Directed Subset Feedback Vertex Set is Fixed-Parameter Tractable*

Rajesh Chitnis[1] **, Marek Cygan[2] ***, MohammadTaghi Hajiaghayi[1†], and Dániel Marx[3‡]

[1] Department of Computer Science, University of Maryland at College Park, USA, {rchitnis, hajiagha}@cs.umd.edu
[2] IDSIA, University of Lugano, Switzerland, marek@idsia.ch
[3] Computer and Automation Research Institute, Hungarian Academy of Sciences (MTA SZTAKI), Budapest, Hungary, dmarx@cs.bme.hu

Abstract. Given a graph G and an integer k, the FEEDBACK VERTEX SET (FVS) problem asks if there is a vertex set T of size at most k that hits all cycles in the graph. Bodlaender (WG '91) gave the first fixed-parameter algorithm for FVS in undirected graphs. The fixed-parameter tractability status of FVS in directed graphs was a long-standing open problem until Chen et al. (STOC '08) showed that it is fixed-parameter tractable by giving an $4^k k! n^{O(1)}$ algorithm. In the subset versions of this problems, we are given an additional subset S of vertices (resp. edges) and we want to hit all cycles passing through a vertex of S (resp. an edge of S). Indeed both the edge and vertex versions are known to be equivalent in the parameterized sense. Recently the SUBSET FEEDBACK VERTEX SET in undirected graphs was shown to be FPT by Cygan et al. (ICALP '11) and Kakimura et al. (SODA '12). We generalize the result of Chen et al. (STOC '08) by showing that SUBSET FEEDBACK VERTEX SET in directed graphs can be solved in time $2^{2^{O(k)}} n^{O(1)}$, i.e., FPT parameterized by size k of the solution. By our result, we complete the picture for feedback vertex set problems and their subset versions in undirected and directed graphs.

The technique of random sampling of important separators was used by Marx and Razgon (STOC '11) to show that UNDIRECTED MULTICUT is FPT and was generalized by Chitnis et al. (SODA '12) to directed graphs to show that DIRECTED MULTIWAY CUT is FPT. In this paper we give a general family of problems (which includes DIRECTED MULTIWAY CUT and DIRECTED SUBSET FEEDBACK VERTEX SET among others) for which we can do random sampling of important separators and obtain a set which is disjoint from a minimum solu-

---

* A full version of the paper is available at http://arxiv.org/pdf/1205.1271v1.pdf

** Supported in part by NSF CAREER award 1053605, ONR YIP award N000141110662, DARPA/AFRL award FA8650-11-1-7162 and a University of Maryland Research and Scholarship Award (RASA).

*** Supported in part by ERC Starting Grant NEWNET 279352, NCN grant N206567140 and Foundation for Polish Science.

† Supported in part by NSF CAREER award 1053605, ONR YIP award N000141110662, DARPA/AFRL award FA8650-11-1-7162 and a University of Maryland Research and Scholarship Award (RASA). The author is also with AT&T Labs–Research

‡ Supported by ERC Starting Grant PARAMTIGHT (No. 280152)

tion and covers its "shadow". We believe this general approach will be useful for showing the fixed-parameter tractability of other problems in directed graphs.

1 Introduction

The FEEDBACK VERTEX SET (FVS) problem has been one of the most extensively studied problems in the parameterized complexity community. Given a graph G and an integer k, it asks if there is a set T of size at most k which hits all cycles in G. FVS in both undirected and directed graphs was shown to be NP-hard by Karp [18]. A generalization of the FVS problem is the SUBSET FEEDBACK VERTEX SET (SFVS) problem: given a subset $S \subseteq V$ (resp. $S \subseteq E$), find a set T of size at most k such that T hits all cycles passing through a vertex of S (resp. an edge of S). It is easy to see that $S = V$ (resp. $S = E$) gives the FVS problem.

As compared to undirected graphs, FVS behaves differently on digraphs. In particular the trick of replacing each edge of an undirected graph G by arcs in both directions does not work: every feedback vertex set of the resulting digraph is a vertex cover of G and vice versa. Any other simple transformation does not seem possible either and thus the directed and undirected versions are very different problems. This is reflected in the best known approximation ratio for the directed versions as compared to the undirected problems: FVS in undirected graphs has an 2-approximation [1] while FVS in directed graphs has an $O(\log|V|\log\log|V|)$-approximation [13,24]. For SFVS in undirected graphs there is an 8-approximation [14] while the best-known approximation in directed graphs is $O(\min\{\log|V|\log\log|V|, \log^2|S|\})$ [13].

Rather than finding approximate solutions in polynomial time, one can look for exact solutions in time that is superpolynomial, but still better than the running time obtained by brute force solutions. In both the directed and the undirected versions of the feedback vertex set problems, brute force can be used to check in time $n^{O(k)}$ if a solution of size at most k exists: one can go through all sets of size at most k. Thus the problem can be solved in polynomial time if the optimum is assumed to be small. In the undirected case, we can do significantly better: since the first FPT algorithm for FVS in undirected graphs by Bodlaender [3] almost 21 years ago, there have been a number of papers [2,5,6,17] giving faster algorithms and the current fastest algorithm runs in $O^*(3^k)$ time [10] (the O^* notation hides all factors which are polynomial in size of input). That is, undirected FVS is fixed-parameter tractable parameterized by the size of the cutset we remove. Recall that a problem is *fixed-parameter tractable* (FPT) with a particular parameter p if it can be solved in time $f(p)n^{O(1)}$, where f is an arbitrary function depending only on p; see [12,15,22] for more background. For digraphs, the fixed-parameter tractability status of FVS was a long-standing open problem (almost 16 years) until Chen et al. [7] resolved it by giving an $O^*(4^k k!)$ algorithm. This was recently generalized by Bonsma and Lokshtanov [4] who gave a $O^*(47.5^k k!)$ algorithm for FVS in mixed graphs, i.e., graphs having both directed and undirected edges.

In the more general SUBSET FEEDBACK VERTEX SET problem, given an additional subset S of vertices and we want to find a set T of size at most k that hits all cycles passing through a vertex of S. In the edge version we are given a subset $S \subseteq E(G)$ and we want to hit all cycles passing through an edge of S. The vertex and edge versions are indeed known to be equivalent in the parameterized sense in both undirected and directed graphs. Recently Cygan et al. [11] and independently Kakimura et al. [16] have shown that SUBSET FEEDBACK VERTEX SET in undirected graphs is FPT parameterized by the size of the solution. Our main result is that SUBSET FEEDBACK VERTEX SET in digraphs is also fixed-parameter tractable parameterized by the size of the solution:

Theorem 1. (main result) SUBSET FEEDBACK VERTEX SET *(*SUBSET-DFVS*) in directed graphs can be solved in* $O^*(2^{2^{O(k)}})$ *time.*

Our techniques. As a first step, we use the standard technique of *iterative compression* [23] to argue that it is sufficient to solve the compression version of SUBSET-DFVS, where we assume that a solution T of size $k+1$ is given in the input and we have to find a solution of size k. Our algorithm for the compression problem is inspired by the algorithm of Marx and Razgon [21] for undirected MULTICUT and Chitnis et al. [8] for DIRECTED MULTIWAY CUT. We define the "shadow" of a solution X as those vertices that are disconnected from T (in either direction) after the removal of X. Our goal is to ensure that there is a solution whose shadow is empty, as finding such a shadowless solution can be a significantly easier task. For this purpose, we use the technique of "random sampling of important separators," which was introduced in [21] for undirected graphs and was generalized to directed graphs in [8]. We present this approach here in generic way that can be used for the following general family of problems:

Finding an $\mathcal{F}$-transversal for some T-connected $\mathcal{F}$
Input : A directed graph $G=(V,E)$, a positive integer k, a set $T \subseteq V$ and a set $\mathcal{F} = \{F_1, F_2, \dots, F_q\}$ of subgraphs such that $\mathcal{F}$ is T-connected, i.e., $\forall\, i \in [q]$ each vertex of F_i can reach some vertex of T by a walk completely contained in F_i and is reachable from some vertex of T by a walk completely contained in F_i.
Parameter : k
Question : Does there exist an $\mathcal{F}$-transversal $W \subseteq V$ with $|W| \leq k$, i.e., a set W such that $F_i \cap W \neq \emptyset$ for every $i \in [q]$?

It is easy to see that the above family includes DIRECTED MULTIWAY CUT (take T as the set of terminals and $\mathcal{F}$ as the set of all walks between different terminals) and the compression version of SUBSET-DFVS (take T as the solution that we want to compress and $\mathcal{F}$ as set of all S-closed-walks). For this family of problems, we can invoke the random sampling of important separators technique and obtain a set which is disjoint from a minimum solution and covers its shadow. Given such a set, we can use (some problem specific variant of) the "torso operation" to find an equivalent instance that has a shadowless solution. Therefore, we can focus on the simpler task of finding a shadowless solution. We believe this will be a useful opening step in the design of FPT algorithms for other transversal and cut problems on digraphs.

In the case of undirected MULTICUT [21], if there was a shadowless solution, then the problem could be reduced to an FPT problem called ALMOST 2SAT. In the case of DIRECTED MULTIWAY CUT [8], if there was a solution whose shadow is empty, then the problem could be reduced to the undirected version which was known to be FPT. For SUBSET-DFVS, the situation is a bit more complicated. As mentioned above, we first use the technique of iterative compression to reduce the problem to an instance where we are given a solution T and we want to find a disjoint solution of size at most k. We define the "shadows" with respect to the solution T that we want to compress whereas in [8], the shadows were defined with respect to the terminal set T. The "torso" operation we define in this paper is specific to the SUBSET-DFVS problem and differs from the one defined in [8]. Even after ensuring that there is a solution T' whose shadow is empty, we are not done unlike in [8]. We then analyze the structure of the graph $G \setminus T'$ and use "pushing" to branch on some important separators. Then for each branch, we need to do the whole process of random sampling of important separators to find a solution whose shadow is empty. This is followed again by branching on important separators. We repeat this two-step process until the budget k becomes zero.

2 Preliminaries

Observe, that a directed graphs contains no cycles if and only if it contains no closed-walks, for this reason throughout the article we use the term closed-walks, since it is sometimes easier to show a closed walk and avoid discussion whether it is a simple cycle or not. A feedback vertex set is a set of vertices that hits all the closed-walks of the graph.

Definition 2. **(feedback vertex set)** *Let G be a directed graph. A set $T \subseteq V(G)$ is a* feedback vertex set *of G if $G \setminus T$ does not contain any closed-walks.*

This gives rise to the DIRECTED FEEDBACK VERTEX SET (DFVS) problem where we are given a directed graph G and we want to find if G has a feedback vertex set of size at most k. DFVS was shown to be FPT by Chen et al. [7], closing a long-standing open problem in the parameterized complexity community.

In this paper we consider a generalization of the DFVS problem where given a set $S \subseteq V(G)$, we ask if there exists a vertex set of size $\leq k$ that hits all closed-walks passing through S.

SUBSET DIRECTED FEEDBACK VERTEX SET (SUBSET-DFVS)
Input : A directed graph $G = (V, E)$, a set $S \subseteq V(G)$ and a positive integer k.
Parameter : k
Question : Does there exist a set $T \subseteq V(G)$ with $|T| \leq k$ such that $G \setminus T$ has no closed walk containing a vertex of S?

It is easy to see that SUBSET-DFVS is a generalization of DFVS by setting $S = V(G)$. We also define an equivalent variant of SUBSET-DFVS where the set S is a subset of edges. First we define a special type of closed-walks:

Definition 3. *(**S-closed-walk**) Let $G = (V, E)$ be a digraph and $S \subseteq E(G)$. A closed walk (starting and ending at same vertex) C in G is said to be a S-*closed-walk *if it contains an edge from S.*

EDGE SUBSET DIRECTED FEEDBACK VERTEX SET (EDGE-SUBSET-DFVS)
Input : A directed graph $G = (V, E)$, a set $S \subseteq E(G)$ and a positive integer k.
Parameter : k
Question : Does there exist a set $T \subseteq V(G)$ with $|T| \leq k$ such that $G \setminus T$ has no S-closed-walks?

2.1 Iterative Compression

We now use the technique of *iterative compression* introduced by Reed et al. [23]. It has been used to obtain faster FPT algorithms for various problems [6,7,21]. In the first step we transform the SUBSET-DFVS problem into the following problem:

SUBSET-DFVS REDUCTION
Input : A directed graph $G = (V, E)$, a set $S \subseteq E(G)$, a positive integer k and a set $T \subseteq V$ such that $G \setminus T$ has no S-closed-walks .
Parameter : $k + |T|$
Question : Does there exist a set $T' \subseteq V(G)$ with $|T'| \leq k$ such that $G \setminus T'$ has no S-closed-walks?

Lemma 4. [$\star$][4] ***(power of iterative compression)*** SUBSET-DFVS *can be solved by $O(n)$ calls to an algorithm for the* SUBSET-DFVS REDUCTION *problem.*

Now we transform the SUBSET-DFVS REDUCTION problem into the following problem whose only difference is that the subset feedback vertex set in the output must be disjoint from the one in the input:

DISJOINT SUBSET-DFVS REDUCTION
Input : A directed graph $G = (V, E)$, a set $S \subseteq E(G)$, a positive integer k and a set $T \subseteq V$ such that $G \setminus T$ has no S-closed-walks.
Parameter : $k + |T|$
Question : Does there exist a set $T' \subseteq V(G)$ with $|T'| \leq k$ such that $T \cap T' = \emptyset$ and $G \setminus T'$ has no S-closed-walks?

Lemma 5. [$\star$] ***(adding disjointness)*** SUBSET-DFVS REDUCTION *can be solved by $O(2^{|T|})$ calls to an algorithm for the* DISJOINT SUBSET-DFVS REDUCTION *problem.*

From Lemmas 4 and 5, an FPT algorithm for DISJOINT SUBSET-DFVS REDUCTION translates into an FPT algorithm for SUBSET-DFVS with an additional blowup factor of $O(2^{|T|}n)$.

3 Covering the Shadow of a Solution

The purpose of this section is to present the "random sampling of important separators" technique used in [8] for DIRECTED MULTIWAY CUT in a generalized way that applies to SUBSET-DFVS as well. The technique consists of two steps:

[4] The proofs of the results labeled with $\star$ have been deferred to the full version of the paper.

1. First find a set Z *small* enough to be disjoint from a solution X (of size $\leq k$) but *large* enough to cover the "shadow" of X.
2. Then define a "torso" operation which uses the set Z to reduce the problem instance in such a way that X becomes a shadowless solution.

In this section, we define a general family of problems for which Step 1 can be efficiently performed. The general technique to execute Step 1 is very similar to what was done for DIRECTED MULTIWAY CUT [8] and so we defer most of the proofs to the full version of the paper. In Section 4, we show how Step 2 can be done for the specific problem of DISJOINT SUBSET-DFVS REDUCTION. First we start by defining shadows:

Definition 6. **(separator)** *Let $G = (V,E)$ be a directed graph. Given two disjoint non-empty sets $X,Y \subseteq V$ we call a set W of vertices as an $X - Y$* separator *if W is disjoint from $X \cup Y$ and there is no walk from X to Y in $G \setminus W$. A set W is a* minimal $X - Y$ separator *if no proper subset of W is an $X - Y$ separator.*

Definition 7. **(shadow)** *Let G be graph and $W \subseteq V(G)$. Then for $v \in V(G)$ we say that v is in the "forward shadow" $f_{G,T}(W)$ of W (with respect to T), if W is a $T - \{v\}$ separator in G. Similarly, we say that v is in the "reverse shadow" $r_{G,T}(W)$ of W (with respect to T), if W is a $\{v\} - T$ separator in G.*

That is, we can imagine T as a light source with light spreading on the directed edges. The forward shadow of W is the set of vertices that remain dark if the set W blocks the light. In the reverse shadow, we imagine that light is spreading on the edges backwards. We abuse the notation slightly and write $v - T$ separator instead of $\{v\} - T$ separator. We also drop G and T from the subscript if they are clear from the context. Note that W itself is not in the shadow of W (as a $T - v$ or $v - T$ separator needs to be disjoint from T and v), that is, W and $f_{G,T}(W) \cup r_{G,T}(W)$ are disjoint.

Let $G = (V,E)$ be a directed graph and $T \subseteq V(G)$. Consider $\mathcal{F} = \{F_1, F_2, \ldots, F_q\}$ which is a set of subgraphs of G. We define the following property:

Definition 8. ***(T-connected)*** *Let $\mathcal{F} = \{F_1, F_2, \ldots, F_q\}$ be a set of subgraphs of G. Then $\mathcal{F}$ is said to be T-*connected *if $\forall\ i \in [q]$, each vertex of the subgraph F_i can reach some vertex of T by a walk completely contained in F_i and is reachable from some vertex of T by a walk completely contained in F_i.*

For a set $\mathcal{F}$ of subgraphs of G, a transversal is a set of vertices which hits each subgraph in $\mathcal{F}$. We note that the subgraphs in $\mathcal{F}$ are given implicitly to us.

Definition 9. ***($\mathcal{F}$-transversal)*** *Let $\mathcal{F} = \{F_1, F_2, \ldots, F_q\}$ be a set of subgraphs of G. Then W is said to be an $\mathcal{F}$-*transversal *if $\forall\ i \in [q]$ we have $F_i \cap W \neq \emptyset$.*

The main theorem of this section is the following:

Theorem 10. $[\star]$**(randomized covering of the shadow)** *Let $T \subseteq V(G)$. In $O^*(4^k)$ time, we can construct a set $Z \subseteq V(G)$ such that for any set of subgraphs $\mathcal{F}$ which is T-connected, if there exists an $\mathcal{F}$-transversal of size $\leq k$, then the following holds with probability $2^{-2^{O(k)}}$: there is an $\mathcal{F}$-transversal X of size $\leq k$ satisfying*

1. $X \cap Z = \emptyset$.
2. Z *covers the shadow of* X.

We also prove the following derandomized version of Theorem 10:

Theorem 11. [$\star$]**(deterministic covering of the shadow)** *Let* $T \subseteq V(G)$. *In* $O^*(2^{2^{O(k)}})$ *time, we can construct a set* $\{Z_1, Z_2, \ldots, Z_t\}$ *where* $t = 2^{2^{O(k)}} \log^2 n$ *such that for any set of subgraphs* $\mathcal{F}$ *which is* T*-connected, if there exists an* $\mathcal{F}$*-transversal of size* $\leq k$*, then there is an* $\mathcal{F}$*-transversal* X *of size* $\leq k$ *such that for at least one* $i \in [t]$ *we have*

1. $X \cap Z_i = \emptyset$.
2. Z_i *covers the shadow of* X.

In DIRECTED MULTIWAY CUT, T was the set of terminals and the set $\mathcal{F}$ was the set of all walks from one vertex of T to another vertex of T. In SUBSET-DFVS , the set T is the solution that we want to compress and $\mathcal{F}$ is the set of all closed S-walks passing through some vertex of T.

We say that an $\mathcal{F}$-transversal T' is *shadowless* if $f(T') \cup r(T') = \emptyset$. Note that if T' is a shadowless solution, then in the graph $G \setminus T'$, each vertex is reachable from some vertex of T and can reach some vertex of T. In Section 5 we will see how we can make progress in DISJOINT SUBSET-DFVS REDUCTION if there exists a shadowless solution. So we would like to transform the instance in such a way that ensures the existence of a shadowless solution, by taking the torso (Section 4) and make progress by using the BRANCH algorithm from Section 5.

4 Reducing the Instance by Torso

We use the algorithm of Theorem 11 to construct a set Z of vertices that we want to get rid of. The second ingredient of our algorithm is an operation that removes a set of vertices without making the problem any easier. This transformation can be conveniently described using the operation of taking the *torso* of a graph. From this point onwards in the paper, we do not follow [8]. In particular, the *torso* operation is problem-specific. For DISJOINT SUBSET-DFVS REDUCTION, we define it as follows:

Definition 12. **(torso)** *Let* (G, S, T, k) *be an instance of* DISJOINT SUBSET-DFVS REDUCTION *and* $C \subseteq V(G)$. *The graph* $\mathrm{torso}(G,C)$ *has vertex set* C *and there is (directed) edge* (a,b) *in* $\mathrm{torso}(G,C)$ *if there is an* $a \to b$ *walk in* G *whose internal vertices are not in* C. *Furthermore, we add the edge* (a,b) *to* S *if there is an* $a \to b$ *walk in* G *which contains an edge from* S *and whose internal vertices are not in* C.

In particular, if $a, b \in C$ and (a,b) is a directed edge of G, then $\mathrm{torso}(G,C)$ contains (a,b) as well. Thus $\mathrm{torso}(G,C)$ is a supergraph of the subgraph of G induced by C. The following lemma shows that the *torso* operation preserves S-closed-walks inside C.

Lemma 13. [$\star$] **(torso preserves S-closed-walks)** *Let* G *be a directed graph and* $C \subseteq V(G)$. *Let* $G' = \mathrm{torso}(G,C)$, $v \in C$ *and* $W \subseteq C$. *Then* $G \setminus W$ *has an* S*-closed-walk passing through* v *if and only if* $G' \setminus W$ *has an* S*-closed-walk passing through* v.

If we want to remove a set Z of vertices, then we create a new instance by taking the torso on the *complement* of Z:

Definition 14. *Let $I = (G,S,T,k)$ be an instance of* DISJOINT SUBSET-DFVS REDUCTION *and $Z \subseteq V(G) \setminus T$. The reduced instance $I/Z = (G',S,T,p)$ is defined as*

- $G' = \mathrm{torso}(G, V(G) \setminus Z)$
- *S is modified as specified in Definition 12.*

The following lemma states that the operation of taking the torso does not make the DISJOINT SUBSET-DFVS REDUCTION problem easier for any $Z \subseteq V(G) \setminus T$ in the sense that any solution of the reduced instance I/Z is a solution of the original instance I. Moreover, if we perform the torso operation for a Z that is large enough to cover the shadow of some solution T^* and also small enough to be disjoint from T^*, then T^* becomes a shadowless solution for the reduced instance I/Z.

Lemma 15. [$\star$] **(creating a shadowless instance)** *Let $I = (G,S,T,k)$ be an instance of* DISJOINT SUBSET-DFVS REDUCTION *and $Z \subseteq V(G) \setminus T$.*

1. *If I is a no-instance, then the reduced instance I/Z is also a no-instance.*
2. *If I has solution T' with $f_{G,T}(T') \cup r_{G,T}(T') \subseteq Z$ and $T' \cap Z = \emptyset$, then T' is a shadowless solution of I/Z.*

For every Z_i in the output of Theorem 11, we use the torso operation to remove the vertices in Z_i. We prove that this procedure is safe by showing the following:

Lemma 16. [$\star$] *Let $I = (G,S,T,k)$ be an instance of* DISJOINT SUBSET-DFVS REDUCTION. *Let the sets in the output of Theorem 11 be $Z_1, Z_2, \ldots, Z_t$. For every $i \in [t]$, let G_i be the reduced instance G/Z_i.*

1. *If I is a no-instance, then G_i is also a no-instance for every $i \in [t]$.*
2. *If I is a yes-instance, then there exists a solution T^* of I which is a shadowless solution of some G_j for some $j \in [t]$.*

5 Finding a Shadowless Solution

Consider an instance (G,S,T,k) of DISJOINT SUBSET-DFVS REDUCTION. First, let us assume that from each vertex of T, we can reach an edge of S, since otherwise we can clearly remove such a vertex from the set T, without violating the assumption that $G \setminus T$ has no S-closed walk. Next, we branch on all $2^{2^{O(k)}} \log^2 n$ choices for Z taken from $\{Z_1, Z_2, \ldots, Z_t\}$ (given by Theorem 11) and build a reduced instance I/Z for each choice of Z. By Lemma 15, if I is a no-instance then I/Z_j is a no-instance for each $j \in [t]$. If I is a yes-instance, then by Lemma 16, there is at least one $i \in [t]$ such that I has a solution T' which is a solution, and in fact a shadowless solution, for the reduced instance I/Z_i.

So for the reduced instance I/Z_i we know that each vertex in $G \setminus T'$ can reach some vertex of T and can be reached from a vertex of T. Since T' is a solution for the

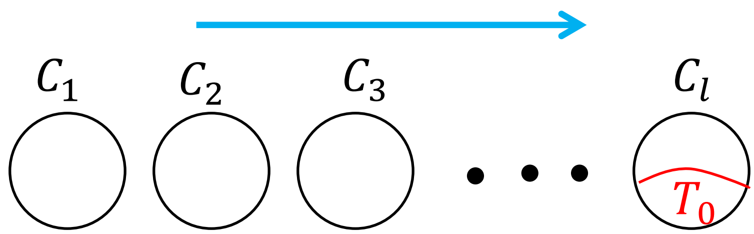

Fig. 1. We arrange the strong components of $G \setminus T'$ in topological order so that the only possible direction of edges between the strong components is as shown by the blue arrow. We will claim later that the last component C_ℓ must contain a non-empty subset T_0 of T and further that no edge of S can be present within C_ℓ. This allows us to make some progress as we shall see in Theorem 21

instance (G,S,T,k) of DISJOINT SUBSET-DFVS REDUCTION, we know that $G \setminus T'$ does not have any S-closed-walks. Consider a topological ordering say $C_1, C_2, \ldots, C_\ell$ of the strong components of $G \setminus T'$, i.e., there can be an edge from C_i to C_j only if $i < j$. We illustrate this in Figure 1.

Definition 17. ***(starting points of** S) Let S^- be the set of starting points of edges in S, i.e., $S^- = \{u \mid (u,v) \in S\}$.*

Lemma 18. $[\star]$ ***(properties of** C_ℓ) Let C_ℓ be the last strong component in the topological ordering of $G \setminus T'$ (refer to Figure 1). Then*

1. *C_ℓ contains a non-empty subset T_0 of T.*
2. *No edge of S is present within C_ℓ.*
3. *S^- is disjoint from C_ℓ.*

Since T_0 is the subset of T present in C_ℓ and only edges between strong components can be from left to right, we have that there are no $T_0 - (T \setminus T_0)$ walks in $G \setminus T'$. Along with the third claim of Lemma 18, this implies that the solution T' contains a $T_0 - (S^- \cup (T \setminus T_0))$ separator. We now define a special type of separators:

Definition 19. **(important separator)** *Let G be a digraph and let $X, Y \subseteq V$ be two disjoint non-empty sets. A minimal $X - Y$ separator W is called an* important $X - Y$ separator *if there is no $X - Y$ separator W' with $|W'| \leq |W|$ and $R^+_{G \setminus W}(X) \subset R^+_{G \setminus W'}(X)$, where $R^+_A(X)$ is the set of vertices reachable from X in A.*

For any $X, Y \subseteq V(G)$, the following lemma (proved in [8]) gives an upper bound the number of important $X - Y$ separators of size at most k:

Lemma 20. $[\star]$**(number of important separators)** *Let $X, Y \subseteq V(G)$ be disjoint sets in a* directed *graph G. Then for every $k \geq 0$ there are at most 4^k important $X - Y$ separators of size at most k. Furthermore, we can enumerate all these separators in time $O^*(4^k)$.*

Algorithm 1 BRANCH

Input: An instance $I = (G, S, T, k)$ of DISJOINT SUBSET-DFVS REDUCTION.
Output: A new set of $2^{O(k+|T|)}$ instances of DISJOINT SUBSET-DFVS REDUCTION where the budget k is reduced.

1: **for** every non-empty subset T_0 of T: **do**
2: Use Lemma 20 to enumerate all the at most 4^k important $T_0 - (S^- \cup (T \setminus T_0))$ separators of size at most k.
3: Let the important separators be $\mathcal{B} = \{B_1, B_2, \ldots, B_m\}$.
4: **for** each $i \in [m]$ **do**
5: Create a new instance $I_{T_0,i} = (G \setminus B_i, S, T, k - |B_i|)$ of DISJOINT SUBSET-DFVS REDUCTION.

By "pushing", we have the following theorem:

Theorem 21. $[\star]$ ***(pushing)*** *Either T' contains an important $T_0 - (S^- \cup (T \setminus T_0))$ separator or there is another solution T'' of the instance (G, S, T, k) such that $|T''| \leq |T'|$ and T'' contains an important $T_0 - (S^- \cup (T \setminus T_0))$ separator.*

Theorem 21 tells us that there is always a minimum solution which contains an important $T_0 - (S^- \cup (T \setminus T_0))$ separator where T_0 is a non-empty subset of T. This gives $2^{|T|} - 1$ choices for T_0. For each guess of T_0 we enumerate all the at most 4^k important $T_0 - (S^- \cup (T \setminus T_0))$ separators of size at most k in time $O^*(4^k)$ as given by Lemma 20. This gives the following natural branching algorithm:

6 FPT Algorithm for DISJOINT SUBSET-DFVS REDUCTION

Lemma 16 and the BRANCH algorithm together combine to give a *bounded-search-tree* FPT algorithm for DISJOINT SUBSET-DFVS REDUCTION as follows:

FPT Algorithm for SUBSET-DFVS

Step 1: At the first step, for a given instance $I = (G, S, T, k)$, use Theorem 11 to obtain a set of instances $\{Z_1, Z_2, \ldots, Z_t\}$ where $2^{2^{O(k)}} \log^2 n$ and Lemma 16 implies

- If I is a no-instance, then all the reduced instances $G_j = G/Z_j$ are no-instances for all $j \in [t]$
- If I is a yes-instance, then there is at least one $i \in [t]$ such that there is a solution T^* for I which is a shadowless solution for the reduced instance $G_i = G/Z_i$.

So at this step we branch into $2^{2^{O(k)}} \log^2 n$ directions.

Step 2 : For each of the instances obtained from the above step, we run the BRANCH algorithm to obtain a set of $2^{O(k+|T|)}$ instances where in each case either the answer is NO, or the budget k is reduced.

We then repeatedly perform Steps 1 and 2. Note that for every instance, one execution of steps 1 and 2 gives rise to $2^{2^{O(k)}} \log^2 n$ instances such that for each instance, either we know that the answer is NO or the budget k has decreased, because we have assumed that from each vertex of T one can reach the set S^-, and hence each important separator is non-empty. Therefore, considering a level as an execution of Step 1 followed by Step 2, the height of the search tree is at most k. Each time we branch into at most $2^{2^{O(k)}} \log^2 n$ directions (as $|T|$ is at most $k+1$). Hence the total number of nodes in the search tree is $\left(2^{2^{O(k)}} \log^2 n\right)^k$.

Lemma 22. [$\star$] *For every n and $k \leq n$, we have* $(\log n)^k \leq (2k \log k)^k + \frac{n}{2^k}$

So the total number of nodes in the search tree is $\left(2^{2^{O(k)}} \log^2 n\right)^k = \left(2^{2^{O(k)}}\right)^k (\log^2 n)^k = (2^{2^{O(k)}})(\log^2 n)^k \leq (2^{2^{O(k)}})\left((2k \log k)^k + \frac{n}{2^k}\right)^2 \leq 2^{2^{O(k)}} n^2$. We then check the leaf nodes and see if there are any S-closed-walks left even after the budget k has become zero. If the graph at least one of the leaf nodes is S-closed-walk free, then the given instance is a yes-instance. Otherwise it is a no-instance. This gives an $O^*(2^{2^{O(k)}})$ algorithm for DISJOINT SUBSET-DFVS REDUCTION. By Lemma 4, we have an $O^*(2^{2^{O(k)}})$ algorithm for the SUBSET-DFVS problem.

7 Conclusion and Open Problems

In this paper we gave the first fixed-parameter algorithm for DIRECTED SUBSET FEEDBACK VERTEX SET parameterized by the size of the solution. Our algorithm used various tools from the FPT world such as iterative compression, bounded-depth search trees, random sampling of important separators, etc. We also gave a general family of problems for which we can do random sampling of important separators and obtain a set which is disjoint from a minimum solution and covers its shadow. We believe this general approach will be useful for deciding the fixed-parameter tractability status of other problems in digraphs where we do not know that much techniques unlike undirected graphs.

The next natural question is whether SUBSET-DFVS has a polynomial kernel or can we rule out such a possibility under some standard assumptions? The recent developments [9,19,20] in the field of kernelization may be useful in answering this question. Another question is to try and reduce the complexity of our algorithm to single exponential. In the field of exact exponential algorithms, Razgon gave a $O(1.9977^n)$ algorithm for DFVS. It would be interesting to break the trivial $2^n n^{O(1)}$ barrier for SUBSET-DFVS.

References

1. Bafna, V., Berman, P., Fujito, T.: A 2-approximation algorithm for the undirected feedback vertex set problem. SIAM J. Discrete Math. 12(3), 289–297 (1999)

2. Becker, A., Bar-Yehuda, R., Geiger, D.: Randomized algorithms for the loop cutset problem. J. Artif. Intell. Res. (JAIR) 12, 219–234 (2000)
3. Bodlaender, H.L.: On disjoint cycles. In: WG. pp. 230–238 (1991)
4. Bonsma, P., Lokshtanov, D.: Feedback vertex set in mixed graphs. In: WADS (2011)
5. Cao, Y., Chen, J., Liu, Y.: On feedback vertex set new measure and new structures. In: SWAT (2010)
6. Chen, J., Fomin, F.V., Liu, Y., Lu, S., Villanger, Y.: Improved algorithms for feedback vertex set problems. J. Comput. Syst. Sci. 74(7), 1188–1198 (2008)
7. Chen, J., Liu, Y., Lu, S., O'Sullivan, B., Razgon, I.: A fixed-parameter algorithm for the directed feedback vertex set problem. In: STOC. pp. 177–186 (2008)
8. Chitnis, R.H., Hajiaghayi, M., Marx, D.: Fixed-parameter tractability of directed multiway cut parameterized by the size of the cutset. In: SODA. pp. 1713–1725 (2012)
9. Cygan, M., Kratsch, S., Pilipczuk, M., Pilipczuk, M., Wahlström, M.: Clique cover and graph separation: New incompressibility results. CoRR abs/1111.0570 (2011)
10. Cygan, M., Nederlof, J., Pilipczuk, M., Pilipczuk, M., van Rooij, J.M.M., Wojtaszczyk, J.O.: Solving connectivity problems parameterized by treewidth in single exponential time. In: FOCS. pp. 150–159 (2011)
11. Cygan, M., Pilipczuk, M., Pilipczuk, M., Wojtaszczyk, J.O.: Subset feedback vertex set is fixed-parameter tractable. In: ICALP (1). pp. 449–461 (2011)
12. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer-Verlag (1999), 530 pp.
13. Even, G., Naor, J., Schieber, B., Sudan, M.: Approximating minimum feedback sets and multi-cuts in directed graphs. In: IPCO. pp. 14–28 (1995)
14. Even, G., Naor, J., Zosin, L.: An 8-approximation algorithm for the subset feedback vertex set problem. SIAM J. Comput. 30(4), 1231–1252 (2000)
15. Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer-Verlag (2006), 493 pp.
16. Kakimura, N., Kawarabayashi, K., Kobayashi, Y.: Erdös-pósa property and its algorithmic applications: parity constraints, subset feedback set, and subset packing. In: SODA. pp. 1726–1736 (2012)
17. Kanj, I.A., Pelsmajer, M.J., Schaefer, M.: Parameterized algorithms for feedback vertex set. In: IWPEC. pp. 235–247 (2004)
18. Karp, R.M.: Reducibility among combinatorial problems. In: Complexity of Computer Computations. pp. 85–103 (1972)
19. Kratsch, S., Wahlström, M.: Representative sets and irrelevant vertices: New tools for kernelization. CoRR abs/1111.2195 (2011)
20. Kratsch, S., Wahlström, M.: Compression via matroids: a randomized polynomial kernel for odd cycle transversal. In: SODA. pp. 94–103 (2012)
21. Marx, D., Razgon, I.: Fixed-parameter tractability of multicut parameterized by the size of the cutset. In: STOC. pp. 469–478 (2011)
22. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford University Press (2006)
23. Reed, B.A., Smith, K., Vetta, A.: Finding odd cycle transversals. Oper. Res. Lett. (2004)
24. Seymour, P.D.: Packing directed circuits fractionally. Combinatorica 15(2), 281–288 (1995)