On the Directed Degree-Preserving Spanning Tree Problem

Daniel Lokshtanov[1], Venkatesh Raman[2], Saket Saurabh[1], and Somnath Sikdar[2]

[1] The University of Bergen, Norway.[1]
{daniello|saket.saurabh}@ii.uib.no
[2] The Institute of Mathematical Sciences, India.[2]
{vraman|somnath}@imsc.res.in

Abstract. In this paper we initiate a systematic study of the Reduced Degree Spanning Tree problem, where given a digraph D and a nonnegative integer k, the goal is to construct a spanning out-tree with at most k vertices of reduced out-degree. This problem is a directed analog of the well-studied Minimum-Vertex Feedback Edge Set problem. We show that this problem is fixed-parameter tractable and admits a problem kernel with at most $8k$ vertices on strongly connected digraphs and $O(k^2)$ vertices on general digraphs. We also give an algorithm for this problem on general digraphs with run-time $O^*(5.942^k)$. This adds the Reduced Degree Spanning Tree problem to the small list of directed graph problems for which fixed-parameter tractable algorithms are known. Finally, we consider the dual of Reduced Degree Spanning Tree, that is, given a digraph D and a nonnegative integer k, the goal is to construct a spanning out-tree of D with at least k vertices of full out-degree. We show that this problem is W[1]-hard on two important digraph classes: directed acyclic graphs and strongly connected digraphs.

1 Introduction

Given a directed graph $D = (V, A)$, we say that a subdigraph T of D is an *out-tree* if it is an oriented tree with exactly one vertex s of in-degree zero (called the *root*). An out-tree that contains all vertices of D is an *out-branching* of D. Given a digraph $D = (V, A)$ and an out-tree T of D, we say that a vertex $v \in V$ is of *full-degree* if its out-degree in T is the same as that in D; otherwise, v is said to be of *reduced-degree*. We define the Directed Reduced Degree Spanning Tree (d-RDST) problem as follows.

Input: Given a directed graph $D = (V, A)$ and a positive integer k.
Parameter: The integer k.
Question: Does there exist an out-branching of D in which at most k vertices are of reduced degree?

We call the dual of this problem the Directed Full-Degree Spanning Tree (d-FDST) problem and it is defined as follows.

Input: Given a directed graph $D = (V, A)$ and a positive integer k.
Parameter: The integer k.
Question: Does there exist an out-branching of D in which at least k vertices are of full degree?

d-RDST and d-FDST are natural generalizations of their undirected counterparts, namely, Vertex Feedback Edge Set and Full-Degree Spanning Tree, respectively. In the Vertex Feedback Edge Set problem, given a connected undirected graph $G = (V, E)$ and a non-negative integer k, the goal is to find an edge subset E' incident on at most k vertices such that the deletion of the edges in E' leaves a tree. Note that the resulting graph will span the entire vertex set. Bhatia et al. [4] show that this problem is MAX SNP-hard and describe a $(2+\epsilon)$-approximation algorithm for it for any fixed $\epsilon > 0$. Guo et al. [14] show that this problem is fixed-parameter tractable by demonstrating a problem kernel with at most $4k$ vertices.

The FULL-DEGREE SPANNING TREE problem asks, given a connected undirected graph $G = (V, E)$ and a non-negative integer k as inputs, whether G has a spanning tree T in which at least k vertices have the same degree in T as in G. This problem has attracted a lot of attention of late [5, 6, 4, 14, 13]. Bhatia et al. [5] studied this problem from the point-of-view of approximation algorithms and gave a factor-$\Theta(\sqrt{|V|})$ algorithm for it. They also showed that this problem admits no factor $O(|V|^{1/2-\epsilon})$ approximation algorithm unless NP = co-R. For planar graphs, a polynomial-time approximation scheme (PTAS) was presented. Independently, Broersma et al. [6] developed a PTAS for planar graphs and showed that this problem can be solved in polynomial time in special classes of graphs such as bounded treewidth graphs and co-comparability graphs. Guo et al. [14] studied the parameterized complexity of this problem and showed it to be W[1]-hard. Gaspers et al. [13] give an $O^*(1.9465^{|V|})$ algorithm for this problem.

These problems on undirected graphs were first studied by Pothof and Schut [17] in the context of water distribution networks where the goal is to determine the flow in a network by installing a small number of flow-meters. It so happens that to measure the flow in each pipe of the network, it is sufficient to find a spanning tree of the network and install flow-meters at those vertices whose degree in the spanning tree is lesser than that in the network. To find an optimal number of flow-meters (which are expensive equipment) one needs to find a spanning tree with the largest number of vertices of full degree. Since networks are best modeled by directed graphs, it is natural to consider the directed analog of the notion of a degree-preserving spanning tree.

Related Results on Constrained Spanning Tree Problems. The FULL DEGREE SPANNING TREE problem is one of the many variants of the generic CONSTRAINED SPANNING TREE problem, where one is required to find a spanning tree of a given (di)graph subject to certain constraints. This class of problems has been studied intensely of late [1, 7, 8, 10, 11, 15, 18].

In [10], the authors consider the problem MAX LEAF SPANNING TREE where one is required to find a spanning tree of an undirected graph with the maximum number of leaves. When parameterized by the solution size, this problem admits a kernel with $3.75k$ vertices. In the directed variant of this problem, one has to decide whether an input digraph D has an out-branching with at least k leaves. This problem admits a kernel with $O(k^3)$ vertices, provided the root of the out-branching is *given as part of the input* [11], and has an algorithm with run-time $O(3.72^k \cdot |V(D)|^{O(1)})$ [8]. Another such problem is MAX INTERNAL SPANNING TREE, where the objective is to find a spanning tree (or an out-branching, in case of digraphs) with at least k internal vertices. For the undirected graphs case, an $O(k^2)$-vertex kernel is known for this problem [18]. In the case of directed graphs, an $O(k^2)$-vertex kernel due to [15] and an algorithm with run-time $O^*(40^k)$ due to [7] is known for this problem.

Our Contribution. We study the directed analogues of the DEGREE-PRESERVING SPANNING TREE problem from the point of view of parameterized complexity. We show that d-RDST is fixed-parameter tractable (FPT) by exhibiting a problem kernel with at most $O(k^2)$ vertices. For strongly connected digraphs, d-RDST admits a kernel with at most $8k$ vertices. We describe a branching algorithm for the d-RDST problem with run time $O^*(5.942^k)$. Our fixed parameter tractable and kernelization algorithms are sufficiently non-trivial and exploit structures provided by the problem in an elegant way. Finally, we show that d-FDST is W[1]-hard, by a reduction from the INDEPENDENT SET problem, on two classes of digraphs: directed acyclic graphs (DAGs) and strongly connected digraphs. This also proves that both d-RDST and d-FDST are NP-complete on these graph classes.

Organization of the Paper. In Section 2 we define the relevant notions related to digraphs and parameterized complexity. In Section 3 we show that the d-RDST admits a kernel with at most $O(k^2)$

vertices. We first demonstrate a kernel with $8k$ vertices for strongly connected digraphs and use the ideas therein to develop the $O(k^2)$ kernel for general digraphs. In Section 4 we develop an algorithm for the d-RDST problem with run-time $O^*(5.942^k)$. In Section 5 we show that d-FDST is W[1]-hard even when the input digraph is restricted to be a DAG.

2 Preliminaries

The notation and terminology for digraphs that we follow are from [3]. Given a digraph D we let $V(D)$ and $A(D)$ denote the vertex set and arc set, respectively, of D. If $u, v \in V(D)$, we say that u is an *in-neighbour* (*out-neighbour*) of v if $(u, v) \in A(D)$ ($(v, u) \in A(D)$). The in-degree $d^-(u)$ (out-degree $d^+(u)$) of u is the number of in-neighbours (out-neighbours) of u. Given a subset $V' \subseteq V(D)$, we let $D[V']$ denote the digraph induced on V'. The *underlying undirected graph* $U(D)$ is the undirected graph obtained from D by disregarding the orientation of arcs and deleting an edge for each pair of parallel edges in the resulting graph. The *connectivity components* of D are the subdigraphs induced by the vertices of components of $U(D)$.

A digraph is *oriented* if every pair of vertices has at most one arc between them. A (v_1, v_s)-*walk* in $D = (V, A)$ is a sequence $v_1, \ldots, v_s$ of vertices such that $(v_i, v_{i+1}) \in A$ for all $1 \leq i \leq s-1$. A *dicycle* is a walk $v_1, v_2, \ldots, v_s$ such that $s \geq 3$, the vertices $v_1, \ldots, v_{s-1}$ are distinct and $v_1 = v_s$. A digraph with no dicycles is called a *directed acyclic graph (DAG)*. A digraph D is *strongly connected* if for every pair of distinct vertices $u, v \in V(D)$, there exists a (u, v)-walk and a (v, u)-walk. A *strong component* of a digraph is a maximal induced subdigraph that is strongly connected. The *strong component digraph* $SC(D)$ is the directed acyclic graph obtained by contracting each strong component to a single vertex and deleting any parallel arcs obtained in this process. A strong component S of a digraph D is a *source strong component* if no vertex in S has an in-neighbour in $V(D) \setminus V(S)$. The following is a necessary and sufficient condition for a digraph to have an out-branching.

Proposition 1 [3] *A digraph D has an out-branching if and only if D has a unique source strong component.*

One can therefore check whether a digraph admits an out-branching in time $O(|V(D)| + |A(D)|)$. We assume that our graphs have no self-loops.

3 The d-RDST Problem: An $O(k^2)$-Vertex Kernel

In this section we show that d-RDST admits a problem-kernel with $O(k^2)$ vertices. We first consider the special case when the input digraph is strongly connected and establish a kernel with $8k$ vertices for this case. This will give some insight as to how to tackle the general case. The fact that d-RDST is NP-hard on the class of strongly connected digraphs follows from the fact that d-FDST is NP-hard on this class of graph (see Theorem 4).

3.1 A Linear Kernel for Strongly Connected Digraphs

We actually establish the $8k$-vertex kernel for a more general class of digraphs, those in which every vertex has out-degree at least one. It is easy to see that this class includes strongly connected digraphs (SCDs) as a proper subclass. Call this class of digraphs *out-degree at least one digraphs.*

A common technique to establish a problem-kernel is to devise a set of *reduction rules* which when applied to the input instance (in some specified sequence) produces the kernel. Reduction rules may be thought of as steps in the kernelization algorithm. Formally,

Definition 1 a ***reduction rule*** for a parameterized problem Q is a polynomial-time algorithm that takes an input (I, k) of Q and outputs either

1. YES or NO, in which case the input instance is a YES- or NO-instance, respectively, or
2. an "equivalent" instance (I', k') of Q such that $k' \leq k$.

Two instances are ***equivalent*** if they are both YES-instances or both NO-instances.

Definition 2 An instance (I, k) of a parameterized problem Q is ***reduced with respect to (wrt) a set $\mathcal{R}$ of reduction rules*** if the instance (D', k') output by any reduction rule in $\mathcal{R}$ is the original instance (D, k) itself.

There are three simple reduction rules for the case where the input is an out-degree at least one digraph. We assume that the input is (D, k).

Rule 1. If there exists $u \in V(D)$ such that $d^-(u) \geq k+2$ then return NO; else return (D, k).
Rule 2. If there are $k+1$ vertices of out-degree at least $k+1$ then return NO; else return (D, k).
Rule 3. Let $x_0, x_1 \ldots, x_{p-1}, x_p$ be a sequence of vertices in D such that for $0 \leq i \leq p-1$ we have $d^+(x_i) = 1$ and $(x_i, x_{i+1}) \in A(D)$. Let Y_0 be the set of in-neighbours of $x_1, \ldots, x_{p-1}$ and let $Y := Y_0 \setminus \{x_0, x_1, \ldots, x_{p-2}\}$. Delete the vertices $x_1, \ldots, x_{p-1}$ and add two new vertices z_1, z_2 and the arcs $(x_0, z_1), (z_1, z_2), (z_2, x_p)$. If $y \in Y$ has at least two out-neighbors in $\{x_1, \ldots, x_{p-1}\}$ then add arcs $(y, z_1), (y, z_2)$. If $y \in Y$ has exactly one out-neighbor in $\{x_1, \ldots, x_{p-1}\}$ then add the arc (y, z_1). Return (D, k).

It is easy to see that Rules 1 and 2 are indeed reduction rules for the d-RDST problem on out-degree at least one digraphs. For if a vertex v has in-degree at least $k+2$ then at least $k+1$ in-neighbors of u must be of reduced degree in any out-branching. This shows that Rule 1 is a reduction rule. If a vertex u has out-degree $k+1$ and is of full degree in some out-branching T then T has at least $k+1$ leaves. Since the input digraph is such that every vertex has out-degree at least one, this means that in T there are at least $k+1$ vertices of reduced degree. This shows that any vertex of out-degree $k+1$ must necessarily be of reduced degree in any solution out-branching. Therefore if there are $k+1$ such vertices the given instance is a NO-instance. This proves that Rule 2 is a reduction rule.

Lemma 1. *Rules 3 is a reduction rule for the d-RDST problem.*

Proof. To show that Rule 3 is a reduction rule we need to show that if (D', k) is the instance obtained by applying Rule 3 to an instance (D, k), then D has an out-branching with at most k vertices of reduced out-degree if and only if D' has an out-branching with at most k vertices of reduced degree.

Suppose D' has an out-branching T' with at most k vertices of reduced degree. There are two cases to consider. In the first case, there are no arcs from Y to z_1 or z_2 in T'. In this case we may assume without loss of generality that the path $x_0 \to z_1 \to z_2$ occurs as sub-path of T'. For if $x_0 \to z_1 \to z_2$ is not a subpath of T', then it must be that either z_1 or z_2 is the root of T'. It is easy to see that one can always transform such an out-branching to one in which x_0 is the root, contains the path $x_0 \to z_1 \to z_2$ and which has at most k vertices of reduced degree. It is possible that z_2 could have x_p as its child. In this case one can simply replace z_1 by the path $x_1 \to \cdots \to x_{p-1}$, delete z_2, and connect x_{p-1} to x_p if $d^+_{T'}(z_2) = 1$, to obtain an out-branching for D with at most k vertices of reduced degree.

In the second case, there exists at least one vertex $y \in Y$ with arcs to $\{z_1, z_2\}$. Suppose that T' contains the arcs $(y_1, z_1), (y_2, z_2)$, where $y_1, y_2 \in Y$ and $y_1 \neq y_2$. Note that both x_0 and z_1 are of out-degree zero in T' and hence of reduced degree. Observe that $T' \setminus \{z_1\}$ is an out-branching for

$D' \setminus \{z_1\}$ as z_1 is a leaf in T'. We transform T' into another out-branching for D' by omitting the arc (y_1, z_1) and inserting the arc (x_0, z_1). In this new out-branching, x_0 is of full degree and y_1 is possibly of reduced degree but the number of vertices of reduced degree does not increase.

Therefore we can assume without loss of generality that in T' there exists exactly one vertex in Y with out-arcs to $\{z_1, z_2\}$. Now suppose that T' contains the arcs (y, z_1) and (y, z_2) where $y \in Y$. Then both x_0 and z_1 are of reduced degree. By deleting the arc (y, z_2) and including (z_1, z_2) we obtain an out-branching of D' in which the number of vertices of reduced degree is at most that in T'. We can therefore assume without loss of generality that in T' the vertex $y \in Y$ has exactly one out-arc to $\{z_1, z_2\}$. Suppose $(y, z_2) \in A(T')$. Then y must be of reduced degree as whenever we have an arc (y, z_2) then we also have an arc (y, z_1). In this case we transform T' by deleting $(y, z_2), (x_0, z_1)$ and introducing $(y, z_1), (z_1, z_2)$. The resulting digraph is an out-branching with at most k vertices of reduced degree as x_0 becomes of reduced degree but z_1 gets full-degree. Therefore we are left to consider the case when y has an arc to z_1 only. Let x_s be the first out-neighbor of y in $\{x_1, \ldots, x_{p-1}\}$. Delete z_1, z_2 and connect x_0 to the dipath $x_1 \to \cdots \to x_{s-1}$ and y to the dipath $x_s \to \cdots \to x_{p-1}$. Add the arc (x_{p-1}, x_p) if $(z_2, x_p) \in A(T')$. The resulting digraph is an out-branching for D with at most k vertices of reduced degree.

To prove the converse, suppose that D has an out-branching T with at most k vertices of reduced degree. Again there are two cases to consider.

Case 1. There are no arcs from Y to any x_i, for $1 \le i \le p-1$, in T. There are two sub-cases here. Either T contains the dipath $x_0 \to x_1 \to \cdots x_{p-1} \to x_p$, in which case we can compress it to the path (x_0, z_1, z_2, x_p) to obtain an out-branching T' for D' with at most k vertices of reduced degree. Otherwise one of the vertices $x_1, \ldots, x_p$ must be the root of T. If x_p is the root, then T contains the dipath $x_0 \to x_1 \to \cdots x_{p-1}$ and we replace it by (x_0, z_1, z_2) to obtain an out-branching T' of D'. If one of $x_1, \ldots, x_{p-1}$ is the root, then delete $x_1, \ldots, x_{p-1}$, make z_1 the root and add the arcs $(z_1, z_2), (z_2, x_p)$. This transforms T into an out-branching of D' with at most k vertices of reduced degree.

Case 2. Now suppose that in T the vertices $y_{i_1}, \ldots, y_{i_s} \in Y$ have out-neighbors among the vertices $x_1, \ldots, x_{p-1}$. Since T is an out-branching, the out-neighbors of y_{i_j} and y_{i_l} are disjoint for all $j \neq l$. Let x_{i_j} be the first out-neighbor of y_{i_j} among $\{x_1, \ldots, x_{p-1}\}$ in the tree T. Transform T by deleting out-arcs from the y_{i_j}'s such that for $1 \le j \le s$, the only out-neighbor of y_{i_j} among $\{x_1, \ldots, x_{p-1}\}$ is x_{i_j}. Sort the vertices $y_{i_1}, \ldots, y_{i_s}$ in increasing order based on x_{i_j}'s. Without loss of generality we assume that the sorted order is also $y_{i_1}, \ldots, y_{i_s}$. Connect the vertices x_is in such a way that the resulting digraph contains the paths: $x_0 \to x_1 \to \cdots \to x_{i_1-1}, y_{i_1} \to x_{i_1} \to \cdots \to x_{i_2-1}, \ldots, y_{i_s} \to x_{i_s} \to \cdots \to x_{p-1} \cdots$, where x_{i_j-1}, for $1 \le j \le s$, are vertices of out-degree zero in T. Note that the last path $y_{i_s} \to x_{i_s} \to \cdots$ in the sequence contains all vertices $x_{i_s}, \ldots, x_{p-1}$ but may or may not contain the vertex x_p. Observe that the only vertices $y \in \{y_{i_1}, \ldots, y_{i_s}\}$ whose out-degree is reduced in this process has at least two out-neighbors among the vertices $\{x_1, \ldots, x_{p-1}\}$. Hence for every y_{i_j} whose out-degree is reduced in this process, there exists a distinct vertex in $x_1, \ldots, x_{p-1}$ of out-degree zero in T which is of full degree in the resulting digraph. Thus the number of vertices of reduced degree does not change in this transformation and it is easy to verify that this new digraph is actually an out-branching T_1 of D. Now delete the arcs $(y_{i_1}, x_{i_1}), \ldots, (y_{i_{s-1}}, x_{i_{s-1}})$, and add $(x_{i_1-1}, x_{i_1}), \ldots, (y_{i_{s-1}-1}, x_{i_{s-1}})$, and obtain another out-branching T_2 of D and with at most k vertices of reduced out-degree. To obtain an out-branching of D' from T_2, we proceed as follows: delete $x_1, \ldots, x_{p-1}$, add the arcs (y_{i_s}, z_1), (z_1, z_2) and connect z_2 to the out-neighbor of x_{p-1}, if any. Note that this transforms T_1 into an out-branching of D' with at most k vertices of reduced degree.

This completes the proof of the lemma. □

Our kernelization algorithm consists in applying Rules 1 to 3 repeatedly, in that order, until the given instance is reduced.

Theorem 1. *Let (D, k) be a yes-instance of the d-RDST problem on out-degree at least one digraphs reduced wrt Rules 1 to 3. Then $|V(D)| \leq 8k$.*

Proof. Since (D, k) is a YES-instance of the problem, let T be an out-branching of D with at most k vertices of reduced degree. Every vertex of D is of out-degree at least one and hence each leaf of T is a vertex of reduced degree. Therefore if T has l leaves then $l \leq k$. It is a well-known fact that in any (undirected) tree with l leaves, the number of vertices of total degree at least three is at most $l - 1$. Therefore the number of vertices of out-degree at least two in T is at most $l - 1 \leq k - 1$. It remains to bound the vertices which have out-degree exactly one in T. Let W be the union of the set of vertices of out-degree at least two in T and the set of vertices of reduced degree in T. Let $\mathcal{P}$ be the set of maximal dipaths in T such that for any dipath $P = x_0 \to x_1 \to \cdots \to x_p$ in $\mathcal{P}$ we have that (1) $d^+_D(x_i) = 1$ for $0 \leq i \leq p - 1$, and (2) $x_p \in W$. Observe that every vertex with out-degree exactly one in T is contained in exactly one path in $\mathcal{P}$. If Z is the set of vertices of out-degree exactly one in T and not contained in W, then $|Z| \leq \sum_{P \in \mathcal{P}}(|P| - 1)$. By Rule 3, any dipath $P \in \mathcal{P}$ has at most four vertices and since the number of dipaths in $\mathcal{P}$ is at most $|W|$, we have $|Z| \leq 3 \cdot |\mathcal{P}| \leq 3 \cdot |W| \leq 3 \cdot (l - 1 + k) \leq 6k$. Therefore the total number of vertices in T is at most $8k$ as claimed. ☐

Observe that the crucial step in the proof above was to bound the number of leaves in the solution out-branching. For out-degree at least one digraphs this is easy since every leaf is a vertex of reduced degree. This is not the case with general digraphs which may have an arbitrary number of vertices of out-degree zero, all of which are of full degree in any out-branching. In the next subsection we present a set of reduction rules for the d-RDST problem in general digraphs which help us bound the number of vertices of out-degree zero in terms of the parameter k.

3.2 An $O(k^2)$-Vertex Kernel in General Digraphs

For general digraphs, the reduction rules that we will describe work with an annotated instance of the problem. We assume that we are given (D, k) and a set $X \subseteq V(D)$ of vertices which will be of reduced degree in *any* out-branching with at most k vertices of reduced degree. The question in this case is to decide whether D admits an out-branching where the set of vertices of reduced degree is $X \cup S$, where $S \subseteq V(D) \setminus X$ and $|S| \leq k$. The parameter in this case is k. Call such an out-branching a *solution out-branching.* To obtain a kernel for d-RDST, we apply the reduction rules to an instance (D, k) after setting $X = \emptyset$.

Given an instance (D, X, k), we define the *conflict set* of a vertex $u \in V(D) \setminus X$ as $C(u) := \{v \in V(D) \setminus X : N^+(u) \cap N^+(v) \neq \emptyset\}$. Clearly vertices of out-degree zero have an empty conflict set. If a vertex v has a non-empty conflict set then in any out-branching either v loses its degree or *every* vertex in $C(v)$ loses its degree. Moreover if $u \in C(v)$ then $v \in C(u)$ and in this case we say that u and v are in conflict. The *conflict number* of D is defined as $c(D) := \sum_{v \in V(D) \setminus X} |C(v)|$.

We assume that the input instance is (D, X, k) and the kernelization algorithm consists in applying each reduction rule repeatedly, in the order given below, until no longer possible. Therefore when we say that Rule i is indeed a reduction rule we assume that the input instance is reduced wrt the rules preceding it. In what follows, we omit formal proofs whenever an intuitive explanation suffices.

Rule 1. If there exists $u \in V(D)$ such that the number of in-neighbors of u in $V(D) \setminus X$ is at least $k + 2$ then return NO; else return (D, X, k).

In the last subsection, we already showed that this rule is indeed a reduction rule.

Rule 2. If $u \in V(D) \setminus X$ such that $|C(u)| > k$, set $X \leftarrow X \cup \{u\}$ and $k \leftarrow k - 1$. Furthermore if $d^+(u) = 1$ then delete the out-arc from u and return (D, X, k).

If the conflict set $C(u)$ of $u \in V(D) \setminus X$ is of size at least $k+1$ and if u is of full degree in some out-branching T, then every vertex in $C(u)$ must be of reduced degree in T. Therefore if (D, X, k) is a YES-instance then u must lose its degree in *any* solution out-branching. Moreover if u has out-degree exactly one, then the out-arc from u must be deleted. This shows that Rule 2 is a reduction rule.

Rule 3. If $c(D) > 2k^2$ then return NO, else return (D, X, k).

Lemma 2. *Rule 3 is a reduction rule for the d-RDST problem.*

Proof. To see why Rule 3 qualifies to be a reduction rule, construct the *conflict graph* $\mathcal{C}_{D,X}$ of the instance (D, X, k) which is defined as follows. The vertex set $V(\mathcal{C}_{D,X}) := V(D) \setminus X$ and two vertices in $V(\mathcal{C}_{D,X})$ have an edge between them if and only if they are in conflict. Since the size of the conflict set of any vertex is at most k, the degree of any vertex in $\mathcal{C}_{D,X}$ is at most k. The key observation is that if T is any solution out-branching of (D, X, k) in which the set of vertices of reduced degree is $X \cup S$ with $S \subseteq V(D) \setminus X$, then S forms a vertex cover of $\mathcal{C}_{D,X}$. Since we require that $|S| \leq k$, the number of edges in $\mathcal{C}_{D,X}$ is at most k^2. For a vertex $v \in V(D) \setminus X$, let $d'(v)$ be the number of neighbors of vertex v in the conflict graph $\mathcal{C}_{D,X}$. Observe that $c(D) := \sum_{v \in V(D) \setminus X} |C(v)| = \sum_{v \in V(D) \setminus X} d'(v) \leq 2k^2$. The last inequality follows from the fact that sum of degrees of vertices in a graph is equal to twice the number of edges. □

Rule 4. If $u \in V(D)$ such that $d^+(u) = 0$ and $d^-(u) = 1$ then delete u from D and return (D, X, k).

It is easy to see that Rule 4 is a reduction rule: vertex u does not determine whether its parent is of full or reduced degree in a solution out-branching and therefore can be safely deleted.

Rule 5. Let $u \in V(D)$ be of out-degree zero and let $v_1, \ldots, v_r$ be its in-neighbors, where $r > 2$. Delete u and add $\binom{r}{2}$ new vertices $u_{12}, u_{13}, \ldots, u_{r-1,r}$; for a newly added vertex u_{ij} add the arcs (v_i, u_{ij}) and (v_j, u_{ij}). Return (D, X, k).

Note that vertex u forces at least $r-1$ vertices from $\{v_1, \ldots, v_r\}$ to be of reduced degree in any out-branching. This situation is captured by deleting u and introducing $\binom{r}{2}$ vertices as described in the rule. The upshot is that each vertex of out-degree zero has in-degree exactly two. A formal proof that Rule 5 is a reduction rule is omitted.

Rule 6. If $u, v \in V(D) \setminus X$ have $p > 1$ common out-neighbors of out-degree zero, delete all but one of them. Return (D, X, k).

Rule 7. If $u \in V(D)$ is of out-degree zero such that at least one in-neighbor of u is in X, delete u. Return (D, X, k).

By Rule 5, it is clear that if $u, v \in V(D) \setminus X$ have at least two common out-neighbors of out-degree zero then these out-neighbors have in-degree exactly two. It is intuitively clear that these out-neighbors are equivalent in some sense and it suffices to preserve just one of them. It is easy to show that the original instance has a solution out-branching if and only if the instance obtained by one application of Rule 6 has a solution out-branching. As for Rule 7, if u has in-neighbors v and w and if $v \in X$, we can delete the arc (v, u) without altering the solution structure. But then v is a private neighbor of w of out-degree zero and hence can be deleted by Rule 4.

Rule 8. Let $x_0, x_1 \ldots, x_{p-1}, x_p$ be a sequence of vertices in D such that for $0 \leq i \leq p-1$ we have $d^+(x_i) = 1$ and $(x_i, x_{i+1}) \in A(D)$. Let Y_0 be the set of in-neighbours of $x_1, \ldots, x_{p-1}$ and let $Y := Y_0 \setminus \{x_0, x_1, \ldots, x_{p-2}\}$. Delete the vertices $x_1, \ldots, x_{p-1}$ and add two new vertices z_1, z_2 and the arcs $(x_0, z_1), (z_1, z_2), (z_2, x_p)$. If $y \in Y$ has at least two out-neighbors in $\{x_1, \ldots, x_{p-1}\}$ then add arcs $(y, z_1), (y, z_2)$. If $y \in Y$ has exactly one out-neighbor in $\{x_1, \ldots, x_{p-1}\}$ then add the arc (y, z_1). Return (D, X, k).

This is Rule 3 from the previous subsection where it was shown to be a reduction rule for the d-RDST problem (note that the proof of Lemma 1 did not use the fact that the input was an out-degree at least one digraph). By Rule 2, no vertex on the path $x_0, x_1, \ldots, x_{p-1}$ is in X and therefore the proof of Lemma 1 continues to hold for the annotated case as well.

We are now ready to bound the number of vertices of out-degree zero in a reduced instance of the annotated problem.

Lemma 3. *Let (D, X, k) be a yes-instance of the annotated d-RDST problem that is reduced wrt Rules 1 through 8 mentioned above. Then the number of vertices of out-degree zero in D is at most k^2.*

Proof. Let u be a vertex of out-degree zero. By Rules 4 and 5, it must have exactly two in-neighbors, say, x and y. By Rule 7, neither x nor y is in X and are therefore still in conflict in the reduced graph. Hence, either x or y must be of reduced degree in any solution out-branching. Furthermore any vertex not in X can have at most k out-neighbors of out-degree zero since, by Rule 2, any vertex not in X is in conflict with at most k other vertices and, by Rule 6, two vertices in conflict have at most one out-neighbor of out-degree zero. Since (D, X, k) is assumed to be a YES-instance, at most k vertices can lose their out-degree in any solution out-branching. Moreover any vertex of out-degree zero is an out-neighbor of at least one vertex of reduced degree. Therefore the total number of vertices of out-degree zero is at most k^2. □

Lemma 4. [⋆][1] *Let (D, k) be a yes-instance of the d-RDST problem and let (D_1, X, k_1) be an instance of the annotated d-RDST problem reduced wrt Rules 1 through 8 by repeatedly applying them on (D, k) by initially setting $X = \emptyset$. Then $|V(D_1)| \leq 8(k^2 + k)$.*

We now show how to obtain a kernel for the original (unannotated) version of the problem. Let (D, k) be an instance of the d-RDST problem and let (D', X, k') be the instance obtained by applying reduction rules 1 through 8 on (D, k) until no longer possible, by initially setting $X = \emptyset$. By Lemma 4, we know that $|V(D')| \leq 8(k^2 + k)$ if (D, k) is a YES-instance, and that $k' + |X| = k$. To get back an instance of the unannotated version, apply the following transformation on (D', X, k'). Let $X = \{x_1, \ldots, x_r\}$. For each $x_i \in X$ add $k + 1$ new vertices $z_{i1}, \ldots, z_{i,k+1}$ and out-arcs (x_i, z_{ij}) for all $1 \leq j \leq k + 1$. Then add $k + 1$ new vertices $u_1, \ldots, u_{k+1}$ and out-arcs (u_j, z_{ij}) for all $1 \leq i \leq r$ and $1 \leq j \leq k + 1$. Finally add a vertex u and out-arcs (u, u_j) for $1 \leq j \leq k + 1$ and (x_i, u) for $1 \leq i \leq r$. Call the resulting digraph D'' and set $k'' = k' + |X|$. It is easy to see that (D', X, k') is a YES-instance of the annotated version of d-RDST if and only if (D'', k'') is a YES-instance of the (unannotated) d-RDST problem.

For if D' has an out-branching T' with at most $k' + |X|$ vertices of reduced degree with all vertices in X of reduced degree, then modify T' into an out-branching T'' for D'' as follows. Add arcs (x_1, u), (u, u_j) for all $1 \leq j \leq k + 1$ and (u_j, z_{ij}) for all i, j. Clearly T'' is an out-branching of D'' and has the same number of vertices of reduced degree as T'. Conversely if D'' admits an out-branching T'' with $k' + |X|$ vertices of reduced degree, then it must be the case that all vertices in X are of reduced degree in T''. For if $x_i \in X$ is of full-degree then the vertices $u_1, \ldots, u_{k+1}$ are of reduced degree, contradicting the fact that T'' has at most $k' + |X| = k$ vertices of reduced degree. Therefore in T'', we may assume that the vertices $u, u_1, \ldots, u_{k+1}$ and the z_{ij}'s are of full-degree. This implies that in T'' there are at most k' vertices from $V(D') \setminus X$ of reduced degree. Furthermore in T'', the vertex u has as in-neighbor a vertex $x_i \in X$. Therefore by deleting $u, u_1, \ldots, u_{k+1}$ and the z_{ij}'s, we obtain an out-branching T' with at most $k' + |X|$ vertices of reduced degree with all vertices of X of reduced degree. This completes the proof of the reduction from the annotated to the unannotated case. Since we add at most $k(k + 1) + k + 2$ vertices in the process, we have

[1] Proofs of results labelled with [⋆] have been moved to the appendix due to space restrictions.

Theorem 2. *The d-RDST problem admits a problem kernel with at most $9k^2 + 10k + 2$ vertices, where the parameter k is the number of vertices of reduced degree.*

4 An Algorithm for the *d*-RDST Problem

In this section we describe a branching algorithm for the d-RDST problem with run-time $O^*(5.942^k)$. We first observe that in order to construct a solution out-branching of a given digraph, it is sufficient to know which vertices will be of reduced degree.

Lemma 5. *Let $D = (V, A)$ be a digraph and let X be the set of vertices of reduced degree in some out-branching of D. Given D and X, one can in polynomial time construct an out-branching of D in which the number of vertices of reduced degree is at most $|X|$.*

Proof. We describe an algorithm that constructs such an out-branching of D. Given D and X, our algorithm first constructs a digraph D' with vertex set $V(D)$ in which

1. all vertices in $V(D) \setminus X$ are connected to their out-neighbors in D by *solid* arcs;
2. a vertex $x \in X$ has a *dotted* out-arc to a vertex y if $(x, y) \in A(D)$ and y has no solid in-arc in D'.

We are guaranteed that there exists an out-branching of D' in which all solid arcs are present but in which one or more dotted arcs may be missing. Note that in D', a vertex with a solid in-arc has no other (solid or dotted) in-arcs.

Our algorithm now runs through all possible choices of the root of the proposed out-branching. For each choice of root, it does a modified breadth-first search starting at the root. In the modified BFS-routine, when the algorithm visits a vertex v, it solidifies all dotted out-arcs from v, if any. For each dotted arc (v, w) that it solidifies, it deletes all dotted in-arcs to w. The algorithm then inserts the neighbors of v in the BFS-queue. If the BFS-tree thus constructed includes all vertices of D', the algorithm outputs this out-branching, or else, moves on to the next choice of root.

Claim. Suppose that r is the root of an out-branching of D' in which X is the vertex-set of reduced degree. Then the above algorithm, on selecting r as root, succeeds in constructing an out-branching in which the number of vertices of reduced degree is at most $|X|$.

In order to prove this claim, it is sufficient to show that in the BFS-tree T constructed by the algorithm with r as root, every vertex of D' is reachable from r. This suffices because every vertex in $V(D') \setminus X$ is of full degree in T.

Therefore let v be a vertex not reachable from r such that the distance, in D', from v to r is the shortest among all vertices not reachable from r in T. Let $r, v_1, \ldots, v_l, v$ be a shortest dipath from r to v in D'. By our choice of v, all vertices $v_1, \ldots, v_l$ are reachable from r in T. Note that the arc (v_l, v) must have been dotted and in fact all in-arcs to v were dotted in D'. When the algorithm visited v_l, the only reason it could not solidify the arc (v_l, v) must have been because v *already* had a solid in-arc into it and hence the arc (v_l, v) had already been deleted. Suppose that v has a solid in-arc from u. Then u must have already been visited *before* v_l at which time the dotted arc (u, v) was solidified. But this means that u, and hence v, is reachable from r in the BFS-tree T, a contradiction. □

By Lemma 5 and Theorem 2, there exists an $O^*(k^{O(k)})$ algorithm for the d-RDST problem. In the rest of this section, we give an improved algorithm with run-time $O^*(c^k)$, for a constant c. Our algorithm (see Figure 1) is based on the simple observation that if two vertices u and v of the input digraph D have a common out-neighbor then one of them must be of reduced degree in *any* out-branching of D. The algorithm recurses on vertex-pairs that have a common out-neighbor and, along each branch of the recursion tree, builds a set X of vertices which would be the candidate vertices of reduced degree in the out-branching that it attempts to construct. When there are no vertices to branch on, it reduces the instance (D, X, k) wrt the following rules.

RDST (D, X, k)
Input: A digraph $D = (V, A)$; $X \subseteq V$, such that the vertices in X will be of reduced degree in the out-branching that is being constructed; an integer parameter k. The algorithm is initially called after setting $X = \emptyset$.
Output: An out-branching of D in which every vertex of X is of reduced degree and with at most k vertices of reduced degree in total, if one exists, or NO, signifying that no such out-branching exists.

1. If $k < 0$ or $|X| > k$ return NO.
2. If no two vertices in $V(D) \setminus X$ have a common out-neighbor then
   (a) Reduce (D, X, k) wrt Rules $1'$ through $5'$.
   (b) For each $(k - |X|)$-sized subset Y of $V(D) \setminus X$, check if there exists an out-branching of D in which the vertex set of reduced degree is $X \cup Y$. If yes, then "expand" this out-branching to an out-branching for the original instance and return the solution; else return NO.
3. Let $u, v \in V(D) \setminus X$ be two vertices with a common out-neighbor then
   (a) $X \leftarrow X \cup \{u\}$; $Z =$ Call $\mathbf{RDST}(D, X, k - 1)$.
   (b) If $Z \neq$ NO then return Z.
   (c) $X \leftarrow X \cup \{v\}$; Return $\mathbf{RDST}(D, X, k - 1)$.

Fig. 1. Algorithm **RDST**.

Rule 1′. If $u \in X$ and $d^+(u) = 1$, delete the out-arc from u and return (D, X, k).
Rule 2′. Let $u \in V(D)$ be of out-degree zero and let $v_1, \ldots, v_r$ be its in-neighbors. If $v_i \in X$ for all $1 \leq i \leq r$, assign v_1 as the parent of u and delete u. If there exists $1 \leq i \leq r$ such that $v_i \notin X$ then assign v_i as the parent of u and delete u. Return (D, X, k).
Rule 3′. This is Rule 8 from Section 3.2.

Rule $1'$ is a reduction rule because a vertex of out-degree exactly one that is of reduced degree must necessarily lose its only out-arc. As for Rule $2'$, we know that in the instance (D, X, k) obtained after the algorithm finishes branching, no two vertices of $V(D) \setminus X$ have a common out-neighbor and therefore at least $r - 1$ in-neighbors of u must be of reduced degree. If all in-neighbors of u are of reduced degree, we arbitrarily fix one of them as parent of u (so that we can construct an out-branching of the original instance later on) and delete u. If exactly $r - 1$ in-neighbors of u are already of reduced degree, we choose that in-neighbor not in X as the parent of u and delete u. Also note that when applying Rule $3'$ to a path $x_0, x_1, \ldots, x_{p-1}, x_p$, the vertices $x_0, x_1, \ldots, x_{p-1}$ are not in X, by Rule $1'$. Therefore if Y is the set of in-neighbors of $x_1, \ldots, x_{p-1}$, excluding $\{x_0, x_1, \ldots, x_{p-2}\}$, then $Y \subseteq X$.

Observe the following:

1. By Rule $2'$, no vertex in the reduced instance (D, X, k) has out-degree zero.
2. Every vertex in the subdigraph induced by $V(D) \setminus X$ has in-degree exactly one and hence each connectivity component (a connected component in the undirected sense) is either a dicycle, or an out-tree or a dicycle which has out-trees rooted at its vertices. Such a digraph is called a *pseudo out-forest* [19].

We now reduce the instance (D, X, k) wrt the following two rules:

Rule 4′. If at least $k + 1 - |X|$ connectivity components of $D[V \setminus X]$ contain dicycles, then return NO; else return (D, X, k).
Rule 5′. If a connectivity component of $D[V \setminus X]$ is a dicycle C such that no vertex in $V(C)$ has an out-neighbor in X, pick a vertex $u \in X$ with an arc to C and fix it as the "entry point" to C; delete C and set $k \leftarrow k - 1$; return (D, X, k).

Rule 4′ is a reduction rule as every connectivity component that has a dicycle contains at least one vertex that will be of reduced degree. If the number of such components if at least $k+1-|X|$, one cannot construct an out-branching with at most k vertices of reduced degree. To see that Rule 5′ is a reduction rule, first note that since C has no out-arcs, it cannot contain the root of the proposed out-branching. Any path from the root to C must necessarily include a vertex from X and it does not matter which arc out of X we use to get to C, since every vertex in X is of reduced degree anyway. Moreover in any out-branching, exactly one vertex of C must be of reduced degree. Therefore if (D', X', k') is the instance obtained by one application of Rule 5′ to the instance (D, X, k), then it is easy to see that these instances must be equivalent.

Lemma 6. *Let (D, X, k) be an instance of the d-RDST problem in which no two vertices of $V(D) \setminus X$ have a common out-neighbor, and reduced wrt Rules 1′ through 5′. Then $|V(D) \setminus X| \leq 7|X|$.*

Proof. Let D' be a digraph obtained from D by deleting all out-arcs from the vertices in X. Therefore in D', every vertex of X has out-degree zero and in-degree at most one. We show that a connectivity component of D' that has p vertices of X has at most $7p$ vertices of $V(D') \setminus X$. This will prove the lemma.

If a connectivity component of D' is an out-tree, then every leaf of this out-tree is a vertex of X. If there are p leaves, then by an argument similar to the one in the proof of Theorem 1, one can show that the number of vertices in the out-tree is at most $8p$. Since exactly p of these vertices are from X, the number of vertices of $V(D') \setminus X$ in the out-tree is at most $7p$. Therefore let R be a connectivity component of D' containing a dicycle such that $|V(R) \cap X| = p$. Define a $d1$-*dipath* to be a dipath $x_0, x_1, \ldots, x_p$ such that (1) $d^+_{D'}(x_i) = 1$ for all $0 \leq i \leq p$, (2) there exists an in-neighbor of x_0 of out-degree at least two, and (3) the out-neighbor of x_p is either in X or has out-degree at least two. Since the vertices of X have out-degree zero in R, vertices of a $d1$-dipath must be from $V(R) \setminus X$. Note that any vertex of $V(R) \setminus X$ is either of out-degree at least two or lies on a $d1$-dipath.

Define $X_r = V(R) \cap X$ and $t = \sum_{v \in V(R) \setminus X_r} (d^+_{D'}(v) - 1)$. Then $t + |V(R) \setminus X_r|$ denotes the total number of arcs that start at $V(R) \setminus X_r$. Since no two vertices of $V(R) \setminus X_r$ have a common out-neighbor, we have $t + |V(R) \setminus X_r| \leq |V(R)|$, which implies that $t \leq |X_r| = p$. The number of vertices in $V(R) \setminus X_r$ of out-degree at least two is bounded above by t. A $d1$-dipath either ends at a vertex of degree at least two or at a vertex in X. Therefore the number of such paths is at most $2p$. By the path rule (Rule 3′), any such path has at most three vertices and so the total number of vertices on such paths is at most $6p$. Hence $|V(R) \setminus X| \leq 7p$.

This completes the proof of the lemma. □

To construct an out-branching, it is sufficient to choose the remaining $k - |X|$ vertices of reduced degree from the vertices in $V(D) \setminus X$. Setting $|X| = c$, the exponential term in the run-time of the algorithm is bounded above by the function

$$\sum_{c=0}^{k} 2^c \cdot \binom{7c}{k-c} \leq k \cdot \max_{0 \leq c \leq k} 2^c \cdot \binom{7c}{k-c}.$$

In the latter function, we must have $k - c \leq 7c$ which implies that $k/8 \leq c$, and one can show that this function attains a maximum at $c = k/2$ where its value is $k \cdot 2^{k/2} \cdot \binom{7k/2}{k/2}$. Using the inequality $\binom{n}{r} \leq n^n/(r^r \cdot (n-r)^{n-r})$, we can bound this by $k \cdot 5.942^k$.

Theorem 3. *Given a digraph D and a nonnegative integer k, one can decide whether D has an out-branching with at most k vertices of reduced degree, and if so, construct such an out-branching in time $O^*(5.942^k)$.*

5 The d-FDST Problem

We now show that d-FDST is W[1]-hard even on DAGs. This is a modification of the reduction presented in [4] (Lemma 3.2).

Theorem 4. [$\star$] *The d-FDST problem parameterized by the solution size is* W[1]*-hard on directed acyclic graphs and strongly connected digraphs. Also the d-RDST problem is* NP*-hard on the class of strongly connected digraphs.*

References

1. N. Alon, F.V. Fomin, G. Gutin, M. Krivelevich and S. Saurabh. *Spanning Directed Trees With Many Leaves.* To appear in SIAM Journal of Discrete Math.
2. S. Arnborg, J. Lagergren and D. Seese. *Easy Problems for Tree-Decomposable Graphs.* Journal of Algorithms, Vol. 12, pp. 308-340, 1991.
3. J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications.* Springer-Verlag, 2000.
4. R. Bhatia, S. Khuller and R. Pless. *On Local Search and Placement of Meters in Networks.* SIAM Journal on Computing Vol. 32, Issue 2, pp. 470-487, 2003.
5. R. Bhatia, S. Khuller, R. Pless and Y. Sussmann. *The Full-Degree Spanning Tree Problem.* Networks, Vol. 36, pp. 203-209, 2000.
6. H.J. Broersma, A. Huck, T. Kloks, O. Koppios, D. Kratsch, H. Müller and H. Tuinstra. *Degree-Preserving Forests.* Networks, Vol. 35, pp. 26-39, 2000.
7. N. Cohen, F.V. Fomin, G. Gutin, E.J. Kim, S. Saurabh and A. Yeo. *Algorithms for Finding k-Vertex Out-Trees and its Applications to the k-Internal Out-Branching Problem.* Manuscript.
8. J. Daligault, G. Gutin, E.J. Kim and A. Yeo. *FPT-Algorithms and Kernels for the Directed k-Leaf Problem.* Manuscript.
9. R.G. Downey and M.R. Fellows. *Parameterized Complexity.* Springer-Verlag, 1999.
10. V. Estivill-Castro, M.R. Fellows, M.A. Langston and F.A. Rosamond. *FPT is P-Time Extremal Structure I.* In Proc. of Algorithms and Complexity in Durham (ACiD), 2005.
11. H. Fernau, F.V. Fomin, D. Lokshtanov, D. Raible, S. Saurabh and Y. Villanger. *Kernel(s) for Problems With No Kernels: On Out-Trees with Many Leaves.* In Proc. of the 26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, pp. 421-432, Schloss Dagstuhl-Leibnitz-Zentrum für Informatik, Germany, 2009.
12. J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer-Verlag. 2006.
13. S. Gaspers, S. Saurabh and A.A. Stepanov. *A Moderately Exponential-Time Algorithm for Full-Degree Spanning Tree.* In Proc. of the 5th Annual Conference on Theory and Applications of Models of Computation (TAMC08), Springer LNCS, Vol. 4978, pp. 478-489, 2008.
14. J. Guo, R. Niedermeier, and S. Wernicke. *Fixed-Parameter Tractability Results for Full-Degree Spanning Tree and Its Dual.* In Proc. the 2nd International Workshop on Parameterized and Exact Computation (IWPEC), Springer LNCS, Vol. 4196, pp. 203-214, 2006.
15. G. Gutin, E.J. Kim and I. Razgon. *Minimum Leaf Out-Branching and Related Problems.* In Proc. of the 4th International Conference on Algorithmic Aspects in Information and Management (AAIM'08), Springer LNCS Vol. 5034, pp. 235-246, 2008.
16. R. Niedermeier. *An Invitation to Fixed-Parameter Algorithms.* Oxford University Press, 2006.
17. I.W.M. Pothof and J. Schut. *Graph-Theoretic Approach to Identifiability in a Water Distribution Network.* Memorandum 1283, Universiteit Twente, Twente, The Netherlands, 1995.
18. E. Prieto and C. Sloper. *Reducing to Independent Set Structure: The Case of k-Internal Spanning Tree.* Nordic Journal of Computing, Vol. 12, Issue 3, pp. 308-318, 2005.
19. S. Saurabh. *Exact Algorithms for Optimization and Parameterized Versions of Some Graph-Theoretic Problems.* PhD Thesis, Homi Bhaba National Institute, Mumbai, India, August 2007.

6 Appendix

6.1 Proof of Lemma 4

Lemma 4. *Let (D, k) be a yes-instance of the d-RDST problem and let (D_1, X, k_1) be an instance of the annotated d-RDST problem reduced wrt Rules 1 through 8 by repeatedly applying them on (D, k) by initially setting $X = \emptyset$. Then $|V(D_1)| \leq 8(k^2 + k)$.*

Proof. Since reduction rules map YES-instances to YES-instances and does not allow the parameter to increase, it is clear that (D_1, X, k_1) is a YES-instance of the annotated d-RDST problem and that $k_1 + |X| \leq k$. Therefore let T_1 be a solution out-branching of (D_1, X, k_1). A leaf of T_1 is either a vertex of out-degree zero in D_1 or a vertex of reduced degree. By Lemma 3, the total number of vertices of out-degree zero is at most $k_1^2 \leq k^2$ and since T_1 is a solution out-branching, the total number of vertices of reduced degree is at most $k_1 + |X| \leq k$. The number of vertices of out-degree at least two is now bounded by $k^2 + k$. As in the proof of Theorem 1, let W be the union of the set of vertices of out-degree at least two in T_1 and the set of vertices of reduced degree in T_1. Observe that $X \subseteq W$. Furthermore, let $\mathcal{P}$ be the set of maximal dipaths in T_1 such that for any dipath $P = x_0 \to x_1 \to \cdots \to x_p$ in $\mathcal{P}$ we have that (1) $d^+_{D_1}(x_i) = 1$ for $0 \leq i \leq p-1$, and (2) $x_p \in W$. By Rule 8, any dipath $P \in \mathcal{P}$ has at most four vertices. Using an argument similar to the one used in the proof of Theorem 1, the number of vertices in the dipaths of $\mathcal{P}$, excluding the last vertex on these paths, can be upper bounded by $6(k^2 + k)$. Hence the total number of vertices in T_1 is at most $8(k^2 + k)$. □

6.2 Proof of Theorem 4

Theorem 4. *The d-FDST problem parameterized by the solution size is* W[1]*-hard on directed acyclic graphs and strongly connected digraphs. Also the d-RDST problem is* NP*-hard on the class of strongly connected digraphs.*

Proof. We first show that k-INDEPENDENT SET, which is known to be W[1]-complete [9], fixed-parameter reduces to the d-FDST problem on directed acyclic graphs. Let (G, k) be an instance of the k-INDEPENDENT SET problem where G is a connected undirected graph on n vertices and m edges. Construct a directed graph D as follows. The vertex set $V(D)$ consists of $m + n + 4$ vertices: $v_1, \ldots, v_n, e_1, \ldots, e_m, a, b, x, y$, where the v_i's and the e_i's "correspond" to the vertices and edges, respectively, of G and a, b, x, y are four special vertices. The digraph D can viewed as a four-layer graph. Layer one consists of vertex a. Layer two consists of the vertices $x, v_1, \ldots, v_n, y$ and the vertex a has an out-arc to each vertex in layer two. Layer three consists of the vertices $e_1, \ldots, e_m$, and layer four consists of the vertex b which has an out-arc to each vertex in layer three and to vertex x in layer two. Each vertex in layer three has an out-arc to the vertex x. If $e = \{u, v\} \in E(G)$ then the vertices u and v in layer two have an out-arc each to the vertex e in layer three. Finally y has an out-arc to vertex b. This completes the description of D. It is easy to verify that D is a DAG. See Figure 2.

Observe the following:

1. Vertex a is the only source of D and hence must be the root of any out-branching and that if a preserves its out-degree in an out-branching then no vertex from layers three and four can preserve their out-degree.
2. Vertices x and y preserve their out-degree in any out-branching. This follows because x is a sink and y's out-neighbor has in-degree exactly one.
3. At most one vertex from layer three can preserve its out-degree in any out-branching because each of them has an arc to x.

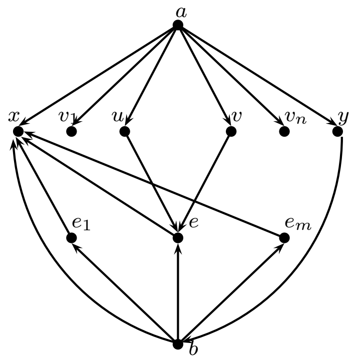

Fig. 2. The digraph D.

4. If b preserves its out-degree, then no vertex from layer three and none of the vertices $v_1, \dots, v_n$ from layer two preserve their out-degree.

Claim. The graph G has an independent set of size k if and only if the digraph D has an out-branching with $k+3$ vertices of full degree.

Suppose that G has an independent set of size k on the vertices $v_{i_1}, \dots, v_{i_k}$. Then it is easy to see that D has an out-branching in which the vertices $a, x, y, v_{i_1}, \dots, v_{i_k}$ preserve their out-degree. Conversely suppose that D admits an out-branching in which at least $k+3$ vertices preserve their out-degree. We consider two cases.

Case 1. Vertex a preserves its out-degree. Then no vertex from layers three and four can preserve its out-degree, as each of these vertices has an out-arc to x. Since x and y are the other two vertices of full degree, it must be that k vertices from among the $v_1, \dots, v_n$ preserve their out-degree. These vertices k vertices form an independent set in G.

Case 2. Vertex a does not preserve its out-degree. If b preserves its out-degree then by Observation 4 above only three vertices preserve their out-degree, namely x, y, b, a contradiction. Therefore assume that b is of reduced degree. By Observation 3, at most one vertex from layer three can preserve its out-degree. Hence at least k vertices from among the $v_1, \dots, v_n$ preserve their out-degree. These vertices from a k-independent set in G.

By modifying the above reduction from k-Independent Set, one can show that d-FDST is W[1]-hard on the class of strongly connected digraphs (and hence that the d-RDST problem is NP-hard on this class of digraphs). Given an instance (G, k) of k-Independent Set, construct the digraph D as in the above proof with just one modification: add the arc (x, a). The resulting digraph is strongly connected. We may assume that G is connected and non-bipartite. One can then show that G has an independent set of size k if and only if D admits an out-branching with $k+3$ vertices of full degree. Suppose G has a k-independent set on the vertex set $\{v_1, \dots, v_k\}$. Since G is non-bipartite, there exists an edge e_j both of whose endpoints are in $V(G) \setminus \{v_1, \dots, v_k\}$. It is easy to see that there is an out-branching with e_j as root in which the vertices $e_j, x, y, v_1, \dots, v_k$ are of full degree. Conversely suppose that D has an out-branching with $k+3$ vertices of full degree. Then, as before, vertex b must of reduced degree. Since y preserves its degree, there are at least $k+2$ vertices from among $a, x, v_1, \dots, v_n, e_1, \dots, e_m$ that preserve their out-degrees. Between a and x, at most one can preserve its degree and among $a, e_1, \dots, e_m$ at most one can preserve its out-degree. Therefore there must at least k vertices from among $v_1, \ldots, v_n$ that preserve their out-degree. These vertices form an independent set in G.

This completes the proof of the theorem. □