Elimination distance to bounded degree on planar graphs

Alexander Lindermayr\textsuperscript{1}, Sebastian Siebertz\textsuperscript{1} and Alexandre Vigny\textsuperscript{1}

\textsuperscript{1}Universität Bremen, Germany

MFCS, August 24-28, 2020
Outline

How trivial can a graph be?

Elimination distance
  Definition and observations

Algorithm
  Tools used
  Proof’s sketch

Conclusion & Futur works
Trivial graphs classes?

Graph problems: Hamilton path, FO / MSO model checking, graph isomorphism,…

Graph classes: edgeless graphs, planar graphs, trees,…

Some hard problems are simple for some graph classes.
Running example

Graph isomorphism problem & Graphs with bounded degree

(parametrized by the degree $d$)

Input: Graphs $G, H$.

Goal: Are $G$ and $H$ isomorphic?

Running time: $O(|G|^f(d))$

Parametrized by $d$: the degree of $G$ and $H$. 
With a small twist

Two graphs with only one node of degree $> d$.

The algorithm can be adapted.

Step 1: Color the neighborhood of the high degree vertex.
Step 2: Remove the high degree vertex.
Step 3: Use the previous algorithm.
Deletion distance

J. Guo, F. Hüffner, R. Niedermeier (2004): Distance from Triviality.

$G$ at deletion distance $k$ from $\mathcal{C}$ iff $G - \{a_1, \ldots, a_k\} \in \mathcal{C}$
Deletion distance

J. Guo, F. Hüffner, R. Niedermeier (2004): Distance from Triviality.

\[ G \text{ at deletion distance } k \text{ from } \mathcal{C} \quad \text{iff} \quad G - \{a_1, \ldots, a_k\} \in \mathcal{C} \]

J. Bulian, A. Dawar (2016): FPT algorithm for graph isomorphism (parametrized by deletion distance to degree \( d \))

Input: Graphs \( G, H \), integer \( d \).

Goal: Are \( G \) and \( H \) isomorphic?

Running time: \( O(f(k,d) \cdot |G|^{g(d)}) \)

Parametrized by \( k \): the deletion distance (of \( G \)) to degree \( d \).

\( k \) is computable in time \( O(f(k,d) \cdot |G|) \).
Elimination distance


$$
ed_{\mathcal{C}}(G) = \begin{cases} 
0 & \text{if } G \in \mathcal{C}, \\
1 + \min \{\ed_{\mathcal{C}}(G - v) | v \in V(G)\} & \text{if } G \text{ is connected}, \\
\max \{\ed_{\mathcal{C}}(H) | H \text{ component of } G\} & \text{otherwise}. 
\end{cases}$$
Example

$\mathcal{C}_d$: all graphs of degree at most $d$.

$ed_{\mathcal{C}_3}(G) = ??$
Example

$C_d$: all graphs of degree at most $d$.

$ed_{C_3}(G) = ??$

Round 1): [1]
Example

$\mathcal{C}_d$: all graphs of degree at most $d$.

$ed_{\mathcal{C}_3}(G) = ??$

Round 1): $[1]$
Round 2): $[3, 7]$
Example

$C_d$: all graphs of degree at most $d$.

$ed_{C_3}(G) = ??$

Round 1) : [1]
Round 2) : [3, 7]
Round 3) : [4, 5]
Example

$C_d$: all graphs of degree at most $d$.

$ed_{C_3}(G) = ??$

Round 1) : [1] 
Round 2) : [3, 7] 
Round 3) : [4, 5]

Round 1) : [7] 
Round 2) : [1] 
Round 3) : [3, 4, 5]
Bounded tree depth

Elimination distance is inspired from tree depth:

\[
\text{td}(G) = \begin{cases} 
0 & \text{if } G \text{ is edgeless}, \\
1 + \min \{\text{td}(G - v) | v \in V(G)\} & \text{if } G \text{ is connected}, \\
\max \{\text{td}(H) | H \text{ component of } G\} & \text{otherwise}.
\end{cases}
\]

Tree depth \( k = \) Elimination distance \( k \) to the edgeless graph.
Graph isomorphism & Graphs with bounded degree

J. Bulian, A. Dawar (2016): FPT algorithm for graph isomorphism (parametrized by elimination distance to degree $d$)

Input : Graphs $G, H$, integer $d$.

Goal : Are $G$ and $H$ isomorphic?

Running time : $O(f(k, d) \cdot |G|^{g(d)})$

Parametrized by $k$ : the elimination distance (of $G$) to degree $d$. 
Questions

When is $\text{ed}_\mathcal{C}(G)$ easily computable?

Restriction on $\mathcal{C}$: Edgeless graphs, Graph with bounded degree, ...

Restriction on $G$: Parametrized by the tree width,

Parametrized by the size of an excluded minor,

Restricted to planar graph, ...
Our result

A. Lindermayr, S. Siebertz, A. Vigny:
Elimination distance to degree $d$ is FPT over $K_5$-minor-free graphs.

---

Input: Graph $G$, integers $k, d$.

Goal: Is $G$ at elimination distance $k$ to degree $d$?

Restriction: $G$ exclude $K_5$ as a minor.

Running time: $O(f(k, d) \cdot |G|^c)$
Tools

We use:

→ Simple combinatoric
→ MSO expressibility
→ Courcelle’s Theorem
→ Grid Theorem
→ Irrelevant vertex technique
Expressible in MSO

“$\text{ed}_{C_d}(G) = k$”:

$$\forall H_1 \preceq G, \exists a_1$$

$$\forall H_2 \preceq (H_1 - a_1), \exists a_2$$

$$\forall H_3 \preceq (H_2 - \{a_1, a_2\}), \exists a_3$$

$$\ldots$$

$$\forall H \preceq (H_k - \{a_1, \ldots, a_k\}), \deg(H) \leq d$$

If $C$ is MSO definable, then “$\text{ed}_C(G) = k$” is also MSO definable.
Courcelle’s Theorem

B. Courcelle (1990): Model checking of MSO formulas is FPT for bounded tree width graphs.

Input: Graph $G$, formula $\phi$.

Goal: Does $G \models \phi$?

Running time: $O(f(|\phi|, k) \cdot |G|)$.

Parametrized by $k$: the tree width of $G$. 
Grid Theorem

N. Robertson, P. D. Seymour (1986): Every graph has either a “small enough” tree width, or a “big enough” grid-minor.

Input: Graph $G$, integer $k$.

Output: Either a tree decomposition of width $O(g(k))$ 
Or a $k \times k$ grid minor

Running time: $O(f(k) \cdot |G|^c)$. 
Proof in a special case

Special case: Graph with degree $k + d$.

In the full proof:

$\rightarrow k + d < \deg(a)$

$\rightarrow d < \deg(a) \leq k + d$

$\rightarrow \deg(a) \leq d$

Here:

$\rightarrow d < \deg(a) \leq k + d$ (red nodes)

$\rightarrow \deg(a) \leq d$ (blue nodes)
A little bit of combinatorics

How many red nodes can there be?

1 round of elimination → \( k + d \) connected components.

\( k \) rounds of elimination → affect \((k + d)^2(k+d)\) nodes.

There are at most \( r = (k + d)^2(k+d) \) nodes of degree > \( d \).

More than \( r \) red nodes → we have \( \text{ed}_{\mathcal{E}_d}(G) > k \).
Otherwise we continue.
Using the grid theorem

We have two cases:

→ Tree decomposition of width $g(r)$.

Courcelle’s Theorem : \( O(f(\phi, g(r)) \cdot |G|) \)
\( O(f(k, d) \cdot |G|) \)

→ Grid minor of size $r \times r$.

Find an irrelevant vertex.
Large grid minor?

If $G$ has a large grid minor, we can find an irrelevant vertex.

Vertex $a$ is solution irrelevant: $\text{ed}_{\mathcal{E}_d}(G) \leq k \iff \text{ed}_{\mathcal{E}_d}(G - \{a\}) \leq k$
Large grid minor?

If $G$ has a large grid minor, we can find an irrelevant vertex.

Vertex $a$ is solution irrelevant:
\[ \text{ed}_{\mathcal{F}_d}(G) \leq k \iff \text{ed}_{\mathcal{F}_d}(G - \{a\}) \leq k \]

In our case a vertex is irrelevant if it is “far enough” from nodes with high degree.
Excluding $K_5$
Excluding $K_5$
Excluding $K_5$
Excluding $K_5$
How trivial can a graph be?

Elimination distance

Algorithm

Conclusion & Future works

Overall picture / The irrelevant vertex technique

\[ \text{tw}(G) < g(r) \] ?

- (yes)
  - Evaluate \( \phi(G) \)
  - (yes)
    - \( \text{ed}_{\mathcal{C}_d}(G) > k \)
  - (no)
    - Find irr. vertex \( a \),
      \( G := (G - \{a\}) \)

- (no)
  - (no)
How trivial can a graph be?

Elimination distance

Algorithm

Conclusion & Future works

Overall picture / The irrelevant vertex technique

1) Simple combinatorics

1) \#red nodes > r ?

(Yes) \(ed_{\mathcal{E}_d}(G) > k\)

(No)

Evaluate \(\phi(G)\)

(Yes)

Find irr. vertex \(a\),

\(G := (G - \{a\})\)

(No)

\(tw(G) < g(r) \) ?
Overall picture / The irrelevant vertex technique

1) Simple combinatorics

2) Grid Theorem

1) Simple combinatorics

2) Grid Theorem

#red nodes > r?

\[ \text{tw}(G) < g(r) \ ? \]

\[ \text{ed}_{C_d}(G) > k \]

Evaluate \( \phi(G) \)

Find irr. vertex \( a \),

\[ G := (G - \{a\}) \]
Overall picture / The irrelevant vertex technique

1) Simple combinatorics
2) Grid Theorem
3) Courcelle’s Theorem

1) #red nodes > r?
   (yes) ed_{C_d}(G) > k
   (no)

2) tw(G) < g(r) ?
   (yes) Evaluate \( \phi(G) \)
   (no)

3) Find irr. vertex \( a \),
   \( G := (G - \{a\}) \)
Overall picture / The irrelevant vertex technique

1) Simple combinatorics
2) Grid Theorem
3) Courcelle’s Theorem
4) Find and remove \{a\}
How trivial can a graph be?

Elimination distance

Algorithm

Conclusion & Futur works

Overall picture / The irrelevant vertex technique

1) Simple combinatorics

2) Grid Theorem

3) Courcelle’s Theorem

4) Find and remove \{a\}

Total: $O(f(k,d) \cdot |G|^c)$
Future work

Open problem 1
FPT algorithm for “ed_{\theta_{d}}(G) = k” for any graph G.

Open problem 2
FPT algorithm for “ed_{\theta_{d}}(G) = k” for any graph G of degree \( k + d \).

Conjecture
Algorithm for open problem 2) \( \rightarrow \) algorithm for open problem 1).
Future work

Open problem 1
FPT algorithm for “ed_{\mathcal{E}_d}(G) = k” for any graph G.

Open problem 2
FPT algorithm for “ed_{\mathcal{E}_d}(G) = k” for any graph G of degree k + d.

Conjecture
Algorithm for open problem 2) → algorithm for open problem 1).

Thank you!