The Parameterized Complexity of Oriented Colouring

Robert Ganian

Masaryk University, Brno

MEMICS 2009
The problem: Oriented Chromatic Number (OCN)

- Introduced by Courcelle, studied by Nešetřil, Raspaud, Sopena and others

- Simple definition – We colour the vertices in a digraph so that:
  - neighbours always have different colours, and
  - all arcs between two colours have the same orientation

- Alternate definition – *homomorphism* of a digraph into a tournament

- Two related problems:
  - Decide oriented colourability by fixed number of colours *(OCN)*
  - Compute minimal number of colours *(OCN)*

- Applications in mobile networks
Oriented colouring examples
Oriented colouring examples
Oriented colouring examples

Robert Ganian
Oriented colouring examples
Oriented colouring examples
Oriented colouring examples

Robert Ganian
Masaryk University, Brno
The tools: A crash course on Parameterized Algorithms

- Many interesting graph problems NP-hard on general graphs
- Input graphs in real-life applications usually specific in some way – often bounded structural parameter (bounded “width”)
- This can be exploited by designing parameterized algorithms with a parameter $t$:
  - **FPT** algorithms are those running in time $O(poly(n) \cdot f(t))$
  - **XP** algorithms are those running in time $O(poly(n)^{f(t)})$
OCN and directed width parameters

- There exists a wide range of studied directed width parameters – D-width, DAG-width, Directed tree-width, Cycle rank, Kelly-width etc.
- Common feature: low, fixed value on DAGs (directed acyclic graphs)
- Even deciding \(\text{OCN}_k\) is NP-hard on DAGs
  \[ \rightarrow \text{“classical” directed width parameters useless for OCN} \]
- Bi-rank-width, a promising new directed width parameter, can decide \(\text{OCN}_k\) in FPT but computing OCN is open
- But we want to compute OCN...
OCN and directed width parameters

- There exists a wide range of studied directed width parameters – D-width, DAG-width, Directed tree-width, Cycle rank, Kelly-width etc.
- Common feature: low, fixed value on DAGs (directed acyclic graphs)
- **Even deciding OCN}_k is NP-hard on DAGs**
  \[ \implies \text{“classical” directed width parameters useless for OCN} \]
- Bi-rank-width, a promising new directed width parameter, can decide OCN}_k in FPT but computing OCN is open
- But we want to compute OCN...
An undirected approach

- Most problems are studied separately on digraphs and undirected graphs
- However OCN only makes sense on digraphs
- Since directed width measures do not help, why not try using the undirected width measures of the underlying graph ("forgetting" edge orientations)?
Some successful undirected width parameters

**Tree-width**
- Robertson and Seymour (1984)
- Measures how close a graph is to being a tree
- Quite restrictive, but many NP-hard problems become FPT on graphs of bounded tree-width.

**Rank-width**
- Oum and Seymour (2004)
- Related to clique-width, but much better for algorithms
- Less restrictive than tree-width, but not as powerful – however still capable of solving many NP-hard problems in FPT or XP time.
Some successful undirected width parameters

**Tree-width**
- Robertson and Seymour (1984)
- Measures how close a graph is to being a tree
- Quite restrictive, but many NP-hard problems become FPT on graphs of bounded tree-width.

**Rank-width**
- Oum and Seymour (2004)
- Related to clique-width, but much better for algorithms
- Less restrictive than tree-width, but not as powerful – however still capable of solving many NP-hard problems in FPT or XP time.
Tree-width and rank-width

Similar scheme:

1. The tree/rank-width of $G$ is the minimum width of a \textit{tree/rank-decomposition} of $G$.

2. The width of a tree/rank-decomposition is the maximum width of “something” in the decomposition.

These decompositions are then used for algorithms as well.
An introduction to rank-width

Robert Ganian
Masaryk University, Brno
Unfortunately rank-width does not help with OCN:

**Theorem**

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

This is in contrast with undirected colouring – computable in XP on rank-width.
OCN on digraphs of bounded rank-width

Proof is based on a reduction from the isomorphism of tournaments (orientations of complete graphs)

- Tournament example:
- Tournaments have rank-width 1 (their underlying graph is a complete graph)
- Tournament isomorphism proved to be DET-hard by Wagner (2007)
- Tournament isomorphism can be solved by computing OCN
OCN on digraphs of bounded rank-width

Sketch of reduction:

1. Create $G$ as disjoint union of two possibly-isomorphic tournaments (each of size $\frac{|V(G)|}{2}$).
2. Compute $OCN(G)$.
3. If $OCN(G) = \frac{V(G)}{2}$ then the tournaments are isomorphic, otherwise they are not.
OCN on digraphs of bounded tree-width

**Theorem**

*OCN can be computed on digraphs of tree-width at most t in FPT time*

- First parameterized algorithm for computing OCN
- Can be divided into three parts:
  - Undirected tree-width bounds OCN
  - Undirected tree-width bounds bi-rank-width
  - A FPT algorithm for deciding $OCN_k$ on bi-rank-width
OCN on digraphs of bounded tree-width

Bounding OCN

- Result of Hell and Nešetřil (2004) – does not give exact bounds
- Precise bounds can be obtained by combining results of Albertson, Chappell, Kierstead, Kndgen and Ramamurthi (2004) with results of Hell and Nešetřil
- Two-step process – Acyclic chromatic number
It is well known that tree-width bounds rank-width, but rank-width doesn’t bound bi-rank-width.

New proof needed.

Core idea: Transform the tree-decomposition in a special way into a bi-rank-decomposition of the digraph.

Tree- and bi-rank-decompositions are not very similar – many technicalities.
OCN on digraphs of bounded tree-width

Bounding bi-rank-width

- Rank-decomposition – each vertex stored in exactly one leaf
- Tree-decomposition – several vertices stored in nodes (bags) of decomposition, can repeat. All edges contained in some bag.

\[ \text{bag}_1 \rightarrow \text{bag}_2 \]

Bound on bi-rank-width comes from the nature of the tree-decomposition if transformation is done correctly.
OCN on digraphs of bounded tree-width

Bounding bi-rank-width

- Rank-decomposition – each vertex stored in exactly one leaf
- Tree-decomposition – several vertices stored in nodes (bags) of decomposition, can repeat. All edges contained in some bag.

\[ bag_1 \longrightarrow bag_2 \]

Bound on bi-rank-width comes from the nature of the tree-decomposition if transformation is done correctly.
OCN on digraphs of bounded tree-width

Deciding $OCN_k$ on bi-rank-width

- Algorithm by Ganian, Hliněný, Kneis, Langer, Obdržálek, Rossmanith (2009)
Conclusions

△ New approach to parameterized algorithms for problems restricted to directed graphs
△ First parameterized algorithm for OCN
△ Proof of tree-width bounding bi-rank-width is of independent interest
  ★ The FPT algorithm could be improved significantly with better bounds on OCN when tree-width is bounded
  ★ Computing OCN is still open on bi-rank-width
Thank you for your attention
$O(2^{b(t)^2+b(t)\cdot(t+1)(2t+3)} \cdot b(t)(2t + 2)^3 \cdot \log b(t) \cdot | V(G)|)$, where $b(t) = 2^{t+1} \cdot 2^{2^{t+1}-1}$