Journal of Computer and System Sciences 92 (2018) 9–21

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Kernels for deletion to classes of acyclic digraphs ☆

Akanksha Agrawal [a], Saket Saurabh [a,b], Roohani Sharma [b], Meirav Zehavi [a,*]

[a] University of Bergen, Norway
[b] Institute of Mathematical Science, HBNI, India

ARTICLE INFO

Article history:
Received 1 October 2016
Received in revised form 1 May 2017
Accepted 27 July 2017
Available online 1 September 2017

Keywords:
Out-forest
Pumpkin
Parameterized complexity
Kernelization

ABSTRACT

Given a digraph D and an integer k, Directed Feedback Vertex Set (DFVS) asks whether there exists a set of vertices S of size at most k such that $F = D \setminus S$ is DAG. Mnich and van Leeuwen [*STACS 2016*] considered the kernelization complexity of DFVS with an additional restriction on F, namely that F must be an out-forest (Out-Forest Vertex Deletion Set), an out-tree (Out-Tree Vertex Deletion Set), or a (directed) pumpkin (Pumpkin Vertex Deletion Set). Their objective was to shed light on the kernelization complexity of DFVS, a well-known open problem in Parameterized Complexity. We improve the kernel sizes of Out-Forest Vertex Deletion Set from $\mathcal{O}(k^3)$ to $\mathcal{O}(k^2)$ and of Pumpkin Vertex Deletion Set from $\mathcal{O}(k^{18})$ to $\mathcal{O}(k^3)$. We also prove that the former kernel size is tight under certain complexity theoretic assumptions.

1. Introduction

Feedback Set problems form a family of fundamental combinatorial optimization problems. The input for Directed Feedback Vertex Set (DFVS) (Directed Feedback Edge Set (DFES)) consists of a directed graph (digraph) D and a positive integer k, and the question is whether there exists a subset $S \subseteq V(D)$ ($S \subseteq E(D)$) such that the graph obtained after deleting the vertices (edges) in S is a directed acyclic graph (DAG). Similarly, the input for Undirected Feedback Vertex Set (UFVS) (Undirected Feedback Edge Set (UFES)) consists of an undirected graph G and a positive integer k, and the question is whether there exists a subset $S \subseteq V(G)$ ($S \subseteq E(G)$) such that the graph obtained after deleting the vertices (edges) in S is a forest.

All of these problems, excluding Undirected Feedback Edge Set, are NP-complete. Furthermore, Feedback Set problems are among Karp's 21 NP-complete problems and have been topic of active research from algorithmic [1–17] as well as structural points of view [18–24]. In particular, such problems constitute one of the most important topics of research in Parameterized Complexity [4,6–10,14,13,15–17], spearheading development of new techniques. In this paper we study the parameterized complexity of restrictions of DFVS.

In Parameterized Complexity each problem instance is accompanied by a parameter k. A central notion in this field is the one of *fixed-parameter tractability (FPT)*. This means, for a given instance (I, k), solvability in time $f(k)|I|^{\mathcal{O}(1)}$ where f is

☆ A preliminary version of this paper appeared in the proceedings of the 27th *International Symposium on Algorithm and Computation (ISAAC 2016)*. The research leading to these results received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013)/ERC Grant Agreements no. 306992 (S. Saurabh). Part of this work was done while M. Zehavi was visiting the Simons Institute for the Theory of Computing.

* Corresponding author.
E-mail addresses: akanksha.agrawal@uib.no (A. Agrawal), saket@imsc.res.in (S. Saurabh), roohani@imsc.res.in (R. Sharma), meirav.zehavi@uib.no (M. Zehavi).

http://dx.doi.org/10.1016/j.jcss.2017.07.008
0022-0000/

some function of k. Another central notion is the one of *kernelization*. A parameterized problem is said to admit a *kernel* of size $f(k)$ for some function f of k if there is a polynomial-time algorithm, called a *kernelization algorithm*, that translates any input instance to an equivalent instance of the same problem whose size is bounded by $f(k)$. In case the function f is polynomial in k, the problem is said to admit a *polynomial kernel*. For more information on these concepts we refer the reader to monographs such as [25,26].

In contrast to UFVS which admits a polynomial kernel, the existence of a polynomial kernel for DFVS is still an open problem. The lack of progress on this question led to the consideration of various restrictions on input instances. In particular, we know of polynomial kernels for DFVS in tournaments as well as various generalizations [27–29]. However, the existence of a polynomial kernel for DFVS is open even for planar digraphs. Recently, in a very interesting article, to make progress on this question Mnich and van Leeuwen [30] considered DFVS with an additional restriction on *the output rather than the input*. Essentially, the basic philosophy of their program is the following: What happens to the kernelization complexity of DFVS when we consider subclasses of DAGs?

Mnich and van Leeuwen [30] inspected this question by considering the classes of out-forests, out-trees and (directed) pumpkins. An *out-tree* is a digraph where each vertex has in-degree at most 1 and the underlying (undirected) graph is a tree. An *out-forest* is a disjoint union of out-trees. On the other hand, a digraph is a *pumpkin* if it consists of a source vertex s and a sink vertex t, $s \neq t$, together with a collection of internally vertex-disjoint induced directed paths from s to t. Here, all vertices except s and t have in-degree 1 and out-degree 1. The examination of the classes of out-forests and out-trees was also motivated by the corresponding questions of UFVS and Tree Deletion Set in the undirected settings. Formally, Mnich and van Leeuwen [30] studied the following problems.

Out-Forest Vertex Deletion Set (OFVDS)	**Parameter:** k
Input: A digraph D and a positive integer k.	
Question: Is there a set $S \subseteq V(D)$ of size at most k such that $F = D \setminus S$ is an out-forest?	

Out-Tree Vertex Deletion Set (OTVDS) and Pumpkin Vertex Deletion Set (PVDS) are defined in a similar manner, where instead of an out-forest, F should be an out-tree or a pumpkin, respectively. Mnich and van Leeuwen [30] showed that OFVDS and OTVDS admit kernels of size $\mathcal{O}(k^3)$ and PVDS admits a kernel of size $\mathcal{O}(k^{18})$.

Our results and methods. The objective of this article is to give improved kernels for OFVDS and PVDS. In this context, we obtain the following results.

- OFVDS admits an $\mathcal{O}(k^2)$ kernel and PVDS admits an $\mathcal{O}(k^3)$ kernel. These results improve upon the best known upper bounds $\mathcal{O}(k^3)$ and $\mathcal{O}(k^{18})$, respectively.
- For any $\epsilon > 0$, OFVDS does not admit a kernel of size $\mathcal{O}(k^{2-\epsilon})$ unless $\mathsf{coNP} \subseteq \mathsf{NP}/\mathsf{poly}$.

To get the improved kernel for OFVDS we incorporate the Expansion Lemma as well as a factor 3-approximation algorithm for OFVDS in the kernelization routine given in [30]. The significance of this improvement also lies in the fact that it is essentially tight.

The kernelization algorithm for PVDS given in [30] works roughly as follows. It has two phases: (a) first it gives an $\mathcal{O}(k^5)$ kernel for a variant of the problem where we know the source and the sink of the pumpkin obtained after deleting the solution vertices; and (b) in the second phase, it reduces PVDS to polynomially many instances of a variant of the problem mentioned in item (a) and then composes these instances to get a kernel of size $\mathcal{O}(k^{18})$. In fact given an instance (D, k) of PVDS, the kernelization algorithm of [30] outputs an equivalent instance (D', k') such that $k' = \mathcal{O}(k^{18})$. We take a completely different route and use “sun-flower style” reduction rules together with a marking strategy to obtain an equivalent instance (D', k') such that $|V(D')| + |E(D')| = \mathcal{O}(k^3)$ and $k' \le k$. We believe the method applied in this algorithm could be useful also in other kernelization algorithms.

2. Preliminaries

We denote the set of natural numbers from 1 to n by $[n]$, and we use standard terminology from the book of Diestel [31] for graph-related terms which are not explicitly defined here. A digraph D is a pair $(V(D), E(D))$ such that $V(D)$ is a set of vertices and $E(D)$ is a set of ordered pairs of vertices. The underlying undirected graph G of D is a pair $(V(G), E(G))$ such that $V(G) = V(D)$ and $E(G)$ is a set of unordered pairs of vertices such that $\{u, v\} \in E(G)$ if and only if either $(u, v) \in E(D)$ or $(v, u) \in E(D)$. Let D be a digraph. For any $v \in V(D)$, we denote by $N^-(v)$ the set of in-neighbors of v, that is, $N^-(v) = \{u \mid (u, v) \in E(D)\}$. Similarly, we denote by $N^+(v)$ the set of out-neighbors of v, that is, $N^+(v) = \{v \mid (v, u) \in E(D)\}$. We denote the in-degree of a vertex v by $d^-(v) = |N^-(v)|$ and its out-degree by $d^+(v) = N^+(v)$. We say that $P = (u_1, \ldots, u_l)$ is a directed path in the digraph D if $u_1, \ldots, u_l \in V(D)$ and for all $i \in [l-1]$, $(u_i, u_{i+1}) \in E(D)$. A *collision* is a triplet (u, w, v) of distinct vertices such that $(u, w), (v, w) \in E(D)$.

3. Improved kernel for Out-Forest Vertex Deletion Set

The aim of this section is to present an $\mathcal{O}(k^2)$ kernel for OFVDS. In Section 3.1 we state definitions and results relevant to our kernelization algorithm. Next, in Section 3.2, we design an algorithm for OFVDS that outputs a 3-approximate solution, which will also be used by our kernelization algorithm. Finally, in Section 3.3, we present our kernelization algorithm.

3.1. Prerequisites

We start by giving the definition of a q-expansion and the statement of the Expansion Lemma.

Definition 1 *(q-Expansion). For a positive integer q, a set of edges $M \subseteq E(G)$ is a q-*expansion of A into B *if (i) every vertex in A is incident to exactly q edges in M, and (ii) M saturates exactly $q|A|$ vertices in B (i.e., there is a set of $q|A|$ vertices in B which are incident to edges in M).*

Lemma 1 *(Expansion Lemma [26,32]). Let q be a positive integer and G be an undirected bipartite graph with vertex bipartition (A, B) such that $|B| \geq q|A|$, and there are no isolated vertices in B. Then, there exist nonempty vertex sets $X \subseteq A$ and $Y \subseteq B$ such that there exists a q-expansion of X into Y, and no vertex in Y has a neighbor outside X (i.e., $N(Y) \subseteq X$). Furthermore, the sets X and Y can be found in time polynomial in the size of G.*

We will also need to rely on the well-known notion of l-flowers.

Definition 2 *(l-Flower). An undirected graph G contains an l-*flower through v *if there is a family of cycles $\{C_1, \dots, C_l\}$ in G such that for all distinct $i, j \in [l]$, $V(C_i) \cap V(C_j) = \{v\}$.*

Lemma 2 *([26,32]). Given an undirected graph G and a vertex $v \in V(G)$, there is a polynomial-time algorithm that either outputs a $(k+1)$-flower through v or, if no such flower exists, outputs a set $Z_v \subseteq V(G) \setminus \{v\}$ of size at most $2k$ that intersects every cycle that passes through v in G.*

3.2. Approximation algorithm for Out-Forest Vertex Deletion Set

This section presents a 3-factor approximation algorithm for OFVDS. Given an instance of OFVDS, let *OPT* be the minimum size of a solution. Formally, we solve the following.

3-Approximate Out-Forest Vertex Deletion Set (Approx-OFVDS) **Input:** A directed graph D. **Output:** A subset $X \subseteq V(D)$ such that $D \setminus X$ is an out-forest and $|X| \leq 3 \cdot OPT$.

Given three distinct vertices $u_1, u_2, u_3 \in V(D)$, we say (u_1, u_2, u_3) is a *collision* if u_1 and u_2 are in-neighbors of u_3. Observe that any solution to OFVDS (and hence, Approx-OFVDS) must intersect any collision in at least 1 vertex. Moreover, it must intersect any cycle in at least 1 vertex. These observations form the basis of this algorithm.

Lemma 3. Approx-OFVDS *can be solved in polynomial time.*

Proof. Given a digraph D, the algorithm first constructs (in polynomial time) a family $\mathcal{F}$ of collisions and induced cycles in D such that the vertex sets of the entities in this family are pairwise disjoint. To do this, it initializes $\mathcal{F} = \emptyset$. Then, as long as there exists a vertex $v \in V(D)$ with at least two in-neighbors, u_1 and u_2, it inserts (v, u_1, u_2) into $\mathcal{F}$ and removes v, u_1 and u_2 from D (only for the purpose of the construction of $\mathcal{F}$). Once there is no vertex $v \in V(D)$ such that $d^-(v) \geq 2$, observe that all the directed cycles in the resulting digraph are pairwise vertex disjoint. $\mathcal{F}$ additionally contains all these pairwise vertex disjoint directed cycles..

Let us now construct a solution, S_{app}, for Approx-OFVDS. For every collision in $\mathcal{F}$, we let S_{app} contain each of the three vertices of this collision. From every cycle C in $\mathcal{F}$ we pick an arbitrary vertex and insert it into S_{app}. Clearly, $|S_{app}| \leq 3|\mathcal{F}|$. Since $\mathcal{F}$ is a collection of pairwise vertex disjoint collisions and directed cycles, $|\mathcal{F}| \leq OPT$. Thus, $|S_{app}| \leq 3 \cdot OPT$. It is now sufficient to prove that $D \setminus S_{app}$ is an out-forest. Observe that no vertex v in $D \setminus S_{app}$ has in-degree at least 2, otherwise the collision consisting of v and two of its in-neighbors would have been inserted into $\mathcal{F}$ and hence also into S_{app}. Moreover, there is no directed cycle C in $D \setminus S_{app}$. Indeed, if the cycle C intersects an collision in $\mathcal{F}$, it is clear that it cannot exist in $D \setminus S_{app}$, and otherwise it would have been inserted into $\mathcal{F}$ and hence one of its vertices would have been inserted into S_{app}. We thus conclude that the theorem is correct. □

3.3. *Kernelization algorithm for* Out-Forest Vertex Deletion Set

We are now ready to present our kernelization algorithm. Let (D, k) be an instance of OFVDS. We note that during the execution of our algorithm, D may become a multigraph.

Preprocessing. We start by applying the following reduction rules exhaustively, where a rule is applied only if its condition is true and the conditions of all of the preceding rules are false. Rule 3.4 is given in [30], and its correctness is proven in that paper. It will be clear that the first five rules can be applied in polynomial time, whereas, for applying the last rule, we call the algorithm given by Lemma 2. Moreover, it is straightforward to verify that each of these rules, except Rule 3.4 (whose safeness follows from [30]), is safe (i.e., the instance it returns is equivalent to the input instance).

Reduction Rule 3.1. *If there exists a vertex $v \in V(D)$ such that $d^+(v) = 0$ and $d^-(v) \le 1$, remove v from D.*

Reduction Rule 3.2. *If there exists a directed path $P = (w_0, w_1, \dots, w_l, w_{l+1})$ in D such that $l \ge 2$ and for all $i \in [l]$, $d^-(w_i) = d^+(w_i) = 1$, remove each vertex in $\{w_1, \dots, w_{l-1}\}$ from D and add the edge (w_0, w_l) to D.*

Reduction Rule 3.3. *If there exists an edge $(u, v) \in E(D)$ with multiplicity at least 3, remove all but two copies of it.*

Reduction Rule 3.4. *If there exist $k + 1$ collisions $(u_1, w_1, v), \dots, (u_{k+1}, w_{k+1}, v)$ that pairwise intersect only at v, remove v from D and decrease k by 1.*

Reduction Rule 3.5. *If there exists a vertex $v \in V(D)$ such that $d^-(v) \ge k + 2$, remove v from D and decrease k by 1.*

Reduction Rule 3.6. *Let G be the underlying graph of D. If there exists a vertex $v \in V(G)$ such that there is a $(k + 1)$-flower through v in G, remove v from D and decrease k by 1.*

Bounding out-degrees. Next, we aim to bound the maximum out-degree of a vertex in D. To this end, suppose that there exists a vertex $v \in V(D)$ with $d^+(v) \ge 16k$. Let G be the underlying graph of D. Since Reduction Rule 3.6 is not applicable, we let Z_v be the set obtained by calling the algorithm given by Lemma 2. Moreover, we let S_{app} be a 3-factor approximate solution obtained by calling the algorithm given by Theorem 3. We can assume that $|S_{app}| \le 3k$, since otherwise the input instance is a NO-instance. Denote $X_v = (S_{app} \cup Z_v) \setminus \{v\}$. Since $|Z_v| \le 2k$, we have that $|X_v| \le 5k$.

We proceed by examining the set $\mathcal{C}_v = \{C_1, C_2, \dots, C_{|\mathcal{C}_v|}\}$ of the connected components in $G \setminus (X_v \cup \{v\})$. Since S_{app} is an approximate solution, each component $C_i \in \mathcal{C}_v$ is an out-tree. Moreover, for any component $C_i \in \mathcal{C}_v$, v has at most one neighbor in C_i, since otherwise there would have been cycle passing through v in $G \setminus Z_v$, contradicting the definition of Z_v. For each component $C_i \in \mathcal{C}_v$, let u_i be the root of C_i. Let $\mathcal{D}_v = \{C_i \mid C_i \in \mathcal{C}_v, (v, u_i) \in E(D)\}$ and $\tilde{\mathcal{D}}_v = \{C_i \mid C_i \in \mathcal{C}, (v, u) \in E(D), u \in V(C_i), u \neq u_i\}$. Observe that $d^+(v) \le |\mathcal{D}_v| + |\tilde{\mathcal{D}}_v| + |X_v|$. Moreover, since Reduction Rule 3.4 is not applicable, $|\tilde{\mathcal{D}}_v| \le k$. Since $d^+(v) \ge 16k$, we have that $|\mathcal{D}_v| \ge 10k$. Without loss of generality, let $\mathcal{D}_v = \{C_1, \dots, C_p\}$ where $p = |\mathcal{D}_v|$. Since Reduction Rule 3.1 is not applicable, for any component $C_i \in \mathcal{D}_v$ there exists an edge in $E(G)$ with one endpoint in C_i and the other in X_v.

We now construct an auxiliary (undirected) bipartite graph H with bipartition (A, B), where $A = X_v$ and B is a set of new vertices denoted by $b_1, \dots, b_p$. For any $u \in A$ and $b_i \in B$, $(u, b_i) \in E(H)$ if and only if there exists an edge in G between u and some vertex in C_i. Since $|B| \ge 2|A|$ and there are no isolated vertices in B, we can use the algorithm given by Lemma 1 to obtain nonempty vertex sets $X'_v \subseteq A$ and $Y'_v \subseteq B$ such that there is a 2-expansion of X'_v into Y'_v and $N(Y'_v) \subseteq X'_v$. Let $\mathcal{D}'_v = \{C_i \mid b_i \in Y'_v\}$.

Reduction Rule 3.7. *Remove each of the edges in D between v and any vertex in a component in $\mathcal{D}'_v$. For every vertex $x_i \in X'_v$, insert two copies of the edge (v, x_i) into $E(D)$.*

Lemma 4. *Reduction Rule 3.7 is safe.*

Proof. Let D' be the graph resulting from the application of the rule. We need to prove that (D, k) is a YES-instance if and only if (D', k) is a YES-instance.

Forward direction. For the forward direction, we first claim that if (D, k) has a solution S such that $v \notin S$, then it has a solution S' such that $X'_v \subseteq S'$. To this end, suppose that (D, k) has a solution S such that $v \notin S$. Let $S' = (S \setminus \bigcup_{C_i \in \mathcal{D}'_v} V(C_i)) \cup X'_v$. It holds that $|S'| \le |S|$ since for each $x \in X'_v \setminus S$, at least one vertex from at least one of the two components in its expansion set must belong to the solution. Suppose for the sake of contradiction that $F = D \setminus S'$ is not an out-forest. First, assume that there exists a vertex in F with in-degree at least 2. Note that $V(D) = \bigcup_{C_i \in \mathcal{C}_v} V(C_i) \cup X_v \cup \{v\}$. Recall that the neighborhood of each of the vertices in the connected components that belong to $\mathcal{D}'_v$, outside their own component, is contained in $\{v\} \cup X'_v$. Moreover, v only has out-neighbors in the components that belong to $\mathcal{D}'_v$ and each $C_i \in \mathcal{C}_v$ is an out-tree. Therefore, since $D \setminus S$ has no vertex of in-degree at least 2, so does $D \setminus S'$. Now, assume that there is a cycle C in F. Then, if $V(C) \cap (S \cap \bigcup_{C_i \in \mathcal{D}'} V(C_i)) = \emptyset$, then C is also a cycle in $D \setminus S$, which is a contradiction. Thus, $V(C) \cap (S \cap \bigcup_{C_i \in \mathcal{D}'} V(C)_i) \neq \emptyset$. However, any cycle that passes through a component in $\mathcal{D}'_v$ also passes through v and a vertex in X'_v. Since $X'_v \subseteq S'$, no such cycle exists. This finishes the proof of the claim.

Let S be a solution to (D, k). If $v \in S$, then it is clear that $D' \setminus S$ is an out-forest. Otherwise, if $v \notin S$, our claim implies that (D, k) has a solution S' such that $X'_v \subseteq S'$. Then, $D' \setminus S'$ is an out-forest.

Backward direction. For the backward direction, let us prove the following claim. If (D', k) has a solution S such that $v \notin S$, then $X'_v \subseteq S$. Suppose, by way of contradiction, that the claim is incorrect. Then, there exists $x \in X'_v$ such that $x \notin S$. However, this implies that $D' \setminus S$ is not an out-forest as it contains the double edges (v, x_i).

Now, let S be a solution to (D', k), and denote $F = D' \setminus S$. Suppose $v \in S$. Then observe that, $D \setminus S = D' \setminus S$ is an out forest and thus S is solution to (D, k). If $v \notin S$, then by our previous claim, $X'_v \subseteq S$. Recall that for each $C_i \in \mathcal{C}_v$, C_i is an out-tree and u_i is the root of the out-tree C_i. Observe that for each $C_i \in \mathcal{D}'_v$ such that $u_i \notin S$, u_i is a root of some out-tree in F. This, is because the neighborhood of such a u_i in $D \setminus C_i$, is contained in $X'_v \cup \{v\}$, $X'_v \subseteq S$ and each edge between v and any vertex in $\mathcal{D}'_v$ is absent in D'. Also observe that, each such vertex u_i and v belong to different out-trees of F because in D', the edges between v and any vertex in $\mathcal{D}'_v$ are absent. This implies that if we add (to D') the edges between v and each vertex u_i that have been removed by the application of the rule, $V(F)$ will still induce an out-forest. Thus, S is a solution to (D, k). $\square$

After an exhaustive application of Reduction Rule 3.7, the out-degree of each vertex in D is at most $16k - 1$. However, since this rule inserts edges into $E(G)$, we need the following lemma.

Lemma 5. *The total number of applications of the reduction rules is bounded by a polynomial in the input size.*

Proof. We associate the measure $\mu(D, k) = |V(D)| + |E^1(D)|$ with every instance, where $E^1(D)$ are those edges in $E(D)$ that have multiplicity 1. Initially, this measure is bounded by a polynomial in the input size. We maintain the invariant that the measure is always positive and it strictly decreases whenever any of the reduction rules is applied. We thus conclude that the lemma is correct. $\square$

Correctness. By relying on counting arguments as well as Lemmas 4 and 5, we obtain the main result of this section.

Theorem 6. OFVDS *admits an $\mathcal{O}(k^2)$-kernel.*

Proof. Since our reduction rules are safe and can be applied only polynomially many times, we next assume that none of them is applicable, and turn to bound the number of vertices and edges in the resulting instance (D, k). To this end, suppose that (D, k) is a YES-instance. Let S be a solution, and denote $F = D \setminus S$. Let $V_{\leq 1}$ be the vertices in F that have degree at most 1 (in F), $V_{=2}$ be the vertices in F that have degree exactly 2, and $V_{\geq 3}$ be the vertices in F that have degree at least 3. Note that $|V(F)| = |V_{\leq 1}| + |V_{=2}| + |V_{\geq 3}|$.

Since Reduction Rules 3.5 and 3.7 are not applicable, the total degree of any vertex is bounded by $\mathcal{O}(k)$. In particular, the number of neighbors in F of each vertex in S is bounded by $\mathcal{O}(k)$. Therefore, the total number of vertices in F that have a neighbor in S is bounded by $\mathcal{O}(k^2)$. Observe that each vertex in $V_{\leq 1}$ is either a leaf in F or a root in F. Also all leaf vertices of F are in $V_{\leq 1}$. If a vertex in $V_{\leq 1}$ is a leaf in F, then such a vertex must have at least one neighbor in S, as otherwise Reduction Rule 3.1 would have been applicable. Thus, the total number of leaf vertices in $|V_{\leq 1}|$ is bounded by $\mathcal{O}(k^2)$. Since the total number of vertices that are a root of some out-tree in F are at most the number of leaves in F, $V_{\leq 1}$ is bounded by $\mathcal{O}(k^2)$. Since the underlying graph of F is a forest, $|V_{\geq 3}| < |V_{\leq 1}|$. Thus, $V_{\geq 3}$ is also bounded by $\mathcal{O}(k^2)$. Let us mark each of the vertices in F which has a neighbor in S as well as the vertices in $V_{\geq 3}$. Note that all other vertices are degree-2 vertices in D. Let $\mathcal{P}$ be the set maximal degree-2 paths in F whose endpoints are not marked. Observe that $|\mathcal{P}|$ is bounded by $\mathcal{O}(k^2)$. Moreover, the length of a maximal degree-2 path in $\mathcal{P}$ is at most 1, else Reduction Rule 3.2 is applicable.

We deduce that the number of vertices in D should be bounded by $\mathcal{O}(k^2)$, otherwise we can conclude that (D, k) be a NO-instance. Since F is a forest, $|E(F)| < |V(F)|$. Moreover, the number of edges incident to vertices in S is bounded by $\mathcal{O}(k^2)$. Therefore, the number of edges in D should be bounded by $\mathcal{O}(k^2)$. $\square$

In the next section, we prove that the size of the kernel given in Theorem 6 is tight, that is OFVDS does not admit an $\mathcal{O}(k^{2-\epsilon})$ size kernel unless coNP $\subseteq$ NP/poly.

4. Kernel lower bound for Out-Forest Vertex Deletion Set

In this section we show that the size of our kernel for OFVDS is optimal, i.e. there is no kernel for OFVDS which can be encoded into $\mathcal{O}(k^{2-\epsilon})$ bits for any $\epsilon > 0$ unless co-NP $\subseteq$ NP/poly. To this end, we will rely on the fact that Vertex Cover (VC) does not admit a kernel which can be encoded into $\mathcal{O}(k^{2-\epsilon})$ bits for any $\epsilon > 0$ unless co-NP $\subseteq$ NP/poly [33].[1]

We give a polynomial-time reduction from VC to OFVDS. Let (G,k) be an instance of VC. We construct an instance of (D,k') of OFVDS as follows. Initially, $V(D) = V(G)$ and $E(D) = \emptyset$. For each edge $\{u,v\} \in E(G)$, we add a vertex e_{uv} to $V(D)$, and we add the edges (u, e_{uv}) and (v, e_{uv}) to $E(D)$. Finally, we set $k' = k$.

Lemma 7. *(G,k) is a YES-instance of* VC *if and only if (D,k') is a YES-instance of* OFVDS.

Proof. In the forward direction, consider a vertex cover S of G of size at most k. We claim that $D \setminus S$ is an out-forest. By construction, each vertex $v \in V(D)$ that also belongs to G has in-degree 0 in D. For each vertex $e_{uv} \in V(D)$, $\{u,v\} \in E(G)$, and therefore at least one vertex among u and v must belong to S. Moreover, u and v are the only in-neighbors of e_{uv}. Therefore, for all $e_{uv} \in V(D)$, the in-degree of e_{uv} is at most 1 in $D \setminus S$. Note that $V(G) \subseteq V(D)$ is an independent set in D. Similarly, $V(D) \setminus V(G)$ is an independent set in D. Moreover, each vertex in $V(D) \setminus V(G)$ has out-degree 0. Therefore, $D \setminus X$ is acyclic.

In the reverse direction, consider an out-forest deletion set X of D of size at most k. Observe that if $e_{uv} \in X$, then $(X \setminus \{e_{uv}\}) \cup \{u\}$ is also an out-forest deletion set in D. Hence, without loss of generality assume that $X \subseteq V(G)$. Suppose, by way of contradiction, that X is not a vertex cover of G. Then, there is an edge $(u,v) \in E(G)$ such that $X \cap \{u,v\} = \emptyset$. By assumption X does not contain e_{uv}. Therefore, e_{uv} is a vertex with in-degree 2 in $D \setminus X$, which contradicts the choice of X. $\square$

We will use the Theorem 8 along with Lemma 7 to rule out an $\mathcal{O}(k^{2-\epsilon})$ kernel for OFVDS. For this purpose, we first state the definition of Oracle Communication Protocol [33].

Definition 3. *An oracle communication protocol for a language L is a communication protocol between two players. The first player is given the input x and has to run in time polynomial in the length of the input; the second player is computationally unbounded but is not given any part of x. At the end of the protocol the first player should be able to decide whether $x \in L$. The cost of the protocol is the number of bits of communication from the first player to the second player.*

Theorem 8 *(Theorem 2 [33]). Let $d \geq 2$ be an integer and ϵ a positive real number. If* coNP $\nsubseteq$ NP/poly, *there is no protocol of cost $\mathcal{O}(n^{d-\epsilon})$ to decide whether a d-uniform hypergraph on n vertices has a vertex cover of at most k vertices, even when the first player is co-nondeterministic.*

Theorem 9. *Let $\epsilon > 0$.* OFVDS *does not admit an oracle communication protocol of cost $\mathcal{O}(k^{2-\epsilon})$ for deciding (D,k) unless* coNP $\subseteq$ NP/poly.

Proof. Suppose OFVDS admits an oracle communication protocol of cost $\mathcal{O}(k^{2-\epsilon})$ for deciding (D,k). Combining this supposition with Lemma 7, we get a protocol of cost $\mathcal{O}(k^{2-\epsilon})$ for VC. By Theorem 8, this implies that coNP $\subseteq$ NP/poly. $\square$

We are now ready to prove the main result of this section.

Theorem 10. OFVDS *does not admit a kernel of size $\mathcal{O}(k^{2-\epsilon})$ for any $\epsilon > 0$ unless* coNP $\subseteq$ NP/poly.

Proof. If OFVDS admits a kernel of size $\mathcal{O}(k^{2-\epsilon})$, then the computationally bounded player computes the kernel and sends it to the computationally unbounded player who can correctly compute and return one bit answer. The cost of this protocol is $\mathcal{O}(k^{2-\epsilon})$. Therefore, by Theorem 9 it implies that coNP $\subseteq$ NP/poly. $\square$

5. Improved kernel for Pumpkin Vertex Deletion Set

In this section we prove the following theorem.

Theorem 11. PVDS *admits an $\mathcal{O}(k^3)$-vertex kernel.*

[1] Given a graph G and a parameter k, VC asks whether one can remove at most k vertices from G so that the resulting graph will be edge-less.

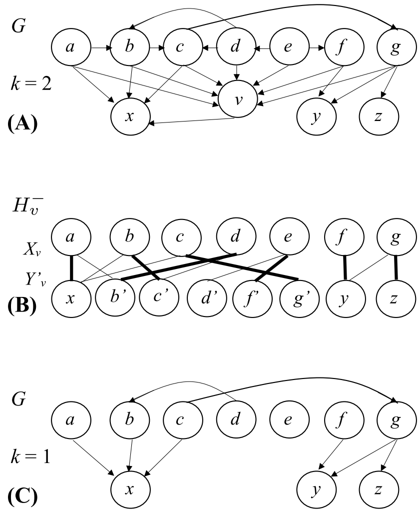

Fig. 1. (A) An instance of PVDS. (B) The graph H_v^-. The bold edges correspond to a maximum matching. (C) An application of Rule 5.2.

Let (D, k) be an instance of PVDS. We assume that $|V(D)| \geq k^3$, else we are done. Let $\mathsf{HO} = \{v \in V \mid d^+(v) \geq k+2\}$ and $\mathsf{HI} = \{v \in V \mid d^-(v) \geq k+2\}$. That is, HO and HI are vertices of high out-degrees and high in-degrees, respectively. Mnich and Leeuwen [30] proved that the following reduction rule is safe.

Reduction Rule 5.1. *If $|HO| > k+1$ or $|HI| > k+1$, return that (D, k) is a NO-instance.*

For the sake of clarity, we divide the presentation of the kernelization algorithm into two subsections. At the end of Section 5.1, we will simplify the instance in a way that will allow us to assume that if there is a solution S, then *both the source and sink of the pumpkin $D \setminus S$ belong to* $\mathsf{HO} \cup \mathsf{HI}$ (Assumption 1). This assumption will be at the heart of the “marking approach” of Section 5.2, which will handle instances which have been reduced with respect to the reduction rules in Section 5.1. An intuitive explanation of the necessity of our marking process is given at the beginning of Section 5.2. Throughout this section, if k becomes negative, we return that (D, k) is a NO-instance, and if D becomes a pumpkin and k is positive or zero, we return that (D, k) is a YES-instance.

5.1. Simplification phase

For any $v \in V(D)$, denote by X_v the set of in-neighbors of v, that is, $X_v = N^-(v)$ and by Y_v the set of every vertex $y \in V(D) \setminus \{v\}$ for which there exists a vertex $x \in X_v$ such that $(x, y) \in E(D)$. Note that X_v and Y_v may or may not be disjoint sets. We now give a construction of an auxiliary graph that will be used to prove the safeness of the upcoming reduction rule. For this, consider a set Y'_v of new vertices such that there is exactly one vertex $y' \in Y'_v$ for any $y \in Y_v$. That is, Y'_v is a set containing a copy for each of the vertex in Y_v. By construction, X_v and Y'_v are disjoint sets. Let H_v^- be the (undirected) bipartite graph on the vertex set $X_v \cup Y'_v$ where for all $x \in X_v$ and $y' \in Y'_v$, $\{x, y'\} \in E(H_v^-)$ if and only if $(x, y) \in E(D)$ (see Fig. 1). Let $\mathsf{match}^-(v)$ be the size of a maximum matching in H_v^-.

An application of the next rule is illustrated in Fig. 1.

Reduction Rule 5.2. *If there exists a vertex $v \in V(D)$ such that* $\mathsf{match}^-(v) \geq 2(k+1)$, *remove v from D and decrease k by* 1.

Lemma 12. *Reduction Rule 5.2 is safe.*

Proof. For the backward direction, trivially if S is a pumpkin deletion set in $D \setminus \{v\}$ of size at most $k-1$, then $S \cup \{v\}$ is a pumpkin deletion set in D of size at most k. For the forward direction, it is *sufficient* to show that if (D, k) is a YES-instance then *every solution S contains v*. For a contradiction, assume that there exists a solution S that does not contain v. Let M be a maximum matching in the graph H_v^-. Observe that for every edge $\{x, y'\} \in M$ where $x \in X_v$, if x is not the source of the pumpkin $D \setminus S$, it holds that $|S \cap \{x, y\}| \geq 1$ (otherwise the pumpkin $D \setminus S$ contains a vertex, which is not its source, and has at least two out-neighbors). Moreover, for every edge $\{x, y'\} \in M$ where $x \in X_v$, if y is the source of the pumpkin $D \setminus S$, it holds that $x \in S$. We thus deduce that *for all but one of the edges $\{x, y'\} \in M$, we have that $|S \cap \{x, y\}| \geq 1$*. Since M is a matching, for every vertex $u \in S$, the vertex u can belong to at most one edge in M, and the vertex u' (if it belongs to Y'_v) can also belong to at most one edge in M. However, $|S| \leq k$, and therefore $S \cup \{y' \in Y'_v : y \in S\}$ can intersect at most $2k$ edges in M. Since $S \cup \{y' \in Y'_v : y \in S\}$ must intersect all but one edge of M and $|M| \geq 2(k+1)$, we obtain a contradiction. $\square$

Now, to present the symmetric rule, for any vertex $v \in V(D)$, denote by X_v the set of out-neighbors of v, that is, $X_v = N^+(v)$. Let Y_v be the set of vertices $y \in V(D)$ for which there exists a vertex $x \in X_v$ such that $(y,x) \in E(D)$. Let Y'_v be a set containing a copy y' of each vertex $y \in Y$. Let H^+_v be the bipartite graph on the vertex-set $X_v \cup Y'_v$ which for all $x \in X_v$ and $y' \in Y'_v$ contains the edge $\{x, y'\}$ if and only if $(y,x) \in E(D)$. Let $\mathsf{match}^+(v)$ be the size of a maximum matching in H^+_v. Then, the following reduction rule is safe.

Reduction Rule 5.3. *If there exists a vertex $v \in V(D)$ such that $\mathsf{match}^+(v) \geq 2(k+1)$, remove v from D and decrement k by 1.*

We also need the following rule, proved by Mnich and Leeuwen [30].

Reduction Rule 5.4. *Let $P = (w_0, \cdots, w_\ell)$ be an induced directed path, that is for all $i \in [\ell - 1]$, $d^-(w_i) = d^+(w_i) = 1$, with $\ell > k+2$ in D. Then, delete w_1 from D and add the edge (w_0, w_2).*

Consider some hypothetical solution S (if such a solution exists). Let s and t denote the source and sink, respectively, of the pumpkin $D \setminus S$. Let A (or B) denote the set of out-neighbors (resp. in-neighbors) of s (resp. t) in the pumpkin. Clearly, $|A| = |B|$. Let $C = V(D) \setminus (S \cup A \cup B \cup \{s,t\})$. Next, we prove a series of useful claims relating to S.

Lemma 13. *(i) Every vertex in $\{s\} \cup A \cup B \cup C$ has in-degree (in D) at most $k+1$, and (ii) every vertex in $\{t\} \cup A \cup B \cup C$ has out-degree (in D) at most $k+1$.*

Proof. The correctness of the claim follows from the observation that every vertex in $\{s\} \cup A \cup B \cup C$ has at most one in-neighbor in the pumpkin $D \setminus S$ and at most $|S| \leq k$ in-neighbors outside this pumpkin, and every vertex in $\{t\} \cup A \cup B \cup C$ has at most one out-neighbor in this pumpkin and at most $|S| \leq k$ in-neighbors outside this pumpkin. $\square$

Lemma 14. *For any vertex $v \in V(D)$, $|N^-(v) \cap C|, |N^+(v) \cap C| \leq 2k+1$.*

Proof. We only show that $|N^-(v) \cap C| \leq 2k+1$, since the proof of the other inequality is symmetric. Clearly, any vertex $v \in \{s,t\} \cup A$ has no in-neighbor among the vertices in C and any vertex $v \in C \cup B$ has at most one in-neighbor among the vertices in C. Thus, we are left with showing the claim only for vertices belonging to S. Let $v \in S$ and for a contradiction assume that $|N^-(v) \cap C| \geq 2k+2$. We will show that in this case Rule 5.2 is applicable. Let X^C_v denote the set of in-neighbors of v in C. That is, $X^C_v = N^-(v) \cap C$. Consider the graph H^-_v. Now observe that $X^C_v \subseteq X_v$ and all the out-neighbors of vertices in X^C_v that belong to $D \setminus S$ are in $C \cup B$ (recall that S is the hypothetical solution we are working with). Observe that each vertex in C has a unique out-neighbor in $D \setminus S$ and in particular they belong to $C \cup B$. Thus, all the out-neighbors of vertices in X^C_v belonging to $C \cup B$, say Y^C_v, belong to Y_v. Recall that H^-_v is a bipartite graph with the vertex bipartition X_v and Y'_v (where in Y'_v we make a copy for each of the vertices in Y_v), and thus we have that $|\mathsf{match}^-(v)| \geq |X^C_v| \geq 2(k+1)$. Therefore, it is not possible that $|X^C_v| = |N^-(v) \cap C| \geq 2k+2$, since in this case Rule 5.2 is applicable. This completes the proof. $\square$

The set of in-neighbors (or out-neighbors) of any vertex $v \in V(D)$ is contained in $A \cup B \cup C \cup S \cup \{s,t\}$. Since $|A| \leq d^+(s)$, $|B| \leq d^-(t)$ and $|S| \leq k$, Lemma 14 gives us the following corollary.

Corollary 15. *For any vertex $v \in V(D)$, $d^-(v), d^+(v) \leq 3k + d^+(s) + d^-(t) + 3$.*

We further strengthen this corollary to obtain the following result.

Lemma 16. *For any vertex $v \in V(D)$, $d^-(v), d^+(v) \leq \min\{4k + 2d^+(s) + 3, 4k + 2d^-(t) + 3\}$.*

Proof. Let $v \in V(D)$. By Corollary 15, $d^-(v), d^+(v) \leq 3k + d^+(s) + d^-(t) + 3$. Thus, to prove the claim, it is sufficient to show that (i) $d^-(t) \leq d^+(s) + k$, and (ii) $d^+(s) \leq d^-(t) + k$. We only consider the first item, since the proof of the second one is symmetric. Observe that the pumpkin $D \setminus S$ contains at most $d^+(s)$ internally vertex disjoint paths from s and t. Therefore, since at most k in-neighbors of t can belong to S, t has (in D) at most $d^+(s) + k$ in-neighbors. $\square$

Let $M = \max_{v \in V(D)}\{d^+(v), d^-(v)\}$. The next corollary (Corollary 17), derived from Lemma 16, and rule (Reduction Rule 5.5) will bring us to the main goal of this subsection, summarized in Assumption 1 below.

Corollary 17. *If $M > 6k + 5$, then $s \in HO$ and $t \in HI$.*

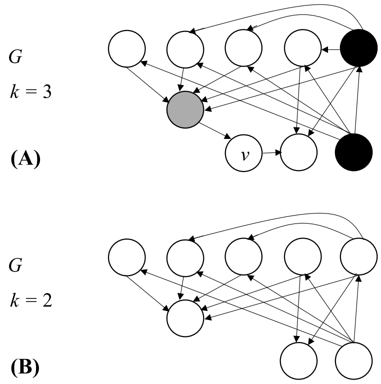

Fig. 2. (A) An instance of PVDS. The gray vertex belongs to HI, and the black vertices belong to HO. (B) An application of Rule 5.7.

Proof. Suppose that $M > 6k + 5$. Let $v \in V(D)$ be a vertex such that $6k + 5 < M = \max\{d^+(v), d^-(v)\}$. By Lemma 16, $d^-(v), d^+(v) \leq \min\{4k + 2d^+(s) + 3, 4k + 2d^-(t) + 3\}$, and therefore $6k + 5 < \min\{4k + 2d^+(s) + 3, 4k + 2d^-(t) + 3\}$. We thus get that $d^+(s), d^-(t) > k + 1$, which implies that $s \in \mathsf{HO}$ and $t \in \mathsf{HI}$. $\square$

Reduction Rule 5.5. *If $|V(D)| > 2k^2M + 4kM + k + 2$, return (D, k) is a NO-instance.*

Lemma 18. *Reduction Rule 5.5 is safe.*

Proof. Suppose that the instance is a YES-instance. Let S be a solution. Then, the pumpkin $D \setminus S$ contains at most $2M|S| \leq 2Mk$ vertices with at least one neighbor in S. Let Z be the set of these vertices. Then, $V(D) \setminus (S \cup Z \cup \{s, t\})$ is a collection of at most $(2k + 1)M$ paths whose vertices (including the endpoints) are vertices of in-degree 1 and out-degree 1 in D. By Rule 5.4, each such path contains at most k vertices. Therefore, $|V(D)| \leq (2k + 1)kM + |Z| + |S \cup \{s, t\}| \leq (2k + 1)kM + 2kM + k + 2 = 2k^2M + 4kM + k + 2$. $\square$

By Rule 5.5, if $M \leq 6k + 4$, we obtain the desired kernel. Thus, by Corollary 17, we have the following observation.

Assumption 1. *From now on, we can assume that if a solution exists, in the resulting pumpkin the source belongs to HO and the target belongs to HI.*

Next, it will be convenient to assume that HI and HO are disjoint sets. To this end, we apply the following rule exhaustively, where safeness follows directly from Lemma 13.

Reduction Rule 5.6. *Remove all vertices in HI $\cap$ HO and decrease k by $|\mathsf{HI} \cap \mathsf{HO}|$.*

We will also assume that the following rule, illustrated in Fig. 2, has been applied exhaustively. This assumption will be used at the end of the following subsection (in the proof of Lemma 26).

Reduction Rule 5.7. *If there exists a vertex $v \notin \mathsf{HI} \cup \mathsf{HO}$ such that $N^-(v) \cap (V(D) \setminus \mathsf{HI}) = \emptyset$ or $N^+(v) \cap (V(D) \setminus \mathsf{HO}) = \emptyset$, delete v from D and decrease k by 1.*

Lemma 19. *Reduction Rule 5.7 is safe.*

Proof. We only consider the case where there exists a vertex $v \notin \mathsf{HI} \cup \mathsf{HO}$ without any in-neighbor from $V(D) \setminus \mathsf{HI}$ since the proof of the second one is symmetric. In this case, by Assumption 1, if there exists a solution S, it does not contain exactly one vertex from HO, which will be a source, and exactly one vertex from HI, which will be the target. Indeed, if S does not contain at least two vertices from HO (HI), then since $|S| \leq k$, the pumpkin $D \setminus S$ will contain at least two vertices with out-degree (in-degree) at least two. Suppose $v \notin S$. Since $v \notin \mathsf{HI} \cup \mathsf{HO}$, from Assumption 1, v is neither the source nor the target of the pumpkin $D \setminus S$. Thus, there exists a vertex $u \in N^-(v)$, such that $u \notin S$. Since $N^-(v) \subseteq \mathsf{HI}$, $u \in \mathsf{HI}$. Since $u \in \mathsf{HI}$ and $u \notin S$, from Assumption 1, u is the target vertex of the pumpkin $D \setminus S$, which is a contradiction because u has an out-neighbor v in $D \setminus S$. $\square$

5.2. Marking approach

We are now ready to present our marking approach, handling instances to which Assumption 1 applies and none of the rules in Section 5.1 is applicable. Let $\mathcal{P}^*$ be the set of connected components in $D \setminus (\mathsf{HO} \cup \mathsf{HI})$ that are directed paths whose internal vertices have in-degree 1 and out-degree 1 in D, and let V^* be the union of the vertex-sets of the paths in $\mathcal{P}^*$. It turns out that by relying on Lemma 14 and Rule 5.4, one can directly bound the number of vertices in $V(D) \setminus V^*$ by $\mathcal{O}(k^3)$, assuming that the input instance is a YES-instance (see the proof of Lemma 24). However, bounding the size of V^* is more tricky, and our marking process is devoted to this cause. In this process, we will mark $\mathcal{O}(k^3)$ vertices from V^*, and prove that because we are handling instances to which Assumption 1 applies, all of the vertices that are not marked are essentially irrelevant. We will perform two "rounds" of marking. Roughly speaking, for each pair of vertices in HO (or HI) the first round aims to capture enough vertices of paths that describe the relation between the vertices in this pair, or, more precisely, why one of the vertices of the pair is a "better choice" than the other when one should decide which vertex (from HO) is the source of the pumpkin. However, this round is not sufficient, since some vertices in HO (or HI) have conflicts (independent of the other vertices in $\mathsf{HO} \cup \mathsf{HI}$) relating to the endpoints of the paths in $\mathcal{P}^*$. In the second round of marking, for each vertex in $\mathsf{HO} \cup \mathsf{HI}$, we mark enough vertices from these problematic paths.

First round of marking. Towards the performance of the first round, we need the following notations. For each vertex $v \in V(D) \setminus (\mathsf{HI} \cup \mathsf{HO})$, let $\widehat{P}(v)$ denote the connected component in $D \setminus (\mathsf{HI} \cup \mathsf{HO})$ containing v. For each $s \in \mathsf{HO}$, let $\widehat{N}(s)$ denote the set of each out-neighbor $v \in V(D) \setminus (\mathsf{HI} \cup \mathsf{HO})$ of s such that $\widehat{P}(v) \in \mathcal{P}^*$ and the first vertex of (the directed path) $\widehat{P}(v)$ is v. Symmetrically, for each $t \in \mathsf{HI}$, let $\widehat{N}(t)$ denote the set of each in-neighbor $v \in V(D) \setminus (\mathsf{HI} \cup \mathsf{HO})$ of t such that $\widehat{P}(v) \in \mathcal{P}^*$ and the last vertex of $\widehat{P}(v)$ is v. By Rule 5.6, $\mathsf{HI} \cap \mathsf{HO} = \emptyset$, and therefore these notations are well defined (i.e., we have not defined $\widehat{N}$ twice for the same vertex). Given $u \in (\mathsf{HI} \cup \mathsf{HO})$, we also denote $\widehat{\mathcal{P}}(u) = \{\widehat{P}(v) \mid v \in \widehat{N}(u)\}$. Observe that the paths in $\widehat{\mathcal{P}}(u)$ are pairwise vertex-disjoint.

Next, we identify enough vertices from paths that capture the relation between each pair of vertices in HO (or HI). For each pair $(s, s') \in \mathsf{HO} \times \mathsf{HO}$, let $\widehat{MK}_P(s, s')$ be an arbitrarily chosen minimal set of paths from $\widehat{\mathcal{P}}(s) \setminus \widehat{\mathcal{P}}(s')$ that together contain at least $k+1$ vertices not having s' as an in-neighbor. In this context, observe that only the last vertex on a path in $\widehat{\mathcal{P}}(s) \setminus \widehat{\mathcal{P}}(s')$ can have s' as an in-neighbor. In this case, the path must contain at least two vertices (since its first vertex cannot have s' as an in-neighbor), and while we insert the entire path into $\widehat{MK}_P(s, s')$, its last vertex is not "counted" when we aim to obtain at least $k+1$ vertices not having s' as an in-neighbor. If there are not enough paths to obtain at least $k+1$ such vertices, let $\widehat{MK}_P(s, s') = \widehat{\mathcal{P}}(s) \setminus \widehat{\mathcal{P}}(s')$. Symmetrically, for each pair $(t, t') \in \mathsf{HI} \times \mathsf{HI}$, let $\widehat{MK}_P(t, t')$ be an arbitrarily chosen minimal set paths from $\widehat{\mathcal{P}}(t) \setminus \widehat{\mathcal{P}}(t')$ that together contain at least $k+1$ vertices not having t' as an out-neighbor. If there are not enough paths, let $\widehat{MK}_P(t, t') = \widehat{\mathcal{P}}(t) \setminus \widehat{\mathcal{P}}(t')$.

Finally, given a pair $(v, v') \in (\mathsf{HO} \times \mathsf{HO}) \cup (\mathsf{HI} \times \mathsf{HI})$, let $\widehat{MK}(v, v')$ denote the union of the vertex-sets of the paths in $\widehat{MK}_P(v, v')$. We have the following claim.

Lemma 20. *For each pair $(v, v') \in (HO \times HO) \cup (HI \times HI)$, $|\widehat{MK}(v, v')| \leq 3(k+1)$.*

Proof. For each path inserted into $\widehat{MK}_P(v, v')$, at most one vertex is not counted in order to reach the desired threshold size of $k+1$, because all internal vertices in this path have in-degree exactly one and out-degree exactly one in D and one of the endpoints is definitely not an in-neighbor/ out-neighbor of s' (because the paths are from $\widehat{\mathcal{P}}(s) \setminus \widehat{\mathcal{P}}(s')$). Also, at least one vertex is counted in order to reach this size, because at least the first vertex in this path (from $\widehat{\mathcal{P}}(s) \setminus \widehat{\mathcal{P}}(s')$) is not an in-neighbor of s'. Once we reach the threshold of $k+1$ during the marking procedure, first note that at most $k+1$ paths are marked. Suppose $i(\leq k+1)$ number of paths are marked in order to reach the threshold of $k+1$. In each of these first $i-1$ paths that are marked, at most one vertex could be an in-neighbor of s'. Thus, in totality, at most $i-1 \leq k$ vertices from these $i-1$ paths do not contribute to the threshold of $k+1$. From the i^{th} path, there might be just a few vertices that are needed to meet the threshold of $k+1$, but others get marked because we pick the entire path and mark its vertices. Since, any such path contains at most $k+3$ vertices (from Rule 5.4), the total number of vertices in this path that so not contribute to the threshold of $k+1$ vertices is at most $k+2$. Thus, the total number of vertices in $\widehat{MK}_P(v, v')$ is at most $k+1+k+k+2 = 3(k+1)$. □

Second round of marking. We proceed to the second round of marking. For this purpose, we need the following notation. For each $u \in \mathsf{HI} \cup \mathsf{HO}$, let $\widetilde{MK}_P(u)$ denote the set of each directed path in $\mathcal{P}^*$ whose first and last vertices are both neighbors of u.

Reduction Rule 5.8. *If there exists $u \in HI \cup HO$ such that $|\widetilde{MK}_P(u)| \geq k+1$, delete u from D and decrease k by 1.*

Lemma 21. *Reduction Rule 5.8 is safe.*

Proof. To prove that the rule is correct, it is sufficient to show that if there exists a solution S, it contains u. Indeed, if S does not contain u, then it must contain at least one vertex from each path in $\widetilde{\mathcal{P}}(u)$, except perhaps one path, since together with u each such path forms a cycle and these cycles intersect only at the vertex u. Since $|\widetilde{\mathcal{P}}(u)| \geq k+1$ and the paths in $\widetilde{\mathcal{P}}(u)$ are vertex-disjoint, the claim follows. $\square$

For each $u \in \mathsf{HI} \cup \mathsf{HO}$, let $\widetilde{MK}(u)$ be the union of the vertex-sets of the paths in $\widetilde{MK}_P(u)$. Since at this point, Rules 5.4 and 5.8 are not applicable, we have the following lemma.

Lemma 22. *For each $u \in HI \cup HO$, $|\widetilde{MK}(u)| \leq k(k+3)$.*

The size of the kernel. For the sake of abbreviation, we define the following sets.

- $MK_{\mathcal{P}} = (\bigcup_{(u,u') \in (\mathsf{HO}\times\mathsf{HO}) \cup (\mathsf{HI}\cup\mathsf{HI})} \widehat{MK}_P(u,u')) \cup (\bigcup_{u \in \mathsf{HO}\cup\mathsf{HI}} \widetilde{MK}_P(u))$, and
- $MK = (\bigcup_{(u,u') \in (\mathsf{HO}\times\mathsf{HO}) \cup (\mathsf{HI}\cup\mathsf{HI})} \widehat{MK}(u,u')) \cup (\bigcup_{u \in \mathsf{HO}\cup\mathsf{HI}} \widetilde{MK}(u))$.

By Lemmas 20 and 22, and since Rule 5.1 is not applicable, we bound $|MK|$ as follows.

Lemma 23. $|MK| \leq 2 \cdot (3(k+1)^3 + k(k+1)(k+3)) \leq 8(k+3)^3$.

Let V^R denote the set of unmarked vertices in V^*, i.e., $V^* \setminus MK$. We construct the graph D' by removing from D all of the vertices in V^R, adding a set N_{k+2} of $k+2$ new vertices, and for each of the new vertices, adding an edge from each vertex in HO as well as an edge to each vertex in HI. If $V(D')$ contains at most $2k+4$ vertices, add to it one-by-one a vertex-set of a path in $\mathcal{P}^*$ until its size becomes at least $2k+5$ (by Rule 5.4, the size will not exceed $3k+7$, and because $|V(D)| \geq k^3$, we will reach the desired size).

Lemma 24. *If $|V(D')| > 30(k+2)^3$, (D',k) is a NO-instance of* PVDS.

Proof. If there exists a solution S, the pumpkin $D' \setminus S$ contains at most $2(k+1)|S| \leq 2(k+1)k$ vertices with at least one neighbor in $S \setminus (\mathsf{HI} \cup \mathsf{HO})$. Moreover, by Lemma 14, the set C associated with $D' \setminus S$ contains at most $4(k+1)|S| \leq 4(k+1)k$ vertices with at least one neighbor in S. Let Z be the set of vertices of these two types. Then, $D' \setminus (S \cup Z \cup \{s,t\})$ is a collection of paths. Among these paths, there are at most $k+2$ paths that are isolated vertices corresponding to the set N_{k+2} of vertices added to D' at its construction. There are paths of $\mathcal{P}^*$. In addition to these, there are paths that are not in $\mathcal{P}^*$. We claim that these additional paths are at most $2|Z| \leq 12(k+1)$. To see this, observe that, if a path in $D' \setminus (S \cup Z \cup \{s,t\})$ is not in $\cap P^*$ then it is a subpath of one of the directed s (source) to t (target) paths in the pumpkin $D' \setminus S$, that intersects with Z. Suppose there are $i \leq |Z|$ directed paths from the source to the target of the pumpkin $D' \setminus S$ that intersect with Z. Consider this set of i paths. Since a deletion of a vertex from any of the paths in this set can split a path into two subpaths, the number of subpaths possible after deleting $|Z|$ vertices is at most $2|Z|$. By Rule 5.4, these additional paths together contain at most $12(k+1)k(k+3)$ vertices. Moreover, from the union of the vertex-sets of paths in $\mathcal{P}^*$, D' contains only vertices from MK, and by Lemma 23, $|MK| \leq 8(k+3)^3$. Summing up, we get that $|V(D)'| \leq (k+2) + 12(k+1)k(k+3) + 8(k+3)^3 + |Z| + |S| + 2 \leq 30(k+3)^3$. $\square$

Correctness. Finally, Theorem 11 follows from Lemma 24 and the two lemmas below.

Lemma 25. *If (D,k) is a YES-instance then (D',k) is a YES-instance.*

Proof. Let S be a solution to (D,k). Let s and t be the source and sink, respectively, of the pumpkin $D \setminus S$. By Assumption 1, $s \in \mathsf{HO}$ and $t \in \mathsf{HI}$. Let $S' = S \cap V(D')$. Note that each path in $\mathcal{P}^*$ must either have its vertex-set contained in S or it must be a path from an out-neighbor of s to an in-neighbor of t which belongs to the pumpkin $D \setminus S$ (rather than only a subpath of a path from an out-neighbor of s to an in-neighbor of t which belongs to this pumpkin). Moreover, the set N_{k+2} of vertices added to D' at its construction are vertices of in-degree one and out-degree one, where s is an in-neighbor and t is an out-neighbor. Therefore, $D' \setminus S'$ is a pumpkin. $\square$

Lemma 26. *If (D',k) is a YES-instance then (D,k) is a YES-instance.*

Proof. Let S be a solution to (D',k). Let s and t be the source and target, respectively, of the pumpkin $D' \setminus S$. Because of the set N_{k+2} of $k+2$ vertices added to D' at its construction, and since $|S| \leq k$, $s \in \mathsf{HO}$ and $t \in \mathsf{HI}$. Moreover, by the definition of HO and HI, $(\mathsf{HO} \cup \mathsf{HI}) \setminus \{s,t\} \subseteq S$. We can also assume that S does not contain any vertex added to D' at its construction since by removing such a vertex from S, we still have a pumpkin. Our goal will be to show that S is also a solution to (D,k), which will imply that the lemma is correct. To this end, we will show that $D \setminus S$ is a pumpkin with source s and sink t.

First, note that we can assume that in $D \setminus S$ there exists a path from s to t. Indeed, if this is not true, then $D' \setminus S$ consists only of s, t and newly added vertices. That is, $V(D')$ contains at most $2k+4$ vertices, which contradicts its construction. By the definition of $\mathcal{P}^*$, each path in $\mathcal{P}^*$ has only internal vertices that have in-degree 1 and out-degree 1 in D, and its endpoints can only be adjacent to vertices in $\mathsf{HI} \cup \mathsf{HO}$ and in the path itself. Thus, to prove the lemma, it is sufficient to show that for each path in $\mathcal{P}^* \setminus MK_{\mathcal{P}}$, its first vertex has s as an in-neighbor, its last vertex has t as an out-neighbor, and if it contains at least two vertices, its first vertex is not a neighbor of t and its last vertex is not a neighbor of s.

Consider some path $P \in \mathcal{P}^* \setminus MK_{\mathcal{P}}$. First suppose, by way of contradiction, that the first vertex v of P does not have s as an in-neighbor. Because Rule 5.7 is not applicable, v has at least one in-neighbor $s' \in \mathsf{HO}$. Thus, since $v \notin MK$, $MK(s', s)$ contains at least $k+1$ vertices that are not out-neighbors of s and such that each of them belongs to a path in $\mathcal{P}^*$ whose first vertex is not an out-neighbor of s. The vertices in $MK(s', s)$ belong to D'. Since $|S| \leq k$, at least one of these vertices, say some u, should belong to the pumpkin $D' \setminus S$. However, in $D' \setminus ((\mathsf{HI} \cup \mathsf{HO}) \setminus \{s\})$, which is a supergraph of $D' \setminus S$, u cannot be reached from s, which contradicts the fact that $D' \setminus S$ is a pumpkin. Symmetrically, it is shown that the last vertex of P has t as an out-neighbor.

Now assume that P contains at least two vertices. Suppose, by way of contradiction, that the first vertex of P has t as a neighbor. We have already shown that the last vertex of P is also a neighbor of t, and therefore $P \in \widetilde{MK}_P(t)$. However, $\widetilde{MK}_P(t) \subseteq MK_{\mathcal{P}}$, which contradicts the fact that $P \in \mathcal{P}^* \setminus MK_{\mathcal{P}}$. Symmetrically, it is shown that the last vertex of P does not have s as a neighbor, concluding the proof of the lemma. $\square$

6. Conclusion

In this paper, we showed that OFVDS admits a kernel of size $\mathcal{O}(k^2)$, and that unless $\mathsf{coNP} \subseteq \mathsf{NP/poly}$, this bound is essentially tight. Furthermore, we showed that PVDS admits a kernel of size $\mathcal{O}(k^3)$. This result significantly improves upon the one by Mnich and van Leeuwen [30], who showed that PVDS admits a kernel of size $\mathcal{O}(k^{18})$.

As a direction for further research, we believe that it is interesting to attempt to further push forward the program initiated by Mnich and van Leeuwen [30]. In particular, we would like to conclude our paper by posing the following intriguing open questions. Here, the techniques we employed to solve PVDS might be relevant.

- For which other families of DAGs $\mathcal{F}$ does $\mathcal{F}$-Vertex Deletion Set admit a polynomial kernel? Here, given a digraph D and a parameter k, $\mathcal{F}$-Vertex Deletion Set asks whether at most k vertices can be removed from D in order to obtain a DAG in $\mathcal{F}$. Observe that when $\mathcal{F}$ is the family of all DAGs, the $\mathcal{F}$-Vertex Deletion Set problem is exactly DFVS. To elucidate the complexity of DFVS and also for purposes of independent interest, it is reasonable to consider families $\mathcal{F}$ which are not the family of all DAGs and yet result in nontrivial problems.
- In particular, it is possible to consider the family $\mathcal{F}$ that contains all DAGs with one source and one sink. This family contain the family of all (directed) pumpkins.
- Finally, we would like to ask whether PVDS admits a kernel of size $\mathcal{O}(k^2)$.

References

[1] V. Bafna, P. Berman, T. Fujito, A 2-approximation algorithm for the undirected feedback vertex set problem, SIAM J. Discrete Math. 12 (1999) 289–297.
[2] R. Bar-Yehuda, D. Geiger, J. Naor, R.M. Roth, Approximation algorithms for the feedback vertex set problem with applications to constraint satisfaction and Bayesian inference, SIAM J. Comput. 27 (1998) 942–959.
[3] A. Becker, D. Geiger, Optimization of pearl's method of conditioning and greedy-like approximation algorithms for the vertex feedback set problem, Artif. Intell. 83 (1996) 167–188.
[4] Y. Cao, J. Chen, Y. Liu, On feedback vertex set new measure and new structures, in: SWAT, 2010, pp. 93–104.
[5] C. Chekuri, V. Madan, Constant factor approximation for subset feedback problems via a new LP relaxation, in: SODA, 2016, pp. 808–820.
[6] J. Chen, F.V. Fomin, Y. Liu, S. Lu, Y. Villanger, Improved algorithms for feedback vertex set problems, J. Comput. Syst. Sci. 74 (2008) 1188–1198.
[7] J. Chen, Y. Liu, S. Lu, B. O'Sullivan, I. Razgon, A fixed-parameter algorithm for the directed feedback vertex set problem, J. ACM 55 (2008).
[8] R.H. Chitnis, M. Cygan, M.T. Hajiaghayi, D. Marx, Directed subset feedback vertex set is fixed-parameter tractable, ACM Trans. Algorithms 11 (2015) 28.
[9] M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J.M.M. Rooij, J.O. Wojtaszczyk, Solving connectivity problems parameterized by treewidth in single exponential time, in: FOCS, 2011, pp. 150–159.
[10] M. Cygan, M. Pilipczuk, M. Pilipczuk, J.O. Wojtaszczyk, Subset feedback vertex set is fixed-parameter tractable, SIAM J. Discrete Math. 27 (2013) 290–309.
[11] G. Even, J. Naor, B. Schieber, M. Sudan, Approximating minimum feedback sets and multicuts in directed graphs, Algorithmica 20 (1998) 151–174.
[12] V. Guruswami, E. Lee, Inapproximability of H-transversal/packing, in: APPROX/RANDOM, 2015, pp. 284–304.
[13] N. Kakimura, K. Kawarabayashi, Y. Kobayashi, Erdös-Pósa property and its algorithmic applications: parity constraints, subset feedback set, and subset packing, in: SODA, 2012, pp. 1726–1736.
[14] K. Kawarabayashi, Y. Kobayashi, Fixed-parameter tractability for the subset feedback set problem and the S-cycle packing problem, J. Comb. Theory, Ser. B 102 (2012) 1020–1034.
[15] T. Kociumaka, M. Pilipczuk, Faster deterministic feedback vertex set, Inf. Process. Lett. 114 (2014) 556–560.
[16] V. Raman, S. Saurabh, C.R. Subramanian, Faster fixed parameter tractable algorithms for finding feedback vertex sets, ACM Trans. Algorithms 2 (2006) 403–415.
[17] M. Wahlström, Half-integrality, LP-branching and FPT algorithms, in: SODA, 2014, pp. 1762–1781.
[18] P. Erdős, L. Pósa, On independent circuits contained in a graph, Can. J. Math. 17 (1965) 347–352.
[19] N. Kakimura, K. Kawarabayashi, D. Marx, Packing cycles through prescribed vertices, J. Comb. Theory, Ser. B 101 (2011) 378–381.
[20] K. Kawarabayashi, D. Král, M. Krcál, S. Kreutzer, Packing directed cycles through a specified vertex set, in: SODA, 2013, pp. 365–377.
[21] M. Pontecorvi, P. Wollan, Disjoint cycles intersecting a set of vertices, J. Comb. Theory, Ser. B 102 (2012) 1134–1141.

[22] B.A. Reed, N. Robertson, P.D. Seymour, R. Thomas, Packing directed circuits, Combinatorica 16 (1996) 535–554.
[23] P.D. Seymour, Packing directed circuits fractionally, Combinatorica 15 (1995) 281–288.
[24] P.D. Seymour, Packing circuits in Eulerian digraphs, Combinatorica 16 (1996) 223–231.
[25] R.G. Downey, M.R. Fellows, Fundamentals of Parameterized Complexity, Springer, 2013.
[26] M. Cygan, F.V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, Springer, 2015.
[27] F.N. Abu-Khzam, A kernelization algorithm for d-hitting set, J. Comput. Syst. Sci. 76 (2010) 524–531.
[28] J. Bang-Jensen, A. Maddaloni, S. Saurabh, Algorithms and kernels for feedback set problems in generalizations of tournaments, Algorithmica (2015) 1–24.
[29] M. Dom, J. Guo, F. Hüffner, R. Niedermeier, A. Truß, Fixed-parameter tractability results for feedback set problems in tournaments, J. Discret. Algorithms 8 (2010) 76–86.
[30] M. Mnich, E.J. van Leeuwen, Polynomial kernels for deletion to classes of acyclic digraphs, in: STACS, 2016, pp. 1–13.
[31] R. Diestel, Graph Theory, 4th edition, Springer, 2012.
[32] S. Thomassé, A $4k^2$ kernel for feedback vertex set, ACM Trans. Algorithms 6 (2010).
[33] H. Dell, D. Melkebeek, Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses, J. ACM 61 (2014) 23.