(q_{\ell-1}, q_{\ell+1})$ .

We are now ready to complete the proof of our linear kernelization by bounding the size of an instance after the exhaustive application of our reduction rules.

► **Lemma 46.** *If none of Reduction Rules 5, 6 and 30-45 apply, then  $|D| = \mathcal{O}(|S|)$ .*

**Proof Sketch.** Let  $\mathcal{P}$  be the set of all acyclic component of  $D \setminus S$ . Recall that since Reduction Rules 5, 6 and 30-42 do not apply, every acyclic component in  $\mathcal{P}$  is a nice road in  $D$  and there are at most  $|\mathcal{P}| = \mathcal{O}(|S|)$  such nice roads. Therefore, it suffices to show that there is a constant  $d$  which bounds the size of any nice road  $P$  in  $\mathcal{P}$ . This is achieved by the following three-step process: (a) we bound the number of vertices with bidirectional arcs to (from) a vertex from  $S$ , (b) we bound the number of sinks and sources on  $P$ , and (c) we bound the number of remaining vertices in  $P$ , i.e., vertices on directed path segments between vertices covered under points (a) and (b).

It is not difficult to show that points (a) and (b) follow from the exhaustive application of Reduction Rule 44. For point (c), we first observe that each vertex on a directed path must have precisely one in- and one out-neighbor in  $S$ , which means that there are only 2 ‘types’ of vertices on these directed path segments. Then Reduction Rule 44 allows us to bound the number of alternations of types on a directed path, while Reduction Rule 45 provides a bound on the number of consecutive types on such paths. ◀

## 5 Conclusions and Future Work

Our results provide a stepping stone towards resolving the existence of a polynomial kernel for DFVS, and to the best of our knowledge also represent the first explicit kernelization results for DFVS with respect to any natural parameter. They also open up several new directions for future research. For instance, can we find reasonable parameters that lie “between” DFVS number and FVS number, and would it be possible to generalize our polynomial kernel to these? What about parameters which are incomparable to the FVS number but also upper-bound the DFVS number? Can our linear kernel be lifted to graph classes of bounded expansion or nowhere dense graphs? Can Theorem 1 be improved to a cubic or quadratic kernel, for instance by using techniques similar to the improvement obtained for the undirected setting by Thomasse [36]? Another related problem of interest is whether DFVS can be solved in time  $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ , which remains open even on planar graphs.

---

### References

- 1 Vineet Bafna, Piotr Berman, and Toshihiro Fujito. A 2-approximation algorithm for the undirected feedback vertex set problem. *SIAM J. Discrete Math.*, 12(3):289–297, 1999.
- 2 Reuven Bar-Yehuda, Dan Geiger, Joseph Naor, and Ron M. Roth. Approximation algorithms for the feedback vertex set problem with applications to constraint satisfaction and bayesian inference. *SIAM J. Comput.*, 27(4):942–959, 1998.
- 3 Ann Becker and Dan Geiger. Optimization of pearl’s method of conditioning and greedy-like approximation algorithms for the vertex feedback set problem. *Artif. Intell.*, 83(1):167–188, 1996.
- 4 Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. (meta) kernelization. *J. ACM*, 63(5):44:1–44:69, 2016.
- 5 Hans L. Bodlaender and Thomas C. van Dijk. A cubic kernel for feedback vertex set and loop cutset. *Theory Comput. Syst.*, 46(3):566–597, 2010.
- 6 Yixin Cao, Jianer Chen, and Yang Liu. On feedback vertex set new measure and new structures. In Haim Kaplan, editor, *Algorithm Theory - SWAT 2010, 12th Scandinavian Symposium and Workshops on Algorithm Theory, Bergen, Norway, June 21-23, 2010. Proceedings*, volume 6139 of *Lecture Notes in Computer Science*, pages 93–104. Springer, 2010.
- 7 Chandra Chekuri and Vivek Madan. Constant factor approximation for subset feedback set problems via a new LP relaxation. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 808–820. SIAM, 2016.
- 8 Jianer Chen, Fedor V. Fomin, Yang Liu, Songjian Lu, and Yngve Villanger. Improved algorithms for feedback vertex set problems. *J. Comput. Syst. Sci.*, 74(7):1188–1198, 2008. doi:10.1016/j.jcss.2008.05.002.
- 9 Jianer Chen, Yang Liu, Songjian Lu, Barry O’Sullivan, and Igor Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55(5), 2008. doi:10.1145/1411509.1411511.
- 10 Rajesh Hemant Chitnis, Marek Cygan, Mohammad Taghi Hajiaghayi, and Dániel Marx. Directed subset feedback vertex set is fixed-parameter tractable. *ACM Transactions on Algorithms*, 11(4):28, 2015.
- 11 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
- 12 Marek Cygan, Daniel Lokshtanov, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. On the hardness of losing width. *Theory Comput. Syst.*, 54(1):73–82, 2014.

- 13 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *FOCS*, pages 150–159, 2011.
- 14 Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. Subset feedback vertex set is fixed-parameter tractable. *SIAM J. Discrete Math.*, 27(1):290–309, 2013. doi:10.1137/110843071.
- 15 Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer Verlag, New York, 2nd edition, 2000.
- 16 Rodney G. Downey and Michael R. Fellows. Fixed-parameter intractability. In *Proceedings of the Seventh Annual Structure in Complexity Theory Conference, Boston, Massachusetts, USA, June 22-25, 1992*, pages 36–49, 1992.
- 17 Rodney G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness I: basic results. *SIAM J. Comput.*, 24(4):873–921, 1995.
- 18 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.
- 19 P Erdős and L Pósa. On independent circuits contained in a graph. *Canad. J. Math.*, 17:347–352, 1965.
- 20 Guy Even, Joseph Naor, Baruch Schieber, and Madhu Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998.
- 21 Jakub Gajarský, Petr Hliněný, Jan Obdržálek, Sebastian Ordyniak, Felix Reidl, Peter Rossmanith, Fernando Sánchez Villaamil, and Somnath Sikdar. Kernelization using structural parameters on sparse graph classes. *J. Comput. Syst. Sci.*, 84:219–242, 2017.
- 22 Jonathan L. Gross and Thomas W. Tucker. *Topological Graph Theory*. Wiley-Interscience, New York, NY, USA, 1987.
- 23 Venkatesan Guruswami and Euiwoong Lee. Inapproximability of h-transversal/packing. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, August 24-26, 2015, Princeton, NJ, USA*, volume 40 of *LIPICs*, pages 284–304. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.
- 24 Bart M. P. Jansen and Hans L. Bodlaender. Vertex cover kernelization revisited - upper and lower bounds for a refined parameter. *Theory Comput. Syst.*, 53(2):263–299, 2013.
- 25 Naonori Kakimura, Ken-ichi Kawarabayashi, and Yusuke Kobayashi. Erdős-pósa property and its algorithmic applications: parity constraints, subset feedback set, and subset packing. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1726–1736. SIAM, 2012.
- 26 Naonori Kakimura, Ken-ichi Kawarabayashi, and Dániel Marx. Packing cycles through prescribed vertices. *J. Comb. Theory, Ser. B*, 101(5):378–381, 2011.
- 27 Richard M. Karp. Reducibility among combinatorial problems. In *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York.*, pages 85–103, 1972. URL: <http://www.cs.berkeley.edu/~luca/cs172/karp.pdf>.
- 28 Ken-ichi Kawarabayashi and Yusuke Kobayashi. 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. doi:10.1016/j.jctb.2011.12.001.
- 29 Ken-ichi Kawarabayashi, Daniel Král', Marek Krcál, and Stephan Kreutzer. Packing directed cycles through a specified vertex set. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 365–377, 2013.
- 30 Tomasz Kociumaka and Marcin Pilipczuk. Faster deterministic feedback vertex set. *Inf. Process. Lett.*, 114(10):556–560, 2014. doi:10.1016/j.ipl.2014.05.001.

- 31 M. Pontecorvi and Paul Wollan. Disjoint cycles intersecting a set of vertices. *J. Comb. Theory, Ser. B*, 102(5):1134–1141, 2012.
- 32 Venkatesh Raman, Saket Saurabh, and C. R. Subramanian. Faster fixed parameter tractable algorithms for finding feedback vertex sets. *ACM Transactions on Algorithms*, 2(3):403–415, 2006. doi:10.1145/1159892.1159898.
- 33 Bruce A. Reed, Neil Robertson, Paul D. Seymour, and Robin Thomas. Packing directed circuits. *Combinatorica*, 16(4):535–554, 1996.
- 34 Paul D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15(2):281–288, 1995.
- 35 Paul D. Seymour. Packing circuits in eulerian digraphs. *Combinatorica*, 16(2):223–231, 1996.
- 36 Stéphan Thomassé. A  $4k^2$  kernel for feedback vertex set. *ACM Trans. Algorithms*, 6(2):32:1–32:8, 2010.
- 37 Magnus Wahlström. Half-integrality, LP-branching and FPT algorithms. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1762–1781. SIAM, 2014.