Noname manuscript No.
(will be inserted by the editor)

Digraph Coloring and Distance to Acyclicity

Ararat Harutyunyan · Michael Lampis · Nikolaos Melissinos

the date of receipt and acceptance should be inserted later

Abstract In k-Digraph Coloring we are given a digraph and are asked to partition its vertices into at most k sets, so that each set induces a DAG. This well-known problem is NP-hard, as it generalizes (undirected) k-Coloring, but becomes trivial if the input digraph is acyclic. This poses the natural parameterized complexity question of what happens when the input is "almost" acyclic. In this paper we study this question using parameters that measure the input's distance to acyclicity in either the directed or the undirected sense.

In the directed sense perhaps the most natural notion of distance to acyclicity is directed feedback vertex set (DFVS). It is already known that, for all $k \geq 2$, k-Digraph Coloring is NP-hard on digraphs of DFVS at most $k+4$. We strengthen this result to show that, for all $k \geq 2$, k-Digraph Coloring is already NP-hard for DFVS exactly k. This immediately provides a dichotomy, as k-Digraph Coloring is trivial if DFVS is at most $k-1$. Refining our reduction we obtain three further consequences: (i) 2-Digraph Coloring is NP-hard for oriented graphs of feedback vertex set (FVS) at most 3; (ii) for all $k \geq 2$, k-Digraph Coloring is NP-hard for graphs of feedback *arc* set (FAS) at most k^2; interestingly, this leads to a second dichotomy, as we show that the problem is FPT by k if FAS is at most k^2-1; (iii) k-Digraph Coloring is NP-hard for graphs of DFVS k, even if the maximum degree Δ is at most $4k-1$; we show that this is also *almost* tight, as the problem becomes FPT for DFVS k and $\Delta \leq 4k-3$.

Since these results imply that the problem is also NP-hard on graphs of bounded directed treewidth, we then consider parameters that measure the distance from acyclicity of the underlying graph. On the positive side, we show that k-Digraph Coloring admits an FPT algorithm parameterized by treewidth, whose parameter dependence is $(\text{tw}!)k^{\text{tw}}$. Since this is considerably worse than the k^{tw} dependence of (undirected) k-Coloring, we pose the question of whether the tw! factor can be eliminated. Our main contribution in this part is to settle this question in the neg-

Michael Lampis was partially supported by a grant from the French National Research Agency under the JCJC program (ASSK: ANR-18-CE40-0025-01).

Université Paris-Dauphine, PSL Research University, CNRS, UMR 7243, LAMSADE, Paris, France E-mail: ararat.harutyunyan@dauphine.fr, michail.lampis@dauphine.fr, nikolaos.melissinos@dauphine.eu

ative and show that our algorithm is essentially optimal, even for the much more restricted parameter treedepth and for $k = 2$. Specifically, we show that an FPT algorithm solving 2-Digraph Coloring with dependence $\mathrm{td}^{o(\mathrm{td})}$ would contradict the ETH.

In the end, we consider the class of tournaments. It is known that deciding whether a tournament is 2-colorable is NP-complete. We present an algorithm that decides if we can 2-color a tournament in $O^*(\sqrt[3]{6}^n)$ time.

Keywords Digraph Coloring · Dichromatic number · NP-completeness · Parameterized complexity · Feedback vertex and arc sets

CR Subject Classification Mathematics of computing→Graph algorithms · Theory of Computation → Design and Analysis of Algorithms → Parameterized Complexity and Exact Algorithms

1 Introduction

In Digraph Coloring, we are given a digraph D and are asked to calculate the smallest k such that the vertices of D can be partitioned into k *acyclic* sets. In other words, the objective of this problem is to color the vertices with the minimum number of colors so that no directed cycle is monochromatic. This notion is called the *dichromatic number* and it was introduced by V. Neumann-Lara [37]. More recently, digraph coloring has received much attention, in part because it turns out that many results about the chromatic number of undirected graphs quite naturally carry over to the dichromatic number of digraphs [1, 2, 4, 7, 11, 20–24, 32, 34, 35, 38]. We note that Digraph Coloring generalizes Coloring (if we simply replace all edges of a graph by pairs of anti-parallel arcs) and is therefore NP-complete.

In this paper we are interested in the computational complexity of Digraph Coloring from the point of view of structural parameterized complexity[1]. Our main motivation for studying this is that (undirected) Coloring is a problem of central importance in this area whose complexity is well-understood, and it is natural to hope that some of the known tractability results may carry over to digraphs – especially because, as we mentioned, Digraph Coloring seems to behave as a very close counterpart to Coloring in many respects. In particular, for undirected graphs, the complexity of Coloring for "almost-acyclic" graphs is very precisely known: for all $k \geq 3$ there is a $O^*(k^{\mathrm{tw}})$ algorithm, where tw is the input graph's treewidth, and this is optimal (under the SETH) even if we replace treewidth by much more restrictive parameters [27, 33]. Can we achieve the same amount of precision for Digraph Coloring?

Our results: The main question motivating this paper is therefore the following: Does Digraph Coloring also become tractable for "almost-acyclic" inputs? We attack this question from two directions.

First, in Section 3, we consider the notion of acyclicity in the digraph sense and study cases where the input digraph is close to being a DAG. Possibly the most

[1] In the remainder, we assume the reader is familiar with the basics of parameterized complexity theory, such as the class FPT, as given in standard textbooks [12].

natural such measure is directed feedback vertex set (DFVS), which is the minimum number of vertices whose removal destroys all directed cycles. The problem is paraNP-hard for this parameter, as for all fixed $k \geq 2$, k-Digraph Coloring is already known to be NP-hard, for inputs of DFVS at most $k+4$ [34]. Our first contribution is to tighten this result by showing that actually k-Digraph Coloring is already NP-hard for DFVS of size *exactly* k. This closes the gap left by the reduction of [34] and provides a complete dichotomy, as the problem is trivially FPT by k when the DFVS has size strictly smaller than k (the only non-trivial part of the problem in this case is to find the DFVS [10]). In the end of this section we consider 2-Digraph Coloring on oriented graphs. We prove that it is NP-hard to decide if an oriented graph is 2-colorable even in cases where the size of DFVS is 3. This is tight as there exists an easy argument showing that all oriented graphs with DFVS k are k-colorable.

In Section 4 we investigate if by considering a more restricted notion of near-acyclicity, or by imposing further restrictions, such as bounding the maximum degree of the graph, could lead to an FPT algorithm. Unfortunately, we show that neither of these suffices to make the problem tractable. In particular, by refining our reduction we obtain the following: First, we show that for all $k \geq 2$, k-Digraph Coloring is NP-hard for digraphs of feedback *arc* set (FAS) k^2, that is, digraphs where there exists a set of k^2 arcs whose removal destroys all cycles (feedback arc set is of course a more restrictive parameter than feedback vertex set). Interestingly, this also leads us to a complete dichotomy, this time for the parameter FAS: we show that k-coloring becomes FPT (by k) on graphs of FAS at most k^2-1, by an argument that reduces this problem to coloring a subdigraph with at most $O(k^2)$ vertices, and hence the correct complexity threshold for this parameter is k^2. Second, we show that k-coloring a digraph with DFVS k remains NP-hard even if the maximum degree is at most $4k-1$. This further strengthens the reduction of [34], which showed that the problem is NP-hard for bounded *degeneracy* (rather than degree). Almost completing the picture, we show that k-coloring a digraph with DFVS k and maximum degree at most $4k-3$ is FPT by k, leaving open only the case where the DFVS is exactly k and the maximum degree exactly $4k-2$.

In Section 5, because of the negative results for DFVS and FAS, we deiced to consider as parameter the treewidth of the underlying graph. It turns out that, finally, this suffices to lead to an FPT algorithm, obtained with standard DP techniques. However, our algorithm has a somewhat disappointing running time of $(\mathrm{tw}!)k^{\mathrm{tw}}n^{O(1)}$, which is significantly worse than the $k^{\mathrm{tw}}n^{O(1)}$ complexity which is known to be optimal for undirected Coloring, especially for small values of k. This raises the question of whether the extra (tw!) factor can be removed. Our main contribution in this part is to show that this is likely impossible, even for a more restricted case. Specifically, we show that if the ETH is true, no algorithm can solve 2-Digraph Coloring in time $td^{o(\mathrm{td})}n^{O(1)}$, where td is the input graph's treedepth, a parameter more restrictive than treewidth (and pathwidth). As a result, this paper makes a counterpoint to the line of research that seeks to find ways in which dichromatic number replicates the behavior of chromatic number in the realm of digraphs by pinpointing one important aspect where the two notions are quite different, namely their complexity with respect to treewidth.

Finally, in Section 6, we consider tournaments. It is already known that 2-Digraph Coloring is NP-hard for tournaments [11]. The exhaustive algorithm to check if a tournament is 2-colorable takes $O^*(2^n)$ time as there exists 2^n possible 2-colorings for a graph. We improve this running time by proposing an algorithm that answers the same question in $O^*(\sqrt[3]{6}^n)$.

Other related work: Structural parameterizations of Digraph Coloring have been studied in [38], who showed that the problem is FPT by modular width generalizing the algorithms of [18,29]; and [20] who showed that the problem is in XP by clique-width (note that hardness results for Coloring rule out an FPT algorithm in this case [16,17,30]). Our results on the hardness of the problem for bounded DFVS and FAS build upon the work of [34]. The fact that the problem is hard for bounded DFVS implies that it is also hard for most versions of directed treewidth, including DAG-width, Kelly-width, and directed pathwidth [6,19,25, 28,31]. Indeed, hardness for FAS implies also hardness for bounded elimination width, a more recently introduced restriction of directed treewidth [15]. For undirected treewidth, a problem with similar behavior is DFVS: (undirected) FVS is solvable in $O^*(3^{\mathrm{tw}})$ [13] but DFVS cannot be solved in time $\mathrm{tw}^{o(\mathrm{tw})}n^{O(1)}$, and this is tight under the ETH [8]. For other natural problems whose complexity by treewidth is $tw^{\Theta(\mathrm{tw})}$ see [3,5,9]

With respect to maximum degree, it is not hard to see that k-Digraph Coloring is NP-hard for graphs of maximum degree $2k+2$, because k-Coloring is NP-hard for graphs of maximum degree $k+1$, for all $k \geq 3$ [2]. On the converse side, using a generalization of Brooks' theorem due to Mohar [36] one can see that k-Digraph Coloring digraphs of maximum degree $2k$ is in P. This leaves as the only open case digraphs of degree $2k+1$, which in a sense mirrors our results for digraphs of DFVS k and degree $4k-2$. We note that the NP-hardness of 2-Digraph Coloring for bounded degree graphs is known even for graphs of large girth, but the degree bound follows the imposed bound on the girth [14].

2 Definitions, Notation and Preliminaries

We use standard graph-theoretic notation. All digraphs are loopless and have no parallel arcs; two oppositely oriented arcs between the same pair of vertices, however, are allowed and are called a *digon*. Oriented graphs are digraphs which do not contain any digons. The in-degree (respectively, out-degree) of a vertex is the number of arcs coming into (respectively going out of) a vertex. The degree of a vertex is the sum of its in-degree and out-degree. For a set of arcs F, $V(F)$ denotes the set of their endpoints. For a set of vertices S of a digraph D, $D[S]$ denotes the digraph induced by S and $N[S]$ denotes the closed neighborhood of S, that is, S and all vertices that have an arc to or from S.

The *chromatic number* of a graph G is the minimum number of colors k needed to color the vertices of G such that each color class is an independent set. We say that a digraph $D=(V,E)$ is *k-colorable* if we can color the vertices of D with k colors such that each color class induces an acyclic subdigraph (such a coloring is called a *proper k-coloring*). The *dichromatic number*, denoted by $\chi(D)$, is the

[2] Note that this argument does not prove that 2-Digraph Coloring is NP-hard for maximum degree 6, but this is not too hard to show. We give a proof in Theorem 1 for the sake of completeness.

minimum number k for which D is k-colorable. The maximum degree of a graph or digraph is denoted with Δ.

A subset of vertices $S \subset V$ of D is called a *feedback vertex set* if $D - S$ is acyclic.

Remark 1 Every digraph $D = (V, E)$ with feedback vertex set of size at most $k - 1$ is k-colorable.

The remark holds because we can use distinct colors for the vertices of the feedback vertex set and the remaining color for the rest of the graph.

A subset of arcs $A \subset E$ of D is called a *feedback arc set* if $D - A$ is acyclic. For the definition of treewidth and nice tree decompositions we refer the reader to [12]. A graph G has treedepth at most k if one of the following holds: (i) G has at most k vertices (ii) G is disconnected and all its components have treedepth at most k (iii) there exists $u \in V(G)$ such that $G - u$ has treedepth at most $k - 1$. We use $\mathrm{tw}(G), \mathrm{td}(G)$ to denote the treewidth and treedepth of a graph. It is known that $\mathrm{tw}(G) \leq \mathrm{td}(G)$ for all graphs G.

The Exponential Time Hypothesis (ETH) [26] states that there is a constant $c > 1$ such that no algorithm which decides if 3-SAT formulas with n variables and m clauses are satisfiable can run in time c^{n+m}. In this paper we will use the simpler (and slightly weaker) version of the ETH which simply states that 3-SAT cannot be solved in time $2^{o(n+m)}$.

Throughout the paper, when n is a positive integer we use $[n]$ to denote the set $\{1, \dots, n\}$. For a set V an ordering of V is an injective function $\sigma : V \to [|V|]$. It is a well-known fact that a digraph D is acyclic if and only if there exists an ordering σ of $V(D)$ such that for all arcs uv we have $\sigma(u) < \sigma(v)$. This is called a topological ordering of D.

We conclude this section with a preliminary theorem. As we mentioned, the argument from (undirected) graph coloring that shows why k-Digraph Coloring is NP-hard for digraphs of $\Delta = 2k + 2$ does not hold for $k = 2$. Our first theorem gives a proof for this case.

Theorem 1 *It is NP-hard to decide if a given digraph with maximum degree* 6 *is* 2*-colorable.*

Proof We perform a reduction from NAE-3-SAT, a variant of 3-SAT where we are asked to find an assignment that sets at least one literal to True and one to False in each clause. First we remark that this problem remains NP-hard if all literals appear at most twice.

To see this, suppose that x appears $\ell \geq 4$ times in ϕ. We replace each appearance of x with a fresh variable x_i, $i \in [\ell]$ and add to the formula the clauses $(\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \dots (\neg x_\ell \vee x_1)$. Repeating this for all variables that appear at least 4 times produces an equivalent instance ϕ' with $O(n + m)$ variables and clauses such that all literals appear at most 2 times. Furthermore, any satisfying assignment the formula forces exactly one true and one false literal in the new clauses.

We construct a digraph as follows: for each variable x_i we make a digon and label its vertices $x_i, \neg x_i$. We call this part of the digraph the assignment part. For each clause we make a directed cycle of size equal to the clause and associate each vertex of the cycle with a literal. We call this part the satisfaction part. Finally, for each vertex of the assignment part we connect it with digons with each vertex of the satisfaction part that represents the opposite literal.

The digraph we constructed has maximum degree 6; indeed, each literal has degree two in the assignment part and since each literal appears in at most two clauses, it has degree at most 4 in the satisfaction part. If there is a satisfying assignment then we give color 1 to all True literals of both parts and color 2 to everything else. Observe that all arcs connecting the two parts are bichromatic and if the assignment is satisfying all directed cycles are also bichromatic. For the converse direction, if there is a 2-coloring we can extract an assignment by setting to True all literals which have color 1 in the assignment part. Note that this implies that in the satisfaction part all literals which have color 1 have been set to True and all literals which have color 2 have been set to False, because of the digons connecting the two parts. But this implies that our assignment is satisfying because all cycles are bichromatic.

3 Bounded Feedback Vertex Set

In this section we study the complexity of the problem parameterized by the size of the feedback vertex set of a digraph. Throughout we will assume that a feedback vertex set is given to us; if not we can use known FPT algorithms to find the smallest such set [10].

As we are mentioned already, a digraph of DFVS $k-1$ can be always colored with k colors. Our main result in this section is that k-Digraph Coloring is NP-hard for digraphs of DFVS k. Observe that Remark 1 indicates that this result will be best possible.

Remark 2 Let $D=(V,E)$ be a digraph with feedback vertex set F of size $|F|=k$. If F does not induce a bi-directed clique, then D is k-colorable.

Indeed, if $u,v \in F$ are not connected by a digon we can use one color for $\{u,v\}$, $k-2$ distinct colors for the rest of F, and the remaining color for the rest of the graph. Remark 2 will also be useful later in designing an algorithm, but at this point it is interesting because it tells us that, since the graphs we construct in our reduction have DFVS k and must in some cases have $\boldsymbol{\chi}(D) > k$, our reduction needs to construct a bi-directed clique of size k.

Before we go on to our reduction let us also mention that we will reduce from a restricted version of 3-SAT with the following properties: (i) all clauses must have either only positive literals or only negative literals (ii) all variables appear at most 2 times positive and 1 time negative. We call this Restricted-3-SAT.

Lemma 1 Restricted-3-SAT *is NP-hard and cannot be solved in $2^{o(n+m)}$ time unless the ETH is false.*

Proof Start with an arbitrary instance ϕ of 3-SAT with n variables and m clauses. We first make sure that every variable appears at most 3 times as follows. First use the trick of Lemma 1 to decrease the number of appearances of each literal to two. We now edit ϕ' as follows: for each variable x of ϕ' we replace every occurence of $\neg x$ with a fresh variable x'. We then add the clause $(\neg x \vee \neg x')$. This gives a new equivalent instance ϕ'' which also has $O(n+m)$ variables and clauses and satisfies all properties of Restricted-3-SAT.

Theorem 2 *For all $k \geq 2$, it is NP-hard to decide if a digraph $D = (V, E)$ is k-colorable even when the size of its feedback vertex set is k. Furthermore, this problem cannot be solved in time $2^{o(n)}$ unless the ETH is false.*

Proof We will prove the theorem for $k = 2$. To obtain the proof for larger values one can add to the construction $k-2$ vertices which are connected to everything with digons: this increases both the dichromatic number and the feedback vertex set by $k-2$. Note that this does indeed construct a "palette" clique of size k, as indicated by Remark 2.

We make a reduction from Restricted-3-SAT, which is NP-hard by Lemma 1. Our reduction will produce an instance of size linear in the input formula, which leads to the ETH-based lower bound. Let ϕ be the given formula with variables $x_1, \ldots, x_n$, and suppose that clauses $c_1, \ldots, c_\ell$ contain only positive literals, while clauses $c_{\ell+1}, \ldots, c_m$ contain only negative literals. We will assume without loss of generality that all variables appear in ϕ both positive and negative (otherwise ϕ can be simplified).

We begin by constructing two "palette" vertices v_1, v_2 which are connected by a digon. Then, for each clause $c_i, i \in [m]$ we do the following: if the clause has size three we construct a directed path with vertices $l_{i,1}, w_{i,1}, l_{i,2}, w_{i,2}, l_{i,3}$, where the vertices $l_{i,1}, l_{i,2}, l_{i,3}$ represent the literals of the clause; if the clause has size two we similarly construct a directed path with vertices $l_{i,1}, w_{i,1}, l_{i,2}$, where again $l_{i,1}, l_{i,2}$ represent the literals of the clause.

For each variable $x_j, j \in [n]$ we do the following: for each clause c_{i_1} where x_j appears positive and clause c_{i_2} where x_j appears negative we construct a vertex w'_{j,i_1,i_2} and add an incoming arc from the vertex that represents the literal x_j in the directed path of c_{i_1} to w'_{j,i_1,i_2}; and an outgoing arc from w'_{j,i_1,i_2} to the vertex that represents the literal $\neg x_j$ in the directed path of c_{i_2}.

Finally, to complete the construction we connect the palette vertices to the rest of the graph as follows: v_1 is connected with a digon to all existing vertices $w_{i,j}$, $i \in [m], j \in [2]$; v_2 is connected with a digon to all existing vertices w'_{j,i_1,i_2}; v_2 has an outgoing arc to the first vertex of each directed path representing a clause and an incoming arc from the last vertex of each such path; v_1 has an outgoing arc to all vertices that represent positive literals and an incoming arc from all vertices representing negative literals. (See Figure 1)

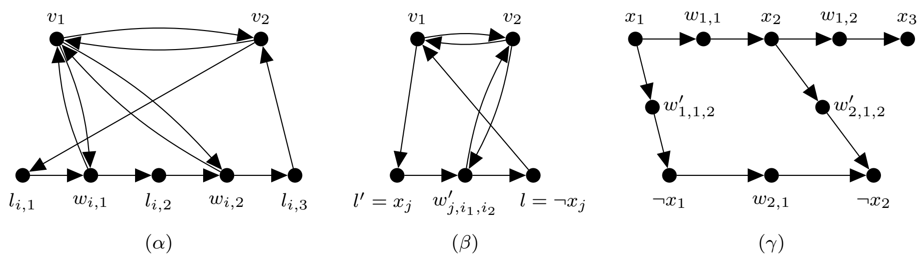

Fig. 1 (α): The cycles created by $\{v_1, v_2\}$ and clauses with three literals. (β): The cycles created by $\{v_1, v_2\}$ and each pair $\{x, \neg x\}$. (γ): An example digraph for the formula $\phi = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2)$, without showing v_1, v_2.

Let us now prove that this reduction implies the theorem. First, we claim that in the digraph we constructed $\{v_1, v_2\}$ is a feedback vertex set. Indeed, suppose we remove these two vertices. Now every arc in the remaining graph either connects vertices that represent the same clause, or is incident on a vertex w'_{j,i_1,i_2}. Observe that these vertices have only one incoming and one outgoing arc and because of the ordering of the clauses $i_1 < i_2$ (since clauses that contain negative literals come later in the numbering). We conclude that every directed path must either stay inside the path representing the same clause or lead to a path the represents a later clause. Hence, the digraph is acyclic.

Let us now argue that if ϕ is satisfiable then the digraph is 2-colorable. We give color 1 to v_1 and 2 to v_2. We give color 2 to each $w_{i,j}$ and color 1 to each w'_{j,i_1,i_2}. Fix a satisfying assignment for ϕ. We give color 1 to all vertices $l_{i,j}$ that represent literals set to True by the assignment and color 2 to all remaining vertices. Let us see why this coloring is acyclic. First, consider a vertex w'_{j,i_1,i_2}. This vertex has color 1 and one incoming and one outgoing arc corresponding to opposite literals. Because the literals are opposite, one of them has color 2, hence w'_{j,i_1,i_2} cannot be in any monochromatic cycle and can be removed. Now, suppose there is a monochromatic cycle of color 1. As $\{v_1, v_2\}$ is a feedback vertex set, this cycle must include v_1. Since v_2 and all $w_{i,j}$ have color 2 the vertex after v_1 in the cycle must be some $l_{i,j}$ representing a positive literal which was set to True by our assignment. The only outgoing arc leaving from $l_{i,j}$ and going to a vertex of color 1 must lead it to a vertex $w'_{j',i,i'}$, which as we said cannot be part of any cycle. Hence, no monochromatic cycle of color 1 exists. Consider then a monochromatic cycle of color 2, which must begin from v_2. The next vertex on this cycle must be a $l_{i,1}$ and since we have eliminated vertices w'_{j,i_1,i_2} the cycle must continue in the directed path of clause i. But, since we started with a satisfying assignment, at least one of the literal vertices of this path has color 1, meaning the cycle cannot be monochromatic.

Finally, let us argue that if the digraph is 2-colorable, then ϕ is satisfiable. Consider a 2-coloring which, without loss of generality, assigns 1 to v_1 and 2 to v_2. The coloring must give color 2 to all $w_{i,j}$ and color 1 to all w_{j,i_1,i_2}, because of the digons connecting these vertices to the palette. Now, we obtain an assignment for ϕ as follows: for each x_j, we find the vertex in our graph that represents the literal $\neg x_j$ (this is unique since each variable appears exactly once negatively): we assign x_j to True if and only if this vertex has color 2. Let us argue that this assignment satisfies all clauses. First, consider a clause with all negative literals. If this clause is not satisfied, then all the vertices representing its literals have color 2. Because vertices $w_{i,j}$ also all have color 2, this creates a monochromatic cycle with v_2, contradiction. Hence, all such clauses are satisfied. Second, consider a clause c_i with all positive literals. In the directed path representing c_i at least one literal vertex must have color 1, otherwise we would get a monochromatic cycle with v_2. Suppose this vertex represents the literal x_j and has an out-neighbor w'_{j,i,i_2}, which is colored 1. If the out-neighbor of w'_{j,i_1,i_2} is also colored 1, we get a monochromatic cycle with v_1. Therefore, that vertex, which represents the literal $\neg x_j$ has color 2. But then, according to our assignment x_j is True and c_i is satisfied.

The last result of this section concerns 2-coloring of oriented graphs.

Theorem 3 *It is NP-hard to decide if an oriented graph $D = (V, E)$ is 2-colorable even when the size of its feedback vertex set is 3.*

Proof We adapt the proof of Theorem 2. First let us give an intuition behind the gadget we are going to use. In the proof of Theorem 2 the digraph we created is not an oriented graph as it contains digons. All the digons of that digraph are connected to vertices v_1 or v_2, and therefore, we want to replace v_1 and v_2 with a gadget that contains two arcs t_1t_2 and f_1f_2 such that the vertices t_1 and t_2 to have the same color as v_1 and the vertices f_1 and f_2 to have the same color as v_2. Then we can replace all cycles that contained v_1 (respectively, v_2) with cycles that contain the arc t_1t_2 (respectively, f_1f_2) and the rest of the proof will remain the same.

The gadget we use in place of $\{v_1, v_2\}$ is the one in the Fig. 2. Furthermore, we will not use the digon between v_1 and v_2 and we replace all the other incoming arcs of v_1 from the previous construction with incoming arcs to t_1, the outgoing arcs of v_1 with outgoing arcs from t_2, the incoming arcs of v_2 with incoming arcs to f_1, the outgoing arcs of v_2 with outgoing arcs from f_2. For example, the digon $v_1w_{i,1}$ in the gadget (α) from the previous theorem becomes a triangle $t_1t_2w_{i,1}$.

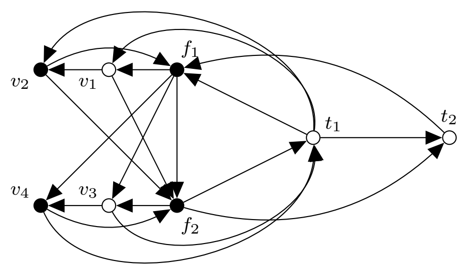

Fig. 2 Gadget H: It is 2-colorable, and in any 2-coloring both pairs $\{f_1, f_2\}$ and $\{t_1, t_2\}$ must be monochromatic but with different color per pair.

Now we need to show that in any proper 2-coloring of this gadget both pairs f_1, f_2 and t_1, t_2 are monochromatic and we use different color per pair.

First observe that there exists such a coloring (see Fig. 2) We will show that the vertices f_1 and t_1 cannot have the same color. Assume that they are both colored 0; then the vertex f_2 must be colored 1 because we have the cycle f_1, f_2, t_1. Because the vertex f_2 is colored 1 and there exists the cycle f_2, v_3, v_4 we know that at least one of v_3, v_4 must be colored 0. Let v_3 (respectively, v_4) be colored 0, then the coloring is not proper because there exists the cycle t_1, f_1, v_3 (resp. t_1, f_1, v_4) with all the vertices colored 0. This is a contradiction so the f_1 and t_1 cannot have the same color. Similarly we can prove that f_2 and t_1 cannot have the same color. So we must color the vertices f_1 and f_2 with one color and t_1 with the second. Furthermore because we have the cycle f_1, f_2, t_2, the vertex t_2 must use the same color as t_1.

It remains to show that the size of minimum feedback vertex set is at most 3; observe that the set $\{f_1, f_2, t_1\}$ is a feedback vertex set (see Fig. 3).

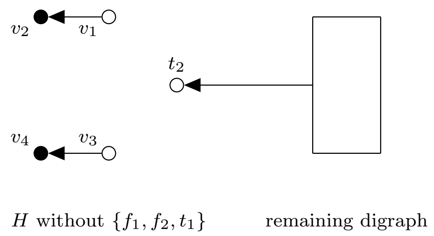

Fig. 3 For the remaining digraph, it has been proved that is acyclic in the previous theorem so $\{f_1, f_2, t_1\}$ is a feedback vertex set of the whole digraph.

This result is tight as, by Remark 2, we know that oriented graphs with DFVS of size k are k-colorable.

4 Bounded Feedback Arc Set and Bounded Degree

In this section we first present two algorithmic results: we show that k-Digraph Coloring becomes FPT (by k) if either the input graph has feedback vertex set k and maximum degree at most $4k-3$; or if it has feedback arc set at most k^2-1 (and unbounded degree). Interestingly, the latter of these results is exactly tight and the former is almost tight: in the second part we refine the reduction of the previous section to show that k-Digraph Coloring is NP-hard for digraphs which have simlutaneously a FAS of size k^2, a feedback vertex set of size k and maximum degree $\Delta = 4k-1$.

4.1 Algorithmic Results

Our first result shows that for k-Digraph Coloring, if we are promised a feedback vertex set of size k (which is the smallest value for which the problem is non-trivial), then the problem remains tractable for degree up to $4k-3$. Observe that in the case of general digraphs (where we do not bound the feedback vertex set) the problem is already hard for maximum degree $2k+2$ (see Other Related Work section), so this seems encouraging. However, we show in Theorem 6 that this tractability cannot be extended much further.

Theorem 4 *Let $D = (V, E)$ be a digraph with feedback vertex set F of size $|F| = k$ and maximum degree $\Delta \le 4k-3$. Then, D is k-colorable if and only if $D[N[F]]$ is k-colorable. Furthermore, a k-coloring of $D[N[F]]$ can be extended to a k-coloring of D in polynomial time.*

Proof Let $D = (V, E)$ be such a digraph. If $D[N[F]]$ is not k-colorable, then D is not k-colorable, so we need to prove that if $D[N[F]]$ is k-colorable then D is k-colorable and we can extend this coloring to D. Assume that $D[N[F]]$ is k-colorable. By Remark 2 we can assume that $D[F]$ is a bi-directed clique. Let $c : N[F] \to [k]$ be the assumed k-coloring and without loss of generality say that $F = \{v_1, \dots, v_k\}$ and $c(v_i) = i$ for all $i \in [k]$.

Before we continue let us define the following sets of vertices: we will call $V_{i,in}$ the set of vertices $v \in N[F] \setminus F$ such that $c(v) = i$ and there exists an arc $vv_i \in E$. Similarly we will call $V_{i,out}$ the set of vertices $v \in N[F] \setminus F$ where $c(v) = i$ and there exists an arc $v_i v \in E$. The sets $V_{i,in}$ and $V_{i,out}$ are disjoint in any proper coloring (otherwise we would have a monochromatic digon). Furthermore, $V_{i,in} \cup V_{i,out}$ is disjoint from $V_{j,in} \cup V_{j,out}$ for $j \neq i$ (because their vertices have different colors), so all these $2k$ sets are pair-wise disjoint. We first show that if one of these $2k$ sets is empty, then we can color D.

Claim If for some $i \in [k]$ one of the sets $V_{i,in}$, $V_{i,out}$ is empty then we can extend c to a k-coloring of D in polynomial time.

Proof We keep c unchanged and color all of $V(D) \setminus N[F]$ with color i. This is a proper k-coloring. Indeed, this cannot create a monochromatic cycle with color $j \neq i$. Furthermore, if a monochromatic cycle of color i exists, since this cycle must intersect F, we conclude that it must contain v_i. However, in the current k-coloring v_i either has in-degree or out-degree 0 in the vertices colored i, so no monochromatic cycle can go through it.

In the remainder we assume that all sets $V_{i,in}, V_{i,out}$ are non-empty. Our strategy will be to edit the k-coloring of $D[N[F]]$ so that we retain a proper k-coloring, but one of these $2k$ sets becomes empty. We will then invoke Claim 4.1 to complete the proof.

We now define, for each pair $i, j \in [k]$ with $i < j$ the set $E_{i,j}$ which contains all arcs with one endpoint in $\{v_i, v_j\}$ and the other in $V_{i,in} \cup V_{i,out} \cup V_{j,in} \cup V_{j,out}$ and whose endpoints have *distinct* colors. We call $E_{i,j}$ the set of *cross* arcs for the pair (i, j). We will now argue that for some pair (i, j) we must have $|E_{i,j}| \leq 3$. For the sake of contradiction, assume that $|E_{i,j}| \geq 4$ for all pairs. Then, by summing up the degrees of vertices of F we have:

$$\sum_{i \in [k]} d(v_i) \geq 2k + k(2k-2) + \sum_{i,j \in [k], i<j} |E_{i,j}| \geq 2k^2 + 4\binom{k}{2} = 4k^2 - 2k$$

In the first inequality we used the fact that each $v_i \in F$ has at least two arcs connecting it to $V_{i,in} \cup V_{i,out}$ (since these sets are non-empty); $2k-2$ arcs connecting it to other vertices of F (since F is a clique); and each cross arc of a set $E_{i,j}$ contributes one to the degree of one vertex of F. From this calculation we infer that the average degree of F is at least $4k-2$, which is a contradiction, since we assumed that the digraph has maximum degre $4k-3$.

Fix now i, j such that $|E_{i,j}| \leq 3$. We will recolor $V_{i,in} \cup V_{i,out} \cup V_{j,in} \cup V_{j,out}$ in a way that allows us to invoke Claim 4.1. Since we do not change any other color, we will only need to prove that our recoloring does not create monochromatic cycles of colors i or j in $D[N[F]]$. We can assume that $|E_{i,j}| = 3$, since if $|E_{i,j}| < 3$ we can add an arbitrary missing cross arc and this can only make the recoloring process harder. Furthermore, without loss of generality, we assume that v_i has strictly more cross arcs of $E_{i,j}$ incident to it than v_j.

We now have to make a case analysis. First, suppose all three arcs of $E_{i,j}$ are incident on v_i. Then, there exists a set among $V_{j,in}, V_{j,out}$ that has at most one arc connecting it to v_i. We color this set i, and leave the other set colored j. We also color $V_{i,in} \cup V_{i,out}$ with j. This creates no monochromatic cycle because: (i) v_i now has at most one neighbor colored i in $V_{i,in} \cup V_{i,out} \cup V_{j,in} \cup V_{j,out}$, so no monochromatic cycle goes through v_i; (ii) v_j has either no out-neighbors or no in-neighbors colored j in $V_{i,in} \cup V_{i,out} \cup V_{j,in} \cup V_{j,out}$. With the new coloring we can invoke Claim 4.1. In the remainder we therefore assume that two arcs of $E_{i,j}$ are incident on v_i and one is incident on v_j.

Second, suppose that one of $V_{j,in}, V_{j,out}$ has no arcs connecting it to v_i. We color this set i and leave the other set colored j. Observe that one of $V_{i,in}, V_{i,out}$ has no arc connecting it to v_j. We color that set j and leave the other set colored i. In the new coloring both v_i, v_j either have no out-neighbor or no in-neighbor with the same color in $V_{i,in} \cup V_{i,out} \cup V_{j,in} \cup V_{j,out}$, so the coloring is proper and we can invoke Claim 4.1. In the remainder we assume that v_i has one arc connecting it to each of $V_{j,in}, V_{j,out}$.

Third, suppose that both arcs of $E_{i,j}$ incident on v_i have the same direction (into or out of v_i). We then color $V_{i,in} \cup V_{i,out}$ with j and $V_{j,in} \cup V_{j,out}$ with i. In the new coloring v_j has at most one neighbor with the same color and v_i has either only in-neighbors or only out-neighbors with color i, so the coloring is acyclic and we again invoke Claim 4.1.

Finally, we have the case where two arcs of $E_{i,j}$ are incident on v_i, they have different directions, one has its other endpoint in $V_{j,in}$ and the other in $V_{j,out}$. Observe that one of $V_{i,in}, V_{i,out}$ has no arc connecting it to v_j and suppose without loss of generality that it is $V_{i,in}$ (the other case is symmetric). We color $V_{i,in}$ with j and leave $V_{i,out}$ with color i. One of $V_{j,in}, V_{j,out}$ has an incoming arc from v_i; we color this set i and leave the other colored j. Now, v_i only has out-neighbors with color i, while v_j has at either only in-neighbors or only out-neighbors colored j, so we are done in this final case.

Our second result concerns a parameter more restricted than feedback vertex set, namely feedback arc set. Note that, in a sense, the class of graphs of bounded feedback arc set contains the class of graphs that have bounded feedback vertex set and bounded degree (selecting all incoming or outgoing arcs of each vertex of a feedback vertex set produces a feedback arc set), so the following theorem may seem more general. However, a closer look reveals that the following result is incomparable to Theorem 4, because graphs of feedback vertex set k and maximum degree $4k-3$ could have feedback arc set of size up to almost $2k^2$ (consider for example a bi-direction of the complete graph $K_{k,2k-2}$), while the following theorem is able to handle graphs of unbounded degree but feedback arc set up to (only) k^2-1. As we show in Theorem 6, this is tight.

Theorem 5 *Let D be a digraph with a feedback arc set F of size at most k^2-1. Then D is k-colorable if and only if $D[V(F)]$ is k-colorable, and such a coloring can be extended to D in polynomial time.*

Proof It is obvious that if $D[V(F)]$ is not k-colorable then D is not k-colorable. We will prove the converse by induction. For $k=1$ it is trivial to see that if $|F|=0$ then D is acyclic so is 1-colorable. Assume then that any digraph D with a feedback arc set F of size at most $(k-1)^2-1$ is $(k-1)$-colorable, if and only if $D[V(F)]$ is $(k-1)$-colorable.

Suppose now that we have D with a feedback arc set F with $|F| \leq k^2 - 1$ and we find that $D[V(F)]$ is k-colorable (this can be tested in $2^{O(k^2)}$ time). Let $c : V(F) \to [k]$ be a coloring of $V(F)$. We distinguish two cases:

Case 1. There exists a color class (say V_k) such that at least $2k-1$ arcs of F are incident on V_k. Then $D - V_k$ has a feedback arc set of size at most $|F| - (2k-1) \leq k^2 - 1 - (2k-1) \leq (k-1)^2 - 1$ and $V_1, \ldots, V_{k-1}$ remains a valid coloring of the remainder of $V(F)$. So by inductive hypothesis $D - V_k$ has a $(k-1)$-coloring. By using the color k on V_k we have a k-coloring for D.

Case 2. Each color class is incident on at most $2k-2$ arcs of F. We then claim that there is a way to color $V(F)$ so that all arcs of F have distinct colors on their endpoints. If we achieve this, we can trivially extend the coloring to the rest of the graph, as arcs of F become irrelevant. Now, let us call $v \in V(F)$ as *type one* if v is incident on at least k arcs of F. We will show that there is at most one *type one* vertex in each color class. Indeed, if $u, v \in V_i$ are both *type one*, then they are incident on at least $2k-1$ arcs of F (there is no digon between u and v because they share a color), which we assumed is not the case, as V_i is incident on at most $2k-2$ arcs of F. Therefore, we can use k distinct colors to color all the *type one* vertices of $V(F)$. Each remaining vertex of $V(F)$ is incident on at most $k-1$ arcs of F. We consider these vertices in some arbitrary order, and give each a color that does not already appear on the other endpoints of its incident arcs from F. Such a color always exists, and proceeding this way we color all arcs of F with distinct colors. This completes the proof.

4.2 Hardness

In this section we improve upon our previous reduction by producing a graph which has bounded degree and bounded feedback arc set. Our goal is to do this efficiently enough to (almost) match the algorithmic bounds given in the previous section.

Theorem 6 *For all $k \geq 2$, it is NP-hard to decide if a digraph $D = (V, E)$ is k-colorable, even if D has a feedback vertex set of size k, a feedback arc set of size k^2, and maximum degree $\Delta = 4k - 1$.*

Proof Recall that in the proof of Theorem 2 for $k \geq 2$ we construct a graph that is made up of two parts: the palette part, which is a bi-directed clique that contains v_1, v_2 and the $k-2$ vertices we have possibly added to increase the chromatic number (call them $v_3, \ldots, v_k$); and the part that represents the formula. We perform the same reduction except that we will now edit the graph to reduce its degree and its feedback arc set. In particular, we delete the palette vertices and replace them with a gadget that we describe below.

We construct a new palette that will be a bi-directed clique of size k, whose vertices are now labeled $v^i, i \in [k]$. Let M be the number of vertices of the graph we constructed for Theorem 2. We construct M "rows" of $2k$ vertices each. More precisely, for each $\ell \in [M], i \in [k]$ we construct the two vertices $v^i_{\ell,in}, v^i_{\ell,out}$. In the remainder, when we refer to row ℓ, we mean the set $\{v^i_{\ell,in}, v^i_{\ell,out} \mid i \in [k]\}$. For all $i, j \in [k], i < j$ we connect the vertices of row 1 to the vertices of the clique as shown in Figure 4. For all $i, j \in [k], i < j$ and $\ell \in [M-1]$ we connect the vertices of rows $\ell, \ell + 1$ as shown in Figure 5.

In more detail we have:

1. For each $i \in [k]$ the vertex v^i has an arc to all $v^j_{1,out}$ for $j \geq i$, an arc to $v^j_{1,in}$ for all $j \neq i$, and an arc from $v^j_{1,in}$ for all $j \leq i$.
2. For each $\ell \in [M]$, for all $i < j$ we have the following four arcs: $v^j_{\ell,out}v^i_{\ell,out}$, $v^i_{\ell,out}v^j_{\ell,in}$, $v^j_{\ell,in}v^i_{\ell,in}$, and $v^j_{\ell,out}v^i_{\ell,in}$. As a result, inside a row arcs are oriented from *out* to *in* vertices; and between vertices of the same type from larger to smaller indices i.
3. For each $\ell \in [M-1]$, for all $i \in [k]$ we have the arcs $v^i_{\ell,out}v^i_{\ell+1,out}$ and $v^i_{\ell+1,in}v^i_{\ell,in}$. As a result, the $v^i_{\ell,out}$ vertices form a directed path going out of v^i and the $v^i_{\ell,in}$ vertices form a directed path going into v^i.
4. For each $\ell \in [M-1]$, for all $i, j \in [k]$ with $i < j$ we have the arcs $v^i_{\ell,out}v^j_{\ell+1,in}$, $v^i_{\ell,out}v^j_{\ell+1,out}$, $v^i_{\ell+1,in}v^j_{\ell,in}$, $v^j_{\ell,out}v^i_{\ell+1,in}$. Again, arcs incident on an *out* and an *in* vertex are oriented towards the *in* vertex.

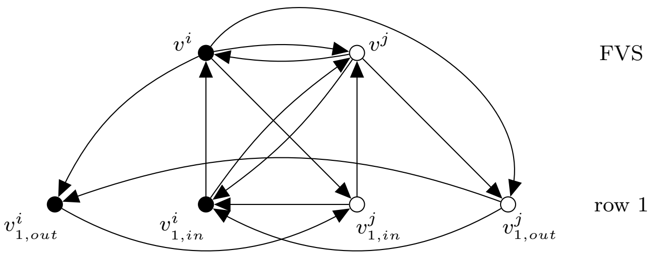

Fig. 4 Graph showing the connections between two vertices of the clique palette (v^i, v^j, where $i < j$) and the corresponding vertices of row 1.

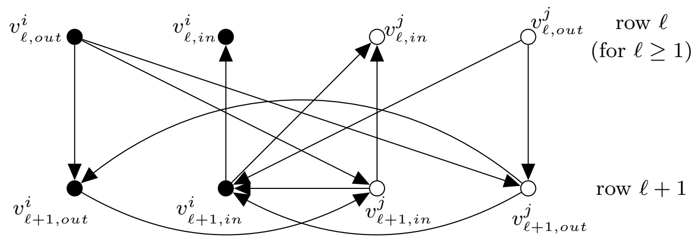

Fig. 5 Here we present the way we are connecting the vertices of the rows i and $i+1$

Let P be the gadget we have constructed so far, consisting of the clique of size k and the M rows of $2k$ vertices each. We will establish the following properties.

1. Deleting all vertices $v^i, i \in [k]$ makes P acyclic and eliminates all directed paths from any vertex $v^i_{\ell,in}$ to any vertex $v^j_{\ell',out}$, for all $i, j \in [k], \ell, \ell' \in [M]$.

2. The maximum degree of any vertex of P is $4k-2$.
3. There is a k-coloring of P that gives all vertices of $\{v^i_{\ell,in}, v^i_{\ell,out} \mid \ell \in [M]\}$ color i, for all $i \in [k]$.
4. In any k-coloring of P, for all i, all vertices of $\{v^i_{\ell,in}, v^i_{\ell,out} \mid \ell \in [M]\}$ receive the same color as v^i.

Before we go on to prove these four properties of P, let us explain why they imply the theorem. To complete the construction, we insert P in our graph in the place of the palette clique we were previously using. To each vertex of the original graph, we associate a distinct row of P (there are sufficiently many rows to do this). Now, if vertex u of the original graph, which is associated to row ℓ, had an arc from (respectively to) the vertex v_i in the palette, we add an arc from $v^i_{\ell,out}$ (respectively to $v^i_{\ell,in}$).

Let us first establish that the new graph has the properties promised in the theorem. The maximum degree of any vertex in the main (non-palette) part remains unchanged and is $2k+2 \leq 4k-1$ while the maximum degree of any vertex of P is now at most $4k-1$, as we added at most one arc to each vertex. Deleting $\{v^i \mid i \in [k]\}$ eliminates all cycles in P, but also all cycles going through P, because such a cycle would need to use a path from a vertex $v^i_{\ell,in}$ (since these are the only vertices with incoming arcs from outside P) to a vertex $v^j_{\ell',out}$. Deleting all of P leaves the graph acyclic (recall that the palette clique was a feedback vertex set in our previous construction), so there is a feedback vertex set of size k.

For the feedback arc set we remove the arcs $\{v^j v^i \mid j > i,\ i,j \in [k]\} \cup \{v^i_{1,in} v^j \mid j > i,\ i,j \in [k]\} \cup \{v^i_{1,in} v^i \mid i \in [k]\}$. Note that these are indeed k^2 arcs. To see that this is a feedback arc set, suppose that the graph contains a directed cycle after its removal. This cycle must contain some vertex v^i, because we argued that $\{v^i \mid i \in [k]\}$ is a feedback vertex set. Among these vertices, select the v^i with minimum i. We now examine the arc of the cycle going into v^i. Its tail cannot be v^j for $j > i$, as we have removed such arcs, nor v_j for $j < i$, as this contradicts the minimality of i. It cannot be $v^i_{1,in}$ as we have also removed these arcs. And it cannot be $v^j_{1,in}$ for $j < i$, as these arcs are also removed. But no other in-neighbor of v^i remains, contradiction.

Let us also argue that using the gadget P instead of the palette clique does not affect the k-colorability of the graph. This is not hard to see because, following Properties 3 and 4 we can assume that any k-coloring of P will give color i to all vertices of $\{v^i\} \cup \{v^i_{\ell,in}, v^i_{\ell,out} \mid \ell \in [M]\}$. The important observation is now that, for all $\ell \in [M]$ there will always exist a monochromatic path from v^i to $v^i_{\ell,out}$ and from $v^i_{\ell,in}$ to v^i. We now note that, if we fix a coloring of the non-palette part of the graph, this coloring contains a monochromatic cycle involving vertex v_i of the original palette if and only if the same coloring gives a monochromatic cycle in the new graph going through v^i.

Finally, we need to prove the four properties.

Property 1. Once we delete $\{v^i \mid i \in [k]\}$ we observe that for every vertex $v^i_{\ell,in}$ its only outgoing arcs are to vertices $v^j_{\ell,in}$ for $j < i$ or vertices $v^j_{\ell-1,in}$ for $j \geq i$. This shows that we have eliminated all directed paths from $v^i_{\ell,in}$ to $v^j_{\ell',out}$. Furthermore, this shows that no cycle can be formed using $v^i_{\ell,in}$ vertices, since all their outgoing arcs either move to a previous row, or stay in the same row but decrease i. In a similar way, no directed cycle can be formed using only $v^i_{\ell,out}$ vertices, as all their outgoing arcs either move to a later row, or stay in the same row but decrease i.

Property 2. For a vertex v^i we have $2k-2$ arcs incident on it from the clique; the two arcs connecting it to $v^i_{1,in}, v^i_{1,out}$; two arcs connecting it to $v^j_{1,in}, v^j_{1,out}$ for $j > i$; two arcs connecting it to $v^j_{1,in}$ for $j < i$. This gives $2k-2+2i+2(k-i) = 4k-2$.

For a vertex $v^i_{1,in}$ we have one arc to v^j for $j \leq i$; two arcs to v^j for $j > i$; arcs to all v^j_{in}, v^j_{out} for $j \neq i$; arcs to $v^j_{2,in}$ for $j \leq i$. This gives $i+2(k-i)+2(k-1)+i = 4k-2$.

For a vertex $v^i_{1,out}$ we have arcs from v^j for $j \leq i$; arcs to $v^j_{1,in}, v^j_{1,out}$ for $j \neq i$; arcs to $v^j_{2,in}$ and $v^j_{2,out}$ for $j \geq i$; arcs to $v^j_{2,in}$ for all $j < i$. This gives $i+2(k-1)+2(k-i)+i = 4k-2$.

For a vertex $v^i_{\ell,in}$, $\ell \geq 2$ we have arcs to $v^j_{\ell-1,in}$, for $j \geq i$; to $v^j_{\ell-1,out}$ for $j \neq i$; to $v^j_{\ell,in}, v^j_{\ell,out}$ for $j \neq i$; from $v^j_{\ell+1,in}$ for $j \leq i$. This gives $(k-i+1)+(k-1)+2(k-2)+i = 4k-2$.

Finally, for a vertex $v^i_{\ell,out}$, $\ell \geq 2$ we have arcs from $v^j_{\ell-1,out}$ for $j \leq i$; to $v^j_{\ell,in}, v^j_{\ell,out}$ for $j \neq i$; to all $v^j_{\ell+1,in}$, for $j \neq i$; to $v^j_{\ell+1,out}$ for $j \geq i$. This gives $i+2(k-1)+(k-1)+(k-i+1) = 4k-2$.

Property 3. We assign color i to v^i and to $\{v^i_{\ell,in}, v^i_{\ell,out} \mid \ell \in [M]\}$. We claim that there is no monochromatic cycle in P with this coloring. Indeed, if such a cycle exists, it must use v^i, as $\{v^i \mid i \in [k]\}$ is a feedback vertex set. But observe that with the coloring we gave, for each $\ell \in [M-1]$ the only out-neighbor of $v^i_{\ell,out}$ with color i is $v^i_{\ell+1,out}$ and $v^i_{M,out}$ has no out-neighbor of color i. Similar examination of $\{v^i_{\ell,in} \mid \ell \in [M]\}$ shows that the part of P colored i induces a directed path on $2M+1$ vertices with v^i in the middle.

Property 4. Since the vertices v^i induce a clique, we may assume without loss of generality that we are given a coloring c where $c(v^i) = i$. We prove the property by induction on ℓ. For $\ell = 1$, we will first prove that $c(v^i_{1,in}) = i$ by induction on i. For the base case we have that $v^1_{1,in}$ is connected with a digon with v^j for all $j > 1$, so $c(v^1_{1,in}) = 1$. Now, fix a j and suppose that for all $i < j$ we have $c(v^i_{1,in}) = i$. Then $v^j_{1,in}$ cannot receive any color $i < j$, because this would make a cycle with $v^i_{1,in}, v^i$. It can also not receive a color $i > j$ because it has a digon to all v^i for $i > j$. Hence, $c(v^j_{1,in}) = j$. Continuing on $\ell = 1$, we will prove by reverse induction on i that $c(v^i_{1,out}) = i$. For $c(v^k_{1,out})$ if we give this vertex any color $j < k$ then we get a cycle with $v^j, v^j_{1,in}$, so we must have $c(v^k_{1,out}) = k$. Now fix an i and suppose that for all $j > i$ we have $c(v^j_{1,out}) = j$. If we give $v^i_{1,out}$ a color $j > i$ this will make a cycle with $v_j, v^j_{1,out}, v^i_{1,out}, v^j_{1,in}$. But if we give $v^i_{1,out}$ a smaller color $j < i$, this will also make a cycle with $v^j, v^j_{1,in}$. Therefore, $c(v^i_{1,out}) = i$ for all i.

Suppose now that the property is true for row ℓ and we want to prove it for row $\ell+1$. We will use similar reasoning as in the previous case. We will also use the observation that for all i, there is a monochromatic path from v^i to $v^i_{\ell,out}$ and a monochromatic path from $v^i_{\ell,in}$ to v^i. First, we show by induction on i that $c(v^i_{\ell+1,in}) = i$ for all i. For $v^1_{\ell+1,in}$ we observe that if we give this vertex color $j > 1$, then using the arcs from $v^j_{\ell,out}$ and to $v^j_{\ell,in}$ we have a monochromatic cycle of color j. Hence, $c(v^1_{\ell+1,in}) = 1$. Fix a j and suppose that for all

$i < j$ we have $c(v^i_{\ell+1,in}) = i$. If we assign $c(v^j_{\ell+1,in})$ a color $i < j$, then we get a cycle using $v^i_{\ell,out}, v^j_{\ell+1,in}, v^i_{\ell+1,in}, v^i_{\ell,in}$. If we assign it a color $i > j$, then we get the cycle using $v^i_{\ell,out}, v^j_{\ell+1,in}, v^i_{\ell,in}$. So, for all i we have $c(v^i_{\ell+1,in}) = i$. To complete the proof, we do reverse induction to show that $c(v^i_{\ell+1,out}) = i$. For $c(v^k_{\ell+1,out})$ we cannot give this vertex color $j < k$ because this will give a cycle using $v^j_{\ell,out}, v^k_{\ell+1,out}, v^j_{\ell+1,in} v^j_{ell,in}$. Now, fix an i and assume that for $j > i$ we have $c(v^j_{\ell+1,out}) = j$. We cannot assign $v^i_{\ell+1,out}$ any color $j > i$ because this would give the cycle $v^j_{\ell,out}, v^j_{\ell+1,out}, v^i_{\ell+1,out}, v^j_{\ell+1,in}, v^j_{\ell,in}$. We can also not assign any color $j < i$ as this gives the cycle using $v^j_{\ell,out}, v^i_{\ell+1,out}, v^j_{\ell+1,in}, v^j_{\ell,in}$. We conclude that for all i we have $c(v^i_{\ell+1,out}) = i$.

5 Treewidth

In this section we consider the complexity of Digraph Coloring with respect to parameters measuring the acyclicity of the underlying graph, namely, treewidth and treedepth. Before we proceed let us recall that in all graphs G we have $\chi(G) \leq \mathrm{tw}(G) + 1 \leq \mathrm{td}(G) + 1$. This means that if our goal is simply to obtain an FPT algorithm then parameterizing by treewidth implies that the graph's chromatic number (and therefore also the digraph's dichromatic number) is bounded. We first present an algorithm with complexity $k^{\mathrm{tw}}(\mathrm{tw}!)$ which, using the above argument, proves that Digraph Coloring is FPT parameterized by treewidth.

Theorem 7 *There is an algorithm which, given a digraph D on n vertices and a tree decomposition of its underlying graph of width* tw *decides if D is k-colorable in time $k^{\mathrm{tw}}(\mathrm{tw}!)n^{O(1)}$.*

Proof The proof uses standard techniques so we sketch some details. In particular we assume that we are given a nice tree decomposition on which we will perform dynamic programming. Before we proceed, let us slightly recast the problem. We will say that a digraph $D = (V, E)$ is k-colorable if there exist two functions c, σ such that (i) $c : V \to [k]$ partitions V into k sets (ii) σ is an ordering of V (iii) for all arcs $uv \in E$ we have either $c(u) \neq c(v)$ or $\sigma(u) < \sigma(v)$. It is not hard to see that this reformulation is equivalent to the original problem. Indeed, if we have a k-coloring, since each color class is acyclic, we can find a topological ordering σ_i of the graph $G[V_i]$ induced by each color class and then concatenate them to obtain an ordering of V. For the converse direction, the existence of c, σ implies that if we look at vertices of each color class, σ must induce a topological ordering, hence each color class is acyclic.

Now, let D be a digraph and S be a subset of its vertices. Let (c, σ) be a pair of coloring and ordering functions that prove that D is k-colorable. Then, we will say that the signature of solution (c, σ) for set S is the pair (c_S, σ_S) where $c_S : S \to [k]$ is defined as $c_S(u) = c(u)$ and $\sigma_S : S \to [|S|]$ is an ordering function such that for all $u, v \in S$ we have $\sigma_S(u) < \sigma_S(v)$ if and only if $\sigma(u) < \sigma(v)$. In other words, the signature of a solution is the restriction of the solution to the set S.

Given a rooted nice tree decomposition of D, let B_t be a bag of the decomposition and denote by $B_t^{\downarrow}$ the set of vertices of D which are contained in B_t and bags in the sub-tree rooted at B_t. Our dynamic programming algorithm stores for each B_t a collection of all pairs (c, σ) such that there exists a k-coloring of $D[B_t^{\downarrow}]$ whose signature is (c, σ). If we manage to construct such a table for each node, it will suffice to check if the collection of signatures of the root is empty to decide if the graph is k-colorable.

The table is easy to initialize for Leaf nodes, as the only valid signature contains the empty coloring and ordering function. For an Introduce node that adds u to a bag containing B_t we consider all signatures (c, σ) of contained in the table of the child bag. For each such signature we construct a signature (c', σ') which is consistent with (c, σ) but also colors u and places it somewhere in the ordering (we consider all such possibilities). For each (c', σ') we delete this signature if u has a neighbor in the bag who is assigned the same color by c' but such that their arc violates the topological ordering σ'. We keep all other produced signatures. To see that this is correct observe that u has no neighbors in $B_t^{\downarrow} \setminus B_t$, because all bags are separators, so if we produce an ordering of $B_t^{\downarrow}$ consistent with σ' the only arcs incident on u that could violate it are contained in the bag (and have been checked). For Forget nodes the table is easily update by keeping only the restrictions of valid signatures to the new bag. Finally, for Join nodes we keep a signature (c, σ) if and only if it is valid for both sub-trees. Again this is correct because nodes of one sub-tree not contained in the bag do not have neighbors in the other sub-tree, so as long as we produce an ordering consistent with σ we can concatenate we cannot violate the topological ordering condition.

For the running time observe that the size of the DP table is $k^{\text{tw}}(\text{tw}!)$, because we consider all colorings and all ordering of each bag. In Introduce nodes we spend polynomial time for each entry of the child node (checking all placements of the new vertex), while computation in Join nodes can be performed in time linear in the size of the table. So the running time is in the end $k^{\text{tw}}(\text{tw}!)n^{O(1)}$.

As we explained, even though Theorem 7 implies that Digraph Coloring is FPT parameterized by treewidth, the complexity it gives is significantly worse than the complexity of Coloring, which is essentially k^{tw}. Our main result in this section is to show that this is likely to be inevitable, even if we focus on the more restricted case of treedepth and 2 colors.

Theorem 8 *If there exists an algorithm which decides if a given digraph on n vertices and (undirected) treedepth* td *is* 2*-colorable in time* $\text{td}^{o(\text{td})}n^{O(1)}$*, then the ETH is false.*

Proof Suppose we are given a 3-SAT formula ϕ with n variables and m clauses. We will produce a digraph G such that $|V(G)| = 2^{O(n/\log n)}m$ and $\text{td}(G) = O(n/\log n)$ and G is 2-colorable if and only if ϕ is satisfiable. Before we proceed, observe that if we can construct such a graph the theorem follows, as an algorithm with running time $O^*(\text{td}^{o(td)})$ for 2-coloring G would decide the satisfiability of ϕ in time $2^{o(n)}$.

To simplify presentation we assume without loss of generality that n is a power of 2 (otherwise adding dummy variables to ϕ can achieve this while increasing n be a factor of at most 2). We begin the construction of G by creating $\log n$ independent sets $V_1, \ldots, V_{\log n}$, each of size $\lceil \frac{2en}{\log^2 n} \rceil$. We add a vertex u and connect it with arcs in both directions to all vertices of $\cup_{i \in [\log n]} V_i$. We also partition the variables of ϕ into $\log n$ sets $X_1, \ldots, X_{\log n}$ of size at most $\lceil \frac{n}{\log n} \rceil$.

The main idea of our construction is that the vertices of V_i will represent an assignment to the variables of X_i. Observe that all vertices of V_i are forced to obtain the same color (as all are forced to have a distinct color from u), therefore the way these vertices represent an assignment is via their topological ordering in the DAG they induce together with other vertices of the graph which obtain the same color.

To continue our construction, for each $i \in [\log n]$ we do the following: we enumerate all the possible truth assignments of the variables of X_i and for each such truth assignment $\sigma : X_i \to \{0,1\}^{|X_i|}$ we define (in an arbitrary way) a distinct ordering $\rho(\sigma)$ of the vertices of V_i. We will say that the ordering $\rho(\sigma)$ is the *translation* of assignment σ. Note that there are $|V_i|! \geq (\frac{2en}{\log^2 n})! \geq (\frac{2n}{\log^2 n})^{\frac{2en}{\log^2 n}} = 2^{\frac{2en}{\log^2 n}(1+\log n - 2\log\log n)} > 2^{\lceil \frac{n}{\log n} \rceil}$ for n sufficiently large, so it is possible to translate truth assignments to X_i to orderings of V_i injectively. Note that enumerating all assignments for each group takes time $2^{O(n/\log n)} = 2^{o(n)}$.

Consider now a clause c_j of ϕ and suppose some variable of the group X_i appears in c_j. For each truth assignment σ to X_i which satisfies c_j we construct an independent set $S_{j,i,\sigma}$ of size $|X_i|-1$, label its vertices $s^{\ell}_{j,i,\sigma}$, for $\ell \in [|X_i|-1]$. For each ℓ we add an arc from $\rho(\sigma)^{-1}(\ell)$ to $s^{\ell}_{j,i,\sigma}$ and an arc from $s^{\ell}_{j,i,\sigma}$ to $\rho(\sigma)^{-1}(\ell+1)$. In other words, the ℓ-th vertex of $S_{j,i,\sigma}$ has an incoming arc from the vertex of V_i which is ℓ-th according to the ordering $\rho(\sigma)$ which is the translation of assignment σ and an outgoing arc to the vertex of V_i which is in position $(\ell+1)$ in the same ordering. Observe that this implies that if all vertices of V_i and of $S_{j,i,\sigma}$ are given the same color, then the topological ordering of the induced DAG will agree with $\rho(\sigma)$ on the vertices of V_i.

To complete the construction, for each clause c_j we do the following: take all independent sets $S_{j,i,\sigma}$ which we have constructed for c_j and order them in a cycle in some arbitrary way. For two sets $S_{j,i,\sigma}, S_{j,i',\sigma'}$ which are consecutive in this cycle add a new "connector" vertex $p_{j,i,\sigma,i',\sigma'}$, all arcs from $S_{j,i,\sigma}$ to this vertex, and all arcs from this vertex to $S_{j,i',\sigma'}$. Finally, we connect each connector vertex $p_{j,i,\sigma,i',\sigma'}$ we have constructed to an arbitrary vertex of V_1 with a digon. This completes the construction.

Let us argue that if ϕ is satisfiable, then G is 2-colorable. We color u with color 2, all the vertices in V_i for $i \in [\log n]$ with 1 and all connector vertices $p_{i,j,\sigma,i',\sigma'}$ with 2. For each clause c_j there exists a group X_i that contains a variable of c_j such that the supposed satisfying assignment of ϕ, when restricted to X_i gives an assignment $\sigma : X_i \to \{0,1\}^{|X_i|}$ which satisfies c_j. Therefore, there exists a corresponding set $S_{j,i,\sigma}$. Color all vertices of this set with 1. After doing this for all clauses, we color all other vertices with 2. We claim this is a valid 2-coloring. Indeed, the graph induced by color 2 is acyclic, as it contains u (but none of its neighbors) and for each c_j, all but one of the sets $S_{j,i,\sigma}$ and the vertices $p_{j,i,\sigma,i',\sigma'}$. Since these sets have been connected in a directed cycle throught connector vertices, and for each c_j we have colored one of these sets with 1, the remaining sets induce a DAG. For the graph induced by color 1 consider for each V_i the ordering $\rho(\sigma)$, where σ is the satisfying assignment restricted to V_i. Every vertex outside V_i which received color 1 and has arcs to V_i, has exactly one incoming and one outgoing arc to V_i. Furthermore, the directions of these arcs agree with the ordering $\rho(\sigma)$. Hence, since $\cup_{i\in[\log n]} V_i$ touches all arcs with both endpoints having color 1 and all such arcs respect the orderings of V_i, the graph induced by color 1 is acyclic.

For the converse direction, suppose we have a 2-coloring of G. Without loss of generality, u has color 2 and $\cup_{i\in[\log n]} V_i$ has color 1. Furthermore, all connectors

$p_{j,i,\sigma,i',\sigma'}$ also have color 2. Consider now a clause c_j. We claim that there must be a group $S_{j,i,\sigma}$ such that $S_{j,i,\sigma}$ does not use color 2. Indeed, if all such groups use color 2, since they are linked in a directed cycle with all possible arcs between consecutive groups and connectors, color 2 would not induce a DAG. So, for each c_j we find a group $S_{j,i,\sigma}$ that is fully colored 1 and infer from this the truth assignment σ for the group X_i. Doing this for all clauses gives us an assignment that satisfies every clause. However, we need to argue that the assignment we extract is consistent, that is, there do not exist $S_{j,i,\sigma}$ and $S_{j',i,\sigma'}$ which are fully colored 1 with $\sigma \neq \sigma'$. For the sake of contradiction, suppose that two such sets exist, and recall that $\rho(\sigma) \neq \rho(\sigma')$. We now observe that if $S_{j,i,\sigma} \cup V_i$ only uses color 1, then any topological ordering of V_i in the graph induced by color 1 must agree with $\rho(\sigma)$, which is a total ordering of V_i. In a similar way, the ordering of V_i must agree with $\rho(\sigma')$, so if $\sigma \neq \sigma'$ we get a contradiction.

Finally, let us argue about the parameters of G. For each clause c_j of ϕ we construct an independent set of size $O(n/\log^2 n)$ for each satisfying assignment of a group X_i containing a variable of c_j. There are at most 3 such groups, and each group has at most $2^{n/\log n}$ satisfying assignments for c_j, so $|V(G)| = 2^{O(n/\log n)}m$.

For the treedepth, recall that deleting a vertex decreases treedepth by at most 1. We delete u and all of $\cup_{i\in[\log n]}V_i$ which are $O(n/\log n)$ vertices in total. It now suffices to prove that in the remainder all components have treedepth $O(n/\log n)$. In the remainder every component is made up of the directed cycle formed by sets $S_{j,i,\sigma}$ and connectors $p_{j,i,\sigma,i',\sigma'}$. We first delete a vertex $p_{j,i,\sigma,i',\sigma'}$ to turn the cycle into a directed "path" of length $L = 2^{O(n/\log n)}$. We now use the standard argument which proves that paths of length L have treedepth $\log L$, namely, we delete the $p_{j,i,\sigma,i',\sigma'}$ vertex that is closest to the middle of the path and then recursively do the same in each component. This shows that the remaining graph has treedepth logarithmic in the length of the path, therefore at most $O(n/\log n)$.

6 2-Coloring Tournaments

In this section we propose an algorithm that decides if a given tournament T is 2-colorable in time $O^*(\sqrt[3]{6}^n)$. Our algorithm starts by removing, arbitrarily, as many disjoint triangles from the tournament as possible and then considers all the proper partial colorings of the tournament induced on these triangles. Then we use a recursive algorithm in order to determine if any of these partial colorings can be extended to a proper 2-coloring for the whole tournament.

As we mentioned, the previous algorithm uses another one in order to decide if a partial coloring is extendable.

In order to decide it, we search for two types of triangles in the tournament - triangles that contain one uncolored vertex and two vertices with the same color and triangles that contain only one colored vertex. For the first type, it is easy to see that we know which color we have to assign to the uncolored vertex. However, for the second type, the algorithm calls itself in order to decide if any of the possible colorings is extendable to this triangle.

Before we continue to the proof, let us recall that any tournament T that has a directed cycle must contain a triangle. Therefore, in the Algorithm 1 we know that the graph $T[V \setminus V_1]$ where V_1 is the set that we use in the line 8, is acyclic as we could not find any other triangles in it.

Algorithm 1 [2-COL(T) decision function]

Input: A tournament $T = (V, E)$.
Output: Is $\vec{\chi}(T) = 2$ or not?

$V_1 \leftarrow \emptyset$, $V_2 \leftarrow V$
$IsTwoDC \leftarrow$ **False**
while there is a triangle $\{v_1, v_2, v_3\}$ in V_2 **do**
  $V_1 \leftarrow V_1 \cup \{v_1, v_2, v_3\}$
  $V_2 \leftarrow V_2 \setminus \{v_1, v_2, v_3\}$
end while
for all 2-coloring $\mathcal{C} : V_1 \to \{1, 2\}$ that are proper **do**
  $IsTwoDC \leftarrow IsTwoDC \vee$ Ext 2-DCN(T,V_1,$\mathcal{C}$)
end for
return $IsTwoDC$

Algorithm 2 [Ext 2-COL(T,V_C,$\mathcal{C}$) decision function]

Input: A tournament $T = (V, E)$, a set of vertices $V_C \subseteq V$ and a function $\mathcal{C} : V_C \to \{1, 2\}$.
Output: Can we find a proper 2-coloring for T by extending $\mathcal{C}$?

$V_{NC} \leftarrow V \setminus V_C$
$Ext \leftarrow$ **False**
while there is a triangle $\{v_1, v_2, v_3\}$ such that $v_1 \in V_{NC}$, $v_2, v_3 \in V_C$ and $\mathcal{C}(v_2) = \mathcal{C}(v_3)$ **do**
  $V_C \leftarrow V_C \cup \{v_1\}$, $V_{NC} \leftarrow V_{NC} \setminus \{v_1\}$
  set $\mathcal{C}(v_1)$ to be the color that is not $\mathcal{C}(v_2) = \mathcal{C}(v_3)$
end while
if $\mathcal{C}$ is a proper coloring for V_C **then**
  while there is a triangle $\{v_1, v_2, v_2\}$ such that $v_1, v_2 \in V_{NC}$ and $v_3 \in V_C$ **do**
    $V_C \leftarrow V_C \cup \{v_1, v_2\}$, $V_{NC} \leftarrow V_{NC} \setminus \{v_1, v_2\}$
    for all the pairs $\{Col_1, Col_2\} \neq \{\mathcal{C}(v_3), \mathcal{C}(v_3)\}$ **do**
      set $\mathcal{C}(v_1) \leftarrow Col_1$ and $\mathcal{C}(v_2) \leftarrow Col_2$
      $Ext \leftarrow Ext \vee$ Ext 2-DNC(T,V_C,$\mathcal{C}$)
    end for
  end while
end if
for all $v \in V_{NC}$ set $\mathcal{C}(v)$ to be 1
if $\mathcal{C}$ is a proper coloring for V **then**
  $Ext \leftarrow$ **True**
end if
return Ext

Now, let us prove that the Algorithm 2 does what we claim.

Lemma 2 *Given a tournament $T = (V, E)$, a set of vertices $S \subseteq V$ such that $T[V \setminus S]$ is acyclic and a function $\mathcal{C} : S \to \{1, 2\}$ of S, Algorithm 2 applied to $V_C = S$ decides if we can find a function $\mathcal{C}^* : V \to \{1, 2\}$ that gives a proper 2-coloring for the tournament T such that $\mathcal{C}^*(v) = \mathcal{C}(v)$ for all $v \in S$.*

Proof If the function cannot be extended the algorithm will return **False** because in order to change the value to **True** that means that at one of the calls of the algorithm we checked an extension $\mathcal{C}^*$ of $\mathcal{C}$ and it was a proper coloring for the tournament which is a contradiction. So we have to prove that if the given function $\mathcal{C}$ can be extended in order to give a proper coloring of the whole tournament then the algorithm will return **True**. For the rest of the proof let us call the triangles that contain one uncolored vertex and two vertices of the same color as *type one* and the triangles with two uncolored vertices as *type two*. Let $\mathcal{C}$ be extendable (i.e., there is an extension $\mathcal{C}^*$ that gives a proper coloring for the tournament);

the algorithm first checks if there exists a triangle of *type one* and gives to the uncolored vertices the other color (in line 5). It is clear that this is the only option for these vertices so that the new color function remains extendable. After that the algorithm checks for triangles of *type two*. In this case we know that the two uncolored vertices cannot have both the same color as the third vertex; so we have a total of $2^2 - 1 = 3$ cases. After that the algorithm checks (between lines 10 and 13) if any of these possibilities can be can be extended and gives us a proper coloring (by calling itself in line 12).

As we mentioned, the algorithm tries to extend all the possible colorings (except those that are not proper) so at some point we have an extendable function $\mathcal{C}$ and either we do not have any uncolored vertices or we do not have any triangles of *type one* or *two*.

Case 1. Suppose $V_{NC} = \emptyset$ when line 16 of Algorithm 2 is executed. Then $\mathcal{C}$ is a proper coloring of V which means that after the check in line 18 we change the value of the variable Ext to **True**.

It remains to show that in the second case if we colored the remaining uncolored vertices with any color we have a proper coloring for T.

Case 2. In this case we do not have any triangles of *type one* or *two*. This combined with the assumption that the coloring is extendable implies that by coloring the remaining vertices with any color we end up with a coloring that does not have any monochromatic triangle. It remains to show the following claim:

Claim Let $T = (V, E)$ be a tournament and $C : V \to \{1, 2\}$ a function that is a 2-coloring of T such that there is no monochromatic triangle. Then C is a proper coloring.

Proof Assume that C does not give a proper coloring. Then there must exist a monochromatic cycle S with length grater than 3. Note that S induces a tournament. But any tournament which contains a directed cycle contains a triangle. This gives a contradiction since there are no monochromatic triangles in T.

So our coloring is a proper 2-coloring; thus the algorithm will change the value of the variable Ext to **True** in line 18 and due to the logic *or* in line 12 this **True** will be kept until the algorithm terminates.

Finally we are going to prove that Algorithm 1 decides if a tournament is 2-colorable and that it runs in $O^*(\sqrt[3]{6}^n)$ time.

Theorem 9 *Given a tournament $T = (V, E)$, Algorithm 1 decides if T is 2-colorable.*

Proof It is easy to see that Algorithm 1 tries to extend any proper coloring of V_1. Now, in order to use Lemma 2 we need to observe that we have no triangles in V_2. Since a tournament without triangles is acyclic, it follows that V_2 is acyclic. So, from lemma 2 we know that if one of these colorings can be extended then the Algorithm 2 will return **True**. Thus, Algorithm 1 returns **True** if the tournament is 2-colorable and **False** otherwise.

Theorem 10 *Let $T = (V, E)$ be a tournament. Then we can decide if the dichromatic number of T is two in time $\mathcal{O}^*(\sqrt[3]{6}^n)$*

Proof Observe that in Algorithm 1 all the steps are polynomial except the number of the proper ways to color the set V_1 and the time Algorithm 2 needs. Now, it is easy to see that the number of proper ways to color V_1 is at most $6^{\frac{|V_1|}{3}}$ since for every triangle in V_1 we know that we have six possible choices to color it (all except the two that give to every vertex the same color). This means that we call Algorithm 2 at most $6^{\frac{|V_1|}{3}}$ times. The running time of the second algorithm depends on the number of times that it will call itself. Now we can see that for the remaining vertices ($V_2 = V \setminus V_1$), in the worst case, we need to check three different colorings (see proof of lemma 2) for two vertices at a time. Thus, the running time of Algorithm 2 is $\mathcal{O}^*(3^{\frac{|V_2|}{2}})$. So, we can decide if T is 2-colorable in time

$$\mathcal{O}^*(6^{\frac{|V_1|}{3}} \cdot 3^{\frac{|V_2|}{2}}) = \mathcal{O}^*(\sqrt[3]{6}^{|V_1|+|V_2|}) = \mathcal{O}^*(\sqrt[3]{6}^{n})$$

7 Conclusions

In this paper we have strengthened known results about the complexity of Digraph Coloring on digraphs which are close to being DAGs, precisely mapping the threshold of tractability for DFVS and FAS; and we precisely bounded the complexity of the problem parameterized by treewidth, uncovering an important discrepancy with its undirected counterpart. One question for further study is to settle the degree bound for which k-Digraph Coloring is NP-hard for DFVS k, and more generally to map out how the tractability threshold for the degree evolves for larger values of the DFVS from $4k - \Theta(1)$ to $2k + \Theta(1)$, which is the correct threshold when the DFVS is unbounded. With regards to undirected structural parameters, it would be interesting to investigate whether a $\mathrm{vc}^{o(\mathrm{vc})}$ algorithm exists for 2-Digraph Coloring, where vc is the input graph's vertex cover, as it seems challenging to extend our hardness result to this more restricted case.

References

1. Pierre Aboulker, Nathann Cohen, Frédéric Havet, William Lochet, Phablo F. S. Moura, and Stéphan Thomassé. Subdivisions in digraphs of large out-degree or large dichromatic number. *Electron. J. Comb.*, 26(3):P3.19, 2019.
2. Stephan Dominique Andres and Winfried Hochstättler. Perfect digraphs. *Journal of Graph Theory*, 79(1):21–29, 2015.
3. Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. A complexity dichotomy for hitting connected minors on bounded treewidth graphs: the chair and the banner draw the boundary. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 951–970. SIAM, 2020.
4. Julien Bensmail, Ararat Harutyunyan, and Ngoc-Khang Le. List coloring digraphs. *Journal of Graph Theory*, 87(4):492–508, 2018.
5. Benjamin Bergougnoux, Édouard Bonnet, Nick Brettell, and O-joung Kwon. Close relatives of feedback vertex set without single-exponential algorithms parameterized by treewidth. *CoRR*, abs/2007.14179, 2020.
6. Dietmar Berwanger, Anuj Dawar, Paul Hunter, Stephan Kreutzer, and Jan Obdrzálek. The dag-width of directed graphs. *J. Comb. Theory, Ser. B*, 102(4):900–923, 2012.
7. Drago Bokal, Gasper Fijavz, Martin Juvan, P. Mark Kayll, and Bojan Mohar. The circular chromatic number of a digraph. *Journal of Graph Theory*, 46(3):227–240, 2004.

8. Marthe Bonamy, Lukasz Kowalik, Jesper Nederlof, Michal Pilipczuk, Arkadiusz Socala, and Marcin Wrochna. On directed feedback vertex set parameterized by treewidth. In Andreas Brandstädt, Ekkehard Köhler, and Klaus Meer, editors, *Graph-Theoretic Concepts in Computer Science - 44th International Workshop, WG 2018, Cottbus, Germany, June 27-29, 2018, Proceedings*, volume 11159 of *Lecture Notes in Computer Science*, pages 65–78. Springer, 2018.
9. Édouard Bonnet, Nick Brettell, O-joung Kwon, and Dániel Marx. Generalized feedback vertex set problems on bounded-treewidth graphs: Chordality is the key to single-exponential parameterized algorithms. *Algorithmica*, 81(10):3890–3935, 2019.
10. 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):21:1–21:19, 2008.
11. Xujin Chen, Xiaodong Hu, and Wenan Zang. A min-max theorem on tournaments. *SIAM J. Comput.*, 37(3):923–937, 2007.
12. Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
13. Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 150–159. IEEE Computer Society, 2011.
14. Tomás Feder, Pavol Hell, and Carlos S. Subi. Complexity of acyclic colorings of graphs and digraphs with degree and girth constraints. *CoRR*, abs/1907.00061, 2019.
15. Henning Fernau and Daniel Meister. Digraphs of bounded elimination width. *Discret. Appl. Math.*, 168:78–87, 2014.
16. Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, and Saket Saurabh. Intractability of clique-width parameterizations. *SIAM J. Comput.*, 39(5):1941–1956, 2010.
17. Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. Clique-width III: hamiltonian cycle and the odd case of graph coloring. *ACM Trans. Algorithms*, 15(1):9:1–9:27, 2019.
18. Jakub Gajarský, Michael Lampis, and Sebastian Ordyniak. Parameterized algorithms for modular-width. In *IPEC*, volume 8246 of *Lecture Notes in Computer Science*, pages 163–176. Springer, 2013.
19. Robert Ganian, Petr Hlinený, Joachim Kneis, Daniel Meister, Jan Obdrzálek, Peter Rossmanith, and Somnath Sikdar. Are there any good digraph width measures? *J. Comb. Theory, Ser. B*, 116:250–286, 2016.
20. Frank Gurski, Dominique Komander, and Carolin Rehs. Acyclic coloring of special digraphs. *CoRR*, abs/2006.13911, 2020.
21. Ararat Harutyunyan. Brooks-type results for coloring of digraphs. *PhD Thesis*, Simon Fraser University, 2011.
22. Ararat Harutyunyan, Mark Kayll, Bojan Mohar, and Liam Rafferty. Uniquely d-colorable digraphs with large girth. *Canad. J. Math.*, 64(6):1310–1328, 2012.
23. Ararat Harutyunyan, Tien-Nam Le, Stéphan Thomassé, and Hehui Wu. Coloring tournaments: From local to global. *J. Comb. Theory, Ser. B*, 138:166–171, 2019.
24. Winfried Hochstättler, Felix Schröder, and Raphael Steiner. On the complexity of digraph colourings and vertex arboricity. *Discret. Math. Theor. Comput. Sci.*, 22(1), 2020.
25. Paul Hunter and Stephan Kreutzer. Digraph measures: Kelly decompositions, games, and orderings. *Theor. Comput. Sci.*, 399(3):206–219, 2008.
26. Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.
27. Lars Jaffke and Bart M. P. Jansen. Fine-grained parameterized complexity analysis of graph coloring problems. In *CIAC*, volume 10236 of *Lecture Notes in Computer Science*, pages 345–356, 2017.
28. Thor Johnson, Neil Robertson, Paul D. Seymour, and Robin Thomas. Directed tree-width. *J. Comb. Theory, Ser. B*, 82(1):138–154, 2001.
29. Michael Lampis. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, 64(1):19–37, 2012.
30. Michael Lampis. Finer tight bounds for coloring on clique-width. In *ICALP*, volume 107 of *LIPIcs*, pages 86:1–86:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.
31. Michael Lampis, Georgia Kaouri, and Valia Mitsou. On the algorithmic effectiveness of digraph decompositions and complexity measures. *Discret. Optim.*, 8(1):129–138, 2011.

32. Zhentao Li and Bojan Mohar. Planar digraphs of digirth four are 2-colorable. *SIAM J. Discret. Math.*, 31(3):2201–2205, 2017.
33. Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Known algorithms on graphs of bounded treewidth are probably optimal. *ACM Trans. Algorithms*, 14(2):13:1–13:30, 2018.
34. Marcelo Garlet Millani, Raphael Steiner, and Sebastian Wiederrecht. Colouring non-even digraphs. *CoRR*, abs/1903.02872, 2019.
35. Bojan Mohar. Circular colorings of edge-weighted graphs. *Journal of Graph Theory*, 43(2):107–116, 2003.
36. Bojan Mohar. Eigenvalues and colorings of digraphs. *Linear Algebra and its Applications*, 432(9):2273 – 2277, 2010. Special Issue devoted to Selected Papers presented at the Workshop on Spectral Graph Theory with Applications on Computer Science, Combinatorial Optimization and Chemistry (Rio de Janeiro, 2008).
37. Victor Neumann-Lara. The dichromatic number of a digraph. *J. Comb. Theory, Ser. B*, 33(3):265–270, 1982.
38. Raphael Steiner and Sebastian Wiederrecht. Parameterized algorithms for directed modular width. In *CALDAM*, volume 12016 of *Lecture Notes in Computer Science*, pages 415–426. Springer, 2020.