

On the Complexity of Singly Connected Vertex Deletion

Avinandan Das¹, Lawqueen Kanesh¹, Jayakrishnan Madathil^{1(⊠)}, Komal Muluk¹, Nidhi Purohit², and Saket Saurabh^{1,2}

¹ The Institute of Mathematical Sciences, HBNI, Chennai, India adas33745@gmail.com, {lawqueen,jayakrishnanm,saket}@imsc.res.in, komalmuluk15@gmail.com

² Department of Informatics, University of Bergen, Bergen, Norway nidhipurohit950gmail.com

Abstract. A digraph D is singly connected if for all ordered pairs of vertices $u, v \in V(D)$, there is at most one path in D from u to v. In this paper, we study the SINGLY CONNECTED VERTEX DELETION (SCVD) problem: Given an *n*-vertex digraph D and a positive integer k, does there exist a set $S \subseteq V(D)$ such that $|S| \leq k$ and D - S is singly connected? This problem may be seen as a directed counterpart of the (UNDIRECTED) FEEDBACK VERTEX SET problem, as an undirected graph is singly connected if and only if it is acyclic. SCVD is known to be NPhard on general digraphs. We study the complexity of SCVD on various classes of digraphs such as tournaments, and various generalisations of tournaments such as digraphs of bounded independence number, inand out-tournaments and local tournaments. We show that unlike the FEEDBACK VERTEX SET ON TOURNAMENTS (FVST) problem, SCVD is polynomial time solvable on tournaments. In addition, we show that SCVD is polynomial time solvable on digraphs of bounded independence number, and on the class of acyclic local tournaments. We also study the parameterized complexity of SCVD, with k as the parameter, on the class of in-tournaments. And we show that on in-tournaments (and outtournaments), SCVD admits a fixed-parameter tractable algorithm and a quadratic kernel. We also show that on the class of local tournaments, which is a sub-class of in-tournaments, SCVD admits a linear kernel.

Keywords: Singly connected digraphs \cdot FPT algorithm \cdot Kernel \cdot Bounded independence number \cdot Tournaments \cdot Local tournaments

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant no. 819416), and the Swarnajayanti Fellowship grant DST/SJF/MSA-01/2017-18.



© Springer Nature Switzerland AG 2020

L. Gasieniec et al. (Eds.): IWOCA 2020, LNCS 12126, pp. 237–250, 2020. https://doi.org/10.1007/978-3-030-48966-3_18

1 Introduction

A digraph D is said to be singly connected if for every (ordered) pair of vertices uand v of D, there is at most one (directed) path in D from u to v. In this paper, we study the SINGLY CONNECTED VERTEX DELETION (SCVD for short) problem, where the goal is to test if a given digraph can be made singly connected by deleting a few vertices. This problem may be seen as a directed counterpart of the FEEDBACK VERTEX SET problem. To see this, let us first define undirected singly connected graphs. An undirected graph G is said to be singly connected if for every pair of vertices u and v of G, there is at most one path in G between u and v. But note that an undirected graph is singly connected if and only if it is acyclic. So, the problem of checking whether it is possible to delete at most k vertices from a given graph to make it singly connected is the same as the problem of checking whether it is possible to delete at most k vertices to make a graph acyclic. This precisely is the FEEDBACK VERTEX SET (FVS) problem. (A feedback vertex set of a graph is a set of vertices whose deletion will render the graph acvclic.) The complexity of FVS has been studied extensively [3,10,12-14,17,21,26,27,31-34,37]. FVS, in fact, was one of Karp's 21 NP-hard problems [28]. As for its algorithmic tractability, FVS is fixed-parameter tractable (when parameterized by the solution size) [21] and it admits a quadratic kernel [40]. FVS also admits constant factor approximation algorithms [3, 8, 16, 24].

Coming back to digraphs, the DIRECTED FEEDBACK VERTEX SET (DFVS) problem asks if a given digraph can be made acyclic by deleting at most k vertices. Naturally, this problem has been deemed the appropriate directed counterpart of FEEDBACK VERTEX SET, and has been studied in the frameworks of approximation algorithms [39] and parameterized algorithms [15]. Although the parameterized complexity of DFVS had been raised as an open problem since the emergence of parameterized algorithms in the early 90s [20,22], it was settled only in 2008 by Chen et al. [15]. They showed that the problem admits a $4^{k}k!n^{\mathcal{O}(1)}$ time algorithm, and hence is fixed-parameter tractable when parameterized by k. If fixed-parameter tractability of DFVS remained open for years, the kernelization complexity of the problem proved even more elusive. While the question whether DFVS (parameterized by k) admits a polynomial kernel still remains unresolved, several attempts have been made to study the kernelization complexity of "DFVS-adjacent" problems. These include studying the problem with larger parameters [9,35], restricting the input digraph to smaller classes [1,7] and imposing more conditions on the acyclic digraph that results from the deletion of a feedback vertex set [2, 36].

While FVS and DFVS generated a large volume of literature, the SCVD problem, already known to be NP-hard [19], received little attention from the parameterized complexity community. In this paper, as a first step, we start an investigation into the complexity of SCVD on various classes of digraphs such as tournaments, local tournaments, digraphs of bounded independence number etc. We formally define the problem below.

SINGLY CONNECTED VERTEX DELETION (SCVD) Parameter: k Input: A digraph D and a non-negative integer k. Question: Does there exist a set $S \subseteq V(D)$ such that $|S| \leq k$ and D - S is singly connected?



Fig. 1. Obstructions to acyclic and singly connected tournaments.

As observed earlier, an undirected graph is singly connected if and only if it is acyclic. But notice that this property does not hold for digraphs. A directed cycle, for instance, is singly connected. And consider a digraph on 3 vertices, say, x, y and z, and with arcs (x, y), (y, z) and (x, z). This digraph, while acyclic, is not singly connected. It is not surprising then that SCVD and DFVS show markedly different behaviour. This is perhaps best illustrated by the fact that while DFVS is NP-hard on tournaments, we show that SCVD is polynomial time solvable on tournaments (Lemma 2). This difference in behaviour appears even starker considering the fact that these two problems require that "obstructions" with a "similar structure" be hit. Notice that obstructions to an acyclic tournament are directed triangles, i.e., all triplets of vertices x, y and z with arcs (x, y), (y, z) and (z, x), whereas obstructions to a singly connected tournament are all triplets of vertices x, y and z with arcs (x, y), (y, z) and (x, z) (see Fig. 1).

A digraph D is not singly connected if and only if there exists a pair of vertices u and v such that D contains two paths from u to v. It is not difficult to see that a digraph D is not singly connected if and only if there exists a pair of vertices u and v such that D contains two *internally vertex disjoint* paths from u to v. (See Lemma 1.) Two internally vertex disjoint paths between a pair of vertices of a digraph constitute a cycle in the underlying undirected graph. That is, the obstructions to a singly connected digraph are cycles in the underlying undirected graph. But notice that not every cycle in the underlying undirected graph is necessarily an obstruction. Thus both DFVS and SCVD require us to examine if a subset of the cycles in the underlying undirected graph can be hit with a few vertices.

Our Contribution. We study the SCVD problem on several well-studied classes of digraphs such as tournaments, α -bounded digraphs, local tournaments, etc.

A digraph D is said to be a *tournament* if for every pair of vertices u and v of D, exactly one of the arcs (u, v) and (v, u) is present in D. The class of α -bounded

digraphs were introduced by Fradkin and Seymour [23] as a generalisation of tournaments. For a fixed positive integer α , a digraph D is said to be α -bounded if the size of a maximum independent set of the underlying undirected graph of D is at most α . Note that tournaments are 1-bounded digraphs. Local tournaments are yet another generalisation of tournaments. A digraph D is said to be an *in-tournament* (resp. *out-tournament*) if for every vertex v of D, the set of in-neighbours (resp. out-neighbours) of v induces a tournament. A digraph D is said to be a *local tournament* if it is both an in-tournament if D is both a directed acyclic graph and a local tournament. (See, for example, the chapter on locally semi-complete digraphs [5] in the monograph edited by Bang-Jensen and Gutin [6] for a survey of literature on these classes of digraphs.)

We show that SINGLY CONNECTED VERTEX DELETION

- is polynomial time solvable on tournaments and α -bounded digraphs,
- is polynomial time solvable on acyclic local tournaments,
- has a $2^k n^{\mathcal{O}(1)}$ algorithm and $\mathcal{O}(k^2)$ vertex kernel on in- and out-tournaments,
- and has an $\mathcal{O}(k)$ vertex kernel on local tournaments.

The polynomial time solvability of SCVD on tournaments follows from a simple observation that no tournament with at least four vertices can be singly connected. A similar result holds for α -bounded digraphs as well: no α -bounded digraph with at least $2\alpha^2 + 4\alpha$ vertices can be singly connected. In order to prove this observation, we use the Gallai-Milgram theorem [25], which says that the vertices of a digraph D can be covered by a disjoint collection of paths, such that the number of paths does not exceed the size of a maximum independent set of the underlying undirected graph of D. On acyclic local tournaments, we design a polynomial time algorithm that computes a minimum-sized vertex subset whose deletion will make the digraph singly connected. Our algorithm uses the fact that every connected local tournament has a Hamiltonian path [4], which in turn, implies that every connected acyclic local tournament has a unique topological ordering. We show that SCVD on in-tournaments (and out-tournaments) can be reduced to the 3-HITTING SET problem, and thus admits a simple $3^k n^{\mathcal{O}(1)}$ time branching algorithm and an $\mathcal{O}(k^2)$ vertex kernel. But we use the technique of iterative compression to design a $2^k n^{\mathcal{O}(1)}$ algorithm for SCVD on in and outtournaments. And our $\mathcal{O}(k)$ vertex kernel for SCVD on local tournaments relies on the fact that for a local tournament D and a set of vertices $S \subseteq V(D)$ such that D-S is singly connected, no vertex in S can have more than a constant number of neighbours in $V(D) \setminus S$.

Related Work on Singly-Connected Digraphs. As noted above, the SCVD problem was shown to be NP-hard by Dietzfelbinger and Jaberi [19]. The reduction in [19], in fact, shows that the problem is NP-hard even on directed acyclic graphs. Their work shows that the arc-deletion version of the problem is also NP-hard, i.e., the problem of testing whether a given digraph can be made singly connected by deleting at most a given number of arcs. As for recognising singly connected digraphs, i.e., the problem of testing whether a given digraph is singly connected digraphs.

connected, Buchsbaum and Carlisle [11] gave an algorithm that runs in $\mathcal{O}(n^2)$ time, where *n* is the number of vertices in the input digraph. Khuller [29,30] gave another $\mathcal{O}(n^2)$ algorithm for this problem. Dietzfelbinger and Jaberi [19] presented a refined version of the algorithm of Buchsbaum and Carlisle [11] that runs in time $\mathcal{O}(s \cdot t + m)$, where *m* is the number of arcs, and *s* and *t* respectively are the number of sources and sinks in the input digraph.

2 Preliminaries

For a positive integer n, we denote the set $\{1, 2, ..., n\}$ by [n]. Let S be a finite set, and let σ be an ordering of the elements of S. For $x, y \in S$, we write $x <_{\sigma} y$ to mean that x appears before y in the ordering σ . And we write $x \leq_{\sigma} y$ to mean that either x = y or $x <_{\sigma} y$.

Digraphs. For a digraph D, V(D) denotes the vertex set and A(D) denotes the arc set of D. For a vertex $v \in V(D)$, $N_D^+(v)$ denotes the set of all out-neighbours of v, and $N_D^-(v)$ denotes the set of all in-neighbours of v, that is, $N_D^+(v) = \{u \in V(D) \mid (v, u) \in A(D)\}$ and $N_D^-(v) = \{u \in V(D) \mid (u, v) \in A(D)\}$. And $N_D(v)$ denotes the set of all neighbours of v in the underlying undirected graph of D, that is, $N_D(v) = N_D^+(v) \cup N_D^-(v)$. Also, we define $N_D^+[v] = N_D^+(v) \cup \{v\}$, $N_D^-[v] = N_D^-(v) \cup \{v\}$ and $N_D[v] = N_D(v) \cup \{v\}$. For a set $X \subseteq V(D)$, we define $N_D(X) = \cup_{v \in X} N_D(v)$.

For a set $A' \subseteq A(D)$, D - A' denotes the digraph $(V(D), A(D) \setminus A')$. For a set $V' \subseteq V(D)$, D[V'] denotes the subgraph of D induced by V'. Similarly, for $S \subseteq V(D)$, D - S denotes the digraph $D[V(D) \setminus S]$.

A digraph D is said to be *connected* if the underlying undirected graph of D is connected. A digraph D on 3 vertices, say, x, y and z, is said to be an *acyclic* triangle if $A(D) = \{(x, y), (y, z), (x, z)\}.$

A path cover \mathcal{P} of a digraph D is a disjoint collection of paths in D such that for every vertex $v \in V(D)$, there is a path $P \in \mathcal{P}$ such that $v \in V(P)$.

For the sake of convenience, we repeat below some of the definitions we introduced in Sect. 1. Recall that a directed graph D is a tournament if for every pair of distinct vertices $u, v \in V(D)$, either $(u, v) \in A(D)$ or $(v, u) \in A(D)$, but not both.

Definition 1 (Out-tournament and In-tournament). A directed graph D is an out-tournament (resp. in-tournament) if for all $v \in V(D)$, $D[N_D^+(v)]$ (resp. $D[N_D^-(v)]$) is a tournament.

Definition 2 (Local tournament). A directed graph D is a local tournament if D is both an out-tournament and an in-tournament.

Note that, by the definition of singly connected digraphs, a digraph D is *not* singly connected if there exist two paths from u to v for $u, v \in V(D)$. Note that these two paths need not be internally vertex disjoint. But the following lemma says that we may as well assume that the two paths are internally vertex disjoint.

Lemma 1 (\star^1). A directed graph D is not singly connected if and only if there exist two vertices $u, v \in V(D)$ such that there exist two internally vertex disjoint paths from u to v.

3 Singly Connected Vertex Deletion on α-bounded Digraphs and Acyclic Local Tournaments

In this section, we study the optimisation version of SCVD restricted to α bounded digraphs and acyclic local tournaments, and prove that the problem is polynomial time solvable on both these classes of digraphs. That is, we consider the following problem.

MINIMUM SINGLY CONNECTED VERTEX DELETION (MIN-SCVD) Input: A digraph D. Output: A minimum-sized set $S \subseteq V(D)$ such that D-S is singly connected.

3.1 MIN-SCVD on α -bounded Digraphs

In this section, we prove that MIN-SCVD is polynomial time solvable on α -bounded digraphs. Specifically, we prove the following theorem.

Theorem 1. MIN-SCVD can be solved in time $\mathcal{O}(n^{\alpha(2\alpha+3)})$ on α -bounded digraphs, where n is the number of vertices of the input α -bounded digraph.

We first consider the problem on tournaments. Although Theorem 1 applies to tournaments as well, as tournaments are 1-bounded digraphs, we consider tournaments separately, and prove that the MIN-SCVD problem can be solved in $\mathcal{O}(n^3)$ time on tournaments. This result follows from a simple observation that no tournament with 4 or more vertices can be singly connected.

Lemma 2 (*). Any tournament on at least 4 vertices is not singly connected.

Using Lemma 2 and the fact that tournaments are hereditary, we get the following corollary.

Corollary 1 (*). MIN-SCVD on tournaments is solvable in $\mathcal{O}(n^3)$ time.

We now move on to α -bounded digraphs, and prove Theorem 1. We prove below that no α -bounded digraph with at least $\alpha(2\alpha + 4)$ vertices is singly connected. Note that this immediately gives an $\mathcal{O}(n^{\alpha(2\alpha+3)})$ time algorithm for MIN-SCVD on α -bounded digraphs, as solving MIN-SCVD reduces to finding a maximum sized induced subgraph that is singly connected, which can be done in the claimed runtime.

We need the following theorem due to Gallai and Milgram [25] to prove our observation that no α -bounded digraph with at least $\alpha(2\alpha + 4)$ vertices can be singly connected.

 $^{^1}$ Due to paucity of space, the proofs of statements marked with a \star have been omitted.



Fig. 2. Forward and backward arcs w.r.t. a path.

Theorem 2 (Gallai and Milgram [18,25]). Every directed graph D has a path cover \mathcal{P} and an independent set $\{v_p \mid P \in \mathcal{P}\}$ of vertices such that $v_p \in P$ for every $P \in \mathcal{P}$.

We can assume that the set $\{v_p \mid P \in \mathcal{P}\}$ in Theorem 2 is a maximal independent set. If not, we can add more vertices to the set until it becomes maximal, and "break" the paths in \mathcal{P} at those newly added vertices to make new paths. The new collection of paths is a path cover of D such that every path contains a vertex of the maximal independent set. We record this fact below.

Observation 1. Every directed graph D has a path cover \mathcal{P} and a maximal independent set $\{v_p \mid P \in \mathcal{P}\}$ of vertices such that $v_p \in P$ for every $P \in \mathcal{P}$.

Let *D* be a digraph. For a path $P = v_1 \dots v_\ell$ in *D*, we define forward arcs and backward arcs with respect to *P* in *D* as follows. An $(v_i, v_j) \in A(D)$ is a forward arc w.r.t. *P* if $v_i, v_j \in V(P)$, and j > i + 1. And $(v_i, v_j) \in A(D)$ is a backward arc w.r.t. *P* if $v_i, v_j \in V(P)$ and i > j + 1 (see Fig. 2).

For a path $P = v_1 \dots v_\ell$ in a digraph D, if $(v_i, v_j) \in A(D)$ is a forward arc w.r.t. P then note that there are two distinct paths from v_i to v_j in D: $v_i \dots v_j$ and $v_i v_j$. Therefore, we have the following observation.

Observation 2. If a digraph D has a path P such that D contains a forward arc w.r.t. P, then D is not singly connected.

We now prove the following lemma, which, in turn proves Theorem 1.

Lemma 3. For each fixed $\alpha \in \mathbb{N}$, every α -bounded digraph with at least $\alpha(2\alpha+4)$ vertices is not singly connected.

Proof. Let D be any α -bounded digraph such that $|V(D)| \geq \alpha(2\alpha + 4)$. Assume that D is singly connected. By Theorem 2 (and Observation 1), there is a maximal independent set I such that D can be decomposed into a collection \mathcal{P} of |I| vertex disjoint paths such that each path contains one vertex from I. Let $|I|(=|\mathcal{P}|) = \alpha'$. Note that $\alpha' \leq \alpha$, as D is an α -bounded digraph. Then, since $|V(D)| \geq \alpha(2\alpha + 4)$, by the pigeonhole principle, there exists a path P in \mathcal{P} with at least $2\alpha + 4$ vertices. Let P be $v_1 \dots v_\ell$, where $\ell \geq (2\alpha + 4)$, be such a path. Let v_P be a vertex of P such that $v_P \in I$. We now prove the following two claims.

Claim 1 (*). With respect to the path P, the vertex v_P can have at most two backward arcs and no forward arcs incident on it.

Claim 2 (*). For a vertex $v \notin V(P)$, there can be at most two arcs between v and V(P).

Now, let $I_P = N[v_P] \cap V(P)$, i.e., the set $I_P \subseteq V(P)$ contains v_P and the vertices in V(P) that are adjacent to v_P . Since P is a path and because of Claim 1, $|I_P| \leq 5$. Let $S = V(P) \setminus I_P$. Then, $|S| \geq 2\alpha - 1$, as $|V(P)| \geq 2\alpha + 4$. Also, observe that no vertex in S is adjacent to v_P . Then, every vertex in S is adjacent to some vertex in $I \setminus \{v_P\}$. To see this, consider $x \in S$. Note first that $x \notin I$, as $I \cap V(P) = \{v_P\}$. And now, if x is not adjacent to any vertex in $I \setminus \{v_P\}$, then $I \cup \{x\}$ is an independent set, which contradicts the maximality of I. Therefore, $|N_D(I \setminus \{v_P\}) \cap S| = |S| \geq 2\alpha - 1$. Now, since $|I \setminus \{v_P\}| \leq \alpha - 1$, by the pigeonhole principle, there is a vertex in $I \setminus \{v_P\}$ which is adjacent to at least three vertices in S, which, by Claim 2, is not possible. This completes the proof of Lemma 3.

3.2 Polynomial Time Algorithm for MIN-SCVD on Acyclic Local Tournaments

In this section, we prove that MIN-SCVD is polynomial time solvable on acyclic local tournaments. Without loss of generality, let us assume that the input acyclic local tournament is connected. Otherwise, we can find an optimal solution in each connected component separately and return the union of the optimal solutions for all the connected components. Specifically, this section is devoted to proving the following theorem.

Theorem 3. MINIMUM SINGLY CONNECTED VERTEX DELETION can be solved in time $\mathcal{O}(n^{\mathcal{O}(1)})$ on acyclic local tournaments, where n is the total number of vertices in the input acyclic local tournament.

The proof of Theorem 3 crucially uses the fact that every connected local tournament has a Hamiltonian path [4], which, in turn, implies that every connected acyclic local tournament has a unique topological ordering.

We first state the following lemma. It is so well-known that we omit its proof.

Lemma 4. Let D be a directed acyclic graph. Then, D has a topological ordering. That is, there exists an ordering $\sigma = (v_1, \ldots, v_n)$ of the vertices of D such that for every arc $(v_i, v_j) \in A(D)$, we have i < j, i.e., v_i appears before v_j in the ordering σ . Moreover, there exists a polynomial time algorithm that, given a directed acyclic graph D as input, finds a topological ordering of D.

It is a folklore result that every tournament contains a Hamiltonian path. Bang-Jensen [4] showed that this applies to connected local tournaments as well. For the sake of completeness, we prove this below.

Lemma 5 (\star) . Let D be a connected local tournament. Then D contains a Hamiltonian path.

The following lemma follows from Lemmas 4 and 5.

Lemma 6 (*). Let D be a connected acyclic local tournament and $P = v_1v_2...v_n$ be a Hamiltonian path of D. Then, $\sigma = (v_1,...,v_n)$ is the unique topological ordering of D.

Notation. Let D be an acyclic local tournament and $\sigma = (v_1, \ldots, v_n)$ be the unique topological ordering of D. For a vertex $u \in V(D)$, by $\ell(u)$, we denote the last vertex v in the ordering σ such that $(u, v) \in A(D)$. For each $i \in [n]$, we define an ordered set $S_i = \{v_i, v_{i+1}, \ldots, \ell(v_i)\}$.

Lemma 7 (*). Let D be a connected acyclic local tournament and $\sigma = (v_1, \ldots, v_n)$ be the topological ordering of D. Then, for all $i \in [n]$, the graph $D[S_i]$ is an acyclic tournament. Moreover, $S_i = N_D^+(v_i) \cup \{v_i\}$.

The following lemma says that any optimal solution to MIN-SCVD on D can exclude at most two vertices from the set S_i for each $i \in [n]$.

Lemma 8 (*). Let D be an acyclic local tournament and S be an optimal solution to MIN-SCVD on D. Let $\sigma = (v_1, v_2, \ldots, v_n)$ be the topological ordering of D. Then, for every $i \in [n]$, we have $|S_i \setminus S| \leq 2$.

Lemma 9 (*). Let D be an acyclic local tournament and $\sigma = (v_1, v_2, \ldots, v_n)$ be the topological ordering of D. Let $v_i, v_j \in V(D)$ such that i < j. Let $\ell(v_i) = v_{p_i}$ and $\ell(v_j) = v_{p_j}$. Then, $p_i \leq p_j$.

The following lemma forms the basis of our algorithm.

Lemma 10 (*). Let D be an acyclic local tournament and $\sigma = (v_1, \ldots, v_n)$ be the topological ordering of D. Then, there exists an optimal solution to MIN-SCVD on D that does not contain the vertices v_1, v_2 .

Proof (Proof Sketch). Let S be an optimal solution to MIN-SCVD on D. If $v_1, v_2 \notin S$, then the lemma holds. So assume that either $v_1 \in S$ or $v_2 \in S$.

By Lemma 7, the graphs $D[S_1]$ and $D[S_2]$ are acyclic tournaments, and $S_1 = N_D^+(v_1) \cup \{v_1\}$ and $S_2 = N_D^+(v_2) \cup \{v_2\}$. By Lemma 9, we have $\ell(v_1) \leq_{\sigma} \ell(v_2)$. This implies that $S_1 \setminus \{v_1\} \subseteq S_2$. By Lemma 8, we have $|S_1 \setminus S| \leq 2$ and $|S_2 \setminus S| \leq 2$. We now consider two cases depending on whether $v_1 \in S$ or $v_2 \in S$. We only prove the case when $v_1 \in S$ here.

Case 1: $v_1 \in S$. If $S_1 \setminus S = \emptyset$, then since $N_D^+(v_1) \subseteq S_1$, the digraph $D - (S \setminus \{v_1\})$ is also singly connected, which contradicts the assumption that S is an optimal solution. Therefore, $|S_1 \setminus S| \ge 1$. Let $v_p \in S_1$ be such that $v_p \notin S$. (Note that $p \ne 1$ as we are in the case when $v_1 \in S$.) We shall show that $(S \setminus \{v_1\}) \cup \{v_p\}$ is also an optimal solution to MIN-SCVD on D.

Now, consider the digraph $D - (S \setminus \{v_1\})$. Since S is an optimal solution, $D - (S \setminus \{v_1\})$ is not singly connected. That is, $D - (S \setminus \{v_1\})$ contains a pair of vertices u and v such that there are two internally vertex disjoint paths in $D - (S \setminus \{v_1\})$ from u to v. We refer to such a pair of paths as a forbidden structure. But since D - S is singly connected, any forbidden structure in $D - (S \setminus \{v_1\})$ must contain v_1 . Also, note that since v_1 is the first vertex in the topological ordering σ , any forbidden structure in $D - (S \setminus \{v_1\})$ must be a pair of paths that start from v_1 .

Now, since, $|S_1 \setminus S| \leq 2$, the vertex v_1 has at most two out-neighbours in the digraph $D - (S \setminus \{v_1\})$, and v_p is one of them. Therefore, if there exists a vertex v_j in $D - (S \setminus \{v_1\})$ such that there are two vertex disjoint paths from v_1 to v_j in $D - (S \setminus \{v_1\})$, then one of those paths must contain the vertex v_p . This implies that $(S \setminus \{v_1\}) \cup \{v_p\}$ is also an optimal solution to MIN-SCVD on D.

```
Algorithm 1: ALGO(D)
```

1 Input: A connected acyclic local tournament D. 2 Output: A solution S to MIN-SCVD for D. 3 Let (v_1, \ldots, v_n) be the topological ordering of D. 4 if D is singly connected then 5 | return $S = \emptyset$; 6 else 7 | return $S = (S_1 \setminus \{v_1, v_2\}) \cup ALGO(D - (S_1 \setminus \{v_2\}));$ 8 end

We are now ready to describe our algorithm, which works as follows. We greedily construct a solution S as follows. First, we add the set $S_1 \setminus \{v_1, v_2\}$ to S, and by doing this, we cover all the forbidden structures containing v_1 . (Note that $D - (S_1 \setminus \{v_1, v_2\})$ could still contain some forbidden structures containing v_2). Next, we recursively find a solution in the digraph $D - (S_1 \setminus \{v_2\})$. A formal description of our algorithm ALGO is in Algorithm 1. It is easy to see that the algorithm runs in polynomial time. The correctness of the algorithm follows from Lemma 10. This completes the proof of Theorem 3.

4 Singly Connected Vertex Deletion on In-Tournaments

In this section, we design an algorithm for SCVD on in-tournaments that runs in time $2^k n^{\mathcal{O}(1)}$. We use the technique of iterative compression, introduced by Reed, Smith and Vetta [38] to design this algorithm. We also show that SCVD on in-tournaments admits a kernel with $\mathcal{O}(k^2)$ vertices.

Remark 1. We note that the classical complexity of SCVD on in-tournaments (and local touranments) is still open. We do not know whether the problem is NP-hard or not on in-tournaments and on local tournaments.

Recall that a directed graph D is said to be an in-tournament if for all vertices $v \in V(D)$, $D[N_D^-(v)]$ is a tournament. We first prove the following preparatory results that will be used to design our algorithm and kernel.

Lemma 11 (*). Let D be an in-tournament. Then D is singly connected if and only if $|N_D^-(v)| \leq 1$ for all $v \in V(D)$.

As an immediate consequence of Lemma 11, we get the following result, which says that singly connected in-tournaments are precisely those digraphs that are acyclic triangle-free.

Lemma 12 (\star). Let D be an in-tournament. Then D is singly connected if and only if D does not contain an acyclic triangle as an induced subgraph.

In light of Lemma 12, it is not difficult to see that the SCVD problem on in-tournaments reduces to the 3-HITTING SET problem. The 3-HITTING SET problem takes as input a set U, a family \mathcal{F} of subsets of U such that $|F| \leq 3$ for every $F \in \mathcal{F}$, and a non-negative integer k. And the question is to determine if there exists $X \subseteq U$ such that $|X| \leq k$ and $X \cap F \neq \emptyset$ for every $F \in \mathcal{F}$. Given an instance (D, k) of SCVD on in-tournaments, where D is an n-vertex intournament, we can construct an equivalent instance (U, \mathcal{F}, k') of 3-HITTING SET by taking U = V(D), $\mathcal{F} = \{\{x, y, z\} \mid \{x, y, z\} \text{ induces an acyclic triangle}\}$, and k' = k. The fastest algorithm for 3-HITTING SET, to the best of our knowledge, is due to Wahlström [41, Corollary 69] and runs in time $2.0755^k n^{\mathcal{O}(1)}$. Thus, we can conclude that SCVD problem on in-tournaments can be solved in time $2.0755^k n^{\mathcal{O}(1)}$ as well. In the remaining part of this section, we show that SCVD on in-tournaments can in fact be solved in time $2^k n^{\mathcal{O}(1)}$. Before that we also note that 3-HITTING SET has a $\mathcal{O}(k^2)$ -sized kernel [1, Remark 1], which can be adapted to SCVD on in-tournaments as well. We record this fact below.

Observation 3. SCVD on in-tournaments admits an $\mathcal{O}(k^2)$ kernel.

We now prove the following theorem.

Theorem 4 (*). SCVD on in-tournaments admits an algorithm that runs in time $2^k n^{\mathcal{O}(1)}$.

To prove Theorem 4, we apply the technique of iterative compression, and show that solving SCVD on in-tournaments boils down to solving $2^k n^{\mathcal{O}(1)}$ many instances of the VERTEX COVER (VC) problem on pseudoforests. A pseudoforest is an undirected graph in which every connected component contains at most one cycle; and VC is polynomial time solvable on pseudoforests. Thus we obtain the runtime claimed in the theorem statement. Theorem 4 implies an analogous result for out-tournaments as well.

Theorem 5 (*). SCVD on out-tournaments admits an algorithm that runs in time $2^k n^{\mathcal{O}(1)}$.

5 A Linear Kernel for SCVD on Local Tournaments

In this section, we prove that SCVD admits a linear vertex kernel on local tournaments. Specifically, we prove the following theorem.

Theorem 6. SCVD on local tournaments admits a kernel with $\mathcal{O}(k)$ vertices.

Let (D, k) be an instance of SCVD, where D is a local tournament. The basis of our kernelization algorithm is Lemma 12. Recall Lemma 12, which says that an in-tournament (and hence a local tournament) is singly connected if and only if it does not contain an acyclic triangle as a subgraph. We give the following reduction rule in order to simplify the input instance (D, k) of SCVD. We apply this reduction rule exhaustively.

Reduction Rule 1. If a vertex $v \in V(D)$ is not contained in any acyclic triangle, then delete v from D. Return instance (D', k), where $D' = D - \{v\}$.

Lemma 13 (*). Reduction Rule 1 is safe.

After an exhaustive application of Reduction Rule 1, every vertex in D is contained in some acyclic triangle.

Next, we prove the following lemma that will help us bound the kernel size.

Lemma 14 (*). Let D be a local tournament and $S \subseteq V(D)$ such that D - S is singly connected. Then, for every vertex $v \in S$, v has at most 3 in-neighbours and at most 3 out-neighbours in $V(D) \setminus S$.

Next, using Lemma 14, we obtain the following lemma.

Lemma 15 (*). Let (D, k) be an instance of SCVD on local tournaments and assume that Reduction Rule 1 is no longer applicable. If (D, k) is a yes-instance of SCVD, then $|V(D)| \leq 7k$.

Reduction Rule 2. If $|V(D)| \ge 7k+1$, then return that (D, k) is a no-instance of SCVD.

The safeness of the above reduction rule follows from Lemma 15. When Reduction Rule 2 is no longer applicable, we obtain our required bound in Theorem 6. Observe that both the reduction rules can be applied in polynomial time and are applied only polynomially many times. The correctness of our kernel follows from Lemmas 13 and 15. This completes the proof of Theorem 6.

6 Conclusion

We studied the SCVD problem on various classes of digraphs such as tournaments, α -bounded digraphs, acyclic local tournaments, in-tournaments and local tournaments. Our algorithm for SCVD on in-tournaments runs in time $2^k n^{\mathcal{O}(1)}$. It remains to be seen if this runtime is optimal or can be improved. In particular, as noted in Remark 1, it is open whether SCVD is NP-hard or polynomial time solvable on in-tournaments. Another class of digraphs that one could consider is the class of locally transitive tournaments. A digraph D is said to be a locally transitive tournament if for every vertex $v \in V(D)$, both $N_D^+(v)$ and $N_D^-(v)$ induce transitive tournaments. Note that locally transitive tournaments are a super-class of acyclic local tournaments, and a sub-class of local tournaments. It would be interesting to see if one can extend the polynomial time algorithm for SCVD on acyclic local tournaments to locally transitive tournaments. As for the parameterized complexity of SCVD, the most interesting open problem is to resolve the complexity of SCVD on general digraphs, i.e., whether SCVD, parameterized by the solution size, admits a fixed-parameter tractable algorithm on general digraphs?

References

- Abu-Khzam, F.N.: A kernelization algorithm for d-hitting set. J. Comput. Syst. Sci. 76(7), 524–531 (2010)
- Agrawal, A., Saurabh, S., Sharma, R., Zehavi, M.: Kernels for deletion to classes of acyclic digraphs. J. Comput. Syst. Sci. 92, 9–21 (2018)
- 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)
- Bang-Jensen, J.: Locally semicomplete digraphs: a generalization of tournaments. J. Graph Theory 14(3), 371–390 (1990)
- Bang-Jensen, J.: Locally semicomplete digraphs and generalizations. In: Classes of Directed Graphs, pp. 245–296 (2018)
- Bang-Jensen, J., Gutin, G. (eds.): Classes of Directed Graphs. SMM. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-71840-8
- Bang-Jensen, J., Maddaloni, A., Saurabh, S.: Algorithms and kernels for feedback set problems in generalizations of tournaments. Algorithmica 76(2), 320–343 (2016)
- Bar-Yehuda, R., Geiger, D., Naor, J., Roth, R.M.: Approximation algorithms for the feedback vertex set problem with applications to constraint satisfaction and bayesian inference. SIAM J. Comput. 27(4), 942–959 (1998)
- Bergougnoux, B., Eiben, E., Ganian, R., Ordyniak, S., Ramanujan, M.S.: Towards a polynomial kernel for directed feedback vertex set. In: MFCS, pp. 36:1–36:15 (2017)
- 10. Bodlaender, H.L.: On disjoint cycles. Int. J. Found. Comput. Sci. 5(1), 59-68 (1994)
- Buchsbaum, A.L., Carlisle, M.C.: Determining uni-connectivity in directed graphs. Inf. Process. Lett. 48(1), 9–12 (1993)
- 12. Cao, Y.: A naive algorithm for feedback vertex set. In: SOSA, pp. 1:1–1:9 (2018)
- Cao, Y., Chen, J., Liu, Y.: On feedback vertex set: new measure and new structures. Algorithmica 73(1), 63–86 (2015)
- 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)
- Chen, J., Liu, Y., Lu, S., O'Sullivan, B., Razgon, I.: A fixed-parameter algorithm for the directed feedback vertex set problem. J. ACM 55(5), 21:1–21:19 (2008)
- Chudak, F.A., Goemans, M.X., Hochbaum, D.S., Williamson, D.P.: A primal-dual interpretation of two 2-approximation algorithms for the feedback vertex set problem in undirected graphs. Oper. Res. Lett. 22(4–5), 111–118 (1998)
- Dehne, F.K.H.A., Fellows, M.R., Langston, M.A., Rosamond, F.A., Stevens, K.: An o(2^{o(k)}n³) FPT algorithm for the undirected feedback vertex set problem. Theory Comput. Syst. 41(3), 479–492 (2007)
- Diestel, R.: Graph Theory. GTM, vol. 173. Springer, Heidelberg (2017). https:// doi.org/10.1007/978-3-662-53622-3
- Dietzfelbinger, M., Jaberi, R.: On testing single connectedness in directed graphs and some related problems. Inf. Process. Lett. 115(9), 684–688 (2015)

- Downey, R.G., Fellows, M.R.: Fixed-parameter intractability. In: Proceedings of the Seventh Annual Structure in Complexity Theory Conference, pp. 36–49 (1992)
- Downey, R.G., Fellows, M.R.: Fixed parameter tractability and completeness. In: Complexity Theory: Current Research, pp. 191–225 (1992)
- Downey, R.G., Fellows, M.R.: Fixed-parameter tractability and completeness I: basic results. SIAM J. Comput. 24(4), 873–921 (1995)
- Fradkin, A., Seymour, P.: Edge-disjoint paths in digraphs with bounded independence number. J. Comb. Theory Ser. B 110, 19–46 (2015)
- Fujito, T.: A note on approximation of the vertex cover and feedback vertex set problems - unified approach. Inf. Process. Lett. 59(2), 59–63 (1996)
- Gallai, T., Milgram, A.N.: Verallgemeinerung eines graphentheoretischen satzes von rédei: Ladislaus rédei zum 60. geburtstag. Acta scientiarum mathematicarum 21(3–4), 181–186 (1960)
- Guo, J., Gramm, J., Hüffner, F., Niedermeier, R., Wernicke, S.: Compressionbased fixed-parameter algorithms for feedback vertex set and edge bipartization. J. Comput. Syst. Sci. 72(8), 1386–1396 (2006)
- Kanj, I.A., Pelsmajer, M.J., Schaefer, M.: Parameterized algorithms for feedback vertex set. In: IWPEC, pp. 235–247 (2004)
- Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W., Bohlinger, J.D. (eds.) Complexity of Computer Computations, pp. 85–103. Springer, Boston (1972). https://doi.org/10.1007/978-1-4684-2001-2_9
- Khuller, S.: An o(|v|²) algorithm for single connectedness. Inf. Process. Lett. 72(3–4), 105–107 (1999)
- Khuller, S.: Addendum to "an o(|v|²) algorithm for single connectedness". Inf. Process. Lett. 74(5–6), 263 (2000)
- Kociumaka, T., Pilipczuk, M.: Faster deterministic feedback vertex set. Inf. Process. Lett. 114(10), 556–560 (2014)
- Li, D., Liu, Y.: A polynomial algorithm for finding the minimum feedback vertex set of a 3-regular simple graph 1. Acta Mathematica Scientia 19(4), 375–381 (1999)
- Liang, Y.D.: On the feedback vertex set problem in permutation graphs. Inf. Process. Lett. 52(3), 123–129 (1994)
- Liang, Y.D., Chang, M.: Minimum feedback vertex sets in cocomparability graphs and convex bipartite graphs. Acta Informatica 34(5), 337–346 (1997)
- Lokshtanov, D., Ramanujan, M.S., Saurabh, S., Sharma, R., Zehavi, M.: Wannabe bounded treewidth graphs admit a polynomial kernel for DFVS. In: WADS, pp. 523–537 (2019)
- Mnich, M., van Leeuwen, E.J.: Polynomial kernels for deletion to classes of acyclic digraphs. Discrete Optim. 25, 48–76 (2017)
- Raman, V., Saurabh, S., Subramanian, C.R.: Faster fixed parameter tractable algorithms for finding feedback vertex sets. ACM Trans. Algorithms 2(3), 403–415 (2006)
- Reed, B.A., Smith, K., Vetta, A.: Finding odd cycle transversals. Oper. Res. Lett. 32(4), 299–301 (2004)
- Seymour, P.D.: Packing directed circuits fractionally. Combinatorica 15(2), 281– 288 (1995)
- 40. Thomassé, S.: A $4k^2$ kernel for feedback vertex set. ACM Trans. Algorithms **6**(2), 32:1–32:8 (2010)
- Wahlström, M.: Algorithms, measures and upper bounds for satisfiability and related problems. Ph.D. thesis, Department of Computer and Information Science, Linköpings universitet (2007)