Hitting minors on bounded treewidth graphs

Ignasi Sau

CNRS, LIRMM, Université de Montpellier, France

Joint work with Julien Baste and Dimitrios M. Thilikos

`arXiv 1704.07284` + `arXiv 1907.04442`

FraNorAC 2019

LIRMM, Montpellier, France

Treewidth behaves very well algorithmically

(Invariant that measures the topological resemblance of a graph to a tree.)

Treewidth behaves very well algorithmically

(Invariant that measures the topological resemblance of a graph to a tree.)

Monadic Second Order Logic (MSOL):
Graph logic that allows quantification over sets of vertices and edges.

Example: $\texttt{DomSet}(S)$: $[\ \forall v \in V(G) \setminus S, \exists u \in S : \{u, v\} \in E(G)\ ]$

Treewidth behaves very well algorithmically

(Invariant that measures the topological resemblance of a graph to a tree.)

Monadic Second Order Logic (MSOL):
Graph logic that allows quantification over sets of vertices and edges.

Example: $\texttt{DomSet}(S)$: $[\ \forall v \in V(G) \setminus S, \exists u \in S : \{u, v\} \in E(G)\ ]$

Theorem (Courcelle. 1990)

Every problem expressible in MSOL can be solved in time $f(\mathrm{tw}) \cdot n$ on graphs on n vertices and treewidth at most tw.

In parameterized complexity: FPT parameterized by treewidth.

Treewidth behaves very well algorithmically

(Invariant that measures the topological resemblance of a graph to a tree.)

Monadic Second Order Logic (MSOL):
Graph logic that allows quantification over sets of vertices and edges.

Example: $\texttt{DomSet}(S): \quad [\, \forall v \in V(G) \setminus S, \exists u \in S : \{u, v\} \in E(G) \,]$

Theorem (Courcelle. 1990)

Every problem expressible in MSOL can be solved in time $f(\mathrm{tw}) \cdot n$ on graphs on n vertices and treewidth at most tw.

In parameterized complexity: FPT parameterized by treewidth.

Examples: Vertex Cover, Dominating Set, Hamiltonian Cycle, Clique, Independent Set, k-Coloring for fixed k, ...

Is it enough to prove that a problem is FPT?

Typically, Courcelle's theorem allows to prove that a problem is FPT...

$$f(\mathrm{tw}) \cdot n^{\mathcal{O}(1)}$$

Is it enough to prove that a problem is FPT?

Typically, Courcelle's theorem allows to prove that a problem is FPT...

... but the running time can (and must) be huge!

$$f(\mathrm{tw}) \cdot n^{\mathcal{O}(1)} = 2^{3^{4^{5^{6^{7^{8^{\mathrm{tw}}}}}}}} \cdot n^{\mathcal{O}(1)}$$

Is it enough to prove that a problem is FPT?

Typically, Courcelle's theorem allows to prove that a problem is FPT...
... but the running time can (and must) be huge!

$$f(\mathrm{tw}) \cdot n^{\mathcal{O}(1)} = 2^{3^{4^{5^{6^{7^{8^{\mathrm{tw}}}}}}}} \cdot n^{\mathcal{O}(1)}$$

$\boxed{\text{Major goal}}$ find the smallest possible function $f(\mathrm{tw})$.

This is a very active area in parameterized complexity.

Remark: Algorithms parameterized by treewidth appear very often as a "black box" in all kinds of parameterized algorithms.

Is it enough to prove that a problem is FPT?

Typically, Courcelle's theorem allows to prove that a problem is FPT...
... but the running time can (and must) be huge!

$$f(\text{tw}) \cdot n^{\mathcal{O}(1)} = 2^{3^{4^{5^{6^{7^{8^{\text{tw}}}}}}}} \cdot n^{\mathcal{O}(1)}$$

$\boxed{\text{Major goal}}$ find the $\boxed{\text{smallest possible}}$ function $f(\text{tw})$. $\boxed{\text{Tool: ETH}}$

This is a very active area in parameterized complexity.

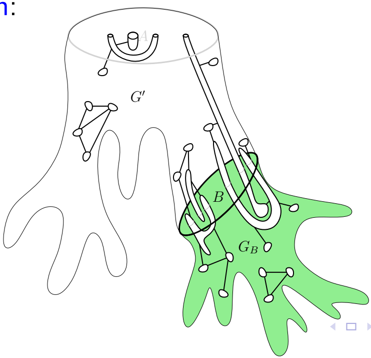
Remark: Algorithms parameterized by treewidth appear very often as a "black box" in all kinds of parameterized algorithms.

Dynamic programming on tree decompositions

- Typically, FPT algorithms parameterized by treewidth are based on dynamic programming (DP) over a tree decomposition.
- Starting from the leaves of the tree decomposition, a set of appropriately defined partial solutions is computed recursively until the root, where a global solution is obtained.

Dynamic programming on tree decompositions

- Typically, FPT algorithms parameterized by treewidth are based on dynamic programming (DP) over a tree decomposition.
- Starting from the leaves of the tree decomposition, a set of appropriately defined partial solutions is computed recursively until the root, where a global solution is obtained.
- The way that these partial solutions are defined depends on each particular problem:



[Figure by Valentin Garnero]

Two distinct types of problems

Two distinct types of problems

- Local problems: solution can be certified locally at each vertex.

  VERTEX COVER, DOMINATING SET, CLIQUE

Two distinct types of problems

- Local problems: solution can be certified locally at each vertex.

  Vertex Cover, Dominating Set, Clique

  Natural DP: $\boxed{2^{\mathcal{O}(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}}$

Two distinct types of problems

- Local problems: solution can be certified locally at each vertex.

  Vertex Cover, Dominating Set, Clique

  Natural DP: $\boxed{2^{\mathcal{O}(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}}$

- Connectivity problems: certificates need a global information.

  Longest Path, Feedback Vertex Set, Steiner Tree

Two distinct types of problems

- Local problems: solution can be certified locally at each vertex.

  Vertex Cover, Dominating Set, Clique

  Natural DP: $\boxed{2^{\mathcal{O}(\text{tw})} \cdot n^{\mathcal{O}(1)}}$

- Connectivity problems: certificates need a global information.

  Longest Path, Feedback Vertex Set, Steiner Tree

  Natural DP: $\boxed{2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})} \cdot n^{\mathcal{O}(1)}}$

The revolution of single-exponential algorithms

It was believed that, except on sparse graphs (planar, surfaces), algorithms in time $2^{\mathcal{O}(\mathrm{tw} \cdot \log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$ were optimal for connectivity problems.

The revolution of single-exponential algorithms

It was believed that, except on sparse graphs (planar, surfaces), algorithms in time $2^{\mathcal{O}(\mathrm{tw}\cdot\log\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$ were optimal for connectivity problems.

$\boxed{\text{This was false!!}}$

Cut&Count technique: [Cygan, Nederlof, Pilipczuk^2, van Rooij, Wojtaszczyk. 2011]

Randomized single-exponential algorithms for connectivity problems.

The revolution of single-exponential algorithms

It was believed that, except on sparse graphs (planar, surfaces), algorithms in time $2^{\mathcal{O}(\mathrm{tw} \cdot \log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$ were optimal for connectivity problems.

$\boxed{\text{This was false!!}}$

Cut&Count technique: [Cygan, Nederlof, Pilipczuk^2, van Rooij, Wojtaszczyk. 2011]

Randomized single-exponential algorithms for connectivity problems.

1. Relax the connectivity requirement by considering a set of cuts that contain the relevant (connected) solutions.
2. Count modulo 2 the number of cuts, because the non-connected solutions will cancel out. By assigning random weights to the vertices/edges, guarantee that w.h.p. the optimal solution is unique (Isolation Lemma).

The revolution of single-exponential algorithms

It was believed that, except on sparse graphs (planar, surfaces), algorithms in time $2^{\mathcal{O}(\mathrm{tw} \cdot \log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$ were optimal for connectivity problems.

$\boxed{\text{This was false!!}}$

Cut&Count technique: [Cygan, Nederlof, Pilipczuk^2, van Rooij, Wojtaszczyk. 2011]

Randomized single-exponential algorithms for connectivity problems.

1. Relax the connectivity requirement by considering a set of cuts that contain the relevant (connected) solutions.
2. Count modulo 2 the number of cuts, because the non-connected solutions will cancel out. By assigning random weights to the vertices/edges, guarantee that w.h.p. the optimal solution is unique (Isolation Lemma).

Deterministic algorithms with algebraic tricks: [Bodlaender, Cygan, Kratsch, Nederlof. 2013]

Representative sets in matroids: [Fomin, Lokshtanov, Saurabh. 2014]

End of the story?

Do all connectivity problems admit single-exponential algorithms (on general graphs) parameterized by treewidth?

End of the story?

Do all connectivity problems admit single-exponential algorithms (on general graphs) parameterized by treewidth?

No!

Cycle Packing: find the maximum number of vertex-disjoint cycles.

End of the story?

Do all connectivity problems admit single-exponential algorithms (on general graphs) parameterized by treewidth?

$\boxed{\text{No!}}$

Cycle Packing: find the maximum number of vertex-disjoint cycles.

An algorithm in time $2^{\mathcal{O}(\mathrm{tw} \cdot \log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$ is optimal under the ETH.

[Cygan, Nederlof, Pilipczuk, Pilipczuk, van Rooij, Wojtaszczyk. 2011]

This reduction uses a framework introduced by [Lokshtanov, Marx, Saurabh. 2011]

End of the story?

Do all connectivity problems admit single-exponential algorithms (on general graphs) parameterized by treewidth?

No!

Cycle Packing: find the maximum number of vertex-disjoint cycles.

An algorithm in time $2^{\mathcal{O}(\mathrm{tw} \cdot \log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$ is optimal under the ETH.

[Cygan, Nederlof, Pilipczuk, Pilipczuk, van Rooij, Wojtaszczyk. 2011]

This reduction uses a framework introduced by [Lokshtanov, Marx, Saurabh. 2011]

There are other examples of such problems...

The $\mathcal{F}$-M-Deletion problem

Let $\mathcal{F}$ be a fixed finite collection of graphs.

The $\mathcal{F}$-M-Deletion problem

Let $\mathcal{F}$ be a fixed finite collection of graphs.

$\mathcal{F}$-M-Deletion
Input: A graph G and an integer k.
Parameter: The treewidth tw of G.
Question: Does G contain a set $S \subseteq V(G)$ with $|S| \leq k$ such that $G - S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?

The $\mathcal{F}$-M-Deletion problem

Let $\mathcal{F}$ be a fixed finite collection of graphs.

$\mathcal{F}$-M-Deletion	
Input:	A graph G and an integer k.
Parameter:	The treewidth tw of G.
Question:	Does G contain a set $S \subseteq V(G)$ with $\lvert S \rvert \leq k$ such that $G - S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?

- $\mathcal{F} = \{K_2\}$: Vertex Cover.

The $\mathcal{F}$-M-DELETION problem

Let $\mathcal{F}$ be a fixed finite collection of graphs.

$\mathcal{F}$-M-DELETION	
Input:	A graph G and an integer k.
Parameter:	The treewidth $\mathbf{tw}$ of G.
Question:	Does G contain a set $S \subseteq V(G)$ with $\lvert S\rvert \leq k$ such that $G - S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?

- $\mathcal{F} = \{K_2\}$: VERTEX COVER.
  Easily solvable in time $2^{\Theta(\mathbf{tw})} \cdot n^{\mathcal{O}(1)}$.

The $\mathcal{F}$-M-Deletion problem

Let $\mathcal{F}$ be a fixed finite collection of graphs.

$\mathcal{F}$-M-Deletion	
Input:	A graph G and an integer k.
Parameter:	The treewidth tw of G.
Question:	Does G contain a set $S \subseteq V(G)$ with $\|S\| \leq k$ such that $G - S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?

- $\mathcal{F} = \{K_2\}$: Vertex Cover.
  Easily solvable in time $2^{\Theta(\mathsf{tw})} \cdot n^{\mathcal{O}(1)}$.
- $\mathcal{F} = \{K_3\}$: Feedback Vertex Set.

The $\mathcal{F}$-M-Deletion problem

Let $\mathcal{F}$ be a fixed finite collection of graphs.

$\mathcal{F}$-M-Deletion	
Input:	A graph G and an integer k.
Parameter:	The treewidth tw of G.
Question:	Does G contain a set $S \subseteq V(G)$ with $\lvert S\rvert \leq k$ such that $G - S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?

- $\mathcal{F} = \{K_2\}$: Vertex Cover.
  Easily solvable in time $2^{\Theta(\text{tw})} \cdot n^{\mathcal{O}(1)}$.
- $\mathcal{F} = \{K_3\}$: Feedback Vertex Set.
  "Hardly" solvable in time $2^{\Theta(\text{tw})} \cdot n^{\mathcal{O}(1)}$. [Cut&Count. 2011]

The $\mathcal{F}$-M-Deletion problem

Let $\mathcal{F}$ be a fixed finite collection of graphs.

$\mathcal{F}$-M-Deletion	
Input:	A graph G and an integer k.
Parameter:	The treewidth tw of G.
Question:	Does G contain a set $S \subseteq V(G)$ with $\lvert S\rvert \leq k$ such that $G - S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?

- $\mathcal{F} = \{K_2\}$: Vertex Cover.
  Easily solvable in time $2^{\Theta(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.
- $\mathcal{F} = \{K_3\}$: Feedback Vertex Set.
  "Hardly" solvable in time $2^{\Theta(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$. [Cut&Count. 2011]
- $\mathcal{F} = \{K_5, K_{3,3}\}$: Vertex Planarization.

The $\mathcal{F}$-M-Deletion problem

Let $\mathcal{F}$ be a fixed finite collection of graphs.

$\mathcal{F}$-M-Deletion	
Input:	A graph G and an integer k.
Parameter:	The treewidth tw of G.
Question:	Does G contain a set $S \subseteq V(G)$ with $\|S\| \leq k$ such that $G - S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?

- $\mathcal{F} = \{K_2\}$: Vertex Cover.
  Easily solvable in time $2^{\Theta(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.
- $\mathcal{F} = \{K_3\}$: Feedback Vertex Set.
  "Hardly" solvable in time $2^{\Theta(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$. [Cut&Count. 2011]
- $\mathcal{F} = \{K_5, K_{3,3}\}$: Vertex Planarization.
  Solvable in time $2^{\Theta(\mathrm{tw} \cdot \log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$. [Jansen, Lokshtanov, Saurabh. 2014 + Pilipczuk. 2015]

Covering topological minors

Let $\mathcal{F}$ be a fixed finite collection of graphs.

$\mathcal{F}$-M-Deletion	
Input:	A graph G and an integer k.
Parameter:	The treewidth $\mathbf{tw}$ of G.
Question:	Does G contain a set $S \subseteq V(G)$ with $\|S\| \leq k$ such that $G - S$ does not contain any graph in $\mathcal{F}$ as a minor?

Covering topological minors

Let $\mathcal{F}$ be a fixed finite collection of graphs.

$\mathcal{F}$-M-Deletion
Input: A graph G and an integer k.
Parameter: The treewidth tw of G.
Question: Does G contain a set $S \subseteq V(G)$ with $|S| \leq k$ such that $G - S$ does not contain any graph in $\mathcal{F}$ as a minor?

$\mathcal{F}$-TM-Deletion
Input: A graph G and an integer k.
Parameter: The treewidth tw of G.
Question: Does G contain a set $S \subseteq V(G)$ with $|S| \leq k$ such that $G - S$ does not contain any graph in $\mathcal{F}$ as a topol. minor?

Covering topological minors

Let $\mathcal{F}$ be a fixed finite collection of graphs.

$\mathcal{F}$-M-Deletion
Input: A graph G and an integer k.
Parameter: The treewidth tw of G.
Question: Does G contain a set $S \subseteq V(G)$ with $|S| \leq k$ such that $G - S$ does not contain any graph in $\mathcal{F}$ as a minor?

$\mathcal{F}$-TM-Deletion
Input: A graph G and an integer k.
Parameter: The treewidth tw of G.
Question: Does G contain a set $S \subseteq V(G)$ with $|S| \leq k$ such that $G - S$ does not contain any graph in $\mathcal{F}$ as a topol. minor?

Both problems are NP-hard if $\mathcal{F}$ contains some edge. [Lewis, Yannakakis. 1980]

FPT by Courcelle's Theorem.

Work with Julien Baste and Dimitrios M. Thilikos (2016-)

Objective

Determine, for every fixed $\mathcal{F}$, the (asymptotically) smallest function $f_{\mathcal{F}}$ such that $\mathcal{F}$-M-Deletion/$\mathcal{F}$-TM-Deletion can be solved in time

$$f_{\mathcal{F}}(\mathsf{tw}) \cdot n^{\mathcal{O}(1)}$$

on n-vertex graphs.

Work with Julien Baste and Dimitrios M. Thilikos (2016-)

Objective

Determine, for every fixed $\mathcal{F}$, the (asymptotically) smallest function $f_{\mathcal{F}}$ such that $\mathcal{F}$-M-Deletion/$\mathcal{F}$-TM-Deletion can be solved in time

$$f_{\mathcal{F}}(\text{tw}) \cdot n^{\mathcal{O}(1)}$$

on n-vertex graphs.

- We do not want to optimize the degree of the polynomial factor.
- We do not want to optimize the constants.
- Our hardness results hold under the ETH.

Summary of our results: arXiv 1704.07284+1907.04442

[1] Connected collection $\mathcal{F}$: all the graphs are connected.

[2] Planar collection $\mathcal{F}$: contains at least one planar graph.

Summary of our results: arXiv 1704.07284+1907.04442

- For every $\mathcal{F}$: $\mathcal{F}$-M/TM-DELETION in time $2^{2^{\mathcal{O}(\mathsf{tw} \cdot \log \mathsf{tw})}} \cdot n^{\mathcal{O}(1)}$.

[1] Connected collection $\mathcal{F}$: all the graphs are connected.

[2] Planar collection $\mathcal{F}$: contains at least one planar graph.

Summary of our results: arXiv 1704.07284+1907.04442

- For every $\mathcal{F}$: $\mathcal{F}$-M/TM-Deletion in time $2^{2^{\mathcal{O}(\mathrm{tw}\cdot\log\mathrm{tw})}}\cdot n^{\mathcal{O}(1)}$.
- $\mathcal{F}$ connected[1] + planar[2]: $\mathcal{F}$-M-Deletion in time $2^{\mathcal{O}(\mathrm{tw}\cdot\log\mathrm{tw})}\cdot n^{\mathcal{O}(1)}$.

[1] Connected collection $\mathcal{F}$: all the graphs are connected.
[2] Planar collection $\mathcal{F}$: contains at least one planar graph.

Summary of our results: arXiv 1704.07284+1907.04442

- For every $\mathcal{F}$: $\mathcal{F}$-M/TM-Deletion in time $2^{2^{\mathcal{O}(\mathrm{tw}\cdot\log \mathrm{tw})}} \cdot n^{\mathcal{O}(1)}$.
- $\mathcal{F}$ connected[1] ~~+ planar~~[2]: $\mathcal{F}$-M-Deletion in time $2^{\mathcal{O}(\mathrm{tw}\cdot\log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.

[1] Connected collection $\mathcal{F}$: all the graphs are connected.
[2] Planar collection $\mathcal{F}$: contains at least one planar graph.

Summary of our results: arXiv 1704.07284+1907.04442

- For every $\mathcal{F}$: $\mathcal{F}$-M/TM-Deletion in time $2^{2^{\mathcal{O}(\mathrm{tw}\cdot\log\mathrm{tw})}}\cdot n^{\mathcal{O}(1)}$.
- $\mathcal{F}$ connected[1] ~~+ planar~~[2]: $\mathcal{F}$-M-Deletion in time $2^{\mathcal{O}(\mathrm{tw}\cdot\log\mathrm{tw})}\cdot n^{\mathcal{O}(1)}$.
- G planar + $\mathcal{F}$ connected: $\mathcal{F}$-M-Deletion in time $2^{\mathcal{O}(\mathrm{tw})}\cdot n^{\mathcal{O}(1)}$.

[1] Connected collection $\mathcal{F}$: all the graphs are connected.

[2] Planar collection $\mathcal{F}$: contains at least one planar graph.

Summary of our results: arXiv 1704.07284+1907.04442

- For every $\mathcal{F}$: $\mathcal{F}$-M/TM-Deletion in time $2^{2^{\mathcal{O}(\mathrm{tw}\cdot\log \mathrm{tw})}} \cdot n^{\mathcal{O}(1)}$.
- $\mathcal{F}$ connected[1] ~~+ planar~~[2]: $\mathcal{F}$-M-Deletion in time $2^{\mathcal{O}(\mathrm{tw}\cdot\log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.
- G planar + $\mathcal{F}$ connected: $\mathcal{F}$-M-Deletion in time $2^{\mathcal{O}(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.

  (For $\mathcal{F}$-TM-Deletion we need: $\mathcal{F}$ contains a subcubic planar graph.)

[1] Connected collection $\mathcal{F}$: all the graphs are connected.

[2] Planar collection $\mathcal{F}$: contains at least one planar graph.

Summary of our results: arXiv 1704.07284+1907.04442

- For every $\mathcal{F}$: $\mathcal{F}$-M/TM-DELETION in time $2^{2^{\mathcal{O}(\mathrm{tw}\cdot\log\mathrm{tw})}}\cdot n^{\mathcal{O}(1)}$.
- $\mathcal{F}$ connected[1] ~~+ planar~~[2]: $\mathcal{F}$-M-DELETION in time $2^{\mathcal{O}(\mathrm{tw}\cdot\log\mathrm{tw})}\cdot n^{\mathcal{O}(1)}$.
- G planar + $\mathcal{F}$ connected: $\mathcal{F}$-M-DELETION in time $2^{\mathcal{O}(\mathrm{tw})}\cdot n^{\mathcal{O}(1)}$.

  (For $\mathcal{F}$-TM-DELETION we need: $\mathcal{F}$ contains a subcubic planar graph.)
- $\mathcal{F}$ (connected): $\mathcal{F}$-M/TM-DELETION not in time $2^{o(\mathrm{tw})}\cdot n^{\mathcal{O}(1)}$ unless the ETH fails, even if G planar.

---

[1] Connected collection $\mathcal{F}$: all the graphs are connected.

[2] Planar collection $\mathcal{F}$: contains at least one planar graph.

Summary of our results: arXiv 1704.07284+1907.04442

- For every $\mathcal{F}$: $\mathcal{F}$-M/TM-Deletion in time $2^{2^{\mathcal{O}(\mathrm{tw}\cdot\log\mathrm{tw})}} \cdot n^{\mathcal{O}(1)}$.

- $\mathcal{F}$ connected[1] ~~+ planar~~[2]: $\mathcal{F}$-M-Deletion in time $2^{\mathcal{O}(\mathrm{tw}\cdot\log\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.

- G planar $+$ $\mathcal{F}$ connected: $\mathcal{F}$-M-Deletion in time $2^{\mathcal{O}(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.

  (For $\mathcal{F}$-TM-Deletion we need: $\mathcal{F}$ contains a subcubic planar graph.)

- $\mathcal{F}$ (connected): $\mathcal{F}$-M/TM-Deletion not in time $2^{o(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$
  unless the ETH fails, even if G planar.

- $\mathcal{F} = \{H\}$, H connected:

[1] Connected collection $\mathcal{F}$: all the graphs are connected.
[2] Planar collection $\mathcal{F}$: contains at least one planar graph.

Summary of our results: arXiv 1704.07284+1907.04442

- For every $\mathcal{F}$: $\mathcal{F}$-M/TM-DELETION in time $2^{2^{\mathcal{O}(\mathrm{tw}\cdot\log \mathrm{tw})}} \cdot n^{\mathcal{O}(1)}$.
- $\mathcal{F}$ connected[1] ~~+ planar~~[2]: $\mathcal{F}$-M-DELETION in time $2^{\mathcal{O}(\mathrm{tw}\cdot\log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.
- G planar + $\mathcal{F}$ connected: $\mathcal{F}$-M-DELETION in time $2^{\mathcal{O}(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.

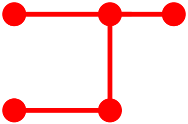
  (For $\mathcal{F}$-TM-DELETION we need: $\mathcal{F}$ contains a subcubic planar graph.)
- $\mathcal{F}$ (connected): $\mathcal{F}$-M/TM-DELETION not in time $2^{o(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$ unless the ETH fails, even if G planar.
- $\mathcal{F} = \{H\}$, H connected: complete tight dichotomy...

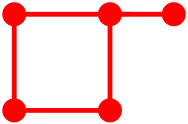
[1] Connected collection $\mathcal{F}$: all the graphs are connected.

[2] Planar collection $\mathcal{F}$: contains at least one planar graph.

A dichotomy for hitting a connected minor

A dichotomy for hitting a connected minor

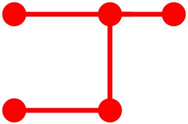


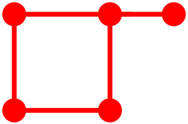


Theorem

Let H be a connected graph.

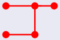
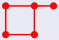
A dichotomy for hitting a connected minor



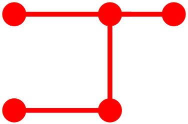


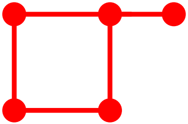
Theorem

Let H *be a* *connected* *graph.*
The $\{H\}$-M-Deletion *problem is solvable in time*

- $2^{\mathcal{O}(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$, *if* $H \preceq_{\mathsf{c}}$  *or* $H \preceq_{\mathsf{c}}$  .

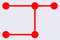
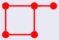
A dichotomy for hitting a connected minor



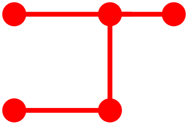
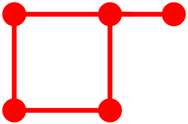


Theorem

Let H be a connected graph.
The $\{H\}$-M-DELETION problem is solvable in time

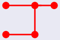
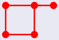
- $2^{\mathcal{O}(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$, *if* $H \preceq_c$  *or* $H \preceq_c$ .
- $2^{\mathcal{O}(\mathrm{tw} \cdot \log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$, *otherwise.*

A dichotomy for hitting a connected minor

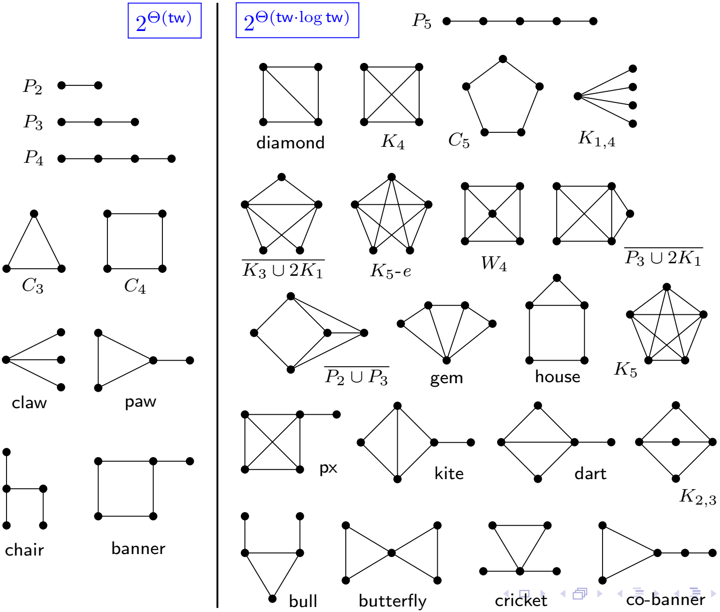
Theorem

Let H *be a connected graph.*
The $\{H\}$-M-DELETION *problem is solvable in time*

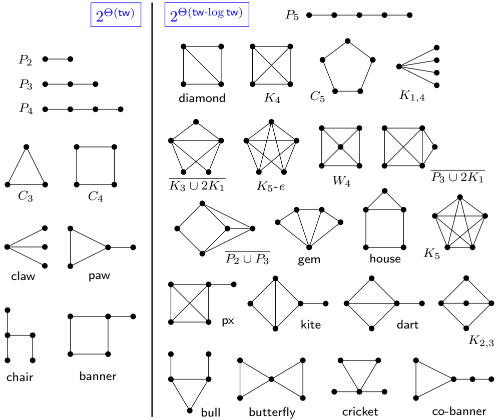
- $2^{\mathcal{O}(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$, *if* $H \preceq_{\mathrm{c}}$  *or* $H \preceq_{\mathrm{c}}$ .
- $2^{\mathcal{O}(\mathrm{tw} \cdot \log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$, *otherwise.*

In both cases, the running time is asymptotically optimal under the ETH.

Complexity of hitting a single connected minor H

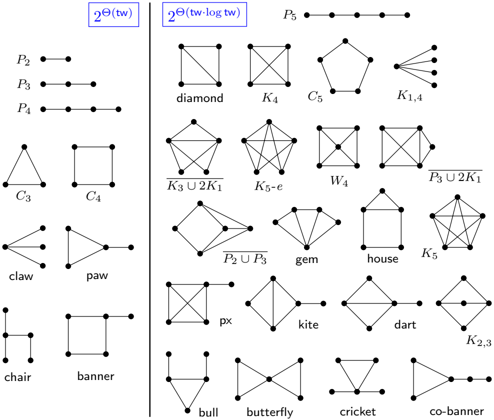


A compact statement for a single connected graph

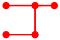


All these cases can be succinctly described as follows:

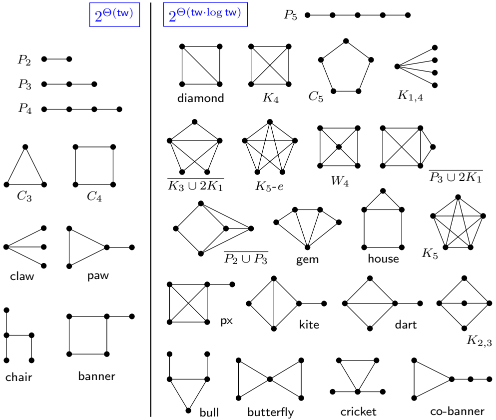
A compact statement for a single connected graph



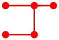
All these cases can be succinctly described as follows:

- All graphs on the left are contractions of  or 

A compact statement for a single connected graph



All these cases can be succinctly described as follows:

- All graphs on the left are contractions of  or 
- All graphs on the right are not contractions of  or 

We have three types of results

We have three types of results

1. $\boxed{\text{General algorithms}}$
   - For every $\mathcal{F}$: time $2^{2^{\mathcal{O}(\text{tw}\cdot\log\text{tw})}} \cdot n^{\mathcal{O}(1)}$.
   - $\mathcal{F}$ connected + planar: time $2^{\mathcal{O}(\text{tw}\cdot\log\text{tw})} \cdot n^{\mathcal{O}(1)}$.
   - $\mathcal{F}$ connected ~~+ planar~~: time $2^{\mathcal{O}(\text{tw}\cdot\log\text{tw})} \cdot n^{\mathcal{O}(1)}$.
   - G planar + $\mathcal{F}$ connected: time $2^{\mathcal{O}(\text{tw})} \cdot n^{\mathcal{O}(1)}$.

We have three types of results

1. General algorithms
   - For every $\mathcal{F}$: time $2^{2^{\mathcal{O}(\mathrm{tw} \cdot \log \mathrm{tw})}} \cdot n^{\mathcal{O}(1)}$.
   - $\mathcal{F}$ connected + planar: time $2^{\mathcal{O}(\mathrm{tw} \cdot \log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.
   - $\mathcal{F}$ connected ~~+ planar~~: time $2^{\mathcal{O}(\mathrm{tw} \cdot \log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.
   - G planar + $\mathcal{F}$ connected: time $2^{\mathcal{O}(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.

2. Ad-hoc single-exponential algorithms
   - Some use "typical" dynamic programming.
   - Some use the rank-based approach. [Bodlaender, Cygan, Kratsch, Nederlof. 2013]

We have three types of results

1. General algorithms
   - For every $\mathcal{F}$: time $2^{2^{\mathcal{O}(\mathrm{tw}\cdot\log \mathrm{tw})}} \cdot n^{\mathcal{O}(1)}$.
   - $\mathcal{F}$ connected + planar: time $2^{\mathcal{O}(\mathrm{tw}\cdot\log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.
   - $\mathcal{F}$ connected ~~+ planar~~: time $2^{\mathcal{O}(\mathrm{tw}\cdot\log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.
   - G planar + $\mathcal{F}$ connected: time $2^{\mathcal{O}(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.

2. Ad-hoc single-exponential algorithms
   - Some use "typical" dynamic programming.
   - Some use the rank-based approach. [Bodlaender, Cygan, Kratsch, Nederlof. 2013]

3. Lower bounds under the ETH
   - $2^{o(\mathrm{tw})}$ is "easy".
   - $2^{o(\mathrm{tw}\cdot\log \mathrm{tw})}$ is much more involved and we get ideas from:

     [Lokshtanov, Marx, Saurabh. 2011] [Marcin Pilipczuk. 2017] [Bonnet, Brettell, Kwon, Marx. 2017]

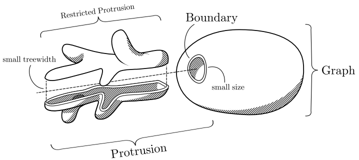
Some ideas of the general algorithms

- For every $\mathcal{F}$: time $2^{2^{\mathcal{O}(\mathsf{tw} \cdot \log \mathsf{tw})}} \cdot n^{\mathcal{O}(1)}$.
- $\mathcal{F}$ connected + planar: time $2^{\mathcal{O}(\mathsf{tw} \cdot \log \mathsf{tw})} \cdot n^{\mathcal{O}(1)}$.
- G planar + $\mathcal{F}$ connected: time $2^{\mathcal{O}(\mathsf{tw})} \cdot n^{\mathcal{O}(1)}$.

Some ideas of the general algorithms

- For every $\mathcal{F}$: time $2^{2^{\mathcal{O}(\mathrm{tw}\cdot\log\mathrm{tw})}} \cdot n^{\mathcal{O}(1)}$.
- $\mathcal{F}$ connected + planar: time $2^{\mathcal{O}(\mathrm{tw}\cdot\log\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.
- G planar + $\mathcal{F}$ connected: time $2^{\mathcal{O}(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.

We build on the machinery of boundaried graphs and representatives:



[Bodlaender, Fomin, Lokshtanov, Penninkx, Saurabh, Thilikos. 2009]

[Fomin, Lokshtanov, Saurabh, Thilikos. 2010]

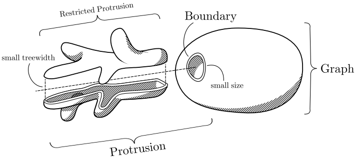
[Kim, Langer, Paul, Reidl, Rossmanith, S., Sikdar. 2013]

[Garnero, Paul, S., Thilikos. 2014]

Some ideas of the general algorithms

- For every $\mathcal{F}$: time $2^{2^{\mathcal{O}(\mathrm{tw}\cdot\log \mathrm{tw})}} \cdot n^{\mathcal{O}(1)}$.
- $\mathcal{F}$ connected + planar: time $2^{\mathcal{O}(\mathrm{tw}\cdot\log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.
- G planar + $\mathcal{F}$ connected: time $2^{\mathcal{O}(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.

We build on the machinery of boundaried graphs and representatives:



[Bodlaender, Fomin, Lokshtanov, Penninkx, Saurabh, Thilikos. 2009]

[Fomin, Lokshtanov, Saurabh, Thilikos. 2010]

[Kim, Langer, Paul, Reidl, Rossmanith, S., Sikdar. 2013]

[Garnero, Paul, S., Thilikos. 2014]

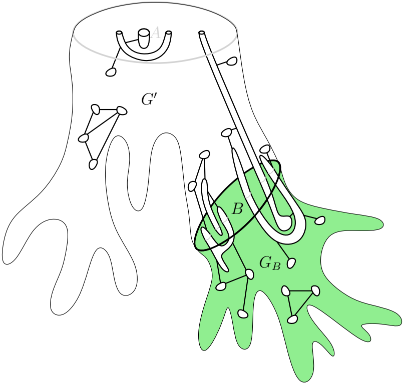
- $\mathcal{F}$ connected ~~+ planar~~: time $2^{\mathcal{O}(\mathrm{tw}\cdot\log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.

  Extra: Bidimensionality, irrelevant vertices, protrusion decompositions...

skip

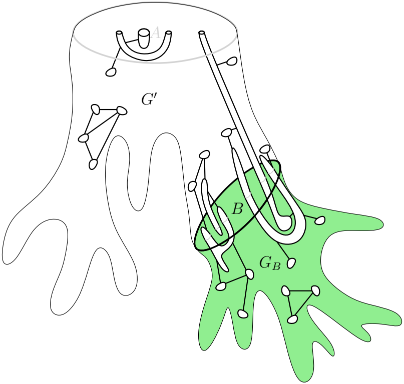
Algorithm for a general collection $\mathcal{F}$

- We see G as a t-boundaried graph.



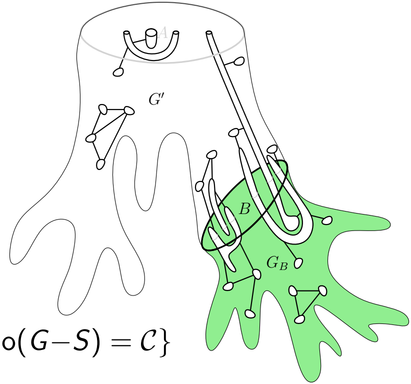
Algorithm for a general collection $\mathcal{F}$

- We see G as a t-boundaried graph.
- folio of G: set of all its $\mathcal{F}$-minor-free minors, up to size $\mathcal{O}_{\mathcal{F}}(t)$.



Algorithm for a general collection $\mathcal{F}$

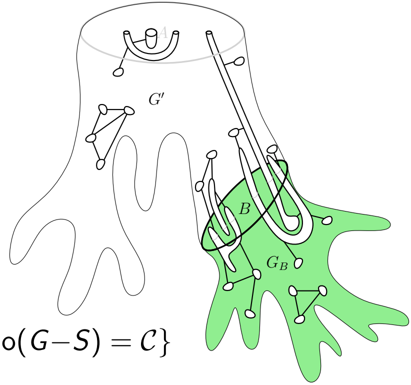
- We see G as a t-boundaried graph.
- folio of G: set of all its $\mathcal{F}$-minor-free minors, up to size $\mathcal{O}_{\mathcal{F}}(t)$.
- We compute, using DP over a tree decomposition of G, the following parameter for every folio $\mathcal{C}$:



$$\mathbf{p}(G, \mathcal{C}) = \min\{|S| \ : \ S \subseteq V(G) \wedge \text{folio}(G - S) = \mathcal{C}\}$$

Algorithm for a general collection $\mathcal{F}$

- We see G as a t-boundaried graph.
- folio of G: set of all its $\mathcal{F}$-minor-free minors, up to size $\mathcal{O}_{\mathcal{F}}(t)$.
- We compute, using DP over a tree decomposition of G, the following parameter for every folio $\mathcal{C}$:

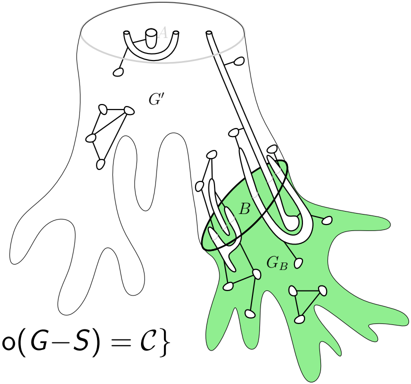


$$\mathbf{p}(G, \mathcal{C}) = \min\{|S| \ : \ S \subseteq V(G) \wedge \text{folio}(G - S) = \mathcal{C}\}$$

- For every t-boundaried graph G,

$$|\text{folio}(G)| = \mathcal{O}_{\mathcal{F}}(1) \cdot \binom{t^2}{t} = 2^{\mathcal{O}_{\mathcal{F}}(t \cdot \log t)}$$

Algorithm for a general collection $\mathcal{F}$



- We see G as a t-boundaried graph.
- folio of G: set of all its $\mathcal{F}$-minor-free minors, up to size $\mathcal{O}_{\mathcal{F}}(t)$.
- We compute, using DP over a tree decomposition of G, the following parameter for every folio $\mathcal{C}$:

$$\mathbf{p}(G, \mathcal{C}) = \min\{|S| \ : \ S \subseteq V(G) \wedge \text{folio}(G - S) = \mathcal{C}\}$$

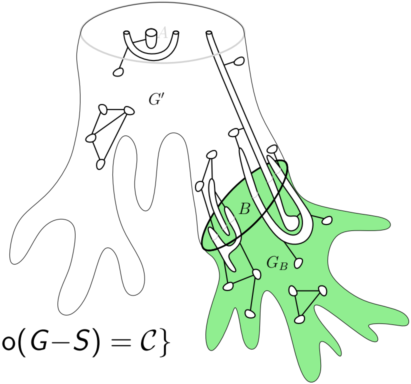
- For every t-boundaried graph G,

$$|\text{folio}(G)| = \mathcal{O}_{\mathcal{F}}(1) \cdot \binom{t^2}{t} = 2^{\mathcal{O}_{\mathcal{F}}(t \cdot \log t)}$$

- The number of distinct folios is $2^{2^{\mathcal{O}_{\mathcal{F}}(t \cdot \log t)}}$.

Algorithm for a general collection $\mathcal{F}$

- We see G as a t-boundaried graph.
- folio of G: set of all its $\mathcal{F}$-minor-free minors, up to size $\mathcal{O}_{\mathcal{F}}(t)$.
- We compute, using DP over a tree decomposition of G, the following parameter for every folio $\mathcal{C}$:



$$\mathbf{p}(G, \mathcal{C}) = \min\{|S| \ : \ S \subseteq V(G) \wedge \text{folio}(G-S) = \mathcal{C}\}$$

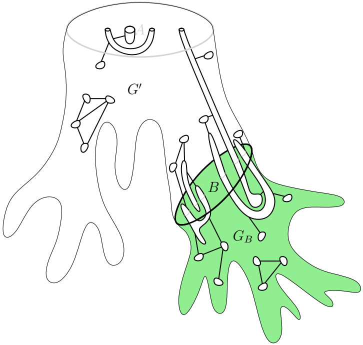
- For every t-boundaried graph G,

$$|\text{folio}(G)| = \mathcal{O}_{\mathcal{F}}(1) \cdot \binom{t^2}{t} = 2^{\mathcal{O}_{\mathcal{F}}(t \cdot \log t)}$$

- The number of distinct folios is $2^{2^{\mathcal{O}_{\mathcal{F}}(t \cdot \log t)}}$.
- This gives an algorithm running in time $2^{2^{\mathcal{O}_{\mathcal{F}}(\text{tw} \cdot \log \text{tw})}} \cdot n^{\mathcal{O}(1)}$.

skip

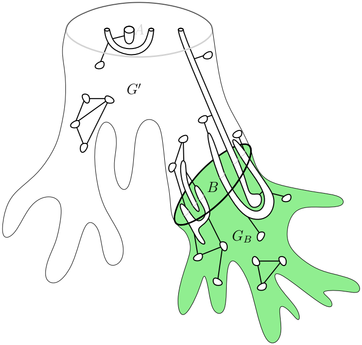
Algorithm for a connected and planar collection $\mathcal{F}$



Algorithm for a connected and planar collection $\mathcal{F}$

- For a fixed $\mathcal{F}$, we define an equivalence relation $\equiv^{(\mathcal{F},t)}$ on t-boundaried graphs:

$$G_1 \equiv^{(\mathcal{F},t)} G_2 \quad \text{if } \forall G' \in \mathcal{B}^t,$$
$$\mathcal{F} \preceq_{\mathrm{m}} G' \oplus G_1 \iff \mathcal{F} \preceq_{\mathrm{m}} G' \oplus G_2.$$

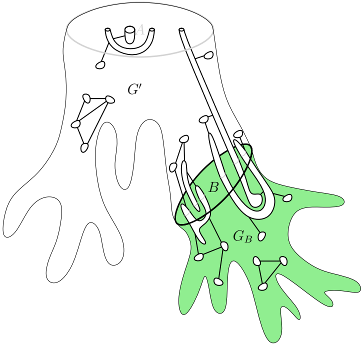


Algorithm for a connected and planar collection $\mathcal{F}$

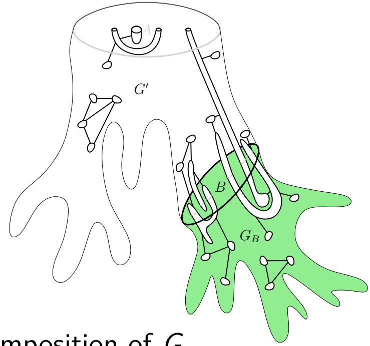
- For a fixed $\mathcal{F}$, we define an equivalence relation $\equiv^{(\mathcal{F},t)}$ on t-boundaried graphs:

$$G_1 \equiv^{(\mathcal{F},t)} G_2 \quad \text{if } \forall G' \in \mathcal{B}^t,$$
$$\mathcal{F} \preceq_m G' \oplus G_1 \iff \mathcal{F} \preceq_m G' \oplus G_2.$$

- $\mathcal{R}^{(\mathcal{F},t)}$: set of minimum-size representatives of $\equiv^{(\mathcal{F},t)}$.



Algorithm for a connected and planar collection $\mathcal{F}$



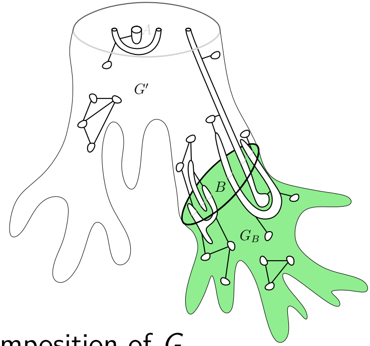
- For a fixed $\mathcal{F}$, we define an equivalence relation $\equiv^{(\mathcal{F},t)}$ on t-boundaried graphs:

$$G_1 \equiv^{(\mathcal{F},t)} G_2 \quad \text{if } \forall G' \in \mathcal{B}^t,$$
$$\mathcal{F} \preceq_{\mathrm{m}} G' \oplus G_1 \iff \mathcal{F} \preceq_{\mathrm{m}} G' \oplus G_2.$$

- $\mathcal{R}^{(\mathcal{F},t)}$: set of minimum-size representatives of $\equiv^{(\mathcal{F},t)}$.

- We compute, using DP over a tree decomposition of G, the following parameter for every representative R:

$$\mathbf{p}(G,R) = \min\{|S| \ : \ S \subseteq V(G) \ \wedge \ \mathrm{rep}_{\mathcal{F},t}(G-S) = R\}$$



- For a fixed $\mathcal{F}$, we define an equivalence relation $\equiv^{(\mathcal{F},t)}$ on t-boundaried graphs:

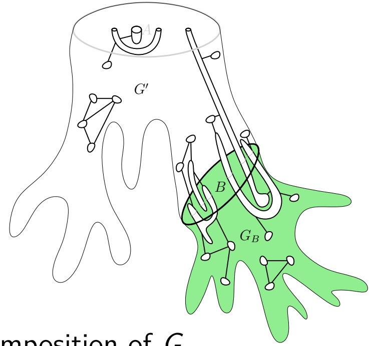
$$G_1 \equiv^{(\mathcal{F},t)} G_2 \quad \text{if } \forall G' \in \mathcal{B}^t,$$
$$\mathcal{F} \preceq_{\mathrm{m}} G' \oplus G_1 \iff \mathcal{F} \preceq_{\mathrm{m}} G' \oplus G_2.$$

- $\mathcal{R}^{(\mathcal{F},t)}$: set of minimum-size representatives of $\equiv^{(\mathcal{F},t)}$.

- We compute, using DP over a tree decomposition of G, the following parameter for every representative R:

$$\mathbf{p}(G,R) = \min\{|S| \ : \ S \subseteq V(G) \ \wedge \ \mathrm{rep}_{\mathcal{F},t}(G-S) = R\}$$

- The number of representatives is $|\mathcal{R}^{(\mathcal{F},t)}| = 2^{\mathcal{O}_{\mathcal{F}}(t \cdot \log t)}$.



- For a fixed $\mathcal{F}$, we define an equivalence relation $\equiv^{(\mathcal{F},t)}$ on t-boundaried graphs:

$$G_1 \equiv^{(\mathcal{F},t)} G_2 \quad \text{if } \forall G' \in \mathcal{B}^t,$$
$$\mathcal{F} \preceq_{\mathrm{m}} G' \oplus G_1 \iff \mathcal{F} \preceq_{\mathrm{m}} G' \oplus G_2.$$

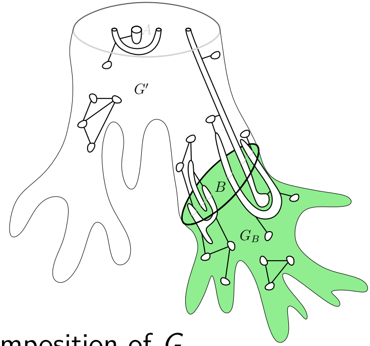
- $\mathcal{R}^{(\mathcal{F},t)}$: set of minimum-size representatives of $\equiv^{(\mathcal{F},t)}$.

- We compute, using DP over a tree decomposition of G, the following parameter for every representative R:

$$\mathbf{p}(G,R) = \min\{|S| \ : \ S \subseteq V(G) \ \wedge \ \mathrm{rep}_{\mathcal{F},t}(G-S) = R\}$$

- The number of representatives is $|\mathcal{R}^{(\mathcal{F},t)}| = 2^{\mathcal{O}_{\mathcal{F}}(t \cdot \log t)}$. $\boxed{\text{Planarity!}}$
  # labeled graphs of size $\leq t$ and tw $\leq h$ is $2^{\mathcal{O}_h(t \cdot \log t)}$. [Baste, Noy, S. 2017]

Algorithm for a connected and planar collection $\mathcal{F}$



- For a fixed $\mathcal{F}$, we define an equivalence relation $\equiv^{(\mathcal{F},t)}$ on t-boundaried graphs:

  $$G_1 \equiv^{(\mathcal{F},t)} G_2 \quad \text{if } \forall G' \in \mathcal{B}^t,$$
  $$\mathcal{F} \preceq_{\mathrm{m}} G' \oplus G_1 \iff \mathcal{F} \preceq_{\mathrm{m}} G' \oplus G_2.$$

- $\mathcal{R}^{(\mathcal{F},t)}$: set of minimum-size representatives of $\equiv^{(\mathcal{F},t)}$.

- We compute, using DP over a tree decomposition of G, the following parameter for every representative R:

  $$\mathbf{p}(G,R) = \min\{|S| \ : \ S \subseteq V(G) \ \wedge \ \mathrm{rep}_{\mathcal{F},t}(G-S) = R\}$$

- The number of representatives is $|\mathcal{R}^{(\mathcal{F},t)}| = 2^{\mathcal{O}_{\mathcal{F}}(t\cdot\log t)}$. $\boxed{\text{Planarity!}}$

  $\#$ labeled graphs of size $\leq t$ and tw $\leq h$ is $2^{\mathcal{O}_h(t\cdot\log t)}$. [Baste, Noy, S. 2017]

- This gives an algorithm running in time $2^{\mathcal{O}_{\mathcal{F}}(\mathsf{tw}\cdot\log\mathsf{tw})} \cdot n^{\mathcal{O}(1)}$.

skip

Algorithm for any connected collection $\mathcal{F}$

Algorithm for any connected collection $\mathcal{F}$

- $\mathcal{R}^{(\mathcal{F},t)}$: set of minimum-size representatives of $\equiv^{(\mathcal{F},t)}$.

Algorithm for any connected collection $\mathcal{F}$

- $\mathcal{R}^{(\mathcal{F},t)}$: set of minimum-size representatives of $\equiv^{(\mathcal{F},t)}$.
- Suppose that we can prove that, for every $R \in \mathcal{R}^{(\mathcal{F},t)}$,

$$|V(R)| = \mathcal{O}_{\mathcal{F}}(t).$$

Algorithm for any connected collection $\mathcal{F}$

- $\mathcal{R}^{(\mathcal{F},t)}$: set of minimum-size representatives of $\equiv^{(\mathcal{F},t)}$.
- Suppose that we can prove that, for every $R \in \mathcal{R}^{(\mathcal{F},t)}$,
$$|V(R)| = \mathcal{O}_{\mathcal{F}}(t).$$
- We are done: $|\mathcal{R}^{(\mathcal{F},t)}| = 2^{\mathcal{O}_{\mathcal{F}}(t \cdot \log t)}$ and the same DP works!

Algorithm for any connected collection $\mathcal{F}$

- $\mathcal{R}^{(\mathcal{F},t)}$: set of minimum-size representatives of $\equiv^{(\mathcal{F},t)}$.
- Suppose that we can prove that, for every $R \in \mathcal{R}^{(\mathcal{F},t)}$,

$$|V(R)| = \mathcal{O}_{\mathcal{F}}(t).$$

- We are done: $|\mathcal{R}^{(\mathcal{F},t)}| = 2^{\mathcal{O}_{\mathcal{F}}(t \cdot \log t)}$ and the same DP works!
- Flat Wall Theorem:

Algorithm for any connected collection $\mathcal{F}$

- $\mathcal{R}^{(\mathcal{F},t)}$: set of minimum-size representatives of $\equiv^{(\mathcal{F},t)}$.
- Suppose that we can prove that, for every $R \in \mathcal{R}^{(\mathcal{F},t)}$,
$$|V(R)| = \mathcal{O}_{\mathcal{F}}(t).$$
- We are done: $|\mathcal{R}^{(\mathcal{F},t)}| = 2^{\mathcal{O}_{\mathcal{F}}(t \cdot \log t)}$ and the same DP works!
- Flat Wall Theorem: As R is $\mathcal{F}$-minor-free, if $\mathrm{tw}(R \setminus B) > c_{\mathcal{F}}$,

Algorithm for any connected collection $\mathcal{F}$

- $\mathcal{R}^{(\mathcal{F},t)}$: set of minimum-size representatives of $\equiv^{(\mathcal{F},t)}$.
- Suppose that we can prove that, for every $R \in \mathcal{R}^{(\mathcal{F},t)}$,
$$|V(R)| = \mathcal{O}_{\mathcal{F}}(t).$$
- We are done: $|\mathcal{R}^{(\mathcal{F},t)}| = 2^{\mathcal{O}_{\mathcal{F}}(t \cdot \log t)}$ and the same DP works!
- Flat Wall Theorem: As R is $\mathcal{F}$-minor-free, if $\mathrm{tw}(R \setminus B) > c_{\mathcal{F}}$, $R \setminus B$ contains a large flat wall,

Algorithm for any connected collection $\mathcal{F}$

- $\mathcal{R}^{(\mathcal{F},t)}$: set of minimum-size representatives of $\equiv^{(\mathcal{F},t)}$.
- Suppose that we can prove that, for every $R \in \mathcal{R}^{(\mathcal{F},t)}$,
$$|V(R)| = \mathcal{O}_{\mathcal{F}}(t).$$
- We are done: $|\mathcal{R}^{(\mathcal{F},t)}| = 2^{\mathcal{O}_{\mathcal{F}}(t \cdot \log t)}$ and the same DP works!
- Flat Wall Theorem: As R is $\mathcal{F}$-minor-free, if $\mathrm{tw}(R \setminus B) > c_{\mathcal{F}}$, $R \setminus B$ contains a large flat wall, where we can find an irrelevant vertex.

Algorithm for any connected collection $\mathcal{F}$

- $\mathcal{R}^{(\mathcal{F},t)}$: set of minimum-size representatives of $\equiv^{(\mathcal{F},t)}$.
- Suppose that we can prove that, for every $R \in \mathcal{R}^{(\mathcal{F},t)}$,
$$|V(R)| = \mathcal{O}_{\mathcal{F}}(t).$$
- We are done: $|\mathcal{R}^{(\mathcal{F},t)}| = 2^{\mathcal{O}_{\mathcal{F}}(t \cdot \log t)}$ and the same DP works!
- Flat Wall Theorem: As R is $\mathcal{F}$-minor-free, if $\mathrm{tw}(R \setminus B) > c_{\mathcal{F}}$, $R \setminus B$ contains a large flat wall, where we can find an irrelevant vertex.
- R has a treewidth modulator of size $\mathcal{O}(t)$ containing its boundary B.

Algorithm for any connected collection $\mathcal{F}$

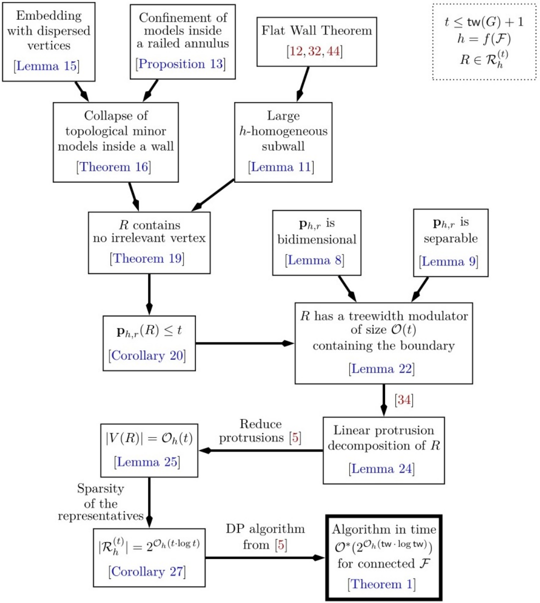
- $\mathcal{R}^{(\mathcal{F},t)}$: set of minimum-size representatives of $\equiv^{(\mathcal{F},t)}$.
- Suppose that we can prove that, for every $R \in \mathcal{R}^{(\mathcal{F},t)}$,
$$|V(R)| = \mathcal{O}_{\mathcal{F}}(t).$$
- We are done: $|\mathcal{R}^{(\mathcal{F},t)}| = 2^{\mathcal{O}_{\mathcal{F}}(t \cdot \log t)}$ and the same DP works!
- Flat Wall Theorem: As R is $\mathcal{F}$-minor-free, if $\mathrm{tw}(R \setminus B) > c_{\mathcal{F}}$, $R \setminus B$ contains a large flat wall, where we can find an irrelevant vertex.
- R has a treewidth modulator of size $\mathcal{O}(t)$ containing its boundary B.
- We can then find a linear protrusion decomposition of R.

Algorithm for any connected collection $\mathcal{F}$

- $\mathcal{R}^{(\mathcal{F},t)}$: set of minimum-size representatives of $\equiv^{(\mathcal{F},t)}$.
- Suppose that we can prove that, for every $R \in \mathcal{R}^{(\mathcal{F},t)}$,
$$|V(R)| = \mathcal{O}_{\mathcal{F}}(t).$$
- We are done: $|\mathcal{R}^{(\mathcal{F},t)}| = 2^{\mathcal{O}_{\mathcal{F}}(t\cdot\log t)}$ and the same DP works!
- Flat Wall Theorem: As R is $\mathcal{F}$-minor-free, if $\mathrm{tw}(R \setminus B) > c_{\mathcal{F}}$, $R \setminus B$ contains a large flat wall, where we can find an irrelevant vertex.
- R has a treewidth modulator of size $\mathcal{O}(t)$ containing its boundary B.
- We can then find a linear protrusion decomposition of R.
- By applying protrusion reduction, we obtain that $|V(R)| = \mathcal{O}_{\mathcal{F}}(t)$.

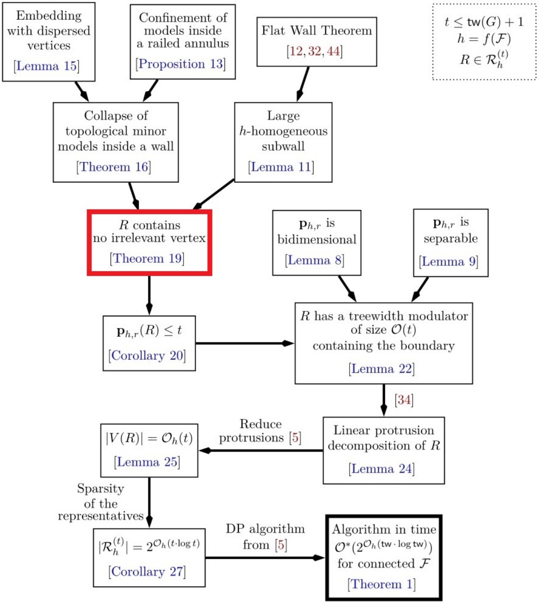
Algorithm for any connected collection $\mathcal{F}$

Algorithm for any connected collection $\mathcal{F}$



skip

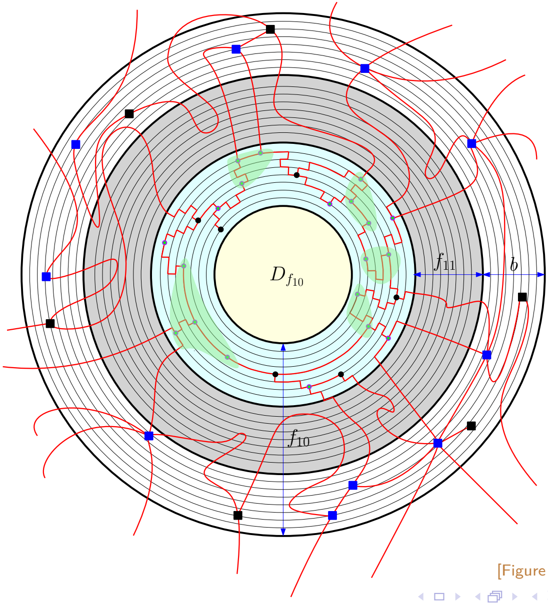
Algorithm for any connected collection $\mathcal{F}$



skip

Hard part: finding an irrelevant vertex inside a flat wall

Hard part: finding an irrelevant vertex inside a flat wall



skip

[Figure by Dimitrios M. Thilikos]

Algorithm when the input graph G is planar

- $\boxed{\text{Idea}}$ get an improved bound on $|\mathcal{R}^{(\mathcal{F},t)}|$.

Algorithm when the input graph G is planar

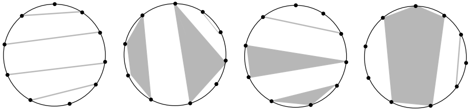
- $\boxed{\text{Idea}}$ get an improved bound on $|\mathcal{R}^{(\mathcal{F},t)}|$.
- We use a sphere-cut decomposition of the input planar graph G.

[Seymour, Thomas. 1994] [Dorn, Penninkx, Bodlaender, Fomin. 2010]

Algorithm when the input graph G is planar

- $\boxed{\text{Idea}}$ get an improved bound on $|\mathcal{R}^{(\mathcal{F},t)}|$.
- We use a sphere-cut decomposition of the input planar graph G.

  [Seymour, Thomas. 1994] [Dorn, Penninkx, Bodlaender, Fomin. 2010]
- Nice topological properties: each separator corresponds to a noose.

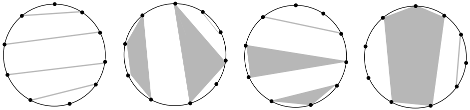


Algorithm when the input graph G is planar

- $\boxed{\text{Idea}}$ get an improved bound on $|\mathcal{R}^{(\mathcal{F},t)}|$.
- We use a sphere-cut decomposition of the input planar graph G.

  [Seymour, Thomas. 1994] [Dorn, Penninkx, Bodlaender, Fomin. 2010]

- Nice topological properties: each separator corresponds to a noose.



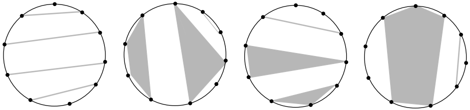
- The number of representatives is $|\mathcal{R}^{(\mathcal{F},t)}| = 2^{\mathcal{O}_{\mathcal{F}}(t)}$.

  Number of planar triangulations on t vertices is $2^{\mathcal{O}(t)}$. [Tutte. 1962]

Algorithm when the input graph G is planar

- $\boxed{\text{Idea}}$ get an improved bound on $|\mathcal{R}^{(\mathcal{F},t)}|$.
- We use a sphere-cut decomposition of the input planar graph G.

  [Seymour, Thomas. 1994] [Dorn, Penninkx, Bodlaender, Fomin. 2010]
- Nice topological properties: each separator corresponds to a noose.

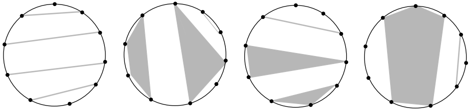


- The number of representatives is $|\mathcal{R}^{(\mathcal{F},t)}| = 2^{\mathcal{O}_{\mathcal{F}}(t)}$.
  Number of planar triangulations on t vertices is $2^{\mathcal{O}(t)}$. [Tutte. 1962]
- This gives an algorithm running in time $2^{\mathcal{O}_{\mathcal{F}}(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.

Algorithm when the input graph G is planar

- $\boxed{\text{Idea}}$ get an improved bound on $|\mathcal{R}^{(\mathcal{F},t)}|$.
- We use a sphere-cut decomposition of the input planar graph G.

  [Seymour, Thomas. 1994] [Dorn, Penninkx, Bodlaender, Fomin. 2010]
- Nice topological properties: each separator corresponds to a noose.



- The number of representatives is $|\mathcal{R}^{(\mathcal{F},t)}| = 2^{\mathcal{O}_{\mathcal{F}}(t)}$.
  Number of planar triangulations on t vertices is $2^{\mathcal{O}(t)}$. [Tutte. 1962]
- This gives an algorithm running in time $2^{\mathcal{O}_{\mathcal{F}}(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.
- We can extend this algorithm to input graphs G embedded in arbitrary surfaces by using surface-cut decompositions. skip [Rué, S., Thilikos. 2014]

What's next about $\mathcal{F}$-Deletion?

What's next about $\mathcal{F}$-Deletion?

- $\boxed{\text{Goal}}$ classify the (asymptotically) tight complexity of $\mathcal{F}$-M-Deletion and $\mathcal{F}$-TM-Deletion for every family $\mathcal{F}$.

What's next about $\mathcal{F}$-Deletion?

- $\boxed{\text{Goal}}$ classify the (asymptotically) tight complexity of $\mathcal{F}$-M-Deletion and $\mathcal{F}$-TM-Deletion for every family $\mathcal{F}$.
- Concerning the $\boxed{\text{minor}}$ version:

What's next about $\mathcal{F}$-Deletion?

- $\boxed{\text{Goal}}$ classify the (asymptotically) tight complexity of $\mathcal{F}$-M-Deletion and $\mathcal{F}$-TM-Deletion for every family $\mathcal{F}$.
- Concerning the $\boxed{\text{minor}}$ version:
  - We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).

What's next about $\mathcal{F}$-Deletion?

- $\boxed{\text{Goal}}$ classify the (asymptotically) tight complexity of $\mathcal{F}$-M-Deletion and $\mathcal{F}$-TM-Deletion for every family $\mathcal{F}$.
- Concerning the $\boxed{\text{minor}}$ version:
  - We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).
  - Missing: When $|\mathcal{F}| \geq 2$ (connected): $2^{\Theta(\mathrm{tw})}$ or $2^{\Theta(\mathrm{tw} \cdot \log \mathrm{tw})}$?

What's next about $\mathcal{F}$-Deletion?

- $\boxed{\text{Goal}}$ classify the (asymptotically) tight complexity of $\mathcal{F}$-M-Deletion and $\mathcal{F}$-TM-Deletion for every family $\mathcal{F}$.
- Concerning the $\boxed{\text{minor}}$ version:
  - We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).
  - Missing: When $|\mathcal{F}| \geq 2$ (connected): $2^{\Theta(\mathsf{tw})}$ or $2^{\Theta(\mathsf{tw} \cdot \log \mathsf{tw})}$?
  - Consider families $\mathcal{F}$ containing disconnected graphs.

What's next about $\mathcal{F}$-Deletion?

- $\boxed{\text{Goal}}$ classify the (asymptotically) tight complexity of $\mathcal{F}$-M-Deletion and $\mathcal{F}$-TM-Deletion for every family $\mathcal{F}$.
- Concerning the $\boxed{\text{minor}}$ version:
  - We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).
  - Missing: When $|\mathcal{F}| \geq 2$ (connected): $2^{\Theta(\mathsf{tw})}$ or $2^{\Theta(\mathsf{tw} \cdot \log \mathsf{tw})}$?
  - Consider families $\mathcal{F}$ containing disconnected graphs.
    Deletion to genus at most g: $2^{\mathcal{O}_g(\mathsf{tw} \cdot \log \mathsf{tw})} \cdot n^{\mathcal{O}(1)}$. [Kociumaka, Pilipczuk. 2017]

What's next about $\mathcal{F}$-Deletion?

- $\boxed{\text{Goal}}$ classify the (asymptotically) tight complexity of $\mathcal{F}$-M-Deletion and $\mathcal{F}$-TM-Deletion for every family $\mathcal{F}$.
- Concerning the $\boxed{\text{minor}}$ version:
  - We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).
  - Missing: When $|\mathcal{F}| \geq 2$ (connected): $2^{\Theta(\mathsf{tw})}$ or $2^{\Theta(\mathsf{tw} \cdot \log \mathsf{tw})}$?
  - Consider families $\mathcal{F}$ containing disconnected graphs.
    Deletion to genus at most g: $2^{\mathcal{O}_g(\mathsf{tw} \cdot \log \mathsf{tw})} \cdot n^{\mathcal{O}(1)}$. [Kociumaka, Pilipczuk. 2017]
- Concerning the $\boxed{\text{topological minor}}$ version:

What's next about $\mathcal{F}$-Deletion?

- $\boxed{\text{Goal}}$ classify the (asymptotically) tight complexity of $\mathcal{F}$-M-Deletion and $\mathcal{F}$-TM-Deletion for every family $\mathcal{F}$.
- Concerning the $\boxed{\text{minor}}$ version:
  - We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).
  - Missing: When $|\mathcal{F}| \geq 2$ (connected): $2^{\Theta(\mathsf{tw})}$ or $2^{\Theta(\mathsf{tw} \cdot \log \mathsf{tw})}$?
  - Consider families $\mathcal{F}$ containing disconnected graphs.
    Deletion to genus at most g: $2^{\mathcal{O}_g(\mathsf{tw} \cdot \log \mathsf{tw})} \cdot n^{\mathcal{O}(1)}$. [Kociumaka, Pilipczuk. 2017]
- Concerning the $\boxed{\text{topological minor}}$ version:
  - Dichotomy for $\{H\}$-TM-Deletion when H connected (+planar).

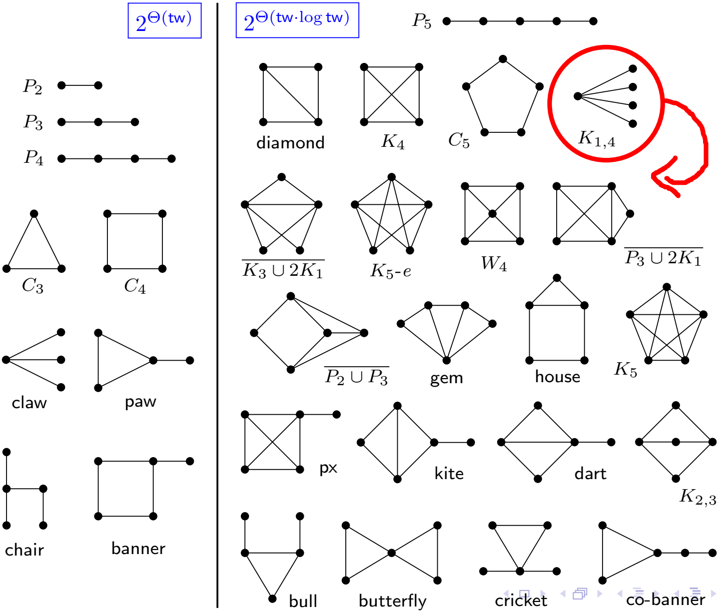
What's next about $\mathcal{F}$-Deletion?

- $\boxed{\text{Goal}}$ classify the (asymptotically) tight complexity of $\mathcal{F}$-M-Deletion and $\mathcal{F}$-TM-Deletion for every family $\mathcal{F}$.
- Concerning the $\boxed{\text{minor}}$ version:
  - We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).
  - Missing: When $|\mathcal{F}| \geq 2$ (connected): $2^{\Theta(\mathsf{tw})}$ or $2^{\Theta(\mathsf{tw} \cdot \log \mathsf{tw})}$?
  - Consider families $\mathcal{F}$ containing disconnected graphs.
    Deletion to genus at most g: $2^{\mathcal{O}_g(\mathsf{tw} \cdot \log \mathsf{tw})} \cdot n^{\mathcal{O}(1)}$. [Kociumaka, Pilipczuk. 2017]
- Concerning the $\boxed{\text{topological minor}}$ version:
  - Dichotomy for $\{H\}$-TM-Deletion when H connected (+planar).
  - We do not know if there exists some $\mathcal{F}$ such that $\mathcal{F}$-TM-Deletion cannot be solved in time $2^{o(\mathsf{tw}^2)} \cdot n^{\mathcal{O}(1)}$ under the ETH.

What's next about $\mathcal{F}$-Deletion?

- $\boxed{\text{Goal}}$ classify the (asymptotically) tight complexity of $\mathcal{F}$-M-Deletion and $\mathcal{F}$-TM-Deletion for every family $\mathcal{F}$.
- Concerning the $\boxed{\text{minor}}$ version:
  - We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).
  - Missing: When $|\mathcal{F}| \geq 2$ (connected): $2^{\Theta(\mathsf{tw})}$ or $2^{\Theta(\mathsf{tw} \cdot \log \mathsf{tw})}$?
  - Consider families $\mathcal{F}$ containing disconnected graphs.
    Deletion to genus at most g: $2^{\mathcal{O}_g(\mathsf{tw} \cdot \log \mathsf{tw})} \cdot n^{\mathcal{O}(1)}$. [Kociumaka, Pilipczuk. 2017]
- Concerning the $\boxed{\text{topological minor}}$ version:
  - Dichotomy for $\{H\}$-TM-Deletion when H connected (+planar).
  - We do not know if there exists some $\mathcal{F}$ such that $\mathcal{F}$-TM-Deletion cannot be solved in time $2^{o(\mathsf{tw}^2)} \cdot n^{\mathcal{O}(1)}$ under the ETH.
  - $\boxed{\text{Conjecture}}$ For every (connected) family $\mathcal{F}$, the $\mathcal{F}$-TM-Deletion problem is solvable in time $2^{\mathcal{O}(\mathsf{tw} \cdot \log \mathsf{tw})} \cdot n^{\mathcal{O}(1)}$.

For topological minors, there is (at least) one change



Gràcies!