Hitting Long Directed Cycles Is Fixed-Parameter Tractable

Alexander Göke
Technische Universität Hamburg, Germany
alexander.goeke@tuhh.de

Dániel Marx
Max Planck Institut für Informatik, Saarland Informatics Campus, Saarbrücken, Germany
dmarx@mpi-inf.mpg.de

Matthias Mnich
Technische Universität Hamburg, Germany
matthias.mnich@tuhh.de

Abstract

In the Directed Long Cycle Hitting Set problem we are given a directed graph G, and the task is to find a set S of at most k vertices/arcs such that $G - S$ has no cycle of length longer than ℓ. We show that the problem can be solved in time $2^{\mathcal{O}(\ell^6+\ell k^3 \log k + k^5 \log k \log \ell)} \cdot n^{\mathcal{O}(1)}$, that is, it is fixed-parameter tractable (FPT) parameterized by k and ℓ. This algorithm can be seen as a far-reaching generalization of the fixed-parameter tractability of Mixed Graph Feedback Vertex Set [Bonsma and Lokshtanov WADS 2011], which is already a common generalization of the fixed-parameter tractability of (undirected) Feedback Vertex Set and the Directed Feedback Vertex Set problems, two classic results in parameterized algorithms. The algorithm requires significant insights into the structure of graphs without directed cycles of length longer than ℓ and can be seen as an exact version of the approximation algorithm following from the Erdős-Pósa property for long cycles in directed graphs proved by Kreutzer and Kawarabayashi [STOC 2015].

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Fixed parameter tractability; Mathematics of computing → Graph algorithms

Keywords and phrases Directed graphs, directed feedback vertex set, circumference

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.59

Category Track A: Algorithms, Complexity and Games

Related Version A full version of the paper is available at `https://arxiv.org/abs/2003.05267`.

Funding *Alexander Göke*: Supported by DFG grant MN 59/1-1.
Dániel Marx: Supported by ERC Consolidator Grant SYSTEMATICGRAPH (No. 725978).
Matthias Mnich: Supported by DFG grant MN 59/4-1.

1 Introduction

Feedback Vertex Set (FVS) and its directed variant Directed FVS (DFVS) are among the most classical problems in algorithmic graph theory: given a (directed) graph G the task is to find a minimum-size set $S \subseteq V(G)$ of vertices such that $G - S$ contains no (directed) cycles. Interestingly, the directed version is *not* a generalization of the undirected one. There is no obvious reduction from FVS to DFVS (replacing each undirected edge with two arcs of opposite directions does not work, as this would create directed cycles of length 2).

Both problems received significant amount of attention from the perspective of parameterized complexity. The main parameter of interest there is the optimal solution size $k = |S|$. Both problems can easily be solved in time $n^{\mathcal{O}(k)}$ by enumerating all size-k vertex subsets $S \subseteq V(G)$ and then checking whether $G - S$ is acyclic. The interesting question is thus whether the problems are *fixed-parameter tractable* with respect to k, i.e. whether there is an algorithm with run time $f(k) \cdot n^{\mathcal{O}(1)}$ for some computable function f depending only on k. FVS is one of the most studied problems in parameterized complexity: starting in the early 1990's, a long series of improved fixed-parameter algorithms [5, 6, 10, 13, 18, 25] lead to the currently fastest (randomized) algorithm from 2020 with run time $2.7^k \cdot n^{\mathcal{O}(1)}$ [20]. The DFVS problem has also received a significant amount of attention from the perspective of parameterized complexity. It was a long-standing open problem whether DFVS admits such an algorithm; the question was finally resolved by Chen et al. who gave a $4^k k! k^4 \cdot \mathcal{O}(nm)$-time algorithm for graphs with n vertices and m edges. Recently, an algorithm for DFVS with run time $4^k k! k^5 \cdot \mathcal{O}(n+m)$ was given by Lokshtanov et al. [22]. A fruitful research direction is trying to extend the algorithm to more general problems than DFVS. On the one hand, Chitnis et al. [8] generalized the result by giving a fixed-parameter algorithm for Directed Subset FVS: here we are given a subset U of arcs and only require the k-vertex set S to hit every cycle that contains an arc of U. On the other hand, Lokshtanov et al. [21] showed that the Directed Odd Cycle Transversal problem, where only the directed cycles of odd length needed to be hit, is W[1]-hard parameterized by solution size.

It is worth noting that very different algorithmic tools form the basis of the fixed-parameter tractability of FVS and DFVS: the undirected version behaves more like a hitting set-type problem, whereas the directed version has a more cut-like flavor. These differences motivated Bonsma and Lokshtanov [4] to consider Mixed FVS, the common generalization of FVS and DFVS where the input graph contains both directed and undirected edges. In such *mixed graphs*, cycles can contain directed arcs and undirected edges, but in particular the walk visiting an undirected edge twice is not a cycle. They obtained an algorithm for Mixed FVS with run time $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$ for k the size of the smallest feedback vertex set.

In this paper we study the following generalization of DFVS: We want to find a minimum size vertex set S such that all cycles of $G - S$ to have length at most ℓ. For $\ell = 1$ this is DFVS in loopless graphs. For $\ell = 2$ this is Mixed FVS in mixed graphs. The length of a longest cycle in a (directed) graph is also known as (directed) circumference of a graph. The parameterized version of our problem thus reads:

Directed Long Cycle Hitting Set	*Parameter:* $k + \ell$.
Input: A directed multigraph G and integers $k, \ell \in \mathbb{N}$.	
Task: Find a set S of at most k vertices such that $G - S$ has circumference at most ℓ.	

Note that Directed Long Cycle Hitting Set for $\ell = 2$ generalizes Mixed FVS (and hence both FVS and DFVS): to see this, subdivide anti-parallel arcs to make all cycles have length at least three and then replace undirected edges by anti-parallel arcs.

In contrast to FVS and DFVS, even checking feasibility of a given solution is a non-trivial task. It amounts to checking, for a digraph G and integer ℓ, whether G contains a cycle of length more than ℓ. This is also known as the Long Directed Cycle problem, which is obviously NP-hard since it contains the Directed Hamiltonian Cycle problem for $\ell = |V(G)| - 1$. However, Long Directed Cycle is fixed-parameter tractable parameterized by ℓ [29], hence *verifying* the solution of Directed Long Cycle Hitting Set is fixed-parameter tractable in ℓ.

Our contributions. Our main result is a fixed-parameter algorithm for Directed Long Cycle Hitting Set.

▶ **Theorem 1.** *There is an algorithm that solves* Directed Long Cycle Hitting Set *in time* $2^{\mathcal{O}(\ell^6 + \ell k^3 \log k + k^5 \log k \log \ell)} \cdot n^{\mathcal{O}(1)}$ *for n-vertex digraphs G and parameters $k, \ell \in \mathbb{N}$.*

The result also extends to the arc deletion variant of the problem, as we show both of them to be equivalent in a parameter preserving way.

The run time in Theorem 1 depends on two parameters, k and ℓ. This is necessary for the following reason. For $\ell = 1$, Directed Long Cycle Hitting Set corresponds to the DFVS problem, which is NP-hard. Moreover, the problem is also NP-hard for $k = 0$, as it contains the Directed Hamiltonian Cycle problem. This also shows that the run time cannot be polynomial in k or ℓ (unless P = NP). Assuming ETH, it is even necessary that the run time depends *exponentially* on both k and ℓ. Our algorithm achieves a run time that is single-exponential in both parameters k and ℓ. It is, in this sense, optimal.

Our algorithm is based on an elaborate combination algorithmic techniques, some of them used previously, some of them new.

- We use the standard opening step of iterative compression, which allows to assume that each directed cycle of length more than ℓ goes through a small number of exceptional vertices.
- We do not want to deal with the situation when there are two exceptional vertices x and y that are in the same strong component of the solution $G - S$. If we guess that this happens in the solution, then a way to avoid this problem is to guess a directed cycle C containing both x and y, and to contract this cycle. In order to guess this cycle, we essentially need a **representative set** of $x \to y$ paths, that is, a collection of paths such that if an (unknown) set S of at most k vertices does not disconnect y from x, then there is at least one $x \to y$ path disjoint from S in our collection. As an interesting self-contained result, we construct such a collection of size $\ell^{\mathcal{O}(k^2 \log k)} \cdot \log n$ on directed graphs without cycles of length greater than ℓ.
- If we can assume that the exceptional vertices are in different strong components of the solution, then this defines a separation problem on the exceptional vertices and makes the **directed shadow removal** technique of Chitnis et al. [8] relevant to simplify the structure of the instance. In particular, a major structural goal that we want to achieve is to ensure that every arc of the input graph lies in a directed cycle of length at most ℓ.
- Removing the exceptional vertices breaks the graph into some number of strong components with no cycle of length longer than ℓ in any of them. We call **portal vertices** the endpoints of the arcs connecting these strong components with each other and with the exceptional vertices. We show that the portal vertices can be partitioned into **clusters**: portals in each cluster are close to each other, while the distance between any two clusters is large. Furthermore, every solution has to separate the clusters from each other, defining another directed multiway cut problem.
- In the final step of the algorithm, we would like to use the technique of **important separators** to solve the directed multiway cut problem defined above: these are separators that are maximally "pushed" towards the target of the separations. However, the exact notion of importance is difficult to define due to the additional constraints of the problem being solved. Ergo, we perform a detailed analysis of the instance structure to identify **outlet vertices** that allows us to represent the additional constraints as separation, and to formally reduce the problem to branching on the choice of an important separator.

Related work. The structure of long cycles in directed graphs has been of interest for long time. For instance, Lewin [19] analyzed the density of such graphs, and Kintali [16] analyzes the directed treewidth of such directed graphs. Algorithmically, though, it was only recently shown by Kawarabayashi and Kreutzer [15] that the vertex version of the Erdős-Posa property holds for long directed cycles: namely, they show that any digraph G

either contains a set of $k+1$ vertex-disjoint directed cycles of length at least ℓ or some set S of at most $f(k,\ell)$ vertices that intersects all directed cycles of G with length at least ℓ. The corresponding questions for directed cycles without length restrictions have also been well-investigated [2, 26].

Note that an algorithmic proof of the Erdős-Pósa property can be a useful opening step for a fixed-parameter algorithm: we either find a set of $k+1$ arc- or vertex-disjoint cycles of length at least ℓ (and thus can reject the instance (G,k,ℓ) as "no"-instance) or obtain a set S which can serve as a feasible approximate solution. Such an opening step was also discussed in the well-known fixed-parameter algorithm for DFVS by Chen et al. [7, Remark 5.3], where the function $f(k,1)$ is known to be near-linear. In our case though, the function $f(k,\ell)$ from the Kawarabayashi-Kreutzer result is too large for us to obtain an algorithm for Directed Long Cycle Hitting Set with run time $2^{\text{poly}(k,\ell)} \cdot n^{\mathcal{O}(1)}$.

Further, directed circumference can be seen as an intermediate step towards a general algorithmic framework for graph optimization problems related to *directed treewidth*. In undirected graphs, treewidth as a graph measure has enjoyed unprecedented success as a tool towards efficient approximation algorithms and fixed-parameter algorithms. For instance, as part of their Graph Minors series, Robertson and Seymour [28] showed that the k-linkage problem is fixed-parameter tractable, heavily relying on the reduction of the problem to graphs of bounded treewidth. In directed graphs, the situation is again much more complicated: Johnson et al. [14] introduced the notion of directed treewidth for digraphs. Yet, for digraphs the k-linkage problem is NP-hard already for $k=2$, and no fixed-parameter algorithm is known which recognizes digraphs of nearly-bounded directed treewidth. On the positive side, though, digraphs of bounded directed circumference are nicely squeezed between acyclic digraphs and digraphs of bounded directed treewidth [16]. Also, the arc version of the k-linkage problem is fixed-parameter tractable on digraphs of directed circumference 2 [3]; the question remains open for digraphs of larger directed circumference.

Returning to the original motivation of studying generalizations of DFVS, Neogi et al. [24] gave a fixed-parameter algorithm for the problem of finding a set S of size at most k in a given digraph G such that every strong component of $G-S$ excludes graphs in a fixed finite family $\mathcal{H}$ as (not necessarily induced) subgraphs, when $\mathcal{H}$ contains only rooted graphs, or contains at least one directed path. Göke et al. [12] considered the problem of finding a set S of size at most k in a given digraph G such that every strong component of $G-S$ has size at most s; they gave a fixed-parameter algorithm for parameter $k+s$.

2 Definitions and Notations

In this paper, we mainly consider finite loop-less directed graphs (or digraphs) G with vertex set $V(G)$ and arc set $A(G)$. We allow multiple arcs and arcs in both directions between the same pairs of vertices. A *walk* is a sequence of vertices $(v_1,\ldots,v_\ell)$ with corresponding arcs (v_i,v_{i+1}) for $i=1,\ldots,\ell-1$ which forms a subgraph of G; the *length* of a walk is its number of arcs. A walk is *closed* if $v_1=v_\ell$; otherwise, it is *open*. A *path* in G is an open walk where all vertices are visited at most once. A *cycle* in G is a closed walk in which every vertex is visited at most once, except for $x_1=x_\ell$ which is visited twice. (Throughout this entire paper, by "cycle" we always mean directed cycle.) We call G *acyclic* if G does not contain any cycle. For two vertices x_i,x_j of a walk W with $i\le j$ we denote by $W[x_i,x_j]$ the subwalk of W starting at x_i and ending in x_j. For a walk W ending in a vertex x and a second walk R starting in x, $W\circ R$ is the walk resulting when concatenating W and R.

For each vertex $v \in V(G)$, its *out-degree* in G is the number $d^+_G(v)$ of arcs of the form (v, w) for some $w \in V(G) \setminus \{v\}$, and its *in-degree* in G is the number $d^-_G(v)$ of arcs of the form (w, v) for some $w \in V(G) \setminus \{v\}$. For each subset $V' \subseteq V(G)$, the subgraph induced by V' is the graph $G[V']$ with vertex set V' and arc set $\{(u, v) \in A(G) \mid u, v \in V'\}$. For a set X of vertices or arcs, let $G - X$ denote the subgraph of G obtained by deleting the elements in X from G. For a subgraph G' and an integer d we denote by $R^+_{G'}(X)$ the set of vertices that are reachable from X in G'.

A digraph G is *strong* if either G consists of a single vertex (then G is called *trivial*), or for any distinct $u, v \in V(G)$ there is a (directed) path from u to v. A *strong component* of G is an inclusion-wise maximal induced subgraph of G that is strong. The *(directed) circumference* of a digraph G is the length $\mathsf{cf}(G)$ of a longest cycle of G; if G is acyclic, then define $\mathsf{cf}(G) = 0$.

3 Directed Long Cycle Hitting Set Algorithm

The goal of this section is to devise an algorithm for Directed Long Cycle Hitting Set and thereby proof Theorem 1. We will only consider the vertex deletion variant. This will suffice, as the arc deletion version can be reduced to the vertex deletion version in a parameter-preserving way, as we show in the full version.

The algorithm performs a sequence of reductions to special cases of the original Bounded Cycle Length Vertex Deletion. All these sections are modular and just need the problem formulation and theorem at the end of the previous section.

Compression. Recall that in the Directed Long Cycle Hitting Set problem we are given a digraph G and integers k and ℓ. The task then is to find a set of at most k vertices such that $G - S$ contains no cycles of length more than ℓ.

As already stated in the introduction, checking a solution for correctness is an non-trivial task. For this we use a fixed-parameter algorithm by Zehavi [29].

▶ **Theorem 2.** *There is an algorithm that decides in time $2^{\mathcal{O}(\ell)} \cdot n^{\mathcal{O}(1)}$ whether an n-vertex digraph G contains a cycle of length more than ℓ.*

This already solves the case for $k = 0$. We now want to design an algorithm for general k.

The goal of this subsection is to get an existing solution T for which we have to find a disjoint solution S of size less than $|T|$. For this we use the standard techniques of iterative compression and disjoint solution.

We start our algorithm by applying the iterative compression technique introduced by Reed, Smith and Vetta [27]. This technique was also used by Chen et al. [7] to show the fixed parameter tractability of DFVS. We choose an arbitrary enumeration $v_1, \ldots, v_n$ of the vertices of G. By G_i we denote the digraph $G[v_1, \ldots, v_i]$. We want to iteratively construct solutions S_{i+1} to (G_{i+1}, k, ℓ) by using the solution S_i of (G_i, k, ℓ). We start with the empty digraph G_0 and the empty solution $S_0 = \emptyset$. This solution is feasible for every choice of G, k and ℓ as the empty digraph contains no cycles.

Now, if S_i hits all cycles of length more than ℓ in G_i, then $T_{i+1} = S_{i+1} \cup \{v_{i+1}\}$ does the same for G_{i+1} (as $G_i - S_i = G_{i+1} - T_{i-1}$). The only problem now is that T_{i+1} may be to large. Therefore, we consider the compression version of our problem: given an instance (G_i, k, ℓ) with a solution T_i of size $k + 1$, find a solution S_i of size at most k. If we can solve this problem, by above procedure we get a solution S_n for the digraph $G_n = G$ and hence have solved the original problem. This adds a factor of n to the run-time, preserving fixed-parameter tractability.

The compression problem can be modified further in two useful ways. The first modification is to get disjointness of solutions T_i and S_i. This can be achieved by guessing the intersection $U_i = T_i \cap S_i$ by taking all possible subsets of T_i. For every choice we can then solve the disjoint compression problem $(G_i - U_i, k - |U_i|, \ell)$ with starting solution $T_i \setminus U_i$. If the non-disjoint instance has a solution, then the instance where we guessed the intersection correctly has a solution. Otherwise, none of the disjoint instances has a solution. This adds a factor of $2^{|T_i|} = 2^{k+1}$ to the run-time, also preserving fixed-parameter tractability.

The other useful modification is about not solving the problem exactly but instead to guess a set $\mathcal{S}$ of bounded size intersecting S_i in at least one vertex. If we have a routine that for an instance (G', k', ℓ', T') of the disjoint compression problem returns us a set $\mathcal{S}$ that is guaranteed to intersect some solution (if a non-empty solution exists), we can branch as follows. First we check whether the empty solution already solves the problem by Theorem 2. This solves the the problem for $k = 0$. Otherwise we call our routine on the instance obtaining a set $\mathcal{S}$. For every $v \in \mathcal{S}$ we recurse on the instance $(G' - v, k' - 1, \ell, T')$. This adds a factor of $f(k', \ell', n')^{k'}$ to the run-time where $f(k', \ell', n')$ is an upper bound on the size of $\mathcal{S}$ on an instance (G', k', ℓ', T') with $n' = |G'|$. This preserves fixed-parameter tractability if we can write $f(k', \ell', n')^{k'}$ as $g(k', \ell') \cdot \mathrm{poly}(n')$ for some appropiate function g. Note that this is the case even if $f(k', \ell', n') = h_1(k', \ell') \cdot \log^{h_2(k', \ell')}(n')$.

After all these reductions we are left with the following problem:

Intersecting Directed Long Cycle Hitting Set	*Parameter:* $k + \ell$.
Input:	A directed multigraph G, $T \subseteq V(G)$ and integers $k, \ell \in \mathbb{N}$.
Properties:	$\mathsf{cf}(G - T) \leq \ell$
Task:	Find a set $\mathcal{S} \subseteq V(G) \setminus T$ that intersects a set $S \subseteq V(G) \setminus T$ of size at most k with $\mathsf{cf}(G - S) \leq \ell$ if such a set exists.

The set $\mathcal{S}$ will often help us to argue about solutions S disjoint from it. This is often described as branching steps throughout the algorithm but we decided to collect all the branching here to be more precise about the assumptions needed in the theorems.

Contraction. In the last section we reduced the original Directed Long Cycle Hitting Set problem to a variant where we are already given a solution T and now want to find a set $\mathcal{S}$ intersecting every solution S disjoint from T. Except for the intersection step, such reductions have been used by Chen et al. [7] in their algorithm for DFVS. The next key observation in their algorithm for DFVS is that every vertex of T must lie in their own strong component of $G - S$. The reason is that for every directed feedback vertex set S of G, each strong component of $G - S$ is a single vertex. For Directed Long Cycle Hitting Set, the situation is way more complicated, as strong components of $G - S$ contain cycles of length up to ℓ. Moreover, those cycles can concatenate to arbitrarily large strong components. So it is possible that $G - S$ contains strong components with more than one vertex of T. In this section, we want to contract (parts of) such components to a single vertex such that eventually, after contraction, every strong component of $G - S$ contains at most one vertex of T. A structural result allowing the contraction is the following lemma:

▶ **Lemma 3.** *Let G be a digraph and let $X \subseteq V(G)$ be such that $G[X]$ is strong and $\mathsf{cf}(G[X]) \leq \ell$. Suppose that the following two properties hold:*

1. *Every cycle of G has length at most ℓ or length at least ℓ^2.*
2. *For any $a, b \in X$ there cannot be both*
   - **a.** *an $a \to b$-path P_{ab} of length at least ℓ in $G[X]$*
   - **b.** *a $b \to a$-path P_{ba} of length at most ℓ in $G - (X \setminus \{a, b\})$*

Let G/X be the digraph obtained by contracting X to a single vertex x.

- *If $\mathsf{cf}(G - S) \leq \ell$ for some $S \subseteq V(G) \setminus X$, then $\mathsf{cf}(G/X - S) \leq \ell$.*
- *If $\mathsf{cf}(G/X - S') \leq \ell$ for some $S' \subseteq V(G/X) \setminus \{x\}$, then $\mathsf{cf}(G - S') \leq \ell$.*

To algorithmically use this result we have to make sure its requirements are fulfilled. This is easy if G has a cycle C with $\ell < |C| < \ell^2$. We can can detect these cycles in time $2^{\mathcal{O}(\ell^2)} \cdot n^{\mathcal{O}(1)}$ by using color coding (see Alon et al. [1]). If there exists such an cycle, any long cycle hitting set has to intersect it. Also it's length is bounded, so we can return it as set $\mathcal{S}$.

To find a set X fulfilling the remaining properties is more difficult. For this we build on a tool called "k-representative set of paths".

▶ **Definition 4.** *Let G be a digraph, $x, y \in V(G)$ and $k \in \mathbb{Z}_{\geq 0}$. A set $\mathcal{P}$ of $x \to y$-paths is a* k-representative set of $x \to y$-paths, *if for every set $S \subseteq V(G)$ of size at most k it holds: If there is an $x \to y$-path in $G - S$ there is an $x \to y$-path $P \in \mathcal{P}$ that is disjoint from S.*

In our case such a k-representative set of paths will be useful for constructing a closed walk visting several vertices of T to use as our set X. Later on the reversed property of k-representative sets of paths will also be handy: if you hit all paths of $\mathcal{P}$ with a set S of size at most k, then there exist no $x \to y$-paths in $G - S$.

Alas, we were not able to find k-representative sets of paths of small size in general graphs. In the case of strong digraphs of bounded circumference, however, we obtain the following:

▶ **Theorem 5.** *Let G be a strong digraph, $x, y \in V(G)$ and $k \in \mathbb{Z}_{\geq 0}$. Then a k-representative set of $x \to y$-paths having size $\mathsf{cf}(G)^{\mathcal{O}(k^2 \log k)} \cdot \log n$ can be found in time $\mathsf{cf}(G)^{\mathcal{O}(k^2 \log k)} \cdot n^{\mathcal{O}(1)}$.*

We need to generalize this tool a bit further as our graph G neither has bounded circumference nor does it need to be strong. However, we have a set T of small size such that $\mathsf{cf}(G - T) \leq \ell$ and we are only interested in the strong components of the graph. This leads us to the following specialized lemma:

▶ **Lemma 6.** *Let G be a strong digraph, $T \subseteq V(G)$ and $k, \ell \in \mathbb{Z}_{\geq 0}$ with $\mathsf{cf}(G - T) \leq \ell$. Then in time $2^{\mathcal{O}(k\ell + k^2 \log k \log \ell)} n^{\mathcal{O}(1)}$, we can find a set $\mathcal{Q}$ of $|T|^2 2^{\mathcal{O}(k\ell + k^2 \log k \log \ell)} \log^2 n$ closed walks with the following property: If there is a set $S \subseteq V(G)$ of size at most k with $\mathsf{cf}(G - S) \leq \ell$ and there are two vertices of T in the same strong component of $G - S$ then there is*

- *a closed walk in $Q \in \mathcal{Q}$ containing two vertices of T that is disjoint from S or*
- *a simple cycle of length at most ℓ containing two vertices of T.*

The lemma allows for a branching procedure that creates several instances. For each instance we assume that there is a solution S such that no two vertices of T lie in the same strong component of $G - S$. We call such a solution *isolating* long cycle hitting set. The instances are created in the following way: We keep our original instance just in case it has an isolating long cycle hitting set. Then we search for a simple cycle C of length at most ℓ in G visiting at least two vertices of T. Such a cycle can be found by color coding in time $2^{\mathcal{O}(\ell)} \cdot n^{\mathcal{O}(1)}$. If such a cycle exists, we branch into two instances: In the first instance, we assume that S intersects C and we can just return $\mathcal{S} = V(C)$ as solution to our intersection problem. In the second instance, C is disjoint from S and we have a candidate X for contraction with Lemma 3. If no such cycle exists we get our candidate applying Lemma 6 by branching on which closed walk of $\mathcal{Q}$ is disjoint from S and take that as a candidate X.

We now have to check whether our candidate X fulfills the assumptions of Lemma 3. If $\mathsf{cf}(G[X]) > \ell$ then X cannot be disjoint from S in contradiction to our branching assumption and we give up on this branch. Then we check for every pair $a, b \in X$, if paths P_{ab} and P_{ba}

as in Lemma 3 exist. If they do we cannot contract X, but $P_{ab} \circ P_{ba}$ forms a cycle of length more than ℓ, so it has to be intersected by S. By branching assumption $V(P_{ab}) \subseteq X$ is disjoint from S and therefore P_{ba} has to intersect S. Luckily, $|V(P_{ba})| \leq \ell + 1$ and we can return $\mathcal{S} = V(P_{ba})$ as set intersecting S. If these paths do not exist we can contract X to a single vertex to obtain a new instance (G', k, ℓ, T'). We continue our branching procedure with this new instance until we guess that our instance has an isolating long cycle hitting separator. As the cardinality of T is decreased in each branching step this is the case at the latest when $|T| = 1$. Note that this may lead to the strange case that we are indeed searching for a solution S that is larger than our set T.

The remaining problem can be stated as:

Isolating Long Cycle Hitting Set Intersection	*Parameter:* $k + \ell + \|T\|$.
Input:	A directed multigraph G, integers $k, \ell \in \mathbb{N}$ and a set $T \subseteq V(G)$
Properties:	$\mathsf{cf}(G - T) \leq \ell$
Task:	Find a set $\mathcal{S}$ intersecting some isolating long cycle hitting set S of size at most k with respect to T if such a set exists.

The above procedure (and the result of this section) can be summarized as follows.

▶ **Theorem 7.** *Instances (G, k, ℓ) of* Directed Long Cycle Hitting Set *can be solved in time $2^{\mathcal{O}(\ell^2 + \ell k^3 \log k)} \cdot (f_{ii}(k, \ell))^k \cdot n^{\mathcal{O}(1)}$ by at most $2^{\mathcal{O}(\ell k^3 \log k)} \cdot (f_{ii}(k, \ell))^k \cdot n^2 \log^{2k+2}(n)$ calls to an algorithm A_{ii} solving the* Isolating Long Cycle Hitting Set Intersection *problem, where $f_{ii}(k, \ell)$ is a size bound on the set produced by A_{ii}.*

Reducing to Important Hitting Separator. In the previous section we reduced the Directed Long Cycle Hitting Set problem to the Isolating Long Cycle Hitting Set Intersection problem, a variant where we are already given a solution T and search for a solution S disjoint from T of size at most k. Additionally, we know that T has at most one vertex in each strong component of $G - S$. For the remainder of this subsection, assume that there is such a solution S of size at most k.

Intuitively, we did the reduction to the isolating variant in order to apply something like Skewed Multiway Cut. This was done by Chen et al. [7] in their algorithm for Directed FVS. They guessed a topological ordering of the vertices of T in $G - S$ and used Skewed Multiway Cut to cut away the backward paths. This also implied that each strong component consisted of a single vertex. In our case, though, we still have cycles left, and a direct construction to Skewed Multiway Cut gives us no control over their length.

We instead guess only a last vertex $t \in T$ in some topological ordering of the strong components of $G - S$. From here our approach differs significantly from that of Chen et al. [7] for Directed FVS. Instead of finding all cuts at once, we focus on the $t \to T \setminus \{t\}$-cuts while still hitting long cycles. This is we want to find the strong component of t in $G - S$.

For this it is useful to see, that two types of arcs may not lie in the strong component of t in $G - S$. The first kind of arcs are simply the arcs having their endpoint in $T \setminus \{t\}$, i.e. arcs in $\delta^-(T \setminus \{t\})$. This is because S is isolating. The other kind of arcs are the arcs that lie only on long cycles, i.e. arcs $a = (v, w) \in A(G)$ with $\mathrm{dist}_{G-a}(w, v) \geq \ell$. This follows from the fact that S hits all long cycles. Therefore, the strong component of t in $G - S$ must be a subset of the connected component of t in G after above arcs are removed. We call the later component $C_t^\star$. We want to focus our search onto that component.

However, there may also be other arcs and vertices which do not lie in the strong component of t in $G - S$ which are still in $C_t^\star$. To make it easier to argue about these, we use the shadow-covering technique introduced by Chitnis et al. [9] to solve the Directed Multiway Cut problem. For this we need to define what the shadow of our solution S with respect to the set T is:

▶ **Definition 8** (shadow). *Let G be a digraph and let $T, S \subseteq V(G)$. A vertex $v \in V(G)$ is* in the forward shadow $f_{G,T}(S)$ of S (with respect to T) *if S is a $T \to v$-separator in G, and v is* in the reverse shadow $r_{G,T}(S)$ of S (with respect to T) *if S is a $v \to T$-separator in G. All vertices of G which are either in the forward shadow or in the reverse shadow of S (with respect to T) are said to be* in the shadow of S (with respect to T)*.*

Note that S itself is not in its own shadow. The most useful property of the shadow for our purposes is that every vertex that is not in the shadow of S with respect to T is reachable from a vertex of T and can reach some (maybe different) vertex of T. In particular, if a vertex v is not in the shadow of S and reachable from t (our last vertex) in $G - S$ then it has to lie in the strong component of t in $G - S$. This is because v can also reach a vertex of T but there is no $T \to T \setminus \{t\}$-walk in $G - S$.

We use the deterministic algorithm for shadow covering by Chitnis et al. [8]. The following corollary follows from their paper:

▶ **Corollary 9.** *Let (G, k, ℓ, T) be an instance of* Isolating Long Cycle Hitting Set Intersection*. One can construct, in time $2^{\mathcal{O}(k^2)} \cdot n^{\mathcal{O}(1)}$, sets $Z_1, \ldots, Z_p$ with $p \leq 2^{\mathcal{O}(k^2)} \log^2 n$ such that if there is an isolating long cycle hitting set of size at most k, there is isolating long cycle hitting set S of size at most k and an $i \in \{1, \ldots, p\}$ such that $Z_i \cap (S \cup T) = \emptyset$ and Z_i includes the shadow of S with respect to T.*

By branching on the possible choices of Z_i we can assume that we have a set Z that covers (read: includes) the shadow of S with respect to T and is disjoint from S and T. Now every vertex outside of Z is reachable from a vertex of T and can also reach a vertex of T.

Consider now the set $V_{\mathsf{out}} \subseteq V(G) \setminus Z$ defined as follows. For every $v \in V_{\mathsf{out}}$ there is a $v \to w$-path P with $w \in V(G) \setminus Z$ whose inner vertices are inside Z and one of the following properties holds. The endpoint w is contained in $T \setminus \{t\}$ or P contains an arc $a = (x, y) \in A(G)$ that only lies on long cycles (i.e. $\mathrm{dist}_{G-a}(y, x) \geq \ell$). The set of these vertices may not be reached from t in $G - S$ if the other endpoint w is not in S. Because then w reaches a vertex of T and that gives us either a $t \to T \setminus \{t\}$ path in $G - S$ or a closed walk in $G - S$ containing an arc that only lies on long cycles (which therefore contains itself a long cycle).

To get rid of the condition that the other endpoint of the path P we apply the tool of critical vertices also used by Chitnis et al. [8]. For this we use an auxiliary graph (called torso) which is created by taking all vertices of $V(G) \setminus Z$ and shrinking paths with interior points in Z to arcs. By remembering the paths which contained arcs only on long cycle we get a set U_t^{long} of dangerous arcs that must not be traversable from t. In case they are not traversable only by the endpoint w lying in S these vertices are called k-critical with respect to U_t^{long}. Now, we can use the following theorem by Chitnis et al. [8].

▶ **Proposition 10** ([8]). *Given a digraph G, a subset U of its arcs, and some $t \in V(G)$, in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ we can find a set F_{critical} of size $2^{\mathcal{O}(k)}$ that contains all k-critical vertices.*

We apply Proposition 10 to $G, U_t^{\mathsf{long}}, t$ and k, and add the resulting set F_{critical} to our solution $\mathcal{S}$. If our solution S is still disjoint from $\mathcal{S}$, it cannot contain a k-critical vertex. This implies that t cannot reach a vertex of V_{out} in $G - S$. So S is a $t \to V_{\mathsf{out}}$-separator in the strong component $C_t^\star$.

But $S \cap C_t^\star$ may also have other properties like hitting long cycles that intersect or lie within $C_t^\star$. To cover these we introduce the notation of important hitting $t \to V_{\mathsf{out}}$-separators. These are $t \to V_{\mathsf{out}}$-separators U that hit all long cycles in a digraph and are backward-range minimal in the sense that there is no other $t \to V_{\mathsf{out}}$-separators U' hitting all cycles with $|U'| \leq |U|$ and $R^-_{G-U'}(V_{\mathsf{out}}) \subsetneq R^-_{G-U}(V_{\mathsf{out}})$.

The main result of this section is that it is enough to find a set $\mathcal{S}$ intersecting one important hitting $t \to V_{\mathsf{out}}$-separator U for every backward range $R^-_{G-U}(V_{\mathsf{out}})$ in a strong digraph (read $G[C_t^\star]$). We call the remaining problem Important Hitting Separator in Strong Digraphs which reads as follows:

Important Hitting Separator in Strong Digraphs	*Parameter:* $k + \ell$.
Input:	A strong digraph G, integers $k, \ell \in \mathbb{N}$, $t \in V(G)$ and $V_{\mathsf{out}} \subseteq V(G)$.
Properties:	$\mathsf{cf}(G - t) \leq \ell$, every arc of G lies on a cycle of length at most ℓ.
Task:	Find a set $\mathcal{S}_{\mathsf{hs}}$ intersecting a important hitting $t \to V_{\mathsf{out}}$-separator of size at most k in every range equivalence class.

Portals and Clusters. In the previous section we reduced Isolating Long Cycle Hitting Set Intersection to Important Hitting Separator in Strong Digraphs. In an intuitive way we replaced the constraint on S of hitting all cycles going through $T \setminus \{t\}$ by a constraint that S needs to be a backward-range minimal $t \to V_{\mathsf{out}}$-separator. Also we simplified our graph to be strongly connected and that every arc lies on a short cycle. We now want to also break the long cycles going only through t into cut constraints. For this we consider the strong components of $G - t$. Let $\mathcal{C}$ the set of all such components. Our main interest are now portal vertices.

▶ **Definition 11.** *Let G be a digraph and let $C \subset V(G)$. A vertex $v \in C$ is a* portal vertex *of C, if $\Delta_G(v) > \Delta_{G[C]}(v)$, where $\Delta_H(v)$ is the number of incident arcs (both in-coming and out-going) of v in a graph H. We denote by X_C the set of all portal vertices of C.*

As all arcs of G lie on a short cycles the arcs going between clusters must cycle back to t.

▶ **Lemma 12.** *Let $C \in \mathcal{C}$ and $v \in X_C$. There is a cycle O_v with $v, t \in V(O_v)$ and $|O_v| \leq \ell$.*

These cycles allow us to transform paths between portal vertices of the same strong component into closed walks. Like in the first part of our algorithm we can eliminate cycles of length between $\ell + 1$ and $2\ell^6$ by detecting them and returning them as set $\mathcal{S}_{\mathsf{hs}}$ as any hitting separator has to intersect these. This gap between small and large cycles however allows us to get a similar gap for the distance between portal vertices.

▶ **Lemma 13.** *For any $v_1, v_2 \in X_C$, either $\mathsf{dist}_{G[C]}(v_1, v_2) \leq 2\ell^2$ or $\mathsf{dist}_{G[C]}(v_1, v_2) \geq 2\ell^6 - 2\ell$.*

This in turn allows us to cluster the portal vertices of every component. For $\ell_{\max} = 2\ell^2$ we put all portal vertices at distance at most $\ell_{\max}$ from a portal vertex v into the set X_v. This defines a partition into clusters:

▶ **Lemma 14.** *For any $C \in \mathcal{C}$ and $v_1, v_2 \in X_C$, sets X_{v_1} and X_{v_2} are either disjoint or equal.*

Consider now a long cycle O in G. Assume it contains for each strong component $C \in \mathcal{C}$ it visits only portal vertices of one of the clusters of C. In strong digraphs of circumference at most ℓ (like $G[C]$) the length of a path can at most be ℓ^2 times the distance between its endpoints. Therefore the length of O inside a single component C can be at most $2\ell^4$. We already made sure that long cycles in G have length more than $2\ell^6$. But we can shortcut O after the visit of C to a cycle with length between $\ell+1$ and $2\ell^6$ – a contradiction. Therefore every long cycle has a path between different clusters of some strong component of $G-t$.

By carefully analyzing the structure of the strong components and guessing vertices of S (by including them into $\mathcal{S}_{\mathsf{hs}}$) we make sure there are neither too many strong components with more than one cluster nor do these have to many clusters themselves. Moreover, we were able to restrict $t \to V_{\mathsf{out}}$-paths to a single strong component.

This means that our sought after solution S forms in each strong component C a multiway cut between the different cluster X_i of C and additionally cuts all $X_i \to V_{\mathsf{out}}$-paths in backward-range minimal way. This structure of S we call important cluster separators:

▶ **Definition 15.** *Let G be a digraph and let $X_1, \dots, X_t, Y \subseteq V(G)$ be pairwise disjoint vertex sets. We call a vertex set $U \subseteq V \setminus (X_1 \cup \dots \cup X_t \cup Y)$ a* cluster separator *if $G-U$ contains*
- *no path from X_i to X_j for $i \neq j$ and*
- *no path from X_i to Y for $i = 1, \dots, t$.*

A cluster separator U is important *if there is no cluster separator U' with $|U'| \leq |U|$ and $R^-_{G-U'}(Y) \subsetneq R^-_{G-U}(Y)$.*

If $V_{\mathsf{out}} = \emptyset$ this boils down to the Directed Multiway Cut problem, which we solve by the algorithm of Chitnis et al. [9]. It thus remains to solve the case where $V_{\mathsf{out}} \neq \emptyset$.

Important Cluster Separator in Strong Digraphs of Bounded Circumference		*Parameter:* $k+\ell$.
Input:	A strong digraph G, integers $k, \ell \in \mathbb{N}$ and sets $X_1, \dots, X_p, V_{\mathsf{out}} \subseteq V(G)$.	
Properties:	$\mathsf{cf}(G) \leq \ell$, $X_i, V_{\mathsf{out}} \neq \emptyset$, $2 \leq p \leq k(k+1)+1$, $\operatorname{dist}(v,w) \leq 2\ell^2 \quad \forall v,w \in X_i, i \in \{1,\dots,p\}$.	
Task:	Find a vertex set $\mathcal{S}_{\mathsf{cluster}}$ intersecting any important cluster separator with respect to $X_1, \dots, X_\ell, V_{\mathsf{out}}$ of size at most k.	

Finding Important Cluster Separators. In this section we want to solve the Important Cluster Separator in Strong Digraphs of Bounded Circumference problem. This problem is strongly related to the Directed Multiway Cut Problem. By adding an additional vertex $v^\star$ that has an incoming arc from ever vertex of V_{out} we see that cluster separator are indeed multiway cuts in this modified graph. The difficulty lies in the notion of importance we introduced (and needed). That is, we want to intersect all cluster separators S where $R^-_{G-S}(V_{\mathsf{out}})$ is minimal for their size.

We introduce two definitions to handle this. The first is the concept of the *frontier* F of a cluster separator S. These are the vertices that define the backward range from V_{out}, i.e. the vertices of S that can reach V_{out} without going through another vertex of S. The other concept is that of an *outlet*. Given an $X_i \to X_j$-path P and two integers $\alpha, \beta \in \mathbb{Z}_{\geq 0}$, an (α,β)-outlet of P is a vertex ω of P with the following property: there is a $\omega \to V_{\mathsf{out}}$-path R_ω such that every vertex on P except for the α-many preceding and following vertices of ω on P have distance at least β from R_ω. These outlets are in some sense key positions where $X_i \to X_j$-paths start to significantly differ from $X_i \to V_{\mathsf{out}}$-paths.

For outlets we differentiate between "open" and "closed" outlets. *Open* outlets are outlets that lie in $R^-_{G-S}(V_{\mathsf{out}})$, i.e. behind the frontier; the other outlets *closed*. The frontier F is therefore separates $\mathcal{X} = \bigcup_{i=1}^t X_i$ from the open outlets. The rest of our efforts now focuses on finding a set V_Ω such that the frontier F is an important $\mathcal{X} \to (V_{\mathsf{out}} \cup V_\Omega)$-separator. This set V_Ω contains (a subset of) the open outlets for some $X_i \to X_j$-paths. As set of paths tho search on we take for every ordered pair of distinct X_i, X_j an arbitrary $X_i \to X_j$-path $P_{i,j}$.

The main property we use for finding the set V_Ω is some kind of locality argument. As our graph G is strong and has bounded circumference, the length of any path cannot differ too much from the distance of its endpoints. So if we know that a path is hit by S not to far from an outlet, we can guess these vertices. This is done with the help of k-representative sets of paths: if we know that an $\omega \to V_{\mathsf{out}}$-path is hit near ω, we construct a k-representative set of $\omega \to V_{\mathsf{out}}$-paths. We can argue that also one of the paths in this set has to be hit near ω. So we can guess all the vertices near ω on the paths of the k-representative set for their intersection with S.

By carefully guessing such potential intersections, and choosing α, β properly, we obtain:

- If a path $P_{i,j}$ has more than $\gamma = \mathrm{poly}(k, \ell)$ outlets, one of them is open.
- If there is a $X_i \to X_j$-path in $G - F$ then $P_{i,j}$ has an open outlet.

The last step is now to get rid of the $X_i \to X_j$-paths in $G - F$. We achieve this by guessing that a so called *landing strip* in front of an open outlet of $P_{i,j}$ (which exists by the second property) is disjoint from S. This landing strip has the task that that if there is a $X_i \to X_j$-path in $G - F$ than also the open outlet would be reachable. This again works by the locality of our strong graphs of bounded circumference.

After all these guessing we obtain (for the right guess) a set V_Ω such that F is an important $\mathcal{X} \to (V_{\mathsf{out}} \cup V_\Omega)$-separator. These we can enumerate by a result of Chitnis et al. [9].

▶ **Proposition 16** ([9]). *Let G be a digraph and let $X, Y \subseteq V(G)$ be disjoint non-empty vertex sets. For every $p \geq 0$ there are at most 4^p important $X - Y$-separators of size at most p, and all these separators can be enumerated in time $\mathcal{O}(4^p \cdot p(n + m))$.*

Putting Everything Together. Finally, we are able to prove our main result. By combining the reductions of the previous sections, we get an overall algorithm solving Directed Long Cycle Hitting Set:

▶ **Theorem 1.** *There is an algorithm that solves* Directed Long Cycle Hitting Set *in time $2^{\mathcal{O}(\ell^6 + \ell k^3 \log k + k^5 \log k \log \ell)} \cdot n^{\mathcal{O}(1)}$ for n-vertex digraphs G and parameters $k, \ell \in \mathbb{N}$.*

4 k-Representative Sets of Paths

In this section, we show how to obtain a k-representative set of paths of small size in strong digraphs of bounded circumference. Let us briefly recall the definition.

▶ **Definition 4.** *Let G be a digraph, $x, y \in V(G)$ and $k \in \mathbb{Z}_{\geq 0}$. A set $\mathcal{P}$ of $x \to y$-paths is a* k-representative set of $x \to y$-paths, *if for every set $S \subseteq V(G)$ of size at most k it holds: If there is an $x \to y$-path in $G - S$ there is an $x \to y$-path $P \in \mathcal{P}$ that is disjoint from S.*

Our goal is to prove the following:

▶ **Theorem 5.** *Let G be a strong digraph, $x, y \in V(G)$ and $k \in \mathbb{Z}_{\geq 0}$. Then a k-representative set of $x \to y$-paths having size $\mathsf{cf}(G)^{\mathcal{O}(k^2 \log k)} \cdot \log n$ can be found in time $\mathsf{cf}(G)^{\mathcal{O}(k^2 \log k)} \cdot n^{\mathcal{O}(1)}$.*

If all $x \to y$-paths are short we can use the following result of Monien [23]:

▶ **Proposition 17** ([23]). *Let G be a digraph, let $x, y \in V(G)$ and let $k \in \mathbb{N}$. If every $x \to y$-path in G has length at most ℓ, then a k-representative set containing at most ℓ^k many $x \to y$-paths can be found in time $\ell^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$.*

Recently, Fomin et al. [11] improved the computation of representative sets of paths, both in terms of the size of the set and the run time, but Proposition 17 will be sufficient for our purposes.

A motivational example. Before we give our construction for k-representative sets of paths in strong digraphs of bounded circumference, we want to consider graphs of treewidth two and a special example of treewidth three. Strong digraphs of treewidth two are trees with bidirected arcs. In this case we have that for every pair $x, y \in V(G)$ there is an unique $x \to y$-path P. Thus, $\{P\}$ is a feasible k-representative set of paths.

The situation is significantly different even for $\mathsf{cf}(G) = 3$. Consider the strong digraph in Figure 1.

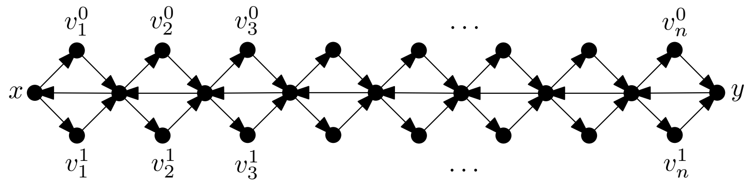

Figure 1 A digraph G with $\mathsf{cf}(G) = 3$ where every k-representative set of $x \to y$-paths has size $2^{\Omega(k)} \log n$.

There are exactly 2^n different $x \to y$-paths in G; each such path corresponds to a $0-1$ vector of length n. Thus, if we remove a vertex v_i^0 or v_i^1, then only those paths survive that have 1 or 0 at the i-th coordinate, respectively. Therefore, a collection of paths in this graph is k-representative only if no matter how we fix the values of k arbitrary coordinates, there is a vector in the collection satisfying these constraints. Such collections of vectors are also known as binary covering arrays. Kleitman and Spencer [17] proved that every collection of vectors of length n satisfying this property has size $2^{\Omega(k)} \cdot \log n$ (more precisely, they gave a lower bound on the dual question of k-independent families, but it can be easily rephrased into this lower bound).

We will now construct a k-representative set of paths for this graph G by using so called k-perfect families of hash functions.

▶ **Definition 18.** *Let $\mathcal{F}$ be a family of functions $f : U \to \{1, \dots, k\}$ on the universe U. We say that $\mathcal{F}$ is a* k-perfect family of hash functions *if for every $X \subseteq U$ of size at most k, there is an $f \in \mathcal{F}$ that is injective on X, i.e. $f(x) \neq f(x')$ for any two distinct $x, x' \in X$.*

We use the following result by Alon et al. [1] for our construction.

▶ **Proposition 19** ([1]). *Let U be a universe and $k \in \mathbb{N}$. Then there exists a k-perfect family $\mathcal{F}$ of size $2^{\mathcal{O}(k)} \log |U|$ that can be constructed in time $2^{\mathcal{O}(k)}|U|^{\mathcal{O}(1)}$.*

Before considering arbitrary strong digraphs of bounded circumference, let us explain how k-perfect families of hash functions can be used for the construction in the case of the digraph G of Figure 1. Let $\mathcal{F}$ be a k-perfect family of hash functions over the universe $U = \{1, \dots, n\}$ as in Proposition 19. Moreover, let $\mathcal{H}$ be the set of all functions $h : \{1, \dots, k\} \to \{0, 1\}$. For $(f, h) \in \mathcal{F} \times \mathcal{H}$ denote by $P_{f,h}$ the $x \to y$-path in G that uses the vertices $v_i^{h(f(i))}$ for $i = 1, \dots, n$. Then we add for every pair $(f, h) \in \mathcal{F} \times \mathcal{H}$ the path $P_{f,h}$ to our set $\mathcal{P}$.

Now consider a deletion set $S \subseteq V(G)$ of size at most k such that there is still an $x \to y$-path in G. Then S contains only vertices of type v_i^j and at most one of v_i^0 and v_i^1 for every $i \in \{1, \dots, n\}$. In other words, for some $X \subseteq U$ of size at most k and function $g : X \to \{0, 1\}$, the vertices $v_i^{g(i)}$ form the set S. As $\mathcal{F}$ is a k-perfect family of hash functions, there is an $f \in \mathcal{F}$ that is injective on X. Now consider the function h defined as follows.

For every $i \in X$, let $h(f(i)) = 1 - g(i)$; as f is injective on X, this is well-defined and gives a function $h : f(X) \to \{0,1\}$. Complete, h to a function $h : \{1,\dots,k\} \to \{0,1\}$ by choosing the remaining values arbitrarily. We claim that the path $P_{f,h}$ introduced for this choice of f and h is disjoint from S. For $i \notin X$, it does not matter if $P_{f,h}$ uses v_i^0 or v_i^1. For $i \in X$, set S contains $v_i^{g(i)}$. By our definition of h, we have $h(f(i)) = 1 - g(i)$, hence $P_{f,h}$ uses $v_i^{1-g(i)}$, avoiding S. Thus $P_{f,h}$ is indeed disjoint from S.

Our proof of Theorem 5 generalizes this construction to arbitrary strong digraphs of bounded circumference: we construct the path by concatenating a series of fairly independent "short jumps." For each of these short jumps, we construct a k-representative set of paths by Proposition 17. The choice of which short path to select is determined by a k-perfect family of hash functions, similarly to the argument in the previous paragraph.

Strong digraphs with bounded circumference. Before we formally start proving Theorem 5, we establish some structural properties of strong digraphs with bounded circumference.

▶ **Lemma 20.** *Let G be a digraph and let $x, y \in V(G)$. If P_1 is an $x \to y$-path and P_2 is a $y \to x$-path, then $|P_1| \leq (\mathsf{cf}(G) - 1)|P_2|$ holds. Consequently, we have $\mathsf{dist}_G(x,y) \leq (\mathsf{cf}(G) - 1)\mathsf{dist}_G(y,x)$.*

Proof. Suppose, for sake of contradiction, that $|P_1| > (\mathsf{cf}(G) - 1)|P_2|$. By $x, y \in V(P_1) \cap V(P_2)$, we can split P_1 into $|P_2|$ pairwise disjoint subpaths whose internal vertices are disjoint from $V(P_2)$. Note that there are $|V(P_1)| - |V(P_2)| = (|P_1| + 1) - (|P_2| + 1) = |P_1| - |P_2|$ of these internal vertices. By pigeonhole principle, at least one of the subpaths has at least $\left\lceil \frac{|P_1|-|P_2|}{|P_2|} \right\rceil = \left\lceil \frac{|P_1|}{|P_2|} \right\rceil - 1$ internal vertices. By our assumption, these are at least $\mathsf{cf}(G) - 1$ many. But than the whole subpath has at least $\mathsf{cf}(G) + 1$ vertices. Since $P_1 \circ P_2$ is a closed walk, our segment is contained in a closed walk. Moreover, P_1 is acyclic and our segment is internally disjoint from P_2. Thus, the segment is even contained in a cycle. But this cycle then has length at least $\mathsf{cf}(G) + 1$, contradicting the definition of circumference. ◀

By using that there is always a backward path in strong digraphs, applying above result twice yields:

▶ **Lemma 21.** *Let G be a strong digraph and $x, y \in V(G)$. Then $|P| \leq (\mathsf{cf}(G)-1)^2 \mathsf{dist}_G(x,y)$ for every $x \to y$-path P.*

Proof. Let W be a shortest $x \to y$-path in G. As G is strong, there is also a $y \to x$-path Q in G. By Lemma 20 we then have

$$|P| \leq (\mathsf{cf}(G) - 1)|Q| \leq (\mathsf{cf}(G) - 1)^2 |W| = (\mathsf{cf}(G) - 1)^2 \mathsf{dist}_G(x,y). \quad ◀$$

Similarly to the length, we can also argue about the distance between two paths.

▶ **Lemma 22.** *Let G be a strong digraph, $x, y \in V(G)$ be two vertices, and P_1, P_2 be two $x \to y$-paths. For every vertex v of P_1, we have $\mathsf{dist}_G(P_2, v) \leq 2(\mathsf{cf}(G) - 2)$ and $\mathsf{dist}_G(v, P_2) \leq 2(\mathsf{cf}(G) - 2)$.*

Proof. As G is strong, there is a $y \to x$-path Q.

▷ Claim 23. For $i \in \{1,2\}$, we have that

(i) $\mathsf{dist}_G(Q,v) \leq \mathsf{cf}(G) - 2$ and $\mathsf{dist}_G(v,Q) \leq \mathsf{cf}(G) - 2$ for every $v \in P_i$, and

(ii) $\mathsf{dist}_G(P_i,v) \leq \mathsf{cf}(G) - 2$ and $\mathsf{dist}_G(v,P_i) \leq \mathsf{cf}(G) - 2$ for every $v \in Q$.

Proof. Let $v \in P_i$. As P_i is acyclic (it is a path), but $P_i \circ Q$ is a closed walk in G, v has to lie on a cycle O in $P_i \circ Q$. This cycle has length at most $\mathsf{cf}(G)$. Furthermore, O has at least two vertices in $V(Q)$, as it contains an arc of Q. So there is a path in O from v to a vertex of Q of length at most $|O| - 2 \leq \mathsf{cf}(G) - 2$, showing $\mathsf{dist}_G(v, Q) \leq \mathsf{cf}(G) - 2$. On the other hand there is also a $V(Q) \to v$-path (from another vertex of $V(Q) \cap V(O)$) in O of length at most $|O| - 2 \leq \mathsf{cf}(G) - 2$, showing $\mathsf{dist}_G(Q, v) \leq \mathsf{cf}(G) - 2$. This shows Statement (i). Statement (ii) can be seen analogously by switching the roles of P_i and Q. ◁

Now fix a $v \in V(P_1)$. By Claim 23, we have that there is a $w \in V(Q)$ such that $\mathsf{dist}_G(w, v) = \mathsf{dist}_G(Q, v) \leq \mathsf{cf}(G) - 2$. Applying Claim 23 another time and using triangle inequality we get

$$\mathsf{dist}_G(P_2, v) \leq \mathsf{dist}_G(P_2, w) + \mathsf{dist}_G(w, v) \leq 2(\mathsf{cf}(G) - 2).$$

Similarly, we get from Claim 23 that there is an $u \in V(Q)$ with $\mathsf{dist}_G(v, u) = \mathsf{dist}_G(v, Q) \leq \mathsf{cf}(G) - 2$. Another application of Claim 23 and the triangle inequality yields

$$\mathsf{dist}_G(v, P_2) \leq \mathsf{dist}_G(v, u) + \mathsf{dist}_G(u, P_2) \leq 2(\mathsf{cf}(G) - 2),$$

concluding the proof. ◀

We are now ready to prove Theorem 5.

▶ **Theorem 5.** *Let G be a strong digraph, $x, y \in V(G)$ and $k \in \mathbb{Z}_{\geq 0}$. Then a k-representative set of $x \to y$-paths having size $\mathsf{cf}(G)^{\mathcal{O}(k^2 \log k)} \cdot \log n$ can be found in time $\mathsf{cf}(G)^{\mathcal{O}(k^2 \log k)} \cdot n^{\mathcal{O}(1)}$.*

Proof. Let us fix an arbitrary $x \to y$-path R (which exists as G is strong) to guide our construction. Denote by r the length of R and by $v_0 = x, v_1, \ldots, v_{r-1}, v_r = y$ its vertices. We only consider a subset of vertices z_i at distance $d = 2\mathsf{cf}(G)^4$ from each other or more formally $z_i = v_{i \cdot d}$. These z_i will be the anchor vertices for our short jumps. We divide then z_i further into $k+1$ subsets Z^o by taking every $(k+1)$st vertex starting at offset o. Formally we define $z_i^o = z_{i(k+1)+o}$ and $Z^o = \{z_i^o\}$. These subsets have the advantage that one of these is far away from a deletion set S of size at most k. For this we fix a set S of size at most k such that an $x \to y$-path in $G - S$ exists.

▷ Claim 24. There is some $o_S \in \{0, \ldots, k\}$ such that
- $\mathsf{dist}_G(Z^{o_S}, S) > 2(\mathsf{cf}(G) - 2)$ and
- $\mathsf{dist}_G(S, Z^{o_S}) > 2(\mathsf{cf}(G) - 2)$.

Proof. We claim that for every $s \in S$ there is at most one value $o \in \{0, \ldots, k\}$ such that $\mathsf{dist}_G(Z^o, s) \leq 2\mathsf{cf}(G)^2$. Suppose that $\mathsf{dist}_G(w_1, s), \mathsf{dist}(w_2, s) \leq 2\mathsf{cf}(G)^2$ for some $w_1 \in Z^{o_1}$ and $w_2 \in Z^{o_2}$ with $o_1 \neq o_2$. Assume, without loss of generality, that w_1 is before w_2 on R; then $R[w_1, w_2]$ has length at least d (as different z_i have distance at least d). By Lemma 20, we have $\mathsf{dist}_G(s, w_1) \leq (\mathsf{cf}(G) - 1)\mathsf{dist}_G(w_1, s) \leq (\mathsf{cf}(G) - 1) \cdot 2\mathsf{cf}(G)^2$, thus $\mathsf{dist}_G(w_2, w_1) \leq \mathsf{dist}_G(w_2, s) + \mathsf{dist}_G(s, w_1) \leq 2\mathsf{cf}(G)^3$. Again by Lemma 20, we have $d \leq |R[w_1, w_2]| \leq (\mathsf{cf}(G) - 1)\mathsf{dist}_G(w_2, w_1) < 2\mathsf{cf}(G)^4$, a contradiction. Thus, we have proven that for each of the k vertices $s \in S$ there is at most one value $o \in \{0, \ldots, k\}$ such that s is at distance at most $2\mathsf{cf}(G)^2$ from Z^o. Therefore, by the pigeon-hole principle there is an $o_S \in \{0, \ldots, k\}$ such that $\mathsf{dist}_G(Z^{o_S}, S) > 2\mathsf{cf}(G)^2$. By Lemma 20 this also implies $\mathsf{dist}_G(S, Z^{o_S}) > 2\mathsf{cf}(G)^2/(\mathsf{cf}(G) - 1) > 2(\mathsf{cf}(G) - 2)$. This completes the proof of Claim 24. ◁

Thus we know that a small surrounding of one of the Z^o's will be disjoint from S. Furthermore, Lemma 21 gives a bound on the length of a path P between two consecutive vertices z_i^o and z_{i+1}^o of Z^o, by $|P| \leq (\mathsf{cf}(G)-1)^2|R[z_i^o, z_{i+1}^o]| = \mathcal{O}(\mathsf{cf}(G)^7 k)$. This allows us to introduce sets $\mathcal{P}_i^o$ of k-representative $z_i^o \to z_{i+1}^o$-paths using the algorithm of Proposition 17 and have their size bounded by some $B = \mathcal{O}(\mathsf{cf}(G)^7 k)^k = \mathsf{cf}(G)^{\mathcal{O}(k \log k)}$ (using $k = 2^{\log k}$ and $\mathsf{cf}(G) \geq 2$). By duplicating paths as necessary we can assume that every P_i^o has size exactly B.

To make sure that our path collections with offset are connected to x and y, we construct additional sets $\mathcal{P}_x^o$ and $\mathcal{P}_y^o$ as follows: Let z_x^o be the first vertex in Z^o *after* x and z_y^o the last vertex *before* y. Then compute, using the algorithm of Proposition 17, $\mathcal{P}_x^o$ as a k-representative set of $x \to z_x^o$-paths and $\mathcal{P}_y^o$ as a k-representative set of $z_y^o \to y$-paths. As the distances between these pairs of vertices are bounded by the distance of neighboring vertices in Z^o we can analogously get a size bound of B for $\mathcal{P}_x^o$ and $\mathcal{P}_y^o$. Note that for some offsets o either $\mathcal{P}_x^o$ or $\mathcal{P}_y^o$ may align with some $\mathcal{P}_i^o$; then we leave out this $\mathcal{P}_i^o$ as we do not need it anymore. For each o, let $\mathcal{P}^o := \{\mathcal{P}_x^o, \mathcal{P}_y^o\} \cup \{\mathcal{P}_i^o\}_i$ be the set of these relevant sets.

▷ Claim 25. Every $\mathcal{P}_T^{o_S} \in \mathcal{P}^{o_S}$ contains a path disjoint from S.

Proof. Consider a set $\mathcal{P}_T^{o_S}$ with $T \in \{x, y, i\}$ such that the paths in $\mathcal{P}_T^{o_S}$ are $x_T \to y_T$-paths. As above sets are k-representative sets of paths, we must only show that there is any $x_T \to y_T$-path in $G - S$. By assumption there is a $x \to y$ path Q in $G - S$. By Lemma 22 we can find a $q_x \in V(Q)$ such that $\mathsf{dist}(x_T, q_x) \leq 2(\mathsf{cf}(G) - 2)$ and a $x_T \to q_x$-path Q_x in G achieving this distance. By Claim 24 we know that Q_x is disjoint from S and therefore, $Q_x \circ Q[q_x, y]$ is a $q_x \to y$ walk disjoint from S. Let $\hat{Q}_x$ be a $q_x \to y$-path contained in this walk. Another application of Lemma 22 yields a vertex $q_y \in V(\hat{Q})$ with $\mathsf{dist}(q_y, y_T) \leq 2(\mathsf{cf}(G) - 2)$ and a $q_y \to y_T$-path Q_y in G achieving this distance. Again, by Claim 24, Q_y is disjoint from S. Then $\hat{Q}_x[x_T, q_y] \circ Q_y$ contains a $x_T \to y_T$-path as proposed. This completes the proof of Claim 25. ◁

Of course, enumerating all possible tuples of paths would construct too many candidates, as the size of $\mathcal{P}^{o_S}$ can be $\Omega(m)$. Therefore, we want to use a $f(k)$-perfect family of hash functions. This is possible if we can bound the number of intersections with the sets $\mathcal{P}^{o_S}$ by $f(k)$.

▷ Claim 26. The set S intersects at most $2k$ sets of $\mathcal{P}^{o_S}$.

Proof. We show that $s \in S$ can intersect for at most two sets that share an endpoint, thus achieving the claimed size bound. Suppose for contradiction that s intersects two paths Q_1 and Q_2 out of sets in $\mathcal{P}^{o_S}$ that do not share an endpoint. Let each Q_i be a $x_i \to y_i$-path, then we can assume without loss on generality that the order the endpoints appear on R is x_1, y_1, x_2, y_2 and $|R[y_1, x_2] \geq 2\mathsf{cf}(G)^5$ (by the distance of the z_i. On the other hand $R[x_i, y_i]$ and Q_i connect the same endpoints, hence Lemma 22 implies that there is a $t_1 \in V(R[x_1, y_1])$ with $\mathsf{dist}(t_1, s) \leq 2(\mathsf{cf}(G) - 2)$ and a $t_2 \in V(R[x_2, y_2])$ with $\mathsf{dist}(s, t_2) \leq 2(\mathsf{cf}(G) - 2)$ as $s \in Q_1 \cap Q_2$. This implies that $\mathsf{dist}(t_1, t_2) \leq \mathsf{dist}(t_1, s) + \mathsf{dist}(s, t_2) \leq 4(\mathsf{cf}(G) - 2)$. If we now consider $R[t_1, t_2]$, we get $|R[t_1, t_2]| \geq |R[y_1, x_2] \geq 2\mathsf{cf}(G)^5 > (\mathsf{cf}(G) - 1)^2 \cdot \mathsf{dist}(t_1, t_2)$ in contradiction to Lemma 21. This completes the proof of Claim 26. ◁

We can now construct a $2k$-perfect family Ψ^o of hash functions over the universe $\mathcal{P}^o$ for each o. For o_S this family contains an element ψ which gives every set of $\mathcal{P}^{o_S}$ that is intersected by S a different number in $\{1, \ldots, 2k\}$ (by Claim 26). Further, there is a map π_{free} that maps the numbers of $\{1, \ldots, 2k\}$ to a number of $\{1, \ldots, B\}$, such that for every $\mathcal{P} \in \mathcal{P}^{o_S}$ which has a path intersected by S, we have that the $\psi \circ \pi_{\mathsf{free}}(\mathcal{P})$th path of $\mathcal{P}$ is not intersected by S. There is such a path by Claim 25. Denote by $Q_{\psi,\pi_{\mathsf{free}}}(\mathcal{P})$ this path. As we cannot know π_{free} in advance we create a set Π of all possible functions from $\{1,\dots,2k\}$ to $\{1,\dots,B\}$.

We know that for the specific choices of o_S, ψ and π_{free} we get a that the union of paths in $\{Q_{\psi,\pi_{\mathsf{free}}}(\mathcal{P}) | \mathcal{P} \in \mathcal{P}^{o_S}\}$ forms a $x \to y$ walk W in $G - S$. Every $x \to y$-path within W is also disjoint from S. Therefore, the set $\mathcal{P}_{x,y,k}$ created as follows contains a path disjoint from S: For every $o \in \{1,\dots,k+1\}$, every $\psi \in \Psi$ and every $\pi \in \Pi$ consider the $x \to y$-walk $\bigcup_{\mathcal{P} \in \mathcal{P}^o} Q_{\psi,\pi}(\mathcal{P})$ and introduce an arbitrary $x \to y$-path contained in it into $\mathcal{P}_{x,y,k}$.

The size bound on $\mathcal{P}_{x,y,k}$ is proven by multiplying the possibilities for each choice:

$$\underbrace{(k+1)}_{\text{choice of } o} \cdot \underbrace{2^{\mathcal{O}(k)} \log m}_{|\Psi|} \cdot \underbrace{B^{2k}}_{|\Pi|} = \mathsf{cf}(G)^{\mathcal{O}(k^2 \log k)} \log n.$$

The run time follows similarly. ◀

We think that k-representative sets of paths could prove a useful tool in many vertex deletion problems. Together with iterative compression, it allows one to argue about connectivity structure of the old solution. In our case it led to contraction argument strengthening the solution structure (Lemma 3) and a focused guessing of solution vertices (subsection about finding important cluster separators).

There may be other use cases as well. Therefore, we decided to present our result on k-representative sets of paths in a self-contained way. We hope that this helps other researchers and improves our understanding of vertex-deletion problems in directed graphs.

References

1 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995.

2 Saeed Akhoondian Amiri, Ken-Ichi Kawarabayashi, Stephan Kreutzer, and Paul Wollan. The Erdős-Pósa property for directed graphs, 2016. `arXiv:1603.02504`.

3 Jørgen Bang-Jensen and Tilde My Larsen. DAG-width and circumference of digraphs. *J. Graph Theory*, 82(2):194–206, 2016.

4 Paul Bonsma and Daniel Lokshtanov. Feedback vertex set in mixed graphs. In *Proc. WADS 2011*, volume 6844 of *Lecture Notes Comput. Sci.*, pages 122–133. Springer, 2011.

5 Yixin Cao, Jianer Chen, and Yang Liu. On feedback vertex set: new measure and new structures. *Algorithmica*, 73(1):63–86, 2015.

6 Jianer Chen, Fedor V. Fomin, Yang Liu, Songjian Lu, and Yngve Villanger. Improved algorithms for feedback vertex set problems. *J. Comput. System Sci.*, 74(7):1188–1198, 2008.

7 Jianer Chen, Yang Liu, Songjian Lu, Barry O'Sullivan, and Igor Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55(5):Art. 21, 19, 2008.

8 Rajesh Chitnis, Marek Cygan, Mohammataghi Hajiaghayi, and Dániel Marx. Directed subset feedback vertex set is fixed-parameter tractable. *ACM Trans. Algorithms*, 11(4):Art. 28, 28, 2015.

9 Rajesh Chitnis, MohammadTaghi Hajiaghayi, and Dániel Marx. Fixed-parameter tractability of directed multiway cut parameterized by the size of the cutset. In *Proc. SODA 2012*, pages 1713–1725, 2012.

10 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michał Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time (extended abstract). In *Proc. FOCS 2011*, pages 150–159, 2011.

11 Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Efficient computation of representative sets with applications in parameterized and exact algorithms. In *Proc. SODA 2014*, pages 142–151, 2014.

12 Alexander Göke, Dániel Marx, and Matthias Mnich. Parameterized algorithms for generalizations of directed feedback vertex set. In *Proc. CIAC 2019*, volume 11485 of *Lecture Notes Comput. Sci.*, pages 249–261, 2019.

13 Jiong Guo, Jens Gramm, Falk Hüffner, Rolf Niedermeier, and Sebastian Wernicke. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. *J. Comput. System Sci.*, 72(8):1386–1396, 2006.

14 Thor Johnson, Neil Robertson, P.D. Seymour, and Robin Thomas. Directed tree-width. *J. Combinat. Theory, Series B*, 82(1):138–154, 2001.

15 Ken-ichi Kawarabayashi and Stephan Kreutzer. The directed grid theorem. In *Proc. STOC 2015*, pages 655–664, 2015.

16 Shiva Kintali. Directed width parameters and circumference of digraphs. *Theoret. Comput. Sci.*, 659:83–87, 2017.

17 Daniel J. Kleitman and Joel Spencer. Families of k-independent sets. *Discrete Math.*, 6:255–262, 1973.

18 Tomasz Kociumaka and Marcin Pilipczuk. Faster deterministic feedback vertex set. *Inform. Process. Lett.*, 114(10):556–560, 2014.

19 Mordechai Lewin. On maximal circuits in directed graphs. *J. Combinatorial Theory, Ser. B*, 18(2):175–179, 1975.

20 Jason Li and Jesper Nederlof. Detecting feedback vertex sets of size k in $O^*(2.7^k)$ time. In *Proc. SODA 2020*, pages 971–989, 2018.

21 Daniel Lokshtanov, M. S. Ramanujan, Saket Saurab, and Meirav Zehavi. Parameterized complexity and approximability of directed odd cycle transversal. In *Proc. SODA 2020*, pages 2181–2200, 2020.

22 Daniel Lokshtanov, M. S. Ramanujan, and Saket Saurabh. When recursion is better than iteration: A linear-time algorithm for acyclicity with few error vertices. In *Proc. SODA 2018*, pages 1916–1933, 2018.

23 B. Monien. How to find long paths efficiently. In *Analysis and design of algorithms for combinatorial problems (Udine, 1982)*, volume 109 of *North-Holland Math. Stud.*, pages 239–254. North-Holland, Amsterdam, 1985.

24 Rian Neogi, M. S. Ramanujan, Saket Saurabh, and Roohani Sharma. On the parameterized complexity of deletion to $\mathcal{H}$-free strong components, 2020. `arXiv:2005.01359`.

25 Venkatesh Raman, Saket Saurabh, and C. R. Subramanian. Faster fixed parameter tractable algorithms for undirected feedback vertex set. In *Proc. ISAAC 2002*, pages 241–248, 2002.

26 Bruce Reed, Neil Robertson, Paul Seymour, and Robin Thomas. Packing directed circuits. *Combinatorica*, 16(4):535–554, 1996.

27 Bruce Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004.

28 Neil Robertson and Paul D. Seymour. Graph minors XIII. The disjoint paths problem. *J. Combinat. Theory, Series B*, 63(1):65–110, 1995.

29 Meirav Zehavi. A randomized algorithm for long directed cycle. *Inform. Process. Lett.*, 116(6):419–422, 2016.