New Results on the Complexity of Oriented Colouring on Restricted Digraph Classes

Robert Ganian and Petr Hliněný ⋆

Faculty of Informatics, Masaryk University
Botanická 68a, Brno, Czech Republic
{xganian1,hlineny}@fi.muni.cz

Abstract. Oriented colouring is a quite intuitive generalization of undirected colouring, yet the problem remains NP-hard even on digraph classes with bounded usual directed width measures. In light of this fact, one might ask whether new width measures are required for efficient dealing with this problem or whether further restriction of traditional directed width measures such as DAG-width would suffice. The K-width and DAG-depth measures (introduced by [Ganian et al, IWPEC'09]) are ideal candidates for tackling this question: They are both closely tied to the cops-and-robber games which inspire and characterize the most renowned directed width measures, while at the same time being much more restrictive.

In this paper, we look at the oriented colouring problem on digraphs of bounded K-width and of bounded DAG-depth. We provide new polynomial algorithms for solving the problem on "small" instances as well as new strong hardness results showing that the input restrictions required by our algorithms are in fact "tight".

Key words: directed graph, complexity, oriented colouring, DAG-depth

1 Preliminaries

1.1 Introduction

The study of ordinary colourings of graphs has become the focus of many authors and lead to a number of interesting results. However, only in the last decade has this been extended to directed graphs. The notion of oriented colouring was first introduced by Courcelle [2], see Definition 1.1. Briefly, while an ordinary colouring is a homomorhpism into a complete graph, an oriented colouring is a homomorhpism into an orientation of a complete graph.

Properties of oriented colouring have been studied by several authors, see e.g. the work of Nešetřil and Raspaud [11] or the survey by Sopena [14]. Similarly to undirected colouring, computing the oriented chromatic number (further

⋆ This research has been supported by the Czech research grants GAČR 201/-08/0308 (P. Hliněný) and 201/09/J021 (R. Ganian), and by the research intent MSM0021622419 of the Czech Ministry of Education.

referred to as OCN) and deciding oriented colourability of digraphs are both NP-hard problems. However, while undirected colouring becomes easy if we restrict the input to the graph class of trees, deciding oriented colourability already by 4 colours (OCN_4) remains NP-hard even on on acyclic digraphs (DAGs) [3]. Apart from being an interesting notion from a theoretical point of view, oriented colouring also has practical applications, e.g. in mobile networks.

There exists a wide range of width parameters for digraphs: directed tree-width [8], DAG-width [1, 12], Kelly-width [7], and cycle rank [4] perhaps being the best known. A shared feature of all these width parameters is that they assume their minimum values on DAGs. Thus, it is impossible to use a bound on any of these width parameters to efficiently decide oriented colourability at all — there will always be instances in which deciding OCN_4 remains NP-hard.

One way to interpret this finding is to ask whether there exist stronger, more restrictive digraph width parameters which could help with computing OCN.

Very recently, a possible lead to answering this question has been given in [6]. Two new directed width measures have been introduced in that article, both related to cops-and-robber games (and thus to the classical directed width measures) and both very restrictive. The first one is K-width (Def. 1.2) which restricts the maximum number of directed paths between pairs of vertices, and the second one is DAG-depth (Def. 1.3) which on the other hand restricts the maximum number of moves in a cops-and-robber game.

These parameters have been successfully used in [6] to design some new FPT algorithms, e.g. for the Hamiltonian Path and c-Paths problems. We analyse the relationship of these new measures to OCN. The first results of this paper (Section 2) are two new polynomial algorithms for computing OCN on digraphs of DAG-depth 2, and on digraphs of K-width 1 with a "single reachable fragment". Then we show that, although our algorithms do seem relatively simple and one would expect there to be more involved variants for (at least slightly) more general cases, the bounds in these algorithms are in fact "tight". To this end we introduce a new reduction proving that the OCN_4 problem is NP-complete already for digraphs with K-width 1 and DAG-depth 3 (Theorem 3.4).

1.2 Definitions

We assume that the reader is familiar with all basic definitions related to undirected and directed graphs. Keep in mind that digraph stands for directed graph and DAG stands for acyclic digraph. Our digraphs are simple; they have no parallel arcs or loops, but can have two arcs in opposite directions.

Let G, H be digraphs. A *homomorphism* of G to H is a mapping $f : V(G) \to V(H)$ such that for all $(a, b) \in E(G)$, it holds $(f(a), f(b)) \in E(H)$.

Definition 1.1 ([2]). The *k-oriented chromatic number* (OCN_k) problem is defined as follows: Given a digraph G, is there a homomorphism from G to H_k, where H_k is some (irreflexive antisymmetric) orientation of edges of the complete graph on k vertices?

There is also the natural optimization variant (OCN) — to find the minimum k such that OCN_k is true.

For simplicity, we will sometimes say that a set of vertices of G have the same colour — meaning that they all map into the same vertex of H. Notice that each colour class is an independent set in G, and that if there is an arc from a vertex coloured a to a vertex coloured b, then there can never be an arc from a vertex coloured b to a vertex coloured a. This is a useful and intuitive way of looking at oriented colouring.

Next, we introduce the first of the two aforementioned width parameters:

Definition 1.2 ([6]). A digraph G has *K-width* k if k is the lowest integer such that, for any pair of vertices $s, t \in V(G)$, the number of distinct directed paths from s to t is at most k. Note that these paths need not be pairwise disjoint.

K-width is related to the better-known DAG-width [1, 12] in the sense that bounded K-width implies bounded DAG-width. More precisely, the K-width of a G is greater or equal to the DAG-width of G minus one [6]. On the other hand, DAGs (which have DAG-width 0) can have arbitrarily high K-width.

The last part of the definitions introduces DAG-depth, an interesting directed counterpart to the better known tree-depth [10]. First, we need to formalize the notion of *reachable fragments*. For a digraph G and any $v \in V(G)$, let G_v denote the subdigraph of G induced by the vertices reachable from v. The maximal elements of the poset $\{\, G_v : v \in V(G) \,\}$ in the digraph-inclusion order are then called *reachable fragments* of G (further referred to as $\mathcal{RF}(G)$). Notice that reachable fragments in the undirected case coincide with connected components.

Definition 1.3 ([6]). The *DAG-depth* $ddp(G)$ of a digraph G is inductively defined as follows: If $|V(G)| = 1$, then $ddp(G) = 1$. If G has a single reachable fragment, then $ddp(G) = 1 + \min\{ddp(G - v) : v \in V(G)\}$. Otherwise, $ddp(G) = \max\{ddp(F) : F \in \mathcal{RF}(G)\}$.

DAG-width has a beautiful characterization [12] via a "cops and robber" game: In this game, on a digraph G, the robber can move between the vertices of G along cop-free directed paths at great speed, while cops move to vertices of G in a helicopter which the robber can see and escape. The DAG-width of G then equals the minimum number t of cops sufficient to catch the robber in G (by landing at him when he has no escape route). Similarly:

Theorem 1.4 ([6]). *The DAG-depth of a digraph G is at most t if, and only if, the cop player has a "lift-free" winning strategy in the t-cops and robber game on G, i.e. a strategy that never moves a cop from a vertex once he has landed.*

Based on the game characterization, it is easy to see that DAG-depth may never be higher than DAG-width. However, DAG-depth is in fact much more restrictive than DAG-width: [6] The number of vertices on the longest directed path in a digraph G is at most $2^t - 1$ where $t = ddp(G)$. Theorem 1.4 will be useful for determining the DAG-depth of some digraphs in the next sections.

2 The Algorithms

First of all, we remark that the problems OCN_2 and OCN_3 are trivially solvable, see e.g.[3]. We present our results for solving OCN_4 on digraphs of K-width 1 consisting of a single reachable fragment, and of DAG-depth 2.

2.1 Digraphs of K-width 1

We begin by proving a few structural properties of digraphs G with K-width 1 consisting of a single reachable fragment (i.e. having $|\mathcal{RF}(G)| = 1$). First, choose any vertex such that the whole digraph G is reachable from that vertex. This will be the unique *source* of G, or s. Then perform a Depth-First search of G from s to create a Depth-First search tree; the paths from s to the leaves of this Depth-First search tree will be called *branches*, and the (x, y) arcs where y is a predecessor of x in some branch will be called *back-arcs*.

Proposition 2.1. *For any two branches* $X = (x_0, x_1, \dots x_a)$ *and* $Y = (y_0, y_1, \dots y_b)$ *starting in* $s = x_0 = y_0$*, the following holds:*

1) *For any two vertices* $x \in V(X) \backslash V(Y)$ *and* $y \in V(Y) \backslash V(X)$*, there is no* (x, y) *arc in* G*.*
2) X *and* Y *intersect in a single path starting in* s*.*
3) *For any back-arc* $(x_i, x_j),\ i > j$*, it holds that no* $x_k,\ i \geq k > j$ *can be the start point of a back-arc, and no* $x_l,\ i > l \geq j$ *can be the endpoint of a back-arc.*
4) *If* $x_i = y_i$ *is the last vertex in common of* X *and* Y*, and there is a back-arc* $(x_m, x_n), m > i > n$*, then there can be no back-arc* $(y_p, y_q), p > i > q$*.*

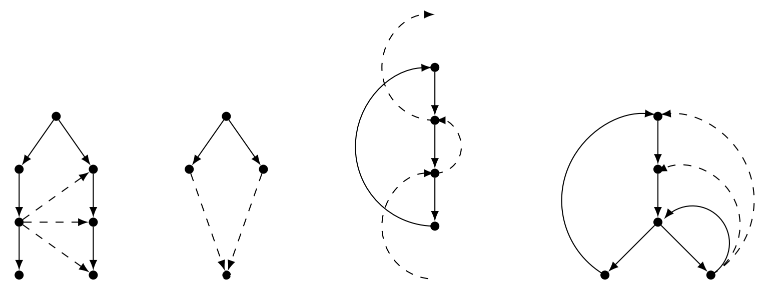


Fig. 1. Forbidden situations by 1), 2), 3) and 4) respectively

Proof. Points 1) and 2) follow trivially from the digraph having K-width 1.

For point 3), if $x_k,\ i \geq k > j$ were the starting point of a back-arc in G, then there would be two paths from x_k to x_{k-1}: One would use the back-arc starting at x_k and then follow down the branch, the other would follow down the branch, use the back-arc (x_i, x_j) and then follow down the branch to x_{k-1}. On the other hand, if $x_l,\ i > l \geq j$ were the endpoint of a back-arc in G, again there would be two paths from x_{l+1} to x_l: One would go down the branch and use the back-arc ending at x_l, the other would go down to x_i, use the back-arc (x_i, x_j) and then follow down to x_l.

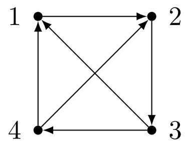
As for point 4), if there was a back-arc $(y_p, y_q), p \geq i > q$, there would be two paths from x_i to x_{i-1}: One going through X and using (x_m, x_n), the other going through Y and using (y_p, y_q). □

This means that our digraph is formed by a set of (non-disjoint) branches from a common source, which at some point disconnect from one another and each end up at separate leaves. Cycles only occur when a back-edge is present, and each back-edge corresponds to precisely one cycle, since there is only one path from the endpoint of the back-edge to its start. And, finally, any two cycles can only intersect in at most one vertex. From these facts, we get:

Theorem 2.2. *A digraph G with K-width 1 consisting of a single reachable fragment either contains a directed cycle of length 2 or 5, or can be orientedly coloured by 4 colours in polynomial time.*

Proof. It is a trivial observation that directed cycles of length 5 require 5 colours for oriented colouring. Cycles of length 2 can not be orientedly coloured at all. We prove the oriented colourability of digraphs without such cycles by providing an algorithm for colouring them using 4 colours:

We start by giving the following orientation of arcs in the target 4-vertex digraph H_4 (cf. Def. 1.1).



Notice that, given any cycle (of length other than 2 and 5) with fixed colouring at one single vertex, such a cycle always remains colourable by using H. For cycles of length 3 and 4, one can fill in the colours by using the 3-cycles and 4-cycle in H, and any number above 5 can be decomposed into a sum of threes and fours – which provides a suitable colouring for such cycles.

Our algorithm works as follows:

- First, find a source s of the reachable fragment by performing a reversed Depth-First search on G.
- Then, start a Depth-First search from s. The only reason for this Depth-First search is to identify back-arcs (we remember whether every arc is *normal* or a back-arc).
- Next, start a new, slightly modified Depth-First search from s. During the search, colour every traversed vertex in accordance with H until an incoming back-arc b is reached. The arc b corresponds to a cycle, and we must ensure that the colouring respects this cycle. So, go to the vertex x starting the

back-arc $b = (x, y)$ and then backtrack via normal arcs all the way up to the end y of b. If we had not avoided back-arcs, we could have ended up backtracking further down the Depth-First search tree. While backtracking, we record the length of the cycle so that we can colour accordingly. Note that even if this means that vertices can be visited more often than in an ordinary Depth-First search, in fact the number of visits only goes up by at most two. Once we reach the end y of the back-arc b (where we had originally started backtracking), start colouring in a manner respecting the length of the cycle and always choose the branch leading to the start of the back-arc.

– If we ever find a 2-cycle or 5-cycle, return false. Otherwise, using Proposition 2.1, we are left with a valid oriented 4-colouring when the algorithm ends.

□

We remark that the previous algorithm can be trivially adjusted to find an oriented 5-colouring for any digraph of K-width 1 consisting of a single reachable fragment, unless a directed 2-cycle is present.

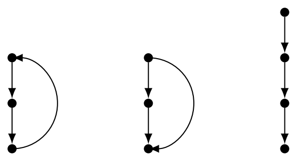
Corollary 2.3. *There is a polynomial algorithm that, given a digraph G of K-width 1 consisting of a single reachable fragment, determines the oriented chromatic number of G.*

Proof. (sketch) First check for directed cycles of length 2. If any are present, the digraph is not orientedly colourable. Otherwise run the algorithms for OCN_2, OCN_3 (always polynomial) and the introduced algorithms for OCN_4 and OCN_5. One of them must suceed. □

2.2 Digraphs of DAG-depth 2

Again, we start by introducing a few structural remarks about digraphs of DAG-depth 2. We then use these remarks to prove that all digraphs of DAG-depth 2 are either orientedly 3-colourable or contain a 2-cycle by providing an algorithm for computing a valid 3-colouring.

Remark 2.4. Digraphs of DAG-depth 2 contain none of the following subgraphs:



Proposition 2.5. *In a digraph of DAG-depth 2, for any two paths of length 2 $P = (a_1, a_2, a_3)$, $Q = (b_1, b_2, b_3)$, and any $v \in V(P) \cap V(Q)$, it holds $v = a_i = b_i$ for some $1 \le i \le 3$. Also, an arc (a_i, b_j) can only exist if $j > i$.*

Proof. (sketch) It is easy to check that all other possibilities result in a path of length 3, which is forbidden by Remark 2.4. □

Theorem 2.6. *Digraphs of DAG-depth 2 without 2-cycles are always orientedly 3-colourable. Furthermore, there exists a simple polynomial algorithm computing a 3-colouring for such digraphs.*

Proof. We utilize the fact that DAG-depth 2 implies no path of length higher than 2. $H = (V, E)$ will be defined as follows: $V = \{1, 2, 3\}$, $E = \{(1, 2),\ (1, 3),\ (2, 3)\}$. Start by colouring all paths of length 2 by colours 1, 2, 3 for the first, second and third vertices respectively. If a 2-cycle is found, return "false" and terminate. If there are no 2-cycles then this is a valid partial oriented colouring by H – paths will remain properly coloured even when they intersect or have arcs between them thanks to Proposition 2.5.

Now, iteratively run through all arcs with at least one endpoint in an uncoloured vertex. Note that all arcs from uncoloured vertices must start at sources and all arcs into uncoloured vertices must end at sinks, since otherwise an uncoloured 2-path would be present. Simply colour all the sinks by 3 and sources by 1, and the remaining disconnected vertices can be coloured arbitrarily. We end up with a valid oriented 3-colouring, assuming the digraph had DAG-depth 2 and no 2-cycles. □

3 Hardness proofs

3.1 Acyclic digraphs

The first NP-hardness proof in this article is for OCN_4 on the class of DAGs. Although the same result was claimed true already by the authors of [3], their paper only sketched a reduction gadget with a picture, and the sketch missed a key point — which would require further work and proving to ensure that no cycles are present in the resulting digraph. So, we decided to include our own reduction here, which is more straightforward and avoids the aforementioned problem. Another reason for proving the acyclic case first is that it serves as a motivation for the DAG-depth and K-width reduction (Theorem 3.4), and allows us to introduce tools which are useful for both of these cases. Please note that the target homomorphism digraph for our reduction is necessarily H of Fig. 3; the reasons will be made clear in the proof of Theorem 3.2.

Lemma 3.1. *Consider the gadget S from Fig. 2 and the target H from Fig. 3.*

1. *For any precolouring $(l_1, l_2, l_3) \mapsto \{T, F\}^3$ of S with the exception of (F, F, F), there is a homomorphism $S \to H$ extending it.*
2. *No homomorphism $S \to H$ maps (l_1, l_2, l_3) to (F, F, F).*

Proof. To explain one issue in advance, we remark that the same statement holds also for a "simpler" gadget S' which results from S by identifying s with s'. It is, however, that this S' has DAG-depth 4 while S has only 3, cf. Theorem 3.4.

1. As the proof, we show a table containing instructions on how to colour S for all combinations of T and F at l_1, l_2, l_3 (except for triple-F). For each l_i, the table contains the colours to be used in the sequence of vertices from s to l_i.

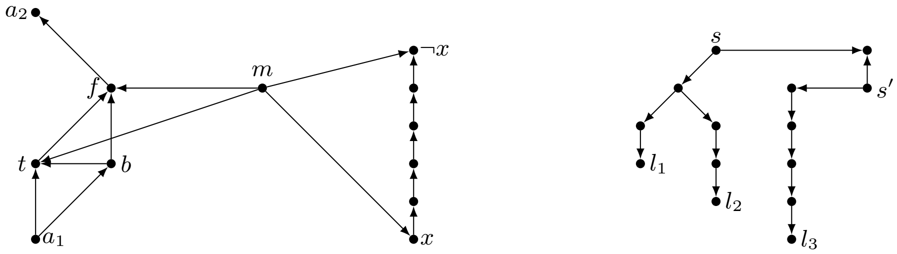


Fig. 2. Gadgets L to the left and S to the right

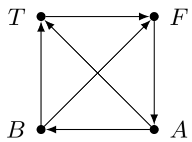


Fig. 3. The unique target colouring digraph H for our reductions

(l_1, l_2, l_3) Evaluation	l_1-branch	l_2-branch	l_3-branch
T,T,T	BFAT	BFABT	BFBTFABT
T,T,F	BFAT	BFABT	BFBTFABF
T,F,T	BFAT	BFABF	BFBTFABT
T,F,F	BFAT	BFABF	BFBTFABF
F,T,T	ABTF	ABFAT	ABABFABT
F,T,F	ABTF	ABFAT	ABABFABF
F,F,T	FABF	FABTF	FAFABFAT

2. Here we show another table, this time describing the relationship between the colour of s and possible l_i colourings. As one can see, all combinations are possible except for triple-F, thus concluding our proof.

Colour at s	Admissible l_1 col.	Admissible l_2 col.	Admissible l_3 col.
A	F	T	T,F
B	T	T,F	T,F
T	T	T,F	T,F
F	T,F	F	T

□

Theorem 3.2. *The OCN_4 problem is NP-complete even on the class of DAGs.*

Proof. We reduce 3-SAT to OCN_4 with the use of two gadgets, S for clauses and L for literals – see Fig. 2.

The reduction works as follows: Given a 3-SAT formula, for every literal we construct a copy of the gadget L consisting of vertices $\{a_1, f, b, t, a_2, m\}$ as depicted by the figure. For every clause we then construct a copy of the gadget

S, where l_1, l_2 and l_3 are identified with the vertices we have created for the appropriate literals or their negations which appear in that particular clause.

Assume we have a 3-SAT evaluation. Then we must show that it is possible to provide a valid oriented 4-colouring of this digraph. Let us name the colours A, B, F, T (Fig. 3). The vertices a_1, a_2, b, t, f will be coloured in accordance to their names, m will be coloured by B and every x and $\neg x$ will be coloured by T and F depending on whether the literal is true or false in the 3-SAT evaluation — if it is true, then the vertex marked x in the figure will be coloured by T and $\neg x$ by F, and otherwise the colours will be switched. The T–F and F–T paths of length 5 are 4-colourable by the sequences (T, A, F, T, A, F) and (F, T, A, B, F, T) respectively. All that remains now is to orientedly colour all S gadgets. Notice that the arcs between colours allow us to use H as the orientation of edges for the colouring. So, the colourability of S is certified by Lemma 3.1(1).

On the other hand, assume we are given an oriented 4-colouring of such a digraph and want to find a valid 3-SAT evaluation. Vertices a_1, t, b, f all need to have distinct colours, and without loss of generality we can again name these colours A, T, B, F. The arcs between these four vertices in L, and the existence of an arc (f, a_2) easily leave the homomorphism image H from Fig 3 as the only admissible variant of colouring. Notice that a_2 and m must then be coloured by A, B respectively.

Now all the vertices x and $\neg x$ have to be coloured by either T or F. Our goal is to have T represent "true" and F represent "false", but for that to make sense x and $\neg x$ may not be both coloured by the same colour — that is where the interconnecting 5-path is used. It is easy to verify that a 5-path starting with T (or F) can not end with T (or F). So right now, we are given an evaluation of literals in the 3-SAT formula by the colouring: If the appropriate literal is coloured by T in x, evaluate it as "true", otherwise evaluate it as "false".

But what certifies that such an evaluation of all literals satisfies the 3-SAT formula? Here the specifics of S come into play. As proved in Lemma 3.1(2), S allows any combination of the colours T, F at l_1, l_2, l_3 except for F, F, F.

So, to recapitulate, it is possible to straightforwardly translate an oriented 4-colouring of such a digraph to the evaluation of the 3-SAT formula. The digraph structure guarantees that the evaluation will be sound (i.e. every literal is "true" iff its negation is "false") and that the evaluation will satisfy the formula. This concludes our proof. □

Remark 3.3. The digraph instances of the OCN_4 problem in Theorem 3.2 are of K-width 3 and DAG-depth 5.

3.2 Digraphs of DAG-depth 3 and K-width 1

Here we prove NP-hardness of the OCN_4 problem on another very restricted (c.f. Remark 3.3) digraph class — those that have simultaneously K-width 1 and DAG-depth 3. Although the constructed instances are not acyclic, all the values of traditional directed width parameters such as directed tree-width [8], DAG-width [1, 12], Kelly-width [7] and cycle rank [4] remain bounded and very small. To recapitulate, these bounds on K-width and DAG-depth mean that there exists at most one path between any two vertices and that the robber can always be caught by cops in 3 moves in the cops-and-robber game of Theorem 1.4. This is just a little less restrictive than in Theorems 2.2 and 2.6.

Theorem 3.4. *The* OCN_4 *problem is NP-complete even on the class of digraphs with K-width 1 and DAG-depth 3.*

Proof. We prove the theorem by a reduction very similar to the case of Theorem 3.2. Notice that if the gadget S is applied on literals which are sinks in the graph, then the conditions on DAG-depth and K-width hold. We will however use a different variable gadget L_1 (Fig. 4) of smaller K-width and DAG-depth.

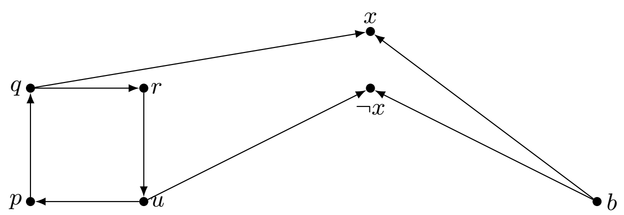


Fig. 4. Gadget L_1

Then, for every literal in the 3-SAT formula we create a separate copy of L_1, for each clause a separate copy of S and merge the copies of vertices x and $\neg x$ with l_1, l_2, l_3 of S in accordance with clauses of the formula. It is easy to verify that such a digraph will have K-width 1 and DAG-depth 3: Since the gadget S only intersects with other gadgets in copies of l_i and all l_i are sinks, K-width can only be increased above 1 by L_1. However, the K-width of L_1 is also 1. As for DAG-depth, the robber can be caught in 3 moves regardless of his starting point in S or L_1. Specifically, if the robber is in L_1 and starts in n, x or $\neg x$, then he can be caught trivially. Otherwise, place cops on q, u and then he is caught by the third one. Catching the robber in S is also simple and we leave the details to the reader as an exercise.

So, assume we have a 3-SAT evaluation. We will use the same H as in the previous reduction. If the literal is true in the evaluation, then we colour (p, q, r, u, b) as (F, A, B, T, B) and $x, \neg x$ as T and F. If it is false, we colour (p, q, r, u, b) as (B, T, F, A, B) and $x, \neg x$ as F and T. Finally, colour S by Lemma 3.1(1).

On the other hand, assume we have a valid oriented 4-colouring of such a digraph. Choose any 4-cycle C in a copy of L_1. C must be coloured by 4 distinct colours, without loss of generality let's say p, q, r, u are coloured by some colours P, Q, R, U respectively. This forces a 4-cycle in H and at this moment the only two orientations of arcs which remain undetermined in H are $\{P, R\}$ and $\{Q, U\}$.

If x and $\neg x$ were to be coloured by R and P (i.e. without using a "cross arc" in the cycle), we would not be able to assign any colour to B. So, only two possibilities can occur:

1. $\neg x$ coloured by P and x coloured by U. Then b must be coloured by R and we obtain $H = H_1$ (Fig. 5 and 3). By identifying $(R, U, P, Q) = (B, T, F, A)$ we see that this is isomorhpic to H as before.
2. $\neg x$ coloured by Q and x coloured by R. Then b must be coloured by P and we obtain $H = H_2$ (Fig. 5 and 3). By identifying $(R, U, P, Q) = (F, A, B, T)$ we again see that this is isomorhpic to H.

Fig. 5. The colour digraphs H_1 and H_2 respectively.

So both admissible cases lead to the same (up to isomorphism) and only possible orientation of arcs in H. Since the aforementioned holds separately for every copy of L_1, each copy of x and $\neg x$ must be coloured only by T or F and never by the same colour as the other. Lemma 3.1(2) already certifies that under these conditions S forces every clause in the 3-SAT formula to hold true, concluding our proof. □

4 Conclusions

There are two possible interpretations of the results of the article. One is optimistic: there are some positive results and the problem can be algorithmically solved for DAG-depth 2 and special cases of K-width 1. This is a step forward, since no such positive results exist for traditional directed width parameters. It also remains an open question whether the algorithm for K-width 1 could be extended to a parameterized FPT algorithm with respect to the number of sources in the digraph.

In light of OCN_4 remaining NP-hard even after such severe restriction of the class of input graphs, we believe that new width parameters are needed for tackling this and perhaps other hard problems on digraphs. The recently introduced bi-rank-width measure [9], a natural directed extension of rank-width, could be a promising candidate. However, bi-rank-width is conceptually quite far away from the aforementioned width measures—these are mostly inspired by cops-and-robber games and undirected tree-width (see e.g. [13])—while bi-rank-width is close to the undirected clique-width and rank-width measures.

A strong positive aspect of bi-rank-width is that it performs much better [6] than the other aforementioned directed width measures with respect to an existence of polynomial algorithms for hard problems on digraphs (such as Directed Steiner Tree or Directed Feedback Vertex Set). Particularly, the OCN_c problem can be solved in FPT time on digraphs of bounded bi-rank-width for every fixed c [6].

There still are many unanswered questions though. One such question is the parameterized complexity of computing the oriented chromatic number (the optimization variant OCN) on digraphs of bounded bi-rank-width, as the algorithm used for computing the ordinary chromatic number on graphs of bounded rank-width [5] can not be straightforwardly extended to oriented colourings.

The major question in this context seems to be the following: Can one find a more restrictive directed width measure which is conceptually related to tree-width (and to cops and robber games), and which at the same time allows to solve the OCN_c problem efficiently?

References

1. D. Berwanger, A. Dawar, P. Hunter, and S. Kreutzer: DAG-width and parity games. In *STACS'06*, volume 3884 of LNCS, pages 542–536. Springer, 2006.
2. B. Courcelle: The monadic second order-logic of graphs VI : on several representations of graphs by relational structures. *Discrete Appl. Math.*, 54:117-149, 1994.
3. J.-F. Culus and M. Demange: Oriented coloring: Complexity and approximation. In *SOFSEM'06*, volume 3831 of LNCS, pages 226–236. Springer, 2006.
4. L. Eggan: Transition graphs and the star-height of regular events. *Michigan Mathematical Journal*, 10(4):385–397, 1963.
5. R. Ganian and P. Hliněný: Better Polynomial Algorithms on Graphs of Bounded Rank-width. Extended abstract in *IWOCA'09*, LNCS to appear. Springer.
6. R. Ganian, P. Hliněný, J. Kneis, A. Langer, J. Obdržálek and P. Rossmanith: On Digraph Width Measures in Parameterized Algorithmics. Extended abstract in *IWPEC'09*, LNCS to appear. Springer.
7. P. Hunter and S. Kreutzer: Digraph measures: Kelly decompositions, games, and orderings. *Theor. Comput. Sci.*, 399(3):206–219, 2008.
8. T. Johnson, N. Robertson, P. D. Seymour and R. Thomas: Directed treewidth. *J. Combin. Theory Ser. B*, 82(1):138–154, 2001.
9. M. Kanté: The rank-width of directed graphs. In arXiv:0709.1433v3, 2008.
10. J. Nešetřil and P. Ossona de Mendez: Tree-depth, subgraph coloring and homomorphism bounds. *European J. Combin.*, 27(6):1024–1041, 2006.
11. J. Nešetřil and A. Raspaud: Colored Homomorphisms of colored mixed graphs. *J. Combin. Theory Ser. B*, 80(1):147–155, 2000.
12. J. Obdržálek: DAG-width: connectivity measure for directed graphs. In *SODA'06*, pages 814–821. ACM-SIAM, 2006.
13. Robertson, N. and Seymour, P.: Graph minors X. Obstructions to tree-decomposition. *J. Combin. Theory Ser. B*, 52(2):153–190, 1991.
14. É. Sopena: Oriented Graph Coloring. *Discrete Math.*, 229:359–369, 2001.