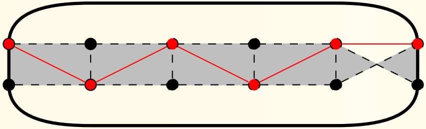
Crossing Number is Hard for Kernelization

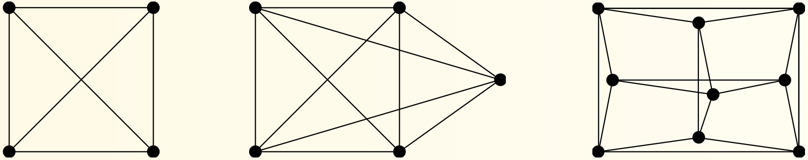
Petr Hliněný

Faculty of Informatics, Masaryk University
Brno, Czech Republic
http://www.fi.muni.cz/~hlineny

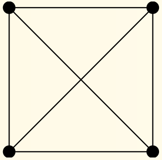
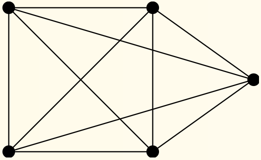
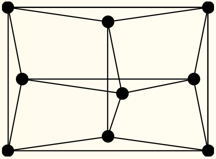
joint work with **Marek Derňár**



1 Crossing Minimization is Hard

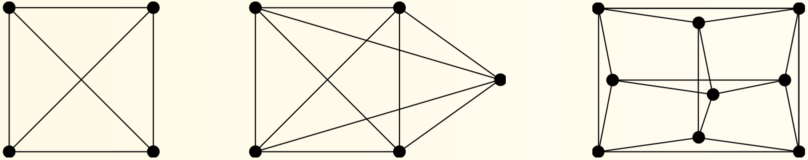


1 Crossing Minimization is Hard

Definition. $CR(k) \equiv$ the problem to draw a graph with $\leq k$ *edge crossings*.

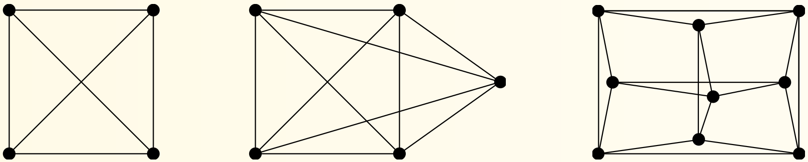
1 Crossing Minimization is Hard



Definition. $CR(k) \equiv$ the problem to draw a graph with $\leq k$ *edge crossings*.

– The vertices of G are distinct points in the plane, and every edge $e = uv \in E(G)$ is a simple curve joining u to v.

1 Crossing Minimization is Hard



Definition. $CR(k) \equiv$ the problem to draw a graph with $\leq k$ *edge crossings*.

– The vertices of G are distinct points in the plane, and every edge $e = uv \in E(G)$ is a simple curve joining u to v.

– No edge passes through another vertex, and no three edges intersect in a common point.

- A very hard algorithmic problem, indeed...

Traditional complexity of $CR(k)$

NP-hardness

- The general case (no surprise); [**Garey and Johnson**, 1983]

Traditional complexity of $CR(k)$

NP-hardness

- The general case (no surprise); [**Garey and Johnson**, 1983]
- The degree-3 and *minor-monotone* cases; [**PH**, 2004]

Traditional complexity of $CR(k)$

NP-hardness

- The general case (no surprise); [**Garey and Johnson**, 1983]
- The degree-3 and *minor-monotone* cases; [**PH**, 2004]
- With fixed *rotation scheme*; [**Pelsmajer, Schaeffer, Štefankovič**, 2007]

Traditional complexity of $CR(k)$

NP-hardness

- The general case (no surprise); [**Garey and Johnson**, 1983]
- The degree-3 and *minor-monotone* cases; [**PH**, 2004]
- With fixed *rotation scheme*; [**Pelsmajer, Schaeffer, Štefankovič**, 2007]
- And even for *almost-planar* (planar graphs plus one edge)! [**Cabello and Mohar**, 2010]

Traditional complexity of $CR(k)$

NP-hardness

- The general case (no surprise); [**Garey and Johnson**, 1983]
- The degree-3 and *minor-monotone* cases; [**PH**, 2004]
- With fixed *rotation scheme*; [**Pelsmajer, Schaeffer, Štefankovič**, 2007]
- And even for *almost-planar* (planar graphs plus one edge)!
  [**Cabello and Mohar**, 2010]

Approximations, at least?

- Up to factor $\log^3 |V(G)|$ ($\log^2 \cdot$) for *cr*$(G) + |V(G)|$ with bounded degs.;
  [**Even, Guha and Schieber**, 2002]

Traditional complexity of $CR(k)$

NP-hardness

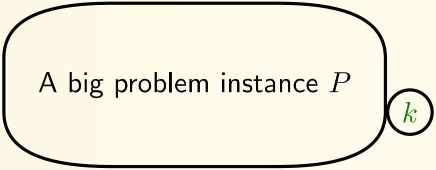
- The general case (no surprise); [**Garey and Johnson**, 1983]
- The degree-3 and *minor-monotone* cases; [**PH**, 2004]
- With fixed *rotation scheme*; [**Pelsmajer, Schaeffer, Štefankovič**, 2007]
- And even for *almost-planar* (planar graphs plus one edge)! [**Cabello and Mohar**, 2010]

Approximations, at least?

- Up to factor $\log^3 |V(G)|$ ($\log^2 \cdot$) for $cr(G) + |V(G)|$ with bounded degs.; [**Even, Guha and Schieber**, 2002]
- No constant factor approximation for some $c > 1$; [**Cabello**, 2013].

Parameterized complexity of $CR(k)$

Definition. Fixed-parameter tractability:



– and a tiny parameter k

– *FPT* $=$ the class of problems which can be solved in time $f(k) \cdot n^c$.

Parameterized complexity of $CR(k)$

Definition. Fixed-parameter tractability:

A big problem instance P

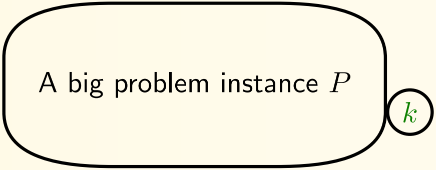
k *– and a tiny parameter k*

– *FPT* $=$ the class of problems which can be solved in time $f(k) \cdot n^c$.

- Yes, $CR(k)$ is in FPT when parameterized by k:
  - [**Grohe**, 2001] with runtime $f(k) \cdot n^2$,
  - [**Kawarabayashi and Reed**, 2007] with linear $f(k) \cdot n$.

Parameterized complexity of $CR(k)$

Definition. Fixed-parameter tractability:



– and a tiny parameter k

– *FPT* $=$ the class of problems which can be solved in time $f(k) \cdot n^c$.

- Yes, $CR(k)$ is in FPT when parameterized by k:
  – [**Grohe**, 2001] with runtime $f(k) \cdot n^2$,
  – [**Kawarabayashi and Reed**, 2007] with linear $f(k) \cdot n$.
- For example, [Grohe] starts with removal of "irrelevant vertices"... (preprocessing)

Going further: Preprocessing

Definition. Kernelization:

equiv. instance

A big instance P k $\rightsquigarrow$ in P-time small P' k' $\leq f'(k)$

Going further: Preprocessing

Definition. Kernelization:

equiv. instance

A big instance P k $\rightsquigarrow$ in P-time small P' k' $\leq f'(k)$

- Actually, it holds

$$\textbf{Computable} + \textbf{``Having a kernelization''} \equiv \textbf{FPT}$$

– with a straightforward proof.

Going further: Preprocessing

Definition. Kernelization:

equiv. instance

A big instance P (k) $\rightsquigarrow$ in P-time small P' (k') $\leq f'(k)$

- Actually, it holds

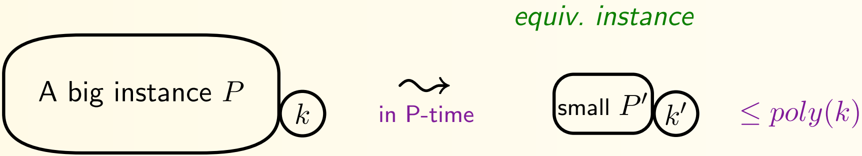
Computable $+$ **"Having a kernelization"** $\equiv$ **FPT**

  – with a straightforward proof.

- The prime question is; how big is the function $f'(k)$?
- A *polynomial kernel* $\longleftrightarrow$ f' is a polynomial.

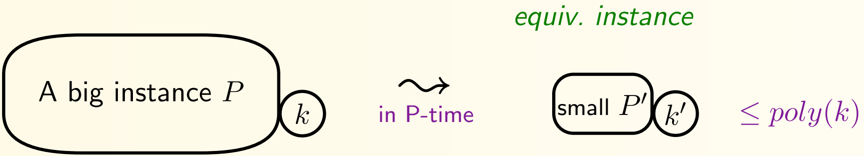
2 Polynomial Kernel for $CR(k)$?

- Recall the concept of polynomial kernelization:



2 Polynomial Kernel for $CR(k)$?

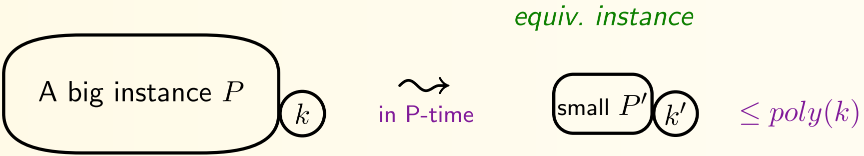
- Recall the concept of polynomial kernelization:



- Can we have a polynomial kernel for $CR(k)$?
  - The way existing FPT algorithms for $CR(k)$ work may suggests so.

2 Polynomial Kernel for $CR(k)$?

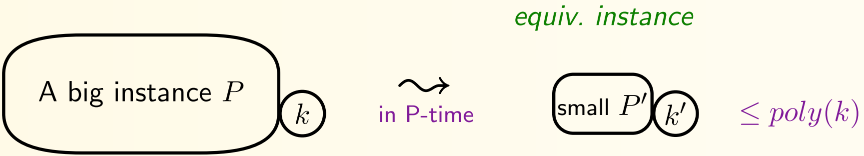
- Recall the concept of polynomial kernelization:



- Can we have a polynomial kernel for $CR(k)$?
  - The way existing FPT algorithms for $CR(k)$ work may suggests so.
  - There has been great advance in algorithmic graph minors theory recently.

2 Polynomial Kernel for $CR(k)$?

- Recall the concept of polynomial kernelization:

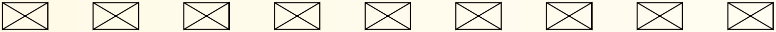


- Can we have a polynomial kernel for $CR(k)$?
  - The way existing FPT algorithms for $CR(k)$ work may suggests so.
  - There has been great advance in algorithmic graph minors theory recently.
  - And yet. . .
- So, why NOT ?

No polynomial kernel for $CR(k)$

– a sketch by means of contradiction:

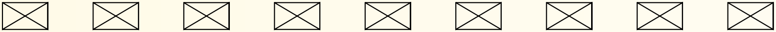
- Imagine a heap of $2^{o(k)}$ small instances of $CR(k)$


,

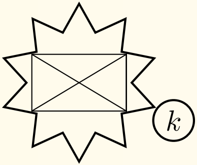
No polynomial kernel for $CR(k)$

– a sketch by means of contradiction:

- Imagine a heap of $2^{o(k)}$ small instances of $CR(k)$


,

- and make them into one large *"OR" composed* instance of $CR(k)$

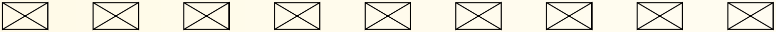


.

No polynomial kernel for $CR(k)$

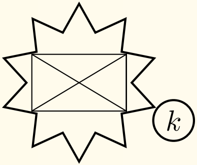
– a sketch by means of contradiction:

- Imagine a heap of $2^{o(k)}$ small instances of $CR(k)$



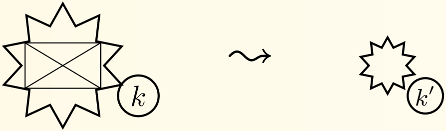
,

- and make them into one large *"OR" composed* instance of $CR(k)$



.

- Now, kernelize to



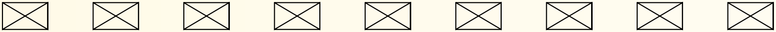
$\leq poly(k) << 2^{o(k)}$

.

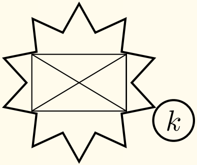
No polynomial kernel for $CR(k)$

– a sketch by means of contradiction:

- Imagine a heap of $2^{o(k)}$ small instances of $CR(k)$

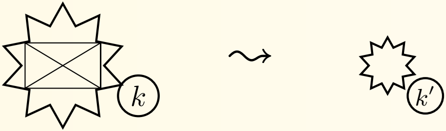

,

- and make them into one large *"OR" composed* instance of $CR(k)$



.

- Now, kernelize to



$\leq poly(k) << 2^{o(k)}$

.

- What does this mean? Most of orig. instances have no bit in the kernel!
  Have some of them been solved? Unlikely in $poly(2^{o(k)})$ time...

The formal tool: OR-cross-composition

Definition. Let $\mathcal{L}$ be an (NP-hard) problem. Consider that,

- for arbitrary instances $x_1, x_2, \ldots, x_t$ of $\mathcal{L}$ (of "related type"),

The formal tool: OR-cross-composition

Definition. Let $\mathcal{L}$ be an (NP-hard) problem. Consider that,

- for arbitrary instances $x_1, x_2, \ldots, x_t$ of $\mathcal{L}$ (of "related type"),
- we can, in time $poly(|x_1| + \ldots |x_t|)$, compute parameterized (y, k) of $\mathcal{P}$
  - such that $k \le poly(\max |x_i| + \log t)$, and

The formal tool: OR-cross-composition

Definition. Let $\mathcal{L}$ be an (NP-hard) problem. Consider that,

- for arbitrary instances $x_1, x_2, \ldots, x_t$ of $\mathcal{L}$ (of "related type"),
- we can, in time $poly(|x_1| + \ldots |x_t|)$, compute parameterized (y, k) of $\mathcal{P}$
  - such that $k \leq poly(\max|x_i| + \log t)$, and
  - $(y, k) \in \mathcal{P}$ if and only if $x_i \in \mathcal{L}$ for some $1 \leq i \leq t$.

Then we say that $\mathcal{L}$ has an OR-*cross-composition* into $\mathcal{P}$.

The formal tool: OR-cross-composition

Definition. Let $\mathcal{L}$ be an (NP-hard) problem. Consider that,

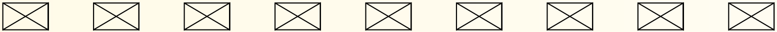
- for arbitrary instances $x_1, x_2, \ldots, x_t$ of $\mathcal{L}$ (of "related type"),
- we can, in time $poly(|x_1| + \ldots |x_t|)$, compute parameterized (y, k) of $\mathcal{P}$
  - such that $k \leq poly(\max|x_i| + \log t)$, and
  - $(y, k) \in \mathcal{P}$ if and only if $x_i \in \mathcal{L}$ for some $1 \leq i \leq t$.

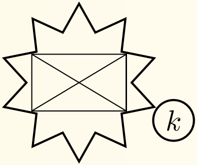
Then we say that $\mathcal{L}$ has an OR-*cross-composition* into $\mathcal{P}$.

Theorem 1. (Bodlaender, Jansen and Kratsch 2014)
If an NP-hard language $\mathcal{L}$ has an OR-cross-composition into the parameterized problem $\mathcal{P}$, then $\mathcal{P}$ does not admit a polynomial kernel unless NP $\subseteq$ coNP/poly.

Though, how to compose $CR(k)$?

- Recall;

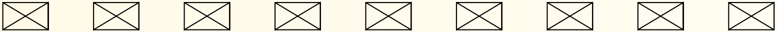
$t \times$ 

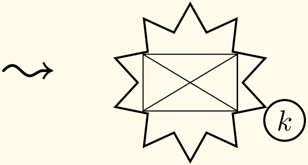
$\rightsquigarrow$ 

of $CR(k)$.

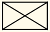
Though, how to compose $CR(k)$?

- Recall;

$t\times$ 



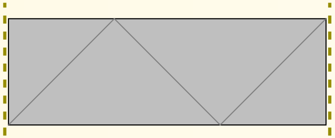
of $CR(k)$.

- If every instance  is nontrivial, then it likely contributes ≥ 1 crossing to the composition, and hence $k = \Omega(t) > poly(\log t)$.

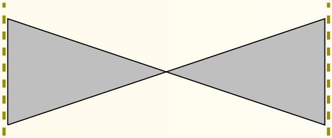
- Does not work straightforwardly yet…

3 $CR(k)$ for Twisted Planar Tiles

- Drawing stretched between the left and right walls of a "tile":



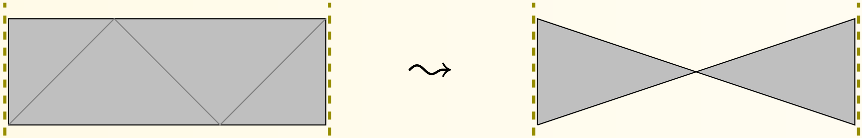
$\rightsquigarrow$



planar tile *twisted tile*

3 $CR(k)$ for Twisted Planar Tiles

- Drawing stretched between the left and right walls of a "tile":

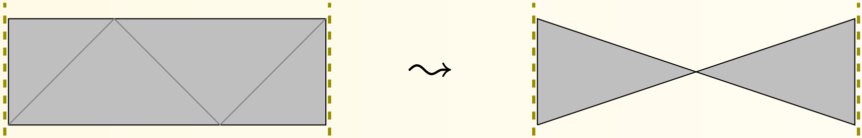


planar tile $\rightsquigarrow$ *twisted tile*

- $TPT\text{-}CR(k) \equiv$ problem to draw a twisted planar tile with $\leq k$ crossings.

3 $CR(k)$ for Twisted Planar Tiles

- Drawing stretched between the left and right walls of a “tile”:



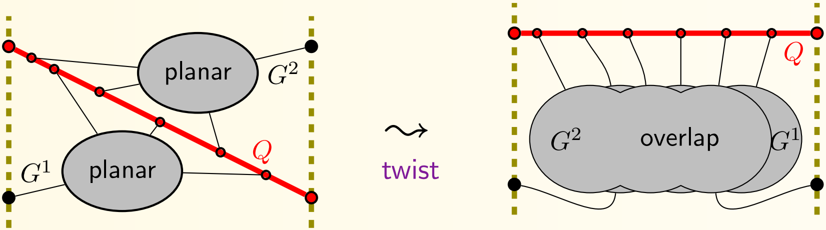
planar tile $\rightsquigarrow$ *twisted tile*

- $TPT\text{-}CR(k) \equiv$ problem to draw a twisted planar tile with $\leq k$ crossings.
- Tiles (and specially twisted planar tiles) have been considered for long time in crossing number research...

  But, no complexity results published so far.

RESULT #1: $\mathit{TPT}\text{-}\mathit{CR}(k)$ is NP-hard

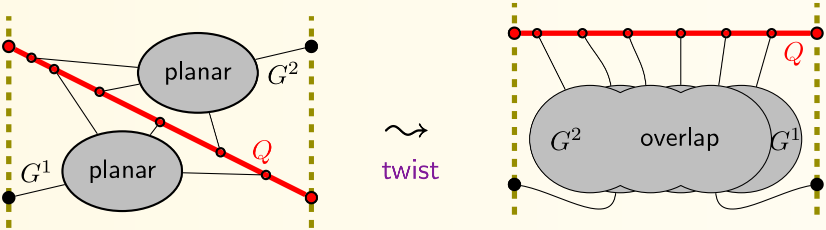
- We borrow the construction from **[Cabello and Mohar, 2013]**:
  (Originally for so called *anchored crossing number*.)



Q a "very thick" red path

RESULT #1: $TPT\text{-}CR(k)$ is NP-hard

- We borrow the construction from **[Cabello and Mohar, 2013]**:
  (Originally for so called *anchored crossing number*.)

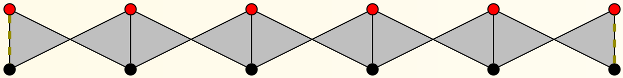


Q a "very thick" red path

- $\to$ Crossing minimization of the "overlap picture" is NP-hard
  (in traditional complexity).

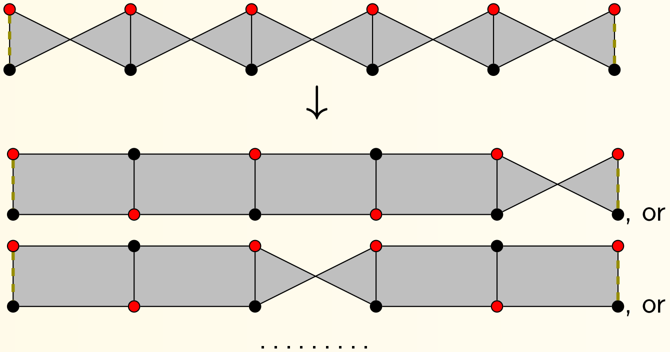
RESULT #2: $\boldsymbol{TPT\text{-}CR(k)}$ is $\mathrm{O{\scriptsize R}}$-composable

- A high-level "picture proof":



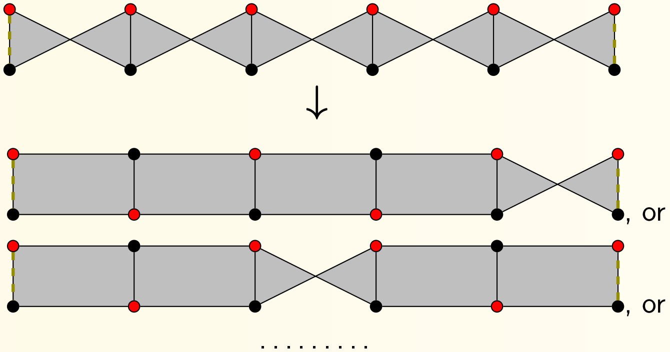
RESULT #2: $\mathit{TPT}\text{-}\mathit{CR}(k)$ is OR-composable

- A high-level "picture proof":



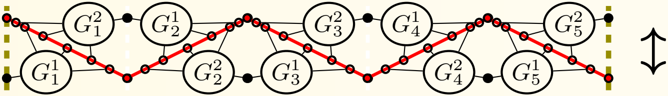
RESULT #2: $TPT\text{-}CR(k)$ is OR-composable

- A high-level "picture proof":



- A schematic realization, ensuring that a "full twist" happens at once:

very thick Q



CONCLUSION: No polynomial kernel for $CR(k)$

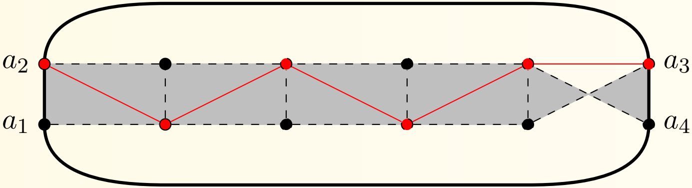
- So far, we have proved that $TPT\text{-}CR(k)$ has no polynomial kernel.

CONCLUSION: No polynomial kernel for $CR(k)$

- So far, we have proved that $TPT\text{-}CR(k)$ has no polynomial kernel.
- Though, the cross-composition framework allows for “embedding” of the composed problem into any other target problem...

CONCLUSION: No polynomial kernel for $CR(k)$

- So far, we have proved that $TPT\text{-}CR(k)$ has no polynomial kernel.
- Though, the cross-composition framework allows for "embedding" of the composed problem into any other target problem. . .
- We hence embed the tiles into an ordinary $CR(k)$ instance:



- Note; the resulting graph is again almost-planar.

4 Final Remarks

- Crossing minimization is NP-hard also under other restrictions, e.g.
  - for cubic graphs, or graphs with fixed (prescribed) rotation system.

4 Final Remarks

- Crossing minimization is NP-hard also under other restrictions, e.g.
  - for cubic graphs, or graphs with fixed (prescribed) rotation system.
- Our construction is incompatible with these restrictions:

4 Final Remarks

- Crossing minimization is NP-hard also under other restrictions, e.g.
  - for cubic graphs, or graphs with fixed (prescribed) rotation system.
- Our construction is incompatible with these restrictions:

- Yet we expect the same to be true in the other cases:

Conjecture. The problem $CR\text{-}\textsc{rot}(k)$, asking for a drawing with $\leq k$ crossings under the restriction of a given rotation system, has no polynomial kernel.

Consequently, $CR(k)$ has no polynomial kernel even for cubic graphs.