The Parameterized Complexity of Oriented Colouring

Robert Ganian*

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
ganian@mail.muni.cz

Abstract The oriented colouring problem is intuitive and related to undirected colouring, yet remains NP-hard even on digraph classes with bounded traditional directed width measures. Recently we have also proved that it remains NP-hard in otherwise severely restricted digraph classes. However, unlike most other problems on directed graphs, the oriented colouring problem is not directly transferable to undirected graphs. In the article we look at the parameterized complexity of computing the oriented colouring of digraphs with bounded undirected width parameters, whereas the parameters "forget" about the orientations of arcs and treat them as edges. Specifically, we provide new complexity results for computing oriented colouring on digraphs of bounded undirected rank-width and a new algorithm for this problem on digraphs of bounded undirected tree-width.

1 Introduction

The study of undirected 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]. Oriented colouring has been studied by several authors, see e.g. the work of Nešetřil and Raspaud [11] or the survey by Sopena [13].

Similarly to undirected colouring, computing the oriented chromatic number (OCN in brief) 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, even deciding oriented colourability by 4 colours (also referred to as OCN_4) remains NP-hard on directed acyclic graphs (further referred to as DAGs) [3]. And since the vast majority of digraph parameters have low, fixed values on DAGs, this alone means that they would not be useful for computing OCN.

* supported by the research grants GAČR 201/09/J021 and MUNI/E/0059/2009

Mathematical and Engineering Methods in Computer Science (MEMICS), Znojmo, Czech Republic, 2009.
Petr Hliněný, Vashek Matyáš, Tomáš Vojnar (Eds.)
Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany
Digital Object Identifier: 10.4230/DROPS.MEMICS.2009.2350

Bi-rank-width (first introduced by Kanté [9]), the digraph equivalent to rank-width, is an exception since it can have high values for DAGs – we have recently shown that deciding OCN_k is in FPT on digraphs of bounded bi-rank-width [7] (FPT stands for fixed parameter tractable, meaning that the time complexity is not only polynomial for any fixed value of the parameter, but also the degree of the polynomial does not depend on the parameter). Unfortunately, the case of computing OCN is worse than for OCN_k: there is no known parameterized algorithm for computing OCN utilizing a digraph parameter. But what about undirected graph parameters?

Most hard problems on directed graphs can be directly translated to undirected graphs. Consider c-Path, Hamiltonian Path, Hamiltonian Cycle, Directed Steiner Tree, Directed Dominating Set, Directed Feedback Vertex Set – all of these directed problems have also been extensively studied on undirected graphs. OCN is different; its definition only makes sense on digraphs. Nevertheless, we show that it is still possible to naturally and intuitively use well-known undirected width parameters for computing OCN on directed graphs. In the article we present new complexity results and a new parameterized algorithm for OCN on digraphs restricted by undirected width parameters.

2 Preliminariess

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 directed acyclic graph.

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 $(f(a), f(b)) \in E(H)$. The k-oriented chromatic number (OCN_k) problem is then defined as follows: Given a digraph G, is there a homomorphism from G to H, where H is some (irreflexive antisymmetric) orientation of edges of the complete graph on k vertices? OCN is the optimization problem of finding the minimum k for a given digraph 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 such vertices with the same colour can never have an arc between them, 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 will need the notions of tree-width and rank-width – both being very successful width parameters of undirected graphs.

Tree-width: A *tree decomposition* of an undirected graph $G = (V, E)$ is a tree T together with a collection of subsets $T_x \subseteq V$ (called bags) labeled by the vertices x of T such that $\bigcup_{x \in T} T_x = V$ and (1) and (2) below hold:
(1): For every edge uv of G, there is some x such that $\{u, v\} \subseteq T_x$.
(2): (Interpolation Property) If y is a vertex on the unique path in T from x to z, then $T_x \cap T_z \subseteq T_y$.

The width of a tree decomposition is the maximum value of $|T_x| - 1$ taken over all the vertices x of the tree T of the decomposition. We then say that a graph G has *tree-width* k if G has a tree-decomposition of width k.

Branch-width and rank-width: A set function $f : 2^M \to \mathbb{Z}$ is called *symmetric* if $f(X) = f(M \setminus X)$ for all $X \subseteq M$. A tree is *subcubic* if all its nodes have degree at most 3. For a symmetric function $f : 2^M \to \mathbb{Z}$ on a finite set M, the branch-width of f is defined as follows.

A *branch-decomposition* of f is a pair (T, μ) of a subcubic tree T and a bijective function $\mu : M \to \{t : t \text{ is a leaf of } T\}$. For an edge e of T, the connected components of $T \setminus e$ induce a bipartition (X, Y) of the set of leaves of T. The *width* of an edge e of a branch-decomposition (T, μ) is $f(\mu^{-1}(X))$. The *width* of (T, μ) is the maximum width over all edges of T. The *branch-width* of f is the minimum of the width of all branch-decompositions of f. (If $|M| \leq 1$, then we define the branch-width of f as $f(\emptyset)$.)

Natural applications of this definition include not only rank-width (introduced by Oum [10]) but also its directed counterpart bi-rank-width (Kanté, [9]) and the branch-width of graphs (Robertson and Seymour, [12]). In the case of rank-width we consider the vertex set $V(G) = M$ of a graph G as the ground set.

For a graph G, let $\boldsymbol{A}_G[U, W]$ be the bipartite adjacency matrix of a bipartition (U, W) of the vertex set $V(G)$ defined over the two-element field GF(2) as follows: the entry $a_{u,w}$, $u \in U$ and $w \in W$, of $\boldsymbol{A}_G[U, W]$ is 1 if and only if uw is an edge of G. The *cut-rank* function $\rho_G(U) = \rho_G(W)$ then equals the rank of $\boldsymbol{A}_G[U, W]$ over GF(2). A *rank-decomposition* and *rank-width* of a graph G is the branch-decomposition and branch-width of the cut-rank function ρ_G of G on $M = V(G)$, respectively.

Another notion we will need later on is bi-rank-width. For a digraph G, let $\boldsymbol{A}_G[U, W]^+$ ($\boldsymbol{A}_G[U, W]^-$) be the bipartite adjacency matrix of a bipartition (U, W) of the vertex set $V(G)$ defined over the two-element field GF(2) as follows: the entry $a_{u,w}$, $u \in U$ and $w \in W$, of $\boldsymbol{A}_G[U,W]^+$ ($\boldsymbol{A}_G[U,W]^-$) is 1 if and only if $(u,w) \in E(G)$ ($(w,u) \in E(G)$). The *bi-cutrank* function of G is defined as the sum of the ranks of these two matrices $brk_G(X) = \text{rank}(\boldsymbol{A}_G[U,W]^+) + \text{rank}(\boldsymbol{A}_G[U,W]^-)$ over the binary field $GF(2)$. A *bi-rank-decomposition* and *bi-rank-width* of a graph G is then the branch-decomposition and branch-width of this bi-cutrank function brk_G.

We have mentioned that for the purposes of this paper, we will apply undirected width measures on directed graphs. So, unless otherwise specified, by tree-width and rank-width we will mean the undirected variants of these measures, even when speaking of digraphs. Formally, given a digraph $G = (V,E)$, we consider an undirected graph $G' = (V(G), E(G'))$ where $E(G') = \{\{a,b\} : (a,b) \in E(G)\}$, and by restricting G to bounded tree-width or rank-width we actually restrict the values of these parameters on G'. Informally this means that we "forget" about the orientations of arcs when computing tree-width and rank-width.

3 OCN on digraphs of bounded rank-width

Although rank-width is not as restrictive as tree-width, in a certain sense bounding rank-width means limiting the complexity of the structure of the graph, and this can be exploited to design powerful parameterized algorithms. For example, computing the "usual" undirected chromatic number can be done in polynomial time (XP to be precise) on graphs of bounded rank-width (see [6]), and deciding colourability can even be done in FPT time ([5]). Unfortunately, despite its successes with undirected colouring, it turns out that rank-width is not useful for computing the more complicated OCN – even on digraphs of bounded rank-width the problem is DET-hard, i.e. as hard as general graph isomorphism. DET is the class of decision problems reducible in logarithmic space to the problem of computing the determinant of an n-by-n matrix of n-bit integers.

Theorem 3.1. *Computing the oriented chromatic number of digraphs is DET-hard even when restricted to digraphs of bounded undirected rank-width.*

Proof. We employ a reduction from a problem involving tournaments. A tournament is, simply put, a complete graph with arbitrary orientation of edges – more precisely, a digraph with precisely one arc between every pair of distinct vertices. The isomorphism of two tournaments has recently been proved to be DET-hard by Wagner [14].

The reduction works as follows: Given two tournaments G_1, G_2 with n vertices each, we construct G as the disjoint union of G_1 and G_2. Note that the rank-width of G is 1, yet we could still solve the problem of isomorphism of G_1 and G_2 by solving OCN on G.

First, assume that the OCN of G is n. Notice that each of G_i contains exactly n vertices and no colour can appear more than once in each G_i. We know that there exists a colouring of G which uniquely identifies each vertex of G_1 with a vertex of G_2 of the same colour. What remains is to argue that such a bijection $f : G_1 \mapsto G_2$ is an isomorphism. Consider any arc $(a, b) \in E(G_1)$. We need to show $(f(a), f(b)) \in E(G_2)$, but by the definition of tournaments either $(f(a), f(b))$ or $(f(b), f(a))$ must be present, and the latter would contradict the oriented colouring of G.

Now assume that G_1 is isomorphic to G_2 by an isomorphism $f : G_1 \mapsto G_2$. We need to show that G can be coloured by n colours. By definition this means proving that there exists a homomorphism from G to some tournament H on n vertices. Choose $H \cong G_2$, $h : G_2 \mapsto H$ being the isomorphism, and consider the following homomorphism: all $v \in G_2$ map to $h(v)$ and all $v \in G_1$ map to $h(f(v))$. Any arc $(a, b) \in E(G)$ must now be also present in $E(H)$, proving that G is orientedly n-colourable. This concludes our proof.

∎

4 *OCN* on digraphs of bounded tree-width

The introduction of tree-width was a breakthrough in the field, and it still remains the most popular graph parameter to this day. Tree-width exploits the fact that almost every problem is easy on the class of trees, and parameterizes the graph by how "tree-like" it is. Powerful tools now exist for designing algorithms on graphs of bounded tree-width, however these are not capable of handling OCN. Nevertheless it still turns out that it is possible to compute OCN on digraphs of bounded tree-width in FPT time. First, we will need a few known results:

Corollary 4.1 ([8], 6.45). *Graphs with bounded degree, or tree-width, or genus have bounded oriented chromatic number.*

More precisely, the authors of [1] have proved that the "acyclic chromatic number" of graphs with tree-width t is at most 2^{t+1}. Hell and

Nešetřil in [8] obtained a bound on the oriented chromatic number of at most $k \cdot 2^{k-1}$, with k as the acyclic chromatic number of the graph. So altogether we get a bound on OCN of $b(t) = 2^{t+1} \cdot 2^{2^{t+1}-1}$ on digraphs of tree-width at most t.

Now, all that remains is to find an FPT algorithm on tree-width which would decide OCN_k. Unfortunately, no such direct algorithm is known, but we have recently developed an FPT algorithm doing just that running on the bi-rank-width of digraphs [7]. What we need to do now is prove that tree-width also bounds bi-rank-width, allowing us to use the aforementioned algorithm. This theorem is of independent interest – the proof that tree-width bounds rank-width does not immediately translate to bi-rank-width, and no result on the relationship of these two parameters has been previously known.

Theorem 4.2. *The bi-rank-width of a digraph G with tree-width t is at most $2 \cdot (t+1)$.*

Proof. We start by normalizing the tree-decomposition of G in a similar way as in [4, Theorem 6.72]:

1. First, we make the decomposition sub-cubic, i.e. bounding the degrees of nodes to 3. This is accomplished by duplicating the nodes of higher degree and inserting them as subdivisions of incident edges. Thus, nodes with high degrees will be duplicated several times.
2. Next, we make all the sets in the tree decomposition uniform of size $t+1$ by adding new vertices to the node if necessary. This can be accomplished by adding vertices from neighbours.
3. We ensure that neighbouring sets differ by at most one. This can be achieved by adding interpolating nodes where necessary.
4. Now we make sure all sets of leaves have bags of size 1. This is done by adding a path to each former leaf and reducing the size of each consecutive bag on the path by one, omitting a random vertex. Notice that this cannot break the interpolation property. The sets on these paths will be smaller than $t+1$, and will be exempt from step 2.
5. Finally, for each node of degree 3 in the decomposition, we create an *attachment node* with the same set by subdividing any of its incident edges.

Now we will transform this tree-decomposition into a bi-rank-decomposition, and argue that such a bi-rank-decomposition has bounded bi-rank-width. First, we perform a Depth-first search starting from any leaf of the tree-decomposition. Every time we come across a new, previously unvisited vertex in a bag at some node, we add it as a pendant vertex to the node if the node has degree at most 2. The decomposition must remain subcubic, so if the degree is already 3, we add it to the node's attachment vertex. In this way, all the vertices previously in bags will be added to the decomposition as leafs in the same order as they appeared in bags.

What remains is to argue that such a bi-rank-decomposition truly has bounded bi-rank-width. Consider any edge of the decomposition. The edges incident to leaves of the decomposition can have a bi-rank-width of at most 2, due to the matrices having a single row or column. All other edges were already present in the tree-decomposition, and due to the nature of tree-decompositions (particularly the interpolation property), only at most $t+1$ vertices could occur in bags on both "sides" of the edge. This means that of all the vertices in the rows of $\boldsymbol{A}_G^+$ (those on one "side" of the edge), only at most $t+1$ could have ever met with the vertices in the columns of $\boldsymbol{A}_G^+$ (i.e. those on the other "side" of the edge) in a bag – and since every edge must be present in some bag, we immediately get that all rows or columns other than those of these $t+1$ vertices will only contain zeros. The same of course holds for the other matrix $\boldsymbol{A}_G^-$. Thus $\mathrm{rank}(\boldsymbol{A}_G^+) + \mathrm{rank}(\boldsymbol{A}_G^-) \leq 2 \cdot (t+1)$.

∎

Recall that on digraphs of tree-width at most t, OCN is bounded by $b(t) = 2^{t+1} \cdot 2^{2^{t+1}-1}$. On the other hand, the algorithm for OCN_k on digraphs of bi-rank-width at most r runs in time $O(2^{k^2} \cdot (2^{kr(r+1)/2} \cdot kr^3 \cdot |V(G)|))$ – the runtime is not written explicitly in [7], however it is based on first considering all orientations of arcs of the tournament on k vertices (2^{k^2} possibilities), and for each it is possible to straightforwardly apply our algorithm for deciding unoriented k-colourability on rank-width [6] which runs in time $(2^{kr(r+1)/2} \cdot kr^3 \cdot |V(G)|)$.

So, to compute OCN, it suffices to simply run through all $b(t)$ admissible colours and for each colour compute OCN_k by the bi-rank-width algorithm. The number of tested colours can be trivially improved to $\log b(t)$ in the same way as one can find a number between 1 and k by using only $O(\log k)$ greater/less-or-equal queries. Altogether, we get:

Corollary 4.3. *OCN can be computed on digraphs of tree-width at most t in time* $O\big(\log b(t) \cdot 2^{b(t)^2} \cdot 2^{b(t)\cdot(t+1)(2t+3)} \cdot b(t)(2t+2)^3 \cdot |V(G)|\big)$.

5 Conclusion

In the article we have introduced new positive as well as negative results for computing the oriented chromatic number of digraphs. The positive result on tree-width is of particular interest. This is the first polynomial parameterized algorithm for OCN. Future research should focus on utilizing undirected width measures on other digraph problems, especially those which do not translate directly to undirected graphs. Another direction for future research would be studying OCN on digraphs of bounded bi-rank-width. If it indeed turns out to be hard, would it be possible to find a new powerful directed measure capable of dealing with such hard problems as OCN?

References

1. Michael O. Albertson, Glenn G. Chappell, H. A. Kierstead, Andr Kndgen, Radhika Ramamurthi: Coloring with no 2-colored P4's. *Electron. J. Combin.*, 11: paper R26, 2004.
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 226236. Springer, 2006.
4. Downey, R. and Fellows, M.: Parameterized complexity. Springer, 1999.
5. Ganian, R. and Hliněný, P.: On Parse Trees and Myhill–Nerode–type Tools for handling Graphs of Bounded Rank-width. Manuscript, to appear. Extended abstract in *IWOCA08*, LNCS. Springer, 2008.
6. R. Ganian and P. Hliněný: Better Polynomial Algorithms on Graphs of Bounded Rank-width. Manuscript, to appear. Extended abstract in *IWOCA09*, LNCS. Springer, 2009.
7. R. Ganian, P. Hliněný, J. Kneis, A. Langer, J. Obdržálek and P. Rossmanith: On Digraph Width Measures in Parameterized Algorithmics. In *IWPEC2009*, to appear.
8. P. Hell and J. Nešetřil: Graphs and Homomorphisms. Oxford University Press, 2004.
9. Kanté, M.: The rank-width of directed graphs. arXiv:0709.1433v3 (2008).
10. Oum, S.: Rank-width and vertex-minors. *J. Combin. Theory Ser. B* **95**(1) (2005) 79–100.
11. J. Nešetřil and A. Raspaud: Colored Homomorphisms of colored mixed graphs. *Journal of Combinatorial Theory*, Series B, 80(1):147-155, 2000.
12. Robertson, N. and Seymour, P.: Graph minors. X. Obstructions to tree-decomposition. *J. Combin. Theory Ser. B* **52**(2) (1991) 153–190.
13. É. Sopena: Oriented Graph Coloring. *Discrete Math.*, 229:359-369, 2001.
14. Wagner, F.: Hardness Results for Tournament Isomorphism and Automorphism. Mathematical foundations of computer science, 572-583, 2007.