Hitting (topological) minors on bounded treewidth graphs - part I

Julien Baste\textsuperscript{1} \quad Ignasi Sau\textsuperscript{2} \quad Dimitrios M. Thilikos\textsuperscript{2,3}

\textbf{GT 2018, Nyborg, Denmark}  
August 29 - September 1, 2018

\textsuperscript{1} Sorbonne Université, Laboratoire d’Informatique de Paris 6, France  
\textsuperscript{2} CNRS, LIRMM, Université de Montpellier, France  
\textsuperscript{3} Dept. of Maths, National and Kapodistrian University of Athens, Greece

[arXiv 1704.07284]
Treewidth behaves very well algorithmically

Monadic Second Order Logic (MSOL):

Graph logic that allows quantification over sets of vertices and edges.

Example:

\[ \text{DomSet}(S) : \forall v \in V(G) \cup S, \exists u \in S : \{u, v\} \in E(G) \]

Theorem (Courcelle, 1990)

Every problem expressible in MSOL can be solved in time \( f(tw) \cdot n \) 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 \( k \), ...
Treewidth behaves very well algorithmically

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

Example: \( \text{DomSet}(S) : \left[ \forall v \in V(G) \setminus S, \exists u \in S : \{u, v\} \in E(G) \right] \)
Treewidth behaves very well algorithmically

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

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

Theorem (Courcelle, 1990)

Every problem expressible in MSOL can be solved in time $f(tw) \cdot n$ 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...

\[ f(tw) \cdot n^{O(1)} \]
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!

\[ f(tw) \cdot n^{O(1)} = 2^{345678^{tw}} \cdot n^{O(1)} \]
Typically, Courcelle’s theorem allows to prove that a problem is FPT... 
... but the running time can (and must) be huge!

\[ f(tw) \cdot n^{O(1)} = 2^{345678^{tw}} \cdot n^{O(1)} \]

**Major goal** find the smallest possible function \( f(tw) \).

This is a very active area in parameterized complexity.
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!

\[ f(tw) \cdot n^{O(1)} = 2^{345678^{tw}} \cdot n^{O(1)} \]

**Major goal** find the smallest possible function \( f(tw) \).

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.
Two behaviors for problems parameterized by treewidth

Typically, FPT algorithms parameterized by treewidth are based on dynamic programming (DP) over a tree decomposition.
Two behaviors for problems parameterized by treewidth

Typically, FPT algorithms parameterized by treewidth are based on dynamic programming (DP) over a tree decomposition.

For many problems, like Vertex Cover or Dominating Set, the “natural” DP algorithms lead to (optimal) single-exponential algorithms:

\[ 2^{O(tw)} \cdot n^{O(1)}. \]
Two behaviors for problems parameterized by treewidth

Typically, FPT algorithms parameterized by treewidth are based on dynamic programming (DP) over a tree decomposition.

For many problems, like Vertex Cover or Dominating Set, the “natural” DP algorithms lead to (optimal) single-exponential algorithms:

$$2^{O(tw)} \cdot n^{O(1)}.$$ 

But for the so-called connectivity problems, like Longest Path or Steiner Tree, the “natural” DP algorithms provide only time

$$2^{O(tw \cdot \log tw)} \cdot n^{O(1)}.$$
Single-exponential algorithms on sparse graphs

On topologically structured graphs (planar, surfaces, minor-free), it is possible to solve connectivity problems in time $2^{O(tw)} \cdot n^{O(1)}$:

- **Planar graphs:**
  
  [Dorn, Penninkx, Bodlaender, Fomin. 2005]

- **Graphs on surfaces:**

  [Dorn, Fomin, Thilikos. 2006]
  
  [Rué, S., Thilikos. 2010]
  
  [Dorn, Fomin, Thilikos. 2008]
  
  [Rué, S., Thilikos. 2012]

- **Minor-free graphs:**

  [Dorn, Fomin, Thilikos. 2008]
  
  [Rué, S., Thilikos. 2012]
Single-exponential algorithms on sparse graphs

On topologically structured graphs (planar, surfaces, minor-free), it is possible to solve connectivity problems in time $2^{O(tw)} \cdot n^{O(1)}$:

- **Planar graphs:**
  [Dorn, Penninkx, Bodlaender, Fomin. 2005]

- **Graphs on surfaces:**
  [Dorn, Fomin, Thilikos. 2006]
  [Rué, S., Thilikos. 2010]
  [Dorn, Fomin, Thilikos. 2008]
  [Rué, S., Thilikos. 2012]

- **Minor-free graphs:**

**Main idea** special type of decomposition with nice topological properties:

```
partial solutions ⇐⇒ non-crossing partitions
```

\[
CN(k) = \frac{1 + 1}{2^k} \sim 4k^{3/2} / \sqrt{\pi k} \leq 4k^{5/2}
\]
Single-exponential algorithms on sparse graphs

On topologically structured graphs (planar, surfaces, minor-free), it is possible to solve connectivity problems in time $2^{O(tw)} \cdot n^{O(1)}$:

- **Planar graphs:**
  - [Dorn, Penninkx, Bodlaender, Fomin. 2005]

- **Graphs on surfaces:**
  - [Dorn, Fomin, Thilikos. 2006]
  - [Rué, S., Thilikos. 2010]
  - [Dorn, Fomin, Thilikos. 2008]
  - [Rué, S., Thilikos. 2012]

**Main idea** special type of decomposition with nice topological properties:

partial solutions $\iff$ non-crossing partitions

$$CN(k) = \frac{1}{k+1} \binom{2k}{k} \sim \frac{4^k}{\sqrt{\pi} k^{3/2}} \leq 4^k.$$
The revolution of single-exponential algorithms

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

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

\begin{center}
This was false!!
\end{center}

\textbf{Cut&Count technique:} \[\text{[Cygan, Nederlof, Pilipczuk}^2, \text{van Rooij, Wojtaszczyk. 2011]}\]

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 $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$ were optimal for connectivity problems.

This was false!!

Cut&Count technique: [Cygan, Nederlof, Pilipczuk, van Rooij, Wojtaszczyk. 2011]
Randomized single-exponential algorithms for connectivity problems.

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

Representative sets in matroids: [Fomin, Lokshtanov, Saurabh. 2014]
Do all connectivity problems admit single-exponential algorithms (on general graphs) parameterized by treewidth?
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.
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 $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$ is optimal under the ETH.

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

**ETH:** The 3-SAT problem on $n$ variables cannot be solved in time $2^{o(n)}$

[Impagliazzo, Paturi. 1999]
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 $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$ is optimal under the ETH.

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

**ETH**: The 3-SAT problem on $n$ variables cannot be solved in time $2^{o(n)}$

[Impagliazzo, Paturi. 1999]

There are other examples of such problems...
The $\mathcal{F}$-\textsc{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.

<table>
<thead>
<tr>
<th>$\mathcal{F}$-M-Deletion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong></td>
</tr>
<tr>
<td><strong>Parameter:</strong></td>
</tr>
<tr>
<td><strong>Question:</strong></td>
</tr>
</tbody>
</table>

- $\mathcal{F} = \{K_2\}$: 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 $tw$ 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 $2^{\Theta(tw)} \cdot n^{O(1)}$. 

$\mathcal{F} = \{C_3\}$: Feedback Vertex Set.
"Hardly" solvable in time $2^{\Theta(tw)} \cdot n^{O(1)}$.

$\mathcal{F} = \{K_5, K_3, 3\}$: Vertex Planarization.
Solvable in time $2^{\Theta(tw) \cdot \log tw} \cdot n^{O(1)}$.

[Jansen, Lokshtanov, Saurabh. 2014 + Pilipczuk. 2015]
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 $2^{\Theta(tw)} \cdot n^{O(1)}$.

- $\mathcal{F} = \{C_3\}$: **Feedback Vertex Set**.
The $\mathcal{F}$-M-Deletion problem

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

\begin{itemize}
  \item $\mathcal{F} = \{ \mathcal{K}_2 \}$: **Vertex Cover**.
    Easily solvable in time $2^{\Theta(tw)} \cdot n^{O(1)}$.

  \item $\mathcal{F} = \{ \mathcal{C}_3 \}$: **Feedback Vertex Set**.
    “Hardly” solvable in time $2^{\Theta(tw)} \cdot n^{O(1)}$.
\end{itemize}

[Cut&Count. 2011]
The \textsc{F-M-Deletion} problem

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

\begin{framed}
\textsc{F-M-Deletion}
\textbf{Input}: A graph $G$ and an integer $k$.
\textbf{Parameter}: The treewidth $tw$ of $G$.
\textbf{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?
\end{framed}

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

- $\mathcal{F} = \{C_3\}$: \textsc{Feedback Vertex Set}.
  “Hardly” solvable in time $2^{\Theta(tw)} \cdot n^{O(1)}$. [Cut&Count. 2011]

- $\mathcal{F} = \{K_5, K_{3,3}\}$: \textsc{Vertex Planarization}. 

- $\mathcal{F} = \{K_3\}$: \textsc{Feedback Vertex Set}.
The \( \mathcal{F}\)-M-DELETION problem

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

\[
\begin{align*}
\text{Input:} & \quad \text{A graph } G \text{ and an integer } k. \\
\text{Parameter:} & \quad \text{The treewidth } tw \text{ of } G. \\
\text{Question:} & \quad \text{Does } G \text{ contain a set } S \subseteq V(G) \text{ with } |S| \leq k \text{ such that } G - S \text{ does not contain any of the graphs in } \mathcal{F} \text{ as a minor?}
\end{align*}
\]

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

- \( \mathcal{F} = \{C_3\} \): **Feedback Vertex Set**. 
  “Hardly” solvable in time \( 2^{\Theta(tw)} \cdot n^{O(1)} \).  
  \([\text{Cut&Count. 2011}]\)

- \( \mathcal{F} = \{K_5, K_{3,3}\} \): **Vertex Planarization**. 
  Solvable in time \( 2^{\Theta(tw \cdot \log tw)} \cdot n^{O(1)} \).  
  \([\text{Jansen, Lokshtanov, Saurabh. 2014 + Pilipczuk. 2015}]\)
Covering topological minors

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

\[
\mathcal{F}\text{-M-Deletion}
\]

**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?

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

FPT by Courcelle, or by Graph Minors theory.]
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, or by Graph Minors theory.
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]
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. 

FPT by Courcelle, or by Graph Minors theory. 

[Lewis, Yannakakis. 1980]
Summary of our results

For every $F$:

$F$-M/TM-Deletion in time $O(tw \cdot \log tw) \cdot n^{O(1)}$.

$F$ connected, $F$-M-Deletion in time $O(tw \cdot \log tw) \cdot n^{O(1)}$.

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

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

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

$F = \{H\}$, $H$ planar + 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.
Summary of our results

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

---

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

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

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

---

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

$^2$Planar collection $\mathcal{F}$: contains at least one planar graph.
Summary of our results

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

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

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

---

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

$^2$Planar collection $\mathcal{F}$: contains at least one planar graph.
Summary of our results

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

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

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

  (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

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

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

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

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

- $\mathcal{F}$ connected: \texttt{F-M/TM-Deletion} not in time $2^{o(tw)} \cdot n^{O(1)}$ 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

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

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

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

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

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

- \( \mathcal{F} = \{H\} \), \( H \) planar + 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

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

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

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

  (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 $2^{\omega(tw)} \cdot n^{O(1)}$ unless the ETH fails, even if $G$ planar.

- $\mathcal{F} = \{H\}$, $H$ planar + 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.
**Complexity of hitting small planar minors $H$**

<table>
<thead>
<tr>
<th>$2\Theta(tw)$</th>
<th>$2\Theta(tw \cdot \log tw)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_2$</td>
<td>$P_5$</td>
</tr>
<tr>
<td>$P_3$</td>
<td>diamond</td>
</tr>
<tr>
<td>$P_4$</td>
<td>$K_4$</td>
</tr>
<tr>
<td>$C_3$</td>
<td>$C_5$</td>
</tr>
<tr>
<td>$C_4$</td>
<td>$K_{1,4}$</td>
</tr>
<tr>
<td>claw</td>
<td>$K_{5-e}$</td>
</tr>
<tr>
<td>paw</td>
<td>$W_4$</td>
</tr>
<tr>
<td>$K_3 \cup 2K_1$</td>
<td>$P_3 \cup 2K_1$</td>
</tr>
<tr>
<td>$P_2 \cup P_3$</td>
<td>gem</td>
</tr>
<tr>
<td>house</td>
<td>$px$</td>
</tr>
<tr>
<td>$K_{2,3}$</td>
<td>kite</td>
</tr>
<tr>
<td>dart</td>
<td>$K_4$</td>
</tr>
<tr>
<td>$K_1,4$</td>
<td>$C_5$</td>
</tr>
</tbody>
</table>

Classification of the complexity of $\{H\}$-M-Deletion for all connected simple planar graphs $H$ with $|V(H)| \leq 5$ and $|E(H)| \geq 1$: for the 9 graphs on the left (resp. 20 graphs on the right), the problem is solvable in time $2\Theta(tw) \cdot n^{O(1)}$ (resp. $2\Theta(tw \cdot \log tw) \cdot n^{O(1)}$). For $\{H\}$-TM-Deletion, $K_{1,4}$ should be on the left.
For topological minors, there is only one change

\[ 2^{\Theta(tw)} \]

\[ 2^{\Theta(tw \cdot \log tw)} \]

\[ P_5 \]

\[
\begin{align*}
P_2 & \quad \text{diamond} \\
P_3 & \quad K_4 \\
P_4 & \quad C_5 \\
C_3 & \quad K_3 \cup 2K_1 \\
C_4 & \quad K_5-e \\
\text{claw} & \quad W_4 \\
\text{paw} & \quad \overline{P_3 \cup 2K_1} \\
\text{chair} & \quad P_2 \cup P_3 \\
\text{banner} & \quad \text{gem} \\
bull & \quad \text{house} \\
\text{butterfly} & \quad px \\
cricket & \quad \text{kite} \\
\text{dart} & \quad \text{K2,3} \\
\co\text{-banner} & \quad \text{K1,4}
\end{align*}
\]
All these cases can be succinctly described as follows:
A compact statement for small planar minors

All these cases can be succinctly described as follows:

- All the graphs on the left are minors of \[ K_2,3 \]
A compact statement for small planar minors

All these cases can be succinctly described as follows:

- All the graphs on the left are minors of \( \square \).
- All the graphs on the right are not minors of \( \square \).
A compact statement for small planar minors

All these cases can be succinctly described as follows:

- All the graphs on the left are minors of $\square$.
- All the graphs on the right are not minors of $\square$ except $P_5$. 

... except $P_5$. ...
A dichotomy for hitting connected planar minors

We can prove that any connected planar $H$ with $|V(H)| \geq 6$ is “hard”.

A dichotomy for hitting connected planar minors

We can prove that any connected planar $H$ with $|V(H)| \geq 6$ is “hard”.

**Theorem**

*Let $H$ be a connected planar graph.*
A dichotomy for hitting connected planar minors

We can prove that any connected planar $H$ with $|V(H)| \geq 6$ is “hard”.

**Theorem**

Let $H$ be a connected planar graph.

The \{H\}-M-DELETION problem is solvable in time

\[ 2^{O(tw)} \cdot n^{O(1)}, \quad \text{if } H \preceq_m \begin{array}{c} \infty \\ \infty \end{array} \text{ and } H \neq P_5. \]
A dichotomy for hitting connected planar minors

We can prove that any connected planar $H$ with $|V(H)| \geq 6$ is “hard”.

**Theorem**

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

- $2^{O(tw)} \cdot n^{O(1)}$, if $H \preceq_m \begin{array}{c}
  * \\
  * \end{array}$ and $H \neq P_5$.

- $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$, otherwise.

In both cases, the running time is asymptotically optimal under the ETH.
We have three types of results

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

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
   - $2^{o(tw)}$ is "easy".
   - $2^{o(tw \cdot \log tw)}$ is much more involved and we get ideas from:
     - [Lokshtanov, Marx, Saurabh. 2011]
     - [Marcin Pilipczuk. 2017]
     - [Bonnet, Brettell, Kwon, Marx. 2017]
We have three types of results

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

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**
   - $2^{o(tw)}$ is "easy".
   - $2^{o(tw \cdot \log tw)}$ is much more involved and we get ideas from:
     - [Lokshtanov, Marx, Saurabh. 2011]
     - [Marcin Pilipczuk. 2017]
     - [Bonnet, Brettell, Kwon, Marx. 2017]
We have three types of results

1. **General algorithms**

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

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

2. **Ad-hoc single-exponential algorithms**
   - Some use “typical” dynamic programming.
   - Some use the rank-based approach.  

3. **Lower bounds under the ETH**
   - $2^{o(tw)}$ is “easy”.
   - $2^{o(tw \cdot \log tw)}$ 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 $2^{2^O(tw \cdot \log tw)} \cdot n^{O(1)}$.
- $\mathcal{F}$ connected + planar: time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.
- $G$ planar + $\mathcal{F}$ connected: time $2^{O(tw)} \cdot n^{O(1)}$.
Some ideas of the general algorithms

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

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]
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 $\mathcal{O}(t)$. 

For every $t$-boundaried graph $G$, $|\text{folio}(G)| = 2^{\mathcal{O}(t \log t)}$.

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

This gives an algorithm running in time $2^{\mathcal{O}(t \log t) \cdot n^{\mathcal{O}(1)}}$. 
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 $O(t)$.
- We compute, using DP over a tree decomposition of $G$, the following parameter for every folio $C$:
  \[
p(G, C) = \min\{|S| : S \subseteq V(G) \land \text{folio}(G - S) = C\}
\]
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 $\mathcal{O}(t)$.
- We compute, using DP over a tree decomposition of $G$, the following parameter for every folio $C$:
  \[
p(G, C) = \min\{|S| : S \subseteq V(G) \wedge \text{folio}(G - S) = C\}
\]
- For every $t$-boundaried graph $G$, $|\text{folio}(G)| = 2^{\mathcal{O}_\mathcal{F}(t \log t)}$. 
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 $O(t)$.

- We compute, using DP over a tree decomposition of $G$, the following parameter for every folio $C$:

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

- For every $t$-boundaried graph $G$, $|\text{folio}(G)| = 2^{O_{\mathcal{F}}(t \log t)}$.

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

We compute, using DP over a tree decomposition of $G$, the following parameter for every folio $C$:

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

- For every $t$-boundaried graph $G$, $|\text{folio}(G)| = 2^{O_F(t \log t)}$.
- The number of distinct folios is $2^{2^{O_F(t \log t)}}$.
- This gives an algorithm running in time $2^{2^{O_F(tw \cdot \log tw)}} \cdot n^{O(1)}$. 
Algorithm for a connected and planar collection \( \mathcal{F} \)

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

\[ G_1 \equiv (\mathcal{F}, t) \iff G_2 \iff \forall G' \in B_t, \mathcal{F} \preceq m G' \oplus G_1 \iff \mathcal{F} \preceq m G' \oplus G_2. \]

\( R(\mathcal{F}, t) \): 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 \( R \):

\[ p(G, R) = \min \{ |S| : S \subseteq V(G) \land \text{rep} \mathcal{F}, t(G - S) = R \} \]

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

This gives an algorithm running in time \( 2^{O(F(tw \cdot \log tw))} \cdot n^{O(1)} \).
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:

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

$$\mathcal{F} \preceq_m G' \uplus G_1 \iff \mathcal{F} \preceq_m G' \uplus G_2.$$
For a fixed $\mathcal{F}$, we define an equivalence relation $\equiv(\mathcal{F}, t)$ on $t$-boundaried graphs:

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

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

This gives an algorithm running in time $2^{O(F(t \cdot \log t))} \cdot n^{O(1)}$. 

[Baste, Noy, S. 2017]
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 $\equiv(\mathcal{F},t)$.

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

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

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

  $$\mathcal{F} \preceq_m G' \oplus G_1 \iff \mathcal{F} \preceq_m G' \oplus G_2.$$  

- $R^{(\mathcal{F},t)}$: 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 $R$:

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

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

\[\text{[Baste, Noy, S. 2017]}\]
For a fixed $\mathcal{F}$, we define an equivalence relation $\equiv(\mathcal{F}, t)$ on $t$-boundaried graphs:

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

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

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

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

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

The number of labeled graphs of size $\leq t$ and $\text{tw} \leq h$ is $2^{O_h(t \cdot \log t)}$. [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:
  \[ G_1 \equiv(\mathcal{F}, t) G_2 \text{ if } \forall G' \in B^t, \]
  \[ \mathcal{F} \preceq_m G' \oplus G_1 \iff \mathcal{F} \preceq_m G' \oplus G_2. \]

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

- We compute, using DP over a tree decomposition of $G$, the following parameter for every representative $R$:
  \[ p(G, R) = \min\{|S| : S \subseteq V(G) \land \text{rep}_{\mathcal{F}, t}(G - S) = R\} \]

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

- The number of labeled graphs of size $\leq t$ and $\text{tw} \leq h$ is $2^{O_h(t \cdot \log t)}$. [Baste, Noy, S. 2017]

- This gives an algorithm running in time $2^{O_F(t \cdot \log tw)} \cdot n^{O(1)}$. 

\[ \]
Algorithm when the input graph $G$ is planar

- **Idea** get an improved bound on $|R^{(F,t)}|$.

  We use a sphere-cut decomposition of the input planar graph $G$. Nice topological properties: each separator corresponds to a noose. The number of representatives is $|R^{(F,t)}| = 2^{O(F(t))}$. Number of planar triangulations on $t$ vertices is $2^{O(t)}$. This gives an algorithm running in time $2^{O(F(tw))} \cdot n^{O(1)}$.

  We can extend this algorithm to input graphs $G$ embedded in arbitrary surfaces by using surface-cut decompositions. [Rué, S., Thilikos. 2014]
Algorithm when the input graph $G$ is planar

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

- We use a **sphere-cut decomposition** of the input **planar graph** $G$.

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

Number of planar triangulations on $t$ vertices is $2^{O(t)}$.

[Tutte. 1962]

This gives an algorithm running in time $2^{O(tw)} \cdot n^{O(1)}$.

We can extend this algorithm to input graphs $G$ embedded in arbitrary surfaces by using surface-cut decompositions.

[Rué, S., Thilikos. 2014]
Algorithm when the input graph $G$ is planar

- **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.

![Diagram of sphere-cut decomposition]

Number of planar triangulations on $t$ vertices is $2^{O(t)}$.

[Tutte. 1962]

This gives an algorithm running in time $2^{O(f(tw)) \cdot n^{O(1)}}$.

We can extend this algorithm to input graphs $G$ embedded in arbitrary surfaces by using surface-cut decompositions.

[Rué, S., Thilikos. 2014]
Algorithm when the input graph $G$ is planar

- **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.

  ![Diagram showing sphere-cut decomposition]

- The number of representatives is $|\mathcal{R}(\mathcal{F}, t)| = 2^{O_{\mathcal{F}}(t)}$.
  
  Number of planar triangulations on $t$ vertices is $2^{O(t)}$. [Tutte. 1962]
Algorithm when the input graph $G$ is planar

- **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.

![Sphere-cut decomposition](image)

- The number of representatives is $|\mathcal{R}(\mathcal{F}, t)| = 2^{O_{\mathcal{F}}(t)}$.
  
  Number of planar triangulations on $t$ vertices is $2^{O(t)}$. [Tutte. 1962]

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

Algorithm when the input graph $G$ is planar

- **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 $|\mathcal{R}(\mathcal{F}, t)| = 2^{O(\mathcal{F})}(t)$.  
  Number of planar triangulations on $t$ vertices is $2^{O(t)}$.  
  [Tutte. 1962]

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

- We can extend this algorithm to input graphs $G$ embedded in **arbitrary surfaces** by using **surface-cut decompositions**.  
  [Rué, S., Thilikos. 2014]
What’s next about $\mathcal{F}$-DELETION?

Ultimate goal: classify the (asymptotically) tight complexity of $\mathcal{F}$-Deletion for every family $\mathcal{F}$. We are still far from it.

Dichotomy for $\{H\}$-TM-Deletion when $H$ planar + connected.

Only “missing” connected graph on at most 5 vertices: $K_5$.

We think that $\{K_5\}$-Deletion is solvable in time $2^{\Theta(tw \log tw)} \cdot n^O(1)$.

We do not even know if there exists some $\mathcal{F}$ such that $\mathcal{F}$-Deletion cannot be solved in time $2^{o(tw^2)} \cdot n^{O(1)}$ under the ETH.

Deletion to genus at most $g$: $2^{O(g)(tw \log tw)} \cdot n^{O(1)}$.

[Kociumaka, Pilipczuk. 2017]

Conjecture: For every connected family $\mathcal{F}$, the $\mathcal{F}$-Deletion problem is solvable in time $2^{O(tw \log tw)} \cdot n^{O(1)}$. Consider families $\mathcal{F}$ containing disconnected graphs.
What’s next about $\mathcal{F}$-DELETION?

- **Ultimate goal**: classify the (asymptotically) tight complexity of $\mathcal{F}$-DELETION for every family $\mathcal{F}$
What’s next about \( \mathcal{F}\text{-DELETION} \)?

- **Ultimate goal**: classify the (asymptotically) tight complexity of \( \mathcal{F}\text{-DELETION} \) for every family \( \mathcal{F} \)... we are still far from it.
What’s next about \( \mathcal{F}\text{-Deletion} \)?

- **Ultimate goal**: classify the (asymptotically) tight complexity of \( \mathcal{F}\text{-Deletion} \) for *every family* \( \mathcal{F} \) ... we are still far from it.

- **Dichotomy for** \( \{H\}\text{-TM-Deletion} \) when \( H \) planar + connected.
What’s next about $\mathcal{F}$-Deletion?

- **Ultimate goal**: classify the (asymptotically) tight complexity of $\mathcal{F}$-Deletion for every family $\mathcal{F}$... we are still far from it.

- Dichotomy for $\{H\}$-TM-Deletion when $H$ planar + connected.

- Only “missing” connected graph on at most 5 vertices: $K_5$.
  We think that $\{K_5\}$-Deletion is solvable in time $2^{\Theta(tw \cdot \log tw)} \cdot n^{O(1)}$.
What’s next about $\mathcal{F}$-Deletion?

- **Ultimate goal**: classify the (asymptotically) tight complexity of $\mathcal{F}$-Deletion for every family $\mathcal{F}$... we are still far from it.

- **Dichotomy for** $\{H\}$-TM-Deletion **when** $H$ **planar + connected**.

- Only “missing” connected graph on at most 5 vertices: $K_5$.
  We think that $\{K_5\}$-Deletion is solvable in time $2^{\Theta(tw \cdot \log tw)} \cdot n^{O(1)}$.

- We do not even know if there exists some $\mathcal{F}$ such that $\mathcal{F}$-Deletion cannot be solved in time $2^{o(tw^2)} \cdot n^{O(1)}$ under the ETH.
What’s next about $\mathcal{F}$-DELETION?

- **Ultimate goal**: classify the (asymptotically) tight complexity of $\mathcal{F}$-DELETION for every family $\mathcal{F}$... we are still far from it.

- Dichotomy for $\{H\}$-TM-DELETION when $H$ planar + connected.

- Only “missing” connected graph on at most 5 vertices: $K_5$. We think that $\{K_5\}$-DELETION is solvable in time $2^{\Theta(tw \cdot \log tw)} \cdot n^{O(1)}$.

- We do not even know if there exists some $\mathcal{F}$ such that $\mathcal{F}$-DELETION cannot be solved in time $2^{o(tw^2)} \cdot n^{O(1)}$ under the ETH.

Deletion to genus at most $g$: $2^{O_g(tw \cdot \log tw)} \cdot n^{O(1)}$. [Kociumaka, Pilipczuk. 2017]
What’s next about \( \mathcal{F}\text{-Deletion} \)?

- **Ultimate goal**: classify the (asymptotically) tight complexity of \( \mathcal{F}\text{-Deletion} \) for every family \( \mathcal{F} \)... we are still far from it.

- **Dichotomy for \( \{H\}\text{-TM-Deletion} \)** when \( H \) planar + connected.

- Only “missing” connected graph on at most 5 vertices: \( K_5 \).
  We think that \( \{K_5\}\text{-Deletion} \) is solvable in time \( 2^{\Theta(tw \cdot \log tw)} \cdot n^{O(1)} \).

- We do not even know if there exists some \( \mathcal{F} \) such that \( \mathcal{F}\text{-Deletion} \) cannot be solved in time \( 2^{o(tw^2)} \cdot n^{O(1)} \) under the ETH.

Deletion to genus at most \( g \): \( 2^{O_g(tw \cdot \log tw)} \cdot n^{O(1)} \). \[Kociumaka, Pilipczuk. 2017\]

- **Conjecture** For every connected family \( \mathcal{F} \), the \( \mathcal{F}\text{-Deletion} \) problem is solvable in time \( 2^{O(tw \cdot \log tw)} \cdot n^{O(1)} \).
What’s next about $\mathcal{F}$-Deletion?

- **Ultimate goal:** classify the (asymptotically) tight complexity of $\mathcal{F}$-Deletion for every family $\mathcal{F}$... we are still far from it.

- **Dichotomy** for $\{H\}$-TM-Deletion when $H$ planar + connected.

- Only “missing” connected graph on at most 5 vertices: $K_5$. We think that $\{K_5\}$-Deletion is solvable in time $2^{\Theta(tw \cdot \log tw)} \cdot n^{O(1)}$.

- We do not even know if there exists some $\mathcal{F}$ such that $\mathcal{F}$-Deletion cannot be solved in time $2^{o(tw^2)} \cdot n^{O(1)}$ under the ETH.

Deletion to genus at most $g$: $2^{O_g(tw \cdot \log tw)} \cdot n^{O(1)}$. [Kociumaka, Pilipczuk. 2017]

- **Conjecture** For every connected family $\mathcal{F}$, the $\mathcal{F}$-Deletion problem is solvable in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.

- Consider families $\mathcal{F}$ containing disconnected graphs.
Gràcies!

FREEDOM FOR ALL CATALAN POLITICAL PRISONERS IN SPAIN