FPT algorithms for hitting forbidden minors

Ignasi Sau

CNRS, LIRMM, Université de Montpellier, France

Joint work with Julien Baste, Giannos Stamoulis, and Dimitrios M. Thilikos

20th Haifa Workshop on
Graph Theory, Combinatorics and Algorithms
June 8th, 2020

Outline of the talk

1. Introduction
   - Parameterized complexity
   - Treewidth

2. Hitting forbidden minors
   - Parameterized by treewidth
   - Parameterized by solution size

Next section is...

1. Introduction
   - Parameterized complexity
   - Treewidth
2. Hitting forbidden minors
   - Parameterized by treewidth
   - Parameterized by solution size

Next subsection is...

1. Introduction
   - Parameterized complexity
   - Treewidth

2. Hitting forbidden minors
   - Parameterized by treewidth
   - Parameterized by solution size

Typical approaches to cope with intractability

P: problems that can be solved in polynomial time.

NP: problems for which a solution can be verified in polynomial time.

Typical approaches to cope with intractability

P: problems that can be solved in polynomial time.

NP: problems for which a solution can be verified in polynomial time.

NP-hard: no algorithm solves all instances optimally in polynomial time.

Typical approaches to cope with intractability

P: problems that can be solved in polynomial time.

NP: problems for which a solution can be verified in polynomial time.

NP-hard: no algorithm solves all instances optimally in polynomial time.

- Approximation algorithms: In polynomial time, find solutions that are "close" to the optimal ones.
- Moderately exponential-time algorithms: Solve the problem in exponential time, but reasonably "fast" (1.15^n vs 2^n).
- Heuristics: Produce "good" solutions, but with no guarantee.

Typical approaches to cope with intractability

P: problems that can be solved in polynomial time.

NP: problems for which a solution can be verified in polynomial time.

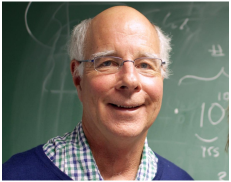
NP-hard: no algorithm solves all instances optimally in polynomial time.

- Approximation algorithms: In polynomial time, find solutions that are "close" to the optimal ones.
- Moderately exponential-time algorithms: Solve the problem in exponential time, but reasonably "fast" (1.15^n vs 2^n).
- Heuristics: Produce "good" solutions, but with no guarantee.
- Parameterized complexity: Topic of this talk...

The area of parameterized complexity

$\boxed{\text{Idea}}$ Measure the complexity of an algorithm in terms of the input size and an additional integer parameter.

This theory started in the late 80's, by Downey and Fellows:



Today, it is a well-established area with hundreds of articles published every year in the most prestigious TCS journals and conferences.

Parameterized problems

In a parameterized problem, an instance is a pair (x, k), where

- x is a typical input (in our setting, a graph).
- k is a positive integer called the parameter.

Parameterized problems

In a parameterized problem, an instance is a pair (x, k), where

- x is a typical input (in our setting, a graph).
- k is a positive integer called the parameter.

Examples of parameterized problems on graphs, with an instance (G, k):

1. k-Vertex Cover: Does G contain a set $S \subseteq V(G)$, with $|S| \leq k$, containing at least an endpoint of every edge?

Parameterized problems

In a parameterized problem, an instance is a pair (x, k), where

- x is a typical input (in our setting, a graph).
- k is a positive integer called the parameter.

Examples of parameterized problems on graphs, with an instance (G, k):

1. k-Vertex Cover: Does G contain a set $S \subseteq V(G)$, with $|S| \leq k$, containing at least an endpoint of every edge?

2. k-Clique: Does G contain a set $S \subseteq V(G)$, with $|S| \geq k$, of pairwise adjacent vertices?

Parameterized problems

In a parameterized problem, an instance is a pair (x, k), where

- x is a typical input (in our setting, a graph).
- k is a positive integer called the parameter.

Examples of parameterized problems on graphs, with an instance (G, k):

1. k-Vertex Cover: Does G contain a set $S \subseteq V(G)$, with $|S| \leq k$, containing at least an endpoint of every edge?

2. k-Clique: Does G contain a set $S \subseteq V(G)$, with $|S| \geq k$, of pairwise adjacent vertices?

3. Vertex k-Coloring: Can $V(G)$ be colored with $\leq k$ colors, so that adjacent vertices get different colors?

Parameterized problems

In a parameterized problem, an instance is a pair (x, k), where

- x is a typical input (in our setting, a graph).
- k is a positive integer called the parameter.

Examples of parameterized problems on graphs, with an instance (G, k):

1. k-Vertex Cover: Does G contain a set $S \subseteq V(G)$, with $|S| \leq k$, containing at least an endpoint of every edge?

2. k-Clique: Does G contain a set $S \subseteq V(G)$, with $|S| \geq k$, of pairwise adjacent vertices?

3. Vertex k-Coloring: Can $V(G)$ be colored with $\leq k$ colors, so that adjacent vertices get different colors?

These three problems are NP-hard, but are they $\boxed{\text{equally}}$ hard?

They behave quite differently...

1. k-Vertex Cover: solvable in time $2^k \cdot n^2$

2. k-Clique: solvable in time $k^2 \cdot n^k$

3. Vertex k-Coloring: NP-hard for every fixed $k \geq 3$

They behave quite differently...

1. k-Vertex Cover: solvable in time $2^k \cdot n^2 \quad = \quad \boxed{f(k) \cdot n^{\mathcal{O}(1)}}$

2. k-Clique: solvable in time $k^2 \cdot n^k \quad = \quad \boxed{f(k) \cdot n^{g(k)}}$

3. Vertex k-Coloring: NP-hard for every fixed $k \geq 3$

They behave quite differently...

1. k-VERTEX COVER: solvable in time $2^k \cdot n^2 \quad = \quad \boxed{f(k) \cdot n^{\mathcal{O}(1)}}$

   $\boxed{\text{The problem is FPT}}$ (fixed-parameter tractable)

2. k-CLIQUE: solvable in time $k^2 \cdot n^k \quad = \quad \boxed{f(k) \cdot n^{g(k)}}$

3. VERTEX k-COLORING: NP-hard for every fixed $k \geq 3$

They behave quite differently...

1. k-Vertex Cover: solvable in time $2^k \cdot n^2 \quad = \quad \boxed{f(k) \cdot n^{\mathcal{O}(1)}}$

   $\boxed{\text{The problem is FPT}}$ (fixed-parameter tractable)

2. k-Clique: solvable in time $k^2 \cdot n^k \quad = \quad \boxed{f(k) \cdot n^{g(k)}}$

   $\boxed{\text{The problem is XP}}$ (slice-wise polynomial)

3. Vertex k-Coloring: NP-hard for every fixed $k \geq 3$

They behave quite differently...

1. k-Vertex Cover: solvable in time $2^k \cdot n^2 \quad = \quad \boxed{f(k) \cdot n^{\mathcal{O}(1)}}$

   $\boxed{\text{The problem is FPT}}$ (fixed-parameter tractable)

2. k-Clique: solvable in time $k^2 \cdot n^k \quad = \quad \boxed{f(k) \cdot n^{g(k)}}$

   $\boxed{\text{The problem is XP}}$ (slice-wise polynomial)

3. Vertex k-Coloring: NP-hard for every fixed $k \geq 3$

   $\boxed{\text{The problem is para-NP-hard}}$

The choice of the parameter is crucial!

Every choice of the parameter defines a different parameterized problem:

The choice of the parameter is crucial!

Every choice of the parameter defines a different parameterized problem:

- Finding a k-clique parameterized by k: W[1]-hard (unlikely FPT).

The choice of the parameter is crucial!

Every choice of the parameter defines a different parameterized problem:

- Finding a k-clique parameterized by k: W[1]-hard (unlikely FPT).
- Finding a k-clique parameterized by Δ: FPT.

The choice of the parameter is crucial!

Every choice of the parameter defines a different parameterized problem:

- Finding a k-clique parameterized by k: W[1]-hard (unlikely FPT).
- Finding a k-clique parameterized by Δ: FPT.

There are mainly two types of parameters:

The choice of the parameter is crucial!

Every choice of the parameter defines a different parameterized problem:

- Finding a k-clique parameterized by k: W[1]-hard (unlikely FPT).
- Finding a k-clique parameterized by Δ: FPT.

There are mainly two types of parameters:

- Parameters concerning the desired solution (output):
  typically, the size of the solution we are looking for.

The choice of the parameter is crucial!

Every choice of the parameter defines a different parameterized problem:

- Finding a k-clique parameterized by k: W[1]-hard (unlikely FPT).
- Finding a k-clique parameterized by Δ: FPT.

There are mainly two types of parameters:

- Parameters concerning the desired solution (output):
  typically, the size of the solution we are looking for.
- Parameters considering structural characteristics of the input graph:
  maximum degree,

The choice of the parameter is crucial!

Every choice of the parameter defines a different parameterized problem:

- Finding a k-clique parameterized by k: W[1]-hard (unlikely FPT).
- Finding a k-clique parameterized by Δ: FPT.

There are mainly two types of parameters:

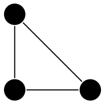
- Parameters concerning the desired solution (output):
  typically, the size of the solution we are looking for.
- Parameters considering structural characteristics of the input graph:
  maximum degree, or treewidth.

Next subsection is...

1. Introduction
    - Parameterized complexity
    - Treewidth
2. Hitting forbidden minors
    - Parameterized by treewidth
    - Parameterized by solution size

Treewidth via k-trees

Example of a 2-tree:

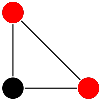


[Figure by Julien Baste]

A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

Treewidth via k-trees

Example of a 2-tree:

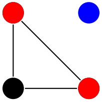


[Figure by Julien Baste]

A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

Treewidth via k-trees

Example of a 2-tree:



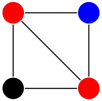
[Figure by Julien Baste]

A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

Treewidth via k-trees

Example of a 2-tree:

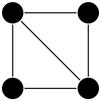
A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.



[Figure by Julien Baste]

Treewidth via k-trees

Example of a 2-tree:

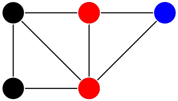


[Figure by Julien Baste]

A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

Treewidth via k-trees

Example of a 2-tree:

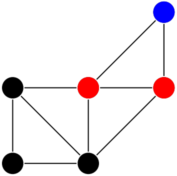


[Figure by Julien Baste]

A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

Treewidth via k-trees

Example of a 2-tree:

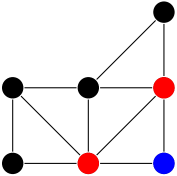


[Figure by Julien Baste]

A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

Treewidth via k-trees

Example of a 2-tree:

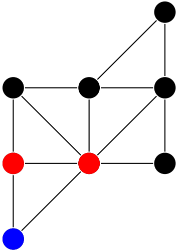


[Figure by Julien Baste]

A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

Treewidth via k-trees

Example of a 2-tree:

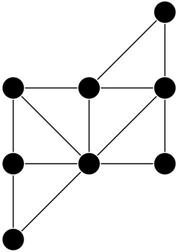


[Figure by Julien Baste]

A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

Treewidth via k-trees

Example of a 2-tree:

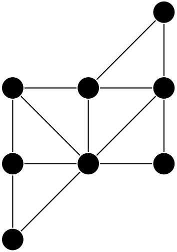


[Figure by Julien Baste]

A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

Treewidth via k-trees

Example of a 2-tree:



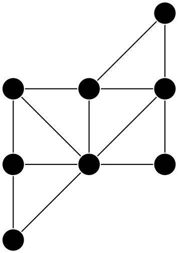
[Figure by Julien Baste]

A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

A partial k-tree is a subgraph of a k-tree.

Treewidth via k-trees

Example of a 2-tree:



[Figure by Julien Baste]

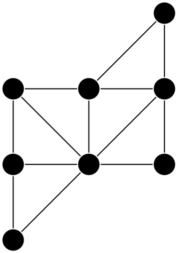
A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

A partial k-tree is a subgraph of a k-tree.

Treewidth of a graph G, denoted $\mathrm{tw}(G)$: smallest integer k such that G is a partial k-tree.

Treewidth via k-trees

Example of a 2-tree:



[Figure by Julien Baste]

A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

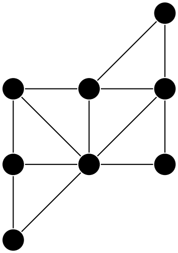
A partial k-tree is a subgraph of a k-tree.

Treewidth of a graph G, denoted $\mathrm{tw}(G)$: smallest integer k such that G is a partial k-tree.

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

Treewidth via k-trees

Example of a 2-tree:



[Figure by Julien Baste]

A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

A partial k-tree is a subgraph of a k-tree.

Treewidth of a graph G, denoted $\mathrm{tw}(G)$: smallest integer k such that G is a partial k-tree.

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

Construction suggests the notion of tree decomposition: small separators.

Why treewidth?

Treewidth is important for (at least) 3 different reasons:

Why treewidth?

Treewidth is important for (at least) 3 different reasons:

1. Treewidth is a fundamental combinatorial tool in graph theory: key role in the Graph Minors project of Robertson and Seymour.

Why treewidth?

Treewidth is important for (at least) 3 different reasons:

1. Treewidth is a fundamental combinatorial tool in graph theory: key role in the Graph Minors project of Robertson and Seymour.

2. Treewidth behaves very well algorithmically, and algorithms parameterized by treewidth appear very often in FPT algorithms.

Why treewidth?

Treewidth is important for (at least) 3 different reasons:

1. Treewidth is a fundamental combinatorial tool in graph theory: key role in the Graph Minors project of Robertson and Seymour.

2. Treewidth behaves very well algorithmically, and algorithms parameterized by treewidth appear very often in FPT algorithms.

3. In many practical scenarios, it turns out that the treewidth of the associated graph is small (programming languages, road networks, ...).

Treewidth behaves very well algorithmically

Treewidth behaves very well algorithmically

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

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.

Treewidth behaves very well algorithmically

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 $f(\text{tw}) \cdot n$ 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, ...

Next section is...

Graph modification problems

Graph modification problems

Let $\mathcal{C}$ be a target graph class (planar graphs, bounded degree, ...).

Graph modification problems

Let $\mathcal{C}$ be a target graph class (planar graphs, bounded degree, ...).

Let $\mathcal{M}$ be a set of allowed graph modification operations (vertex deletion, edge deletion/addition/contraction, ...).

Graph modification problems

Let $\mathcal{C}$ be a target graph class (planar graphs, bounded degree, ...).

Let $\mathcal{M}$ be a set of allowed graph modification operations (vertex deletion, edge deletion/addition/contraction, ...).

$\mathcal{M}$-Modification to $\mathcal{C}$

Input: A graph G and an integer k.

Question: Can we transform G to a graph in $\mathcal{C}$ by applying at most k operations from $\mathcal{M}$?

Graph modification problems

Let $\mathcal{C}$ be a target graph class (planar graphs, bounded degree, ...).

Let $\mathcal{M}$ be a set of allowed graph modification operations (vertex deletion, edge deletion/addition/contraction, ...).

$\mathcal{M}$-MODIFICATION TO $\mathcal{C}$	
Input:	A graph G and an integer k.
Question:	Can we transform G to a graph in $\mathcal{C}$ by applying at most k operations from $\mathcal{M}$?

We focus on:

- $\mathcal{M} = \{\text{vertex deletion}\}$.

Graph modification problems

Let $\mathcal{C}$ be a target graph class (planar graphs, bounded degree, ...).

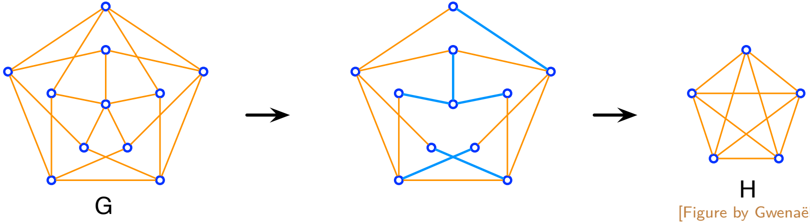
Let $\mathcal{M}$ be a set of allowed graph modification operations (vertex deletion, edge deletion/addition/contraction, ...).

$\mathcal{M}$-MODIFICATION TO $\mathcal{C}$	
Input:	A graph G and an integer k.
Question:	Can we transform G to a graph in $\mathcal{C}$ by applying at most k operations from $\mathcal{M}$?

We focus on:

- $\mathcal{M} = \{\text{vertex deletion}\}$.
- $\mathcal{C}$ is a minor-closed graph class.

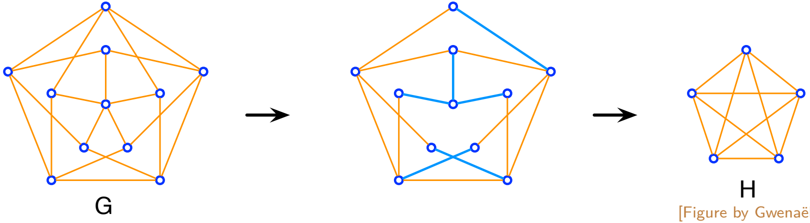
Graph minors



[Figure by Gwenaël Joret]

A graph H is a minor of a graph G if H, denoted by $H \leqslant_{\mathrm{m}} G$, can be obtained from a subgraph of G by contracting edges.

Graph minors

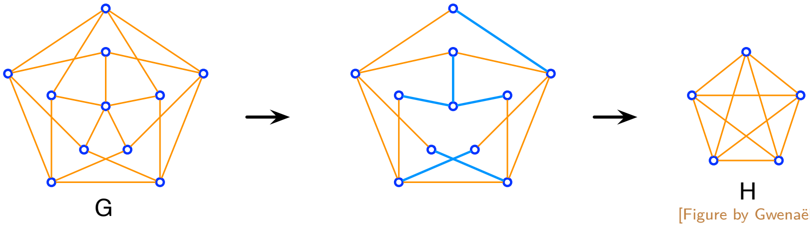


[Figure by Gwenaël Joret]

A graph H is a minor of a graph G if H, denoted by $H \leqslant_{\mathrm{m}} G$, can be obtained from a subgraph of G by contracting edges.

A graph class $\mathcal{C}$ is minor-closed if $(G \in \mathcal{C} \text{ and } H \leqslant_{\mathrm{m}} G) \implies H \in \mathcal{C}$.

Graph minors



[Figure by Gwenaël Joret]

A graph H is a minor of a graph G if H, denoted by $H \leqslant_{\mathrm{m}} G$, can be obtained from a subgraph of G by contracting edges.

A graph class $\mathcal{C}$ is minor-closed if $(G \in \mathcal{C}$ and $H \leqslant_{\mathrm{m}} G) \Longrightarrow H \in \mathcal{C}$.

Theorem (Robertson and Seymour. 1983-2004)

For every minor-closed graph class $\mathcal{C}$ there exists a finite collection $\mathcal{F}$ of forbidden minors such that, for every graph G,

$$G \in \mathcal{C} \iff F \not\leqslant_{\mathrm{m}} G \text{ for every } F \in \mathcal{F}.$$

Hitting forbidden minors

- If $\mathcal{C}$ = {edgeless graphs}, then $\mathcal{F} = \{K_2\}$.
- If $\mathcal{C}$ = {forests}, then $\mathcal{F} = \{K_3\}$.
- If $\mathcal{C}$ = {outerplanar graphs}, then $\mathcal{F} = \{K_4, K_{2,3}\}$.
- If $\mathcal{C}$ = {planar graphs}, then $\mathcal{F} = \{K_5, K_{3,3}\}$.

Hitting forbidden minors

- If $\mathcal{C}$ = {edgeless graphs}, then $\mathcal{F} = \{K_2\}$.
- If $\mathcal{C}$ = {forests}, then $\mathcal{F} = \{K_3\}$.
- If $\mathcal{C}$ = {outerplanar graphs}, then $\mathcal{F} = \{K_4, K_{2,3}\}$.
- If $\mathcal{C}$ = {planar graphs}, then $\mathcal{F} = \{K_5, K_{3,3}\}$.

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

Hitting forbidden minors

- If $\mathcal{C} = \{$edgeless graphs$\}$, then $\mathcal{F} = \{K_2\}$.
- If $\mathcal{C} = \{$forests$\}$, then $\mathcal{F} = \{K_3\}$.
- If $\mathcal{C} = \{$outerplanar graphs$\}$, then $\mathcal{F} = \{K_4, K_{2,3}\}$.
- If $\mathcal{C} = \{$planar graphs$\}$, then $\mathcal{F} = \{K_5, K_{3,3}\}$.

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

$\mathcal{F}$-M-Deletion

Input: A graph G and an integer k.

Question: Does G contain a set $S \subseteq V(G)$ with $|S| \leqslant k$ such that $G \setminus S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?

Hitting forbidden minors

- If $\mathcal{C}$ = {edgeless graphs}, then $\mathcal{F} = \{K_2\}$.
- If $\mathcal{C}$ = {forests}, then $\mathcal{F} = \{K_3\}$.
- If $\mathcal{C}$ = {outerplanar graphs}, then $\mathcal{F} = \{K_4, K_{2,3}\}$.
- If $\mathcal{C}$ = {planar graphs}, then $\mathcal{F} = \{K_5, K_{3,3}\}$.

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

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

- $\mathcal{F} = \{K_2\}$: Vertex Cover.
- $\mathcal{F} = \{K_3\}$: Feedback Vertex Set.
- $\mathcal{F} = \{K_5, K_{3,3}\}$: Vertex Planarization.
- $\mathcal{F}$ = {diamond}: Cactus Vertex Deletion.

Hitting forbidden minors

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

$\mathcal{F}$-M-DELETION

Input: A graph G and an integer k.

Question: Does G contain a set $S \subseteq V(G)$ with $|S| \leqslant k$ such that $G \setminus S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?

Hitting forbidden minors

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

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

NP-hard if $\mathcal{F}$ contains a graph with some edge. [Lewis, Yannakakis. 1980]

Hitting forbidden minors

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

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

NP-hard if $\mathcal{F}$ contains a graph with some edge. [Lewis, Yannakakis. 1980]

We consider the following two parameterizations of $\mathcal{F}$-M-Deletion:

1. Structural parameter: $\mathrm{tw}(G)$.
2. Solution size: k.

Next subsection is...

1. Introduction
   - Parameterized complexity
   - Treewidth
2. Hitting forbidden minors
   - Parameterized by treewidth
   - Parameterized by solution size

Theorem (Courcelle. 1990)

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

Theorem (Courcelle. 1990)

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

It is not difficult to see that can $\mathcal{F}$-M-DELETION be expressed in MSOL:

$\mathcal{F}$-M-DELETION is FPT parameterized by tw...

Theorem (Courcelle. 1990)

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

It is not difficult to see that can $\mathcal{F}$-M-Deletion be expressed in MSOL:

$\mathcal{F}$-M-Deletion is FPT parameterized by tw...

$$f_{\mathcal{F}}(\mathrm{tw}) \cdot n$$

Theorem (Courcelle. 1990)

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

It is not difficult to see that can $\mathcal{F}$-M-DELETION be expressed in MSOL:

$\mathcal{F}$-M-DELETION is FPT parameterized by tw...

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

Theorem (Courcelle. 1990)

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

It is not difficult to see that can $\mathcal{F}$-M-Deletion be expressed in MSOL:

$\mathcal{F}$-M-Deletion is FPT parameterized by tw...

$$f_{\mathcal{F}}(\mathsf{tw}) \cdot n = 2^{3^{4^{5^{6^{7^{8^{\mathsf{tw}}}}}}}} \cdot n$$

$\boxed{\text{Goal}}$ For every $\mathcal{F}$, find the smallest possible function $f_{\mathcal{F}}(\mathsf{tw})$.

Theorem (Courcelle. 1990)

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

It is not difficult to see that can $\mathcal{F}$-M-Deletion be expressed in MSOL:

$\mathcal{F}$-M-Deletion is FPT parameterized by tw...

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

$\boxed{\text{Goal}}$ For every $\mathcal{F}$, find the $\boxed{\text{smallest possible}}$ function $f_{\mathcal{F}}(\mathrm{tw})$.

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

[Impagliazzo, Paturi. 1999]

Theorem (Courcelle. 1990)

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

It is not difficult to see that can $\mathcal{F}$-M-Deletion be expressed in MSOL:

$\mathcal{F}$-M-Deletion is FPT parameterized by tw...

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

$\boxed{\text{Goal}}$ For every $\mathcal{F}$, find the $\boxed{\text{smallest possible}}$ function $f_{\mathcal{F}}(\mathrm{tw})$.

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

[Impagliazzo, Paturi. 1999]

Very active area in parameterized complexity during the last decade.

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

What was known for particular collections $\mathcal{F}$

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 $|S| \leqslant k$ such that $G \setminus S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?

What was known for particular collections $\mathcal{F}$

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 $|S| \leqslant k$ such that $G \setminus S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?

- $\mathcal{F} = \{K_2\}$: VERTEX COVER.

What was known for particular collections $\mathcal{F}$

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 $\lvert S \rvert \leqslant k$ such that $G \setminus S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?

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

What was known for particular collections $\mathcal{F}$

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 $\|S\| \leqslant k$ such that $G \setminus S$ 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.

What was known for particular collections $\mathcal{F}$

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 $\lvert S\rvert \leqslant k$ such that $G \setminus S$ 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**: Cygan, Nederlof, Pilipczuk, Pilipczuk, van Rooij, Wojtaszczyk. 2011]

What was known for particular collections $\mathcal{F}$

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 $\|S\| \leqslant k$ such that $G \setminus S$ 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**: Cygan, Nederlof, Pilipczuk, Pilipczuk, van Rooij, Wojtaszczyk. 2011]

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

What was known for particular collections $\mathcal{F}$

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 $\lvert S \rvert \leqslant k$ such that $G \setminus S$ 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**: Cygan, Nederlof, Pilipczuk, Pilipczuk, van Rooij, Wojtaszczyk. 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]

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 on n-vertex graphs can be solved in time

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

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 on n-vertex graphs can be solved in time

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

- 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.

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 on n-vertex graphs can be solved in time

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

- 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.

[Baste, S., Thilikos. **Hitting minors on bounded treewidth graphs. I. General upper bounds**. 2020]

[Baste, S., Thilikos. **Hitting minors on bounded treewidth graphs. II. Single-exponential algorithms**. 2020]

[Baste, S., Thilikos. **Hitting minors on bounded treewidth graphs. III. Lower bounds**. 2020]

[Baste, S., Thilikos. **Hitting minors on bounded treewidth graphs. IV. An optimal algorithm**. 2020-]

Summary of our results

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

Summary of our results

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

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

Summary of our results

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

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

Summary of our results

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

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

Summary of our results

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

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

Summary of our results

- For every $\mathcal{F}$: $\mathcal{F}$-M-DELETION in time $2^{2^{\mathcal{O}(\mathsf{tw} \cdot \log \mathsf{tw})}} \cdot n^{\mathcal{O}(1)}$.
- For every ~~planar~~[1] $\mathcal{F}$: $\mathcal{F}$-M-DELETION in time $2^{\mathcal{O}(\mathsf{tw} \cdot \log \mathsf{tw})} \cdot n^{\mathcal{O}(1)}$.
- G planar: $\mathcal{F}$-M-DELETION in time $2^{\mathcal{O}(\mathsf{tw})} \cdot n^{\mathcal{O}(1)}$.
- For every $\mathcal{F}$: $\mathcal{F}$-M-DELETION not solvable in time $2^{o(\mathsf{tw})} \cdot n^{\mathcal{O}(1)}$ unless the ETH fails, even if G planar.

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

Summary of our results

- For every $\mathcal{F}$: $\mathcal{F}$-M-DELETION in time $2^{2^{\mathcal{O}(\mathrm{tw}\cdot\log \mathrm{tw})}} \cdot n^{\mathcal{O}(1)}$.
- For every ~~planar~~[1] $\mathcal{F}$: $\mathcal{F}$-M-DELETION in time $2^{\mathcal{O}(\mathrm{tw}\cdot\log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.
- G planar: $\mathcal{F}$-M-DELETION in time $2^{\mathcal{O}(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$.
- For every $\mathcal{F}$: $\mathcal{F}$-M-DELETION not solvable in time $2^{o(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$ unless the ETH fails, even if G planar.
- $\mathcal{F} = \{H\}$, H connected:

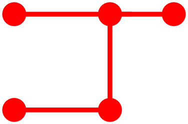
[1] Planar collection $\mathcal{F}$: contains at least one planar graph.

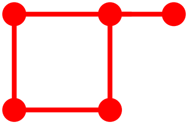
Summary of our results

- For every $\mathcal{F}$: $\mathcal{F}$-M-Deletion in time $2^{2^{\mathcal{O}(\mathsf{tw} \cdot \log \mathsf{tw})}} \cdot n^{\mathcal{O}(1)}$.
- For every ~~planar~~[1] $\mathcal{F}$: $\mathcal{F}$-M-Deletion in time $2^{\mathcal{O}(\mathsf{tw} \cdot \log \mathsf{tw})} \cdot n^{\mathcal{O}(1)}$.
- G planar: $\mathcal{F}$-M-Deletion in time $2^{\mathcal{O}(\mathsf{tw})} \cdot n^{\mathcal{O}(1)}$.
- For every $\mathcal{F}$: $\mathcal{F}$-M-Deletion not solvable in time $2^{o(\mathsf{tw})} \cdot n^{\mathcal{O}(1)}$ unless the ETH fails, even if G planar.
- $\mathcal{F} = \{H\}$, H connected: complete tight dichotomy...

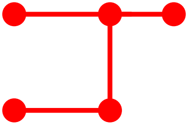
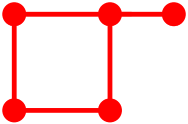
[1] Planar collection $\mathcal{F}$: contains at least one planar graph.

A dichotomy for hitting a connected minor





A dichotomy for hitting a connected minor

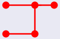
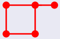
Theorem (Baste, S., Thilikos. 2016-2020)

Let H be a connected graph.

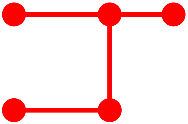
A dichotomy for hitting a connected minor

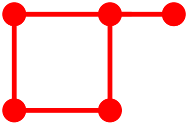
Theorem (Baste, S., Thilikos. 2016-2020)

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* $H \leqslant_{\mathrm{c}}$  *or* $H \leqslant_{\mathrm{c}}$  .

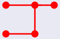
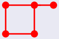
A dichotomy for hitting a connected minor



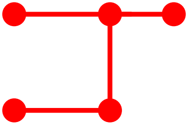


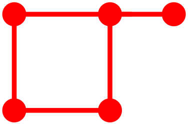
Theorem (Baste, S., Thilikos. 2016-2020)

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

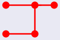
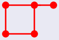
A dichotomy for hitting a connected minor





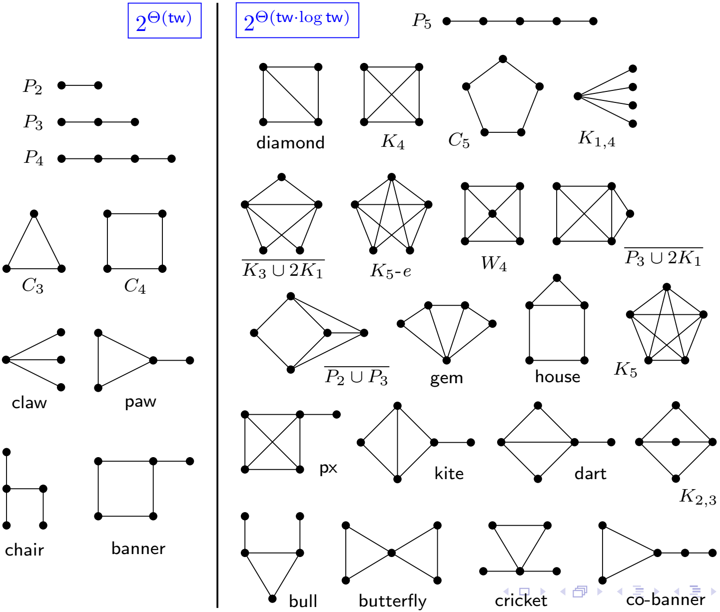
Theorem (Baste, S., Thilikos. 2016-2020)

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

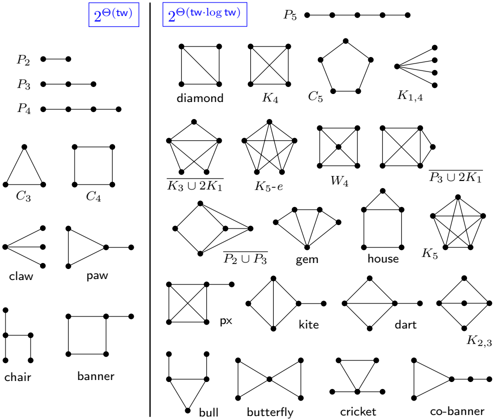
- $2^{\mathcal{O}(\mathrm{tw})} \cdot n^{\mathcal{O}(1)}$, *if* $H \leqslant_c$  *or* $H \leqslant_c$ .
- $2^{\mathcal{O}(\mathrm{tw} \cdot \log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$, *otherwise.*

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

Complexity of hitting a single connected minor H

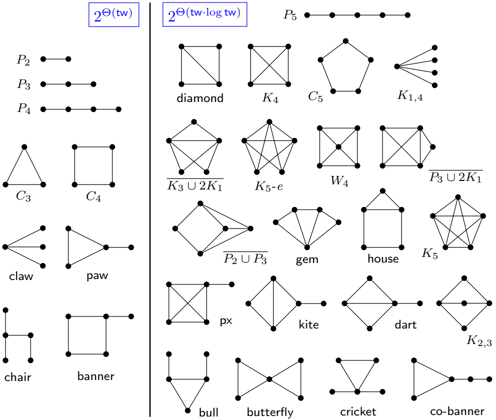


A compact statement for a single connected graph

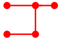
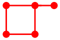


All these cases can be succinctly described as follows:

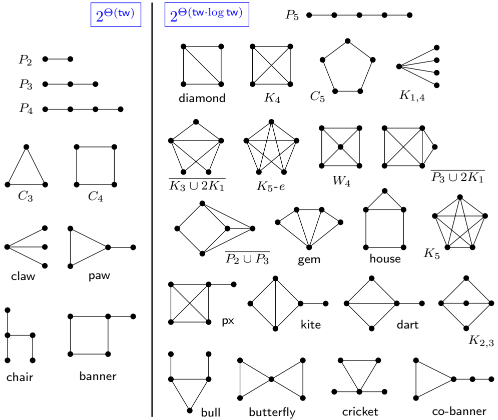
A compact statement for a single connected graph



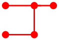
All these cases can be succinctly described as follows:

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

We have three types of results

1. General algorithms

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

skip

We have three types of results

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

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

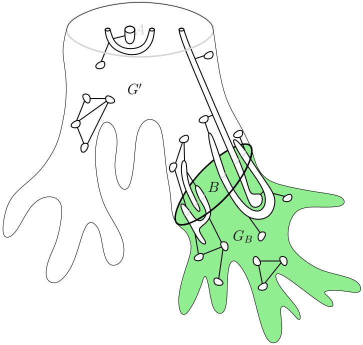
3. Lower bounds under the ETH
   - $2^{o(\mathrm{tw})}$ is "easy".
   - $2^{o(\mathrm{tw}\cdot\log\mathrm{tw})}$ is much more involved and we get ideas from:

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

skip

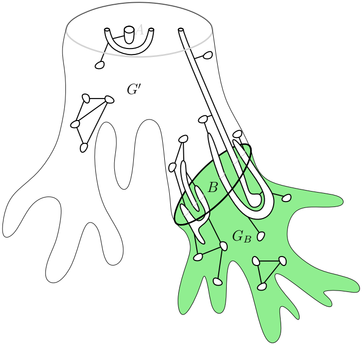
Algorithm in time $2^{\mathcal{O}_{\mathcal{F}}(\mathsf{tw} \cdot \log \mathsf{tw})} \cdot n^{\mathcal{O}(1)}$ for any collection $\mathcal{F}$

[Fig. by Valentin Garnero]



Algorithm in time $2^{\mathcal{O}_{\mathcal{F}}(\mathrm{tw}\cdot\log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$ for any collection $\mathcal{F}$

[Fig. by Valentin Garnero]



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

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

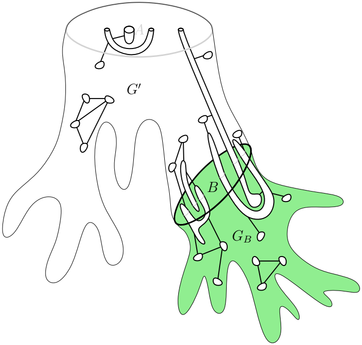
Algorithm in time $2^{\mathcal{O}_{\mathcal{F}}(\text{tw}\cdot\log \text{tw})} \cdot n^{\mathcal{O}(1)}$ for any collection $\mathcal{F}$

[Fig. by Valentin Garnero]

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

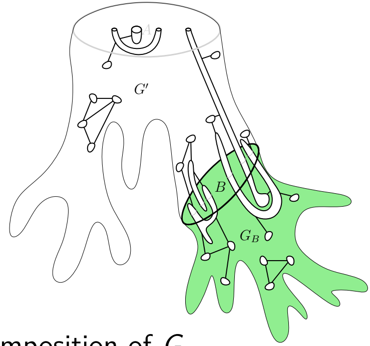
$$G_1 \equiv^{(\mathcal{F},t)} G_2 \quad \text{if } \forall G' \in \mathcal{B}^t,$$
$$\mathcal{F} \leqslant_{\mathrm{m}} G' \oplus G_1 \iff \mathcal{F} \leqslant_{\mathrm{m}} G' \oplus G_2.$$

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



Algorithm in time $2^{\mathcal{O}_{\mathcal{F}}(\mathrm{tw}\cdot\log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$ for any collection $\mathcal{F}$

[Fig. by Valentin Garnero]



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

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

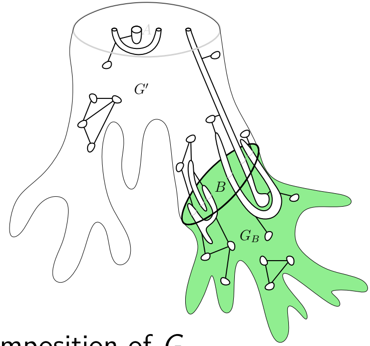
- $\mathcal{R}^{(\mathcal{F},t)}$: set of minimum-size representatives of $\equiv^{(\mathcal{F},t)}$.

- We compute, using DP over a tree decomposition of G, the following parameter for every representative $R \in \mathcal{R}^{(\mathcal{F},t)}$:

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

Algorithm in time $2^{\mathcal{O}_{\mathcal{F}}(\mathrm{tw}\cdot\log \mathrm{tw})} \cdot n^{\mathcal{O}(1)}$ for any collection $\mathcal{F}$

[Fig. by Valentin Garnero]



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

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

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

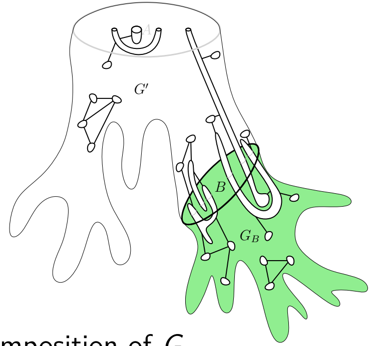
- We compute, using DP over a tree decomposition of G, the following parameter for every representative $R \in \mathcal{R}^{(\mathcal{F},t)}$:

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

- This gives an algorithm running in time $|\mathcal{R}^{(\mathcal{F},t)}|^{\mathcal{O}(1)} \cdot n^{\mathcal{O}(1)}$.

Algorithm in time $2^{\mathcal{O}_{\mathcal{F}}(\text{tw}\cdot\log \text{tw})} \cdot n^{\mathcal{O}(1)}$ for any collection $\mathcal{F}$

[Fig. by Valentin Garnero]



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

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

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

$$\mathbf{p}(G_B, R) = \min\{|S| \ : \ S \subseteq V(G_B) \wedge \text{rep}_{\mathcal{F},t}(G_B \setminus S) = R\}$$

- This gives an algorithm running in time $|\mathcal{R}^{(\mathcal{F},t)}|^{\mathcal{O}(1)} \cdot n^{\mathcal{O}(1)}$.

- $\boxed{\text{Goal}}$ Bound the number of representatives: $|\mathcal{R}^{(\mathcal{F},t)}| = 2^{\mathcal{O}_{\mathcal{F}}(\text{tw}\cdot\log \text{tw})}$.

Bounding the set of representatives

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

Bounding the set of representatives

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

Bounding the set of representatives

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

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

- Then, by the sparsity of the representatives,

$$|\mathcal{R}^{(\mathcal{F},t)}| = \mathcal{O}_{\mathcal{F}}(1) \cdot \binom{t^2}{t} = 2^{\mathcal{O}_{\mathcal{F}}(t \cdot \log t)},$$

and we are done!

Bounding the set of representatives

- $\mathcal{R}^{(\mathcal{F},t)}$: set of minimum-size representatives of $\equiv^{(\mathcal{F},t)}$.
- Suppose that we can prove that, for every $R \in \mathcal{R}^{(\mathcal{F},t)}$,
$$|V(R)| = \mathcal{O}_{\mathcal{F}}(t).$$
- Then, by the sparsity of the representatives,
$$|\mathcal{R}^{(\mathcal{F},t)}| = \mathcal{O}_{\mathcal{F}}(1) \cdot \binom{t^2}{t} = 2^{\mathcal{O}_{\mathcal{F}}(t \cdot \log t)},$$
and we are done!
- $\boxed{\text{Flat Wall Theorem}}$ [Robertson, Seymour. GMXIII. 1995]

Bounding the set of representatives

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

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

- Then, by the sparsity of the representatives,

$$|\mathcal{R}^{(\mathcal{F},t)}| = \mathcal{O}_{\mathcal{F}}(1) \cdot \binom{t^2}{t} = 2^{\mathcal{O}_{\mathcal{F}}(t \cdot \log t)},$$

  and we are done!

- $\boxed{\text{Flat Wall Theorem}}$ [Robertson, Seymour. GMXIII. 1995]

  As a representative R is $\mathcal{F}$-minor-free, if $\mathrm{tw}(R \setminus B) > c_{\mathcal{F}}$,

Bounding the set of representatives

- $\mathcal{R}^{(\mathcal{F},t)}$: set of minimum-size representatives of $\equiv^{(\mathcal{F},t)}$.
- Suppose that we can prove that, for every $R \in \mathcal{R}^{(\mathcal{F},t)}$,
$$|V(R)| = \mathcal{O}_{\mathcal{F}}(t).$$
- Then, by the sparsity of the representatives,
$$|\mathcal{R}^{(\mathcal{F},t)}| = \mathcal{O}_{\mathcal{F}}(1) \cdot \binom{t^2}{t} = 2^{\mathcal{O}_{\mathcal{F}}(t \cdot \log t)},$$
and we are done!

- $\boxed{\text{Flat Wall Theorem}}$ [Robertson, Seymour. GMXIII. 1995]

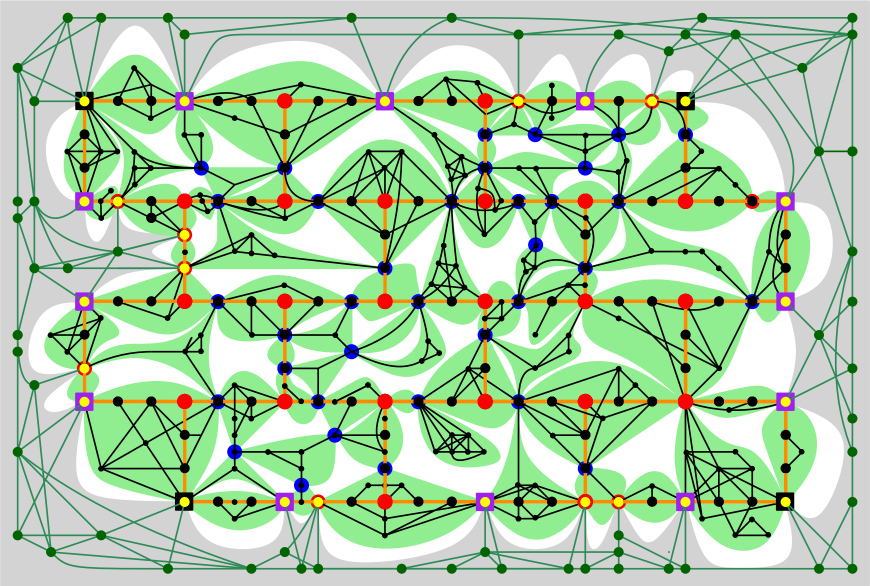
  As a representative R is $\mathcal{F}$-minor-free, if $\mathrm{tw}(R \setminus B) > c_{\mathcal{F}}$, $R \setminus B$ contains a large flat wall,

Bounding the set of representatives

- $\mathcal{R}^{(\mathcal{F},t)}$: set of minimum-size representatives of $\equiv^{(\mathcal{F},t)}$.
- Suppose that we can prove that, for every $R \in \mathcal{R}^{(\mathcal{F},t)}$,
$$|V(R)| = \mathcal{O}_{\mathcal{F}}(t).$$
- Then, by the sparsity of the representatives,
$$|\mathcal{R}^{(\mathcal{F},t)}| = \mathcal{O}_{\mathcal{F}}(1) \cdot \binom{t^2}{t} = 2^{\mathcal{O}_{\mathcal{F}}(t \cdot \log t)},$$
and we are done!
- Flat Wall Theorem [Robertson, Seymour. GMXIII. 1995]

  As a representative 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.

A flat wall can in fact be quite wild...

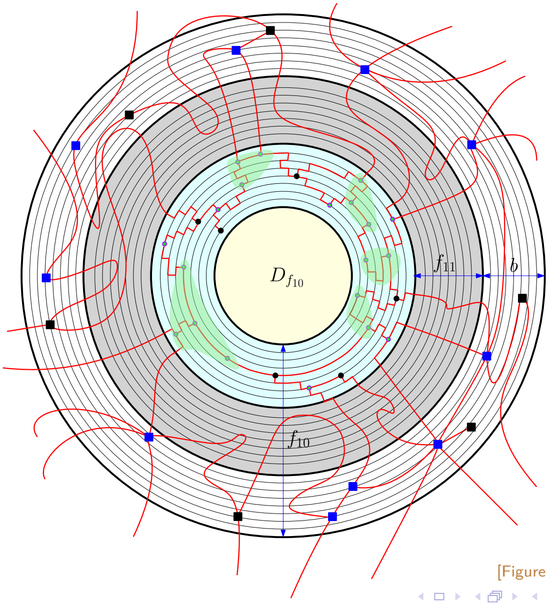


skip

[Figure by Dimitrios M. Thilikos]

Hard part: finding an irrelevant vertex inside a flat wall

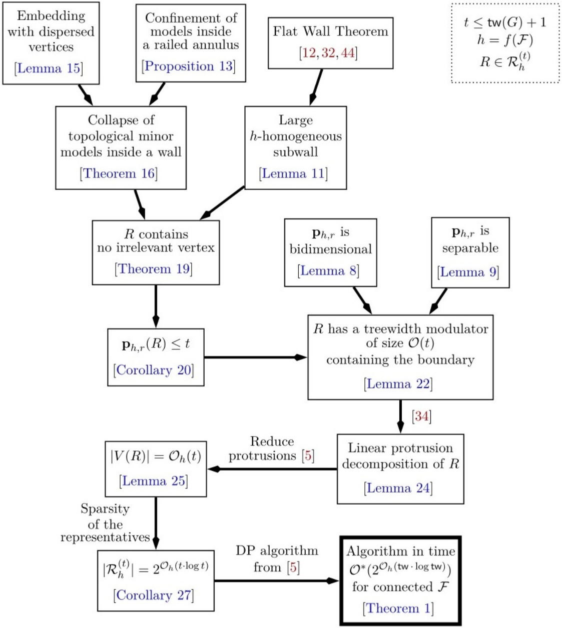
Hard part: finding an irrelevant vertex inside a flat wall



[Figure by Dimitrios M. Thilikos]

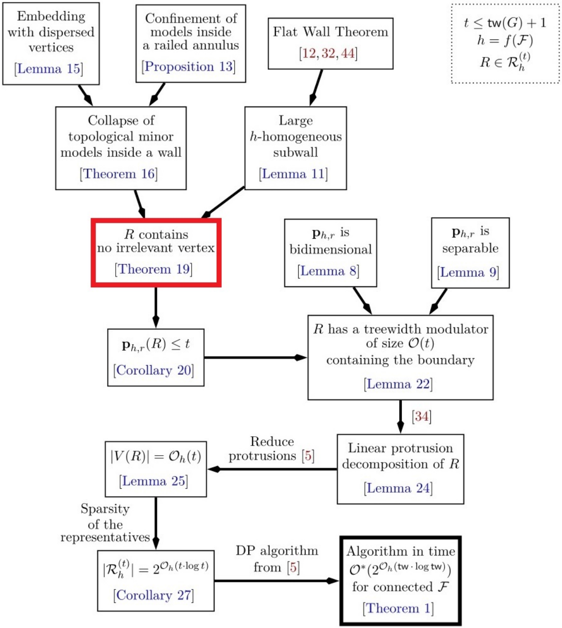
skip

Diagram of the algorithm for a general collection $\mathcal{F}$



skip

Diagram of the algorithm for a general collection $\mathcal{F}$



skip

Next subsection is...

1. Introduction
   - Parameterized complexity
   - Treewidth
2. Hitting forbidden minors
   - Parameterized by treewidth
   - Parameterized by solution size

We parameterize by the size of the desired solution

$\mathcal{F}$-M-DELETION

Input: A graph G and an integer k.

Parameter: k.

Question: Does G contain a set $S \subseteq V(G)$ with $|S| \leqslant k$ such that $G \setminus S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?

We parameterize by the size of the desired solution

$\mathcal{F}$-M-DELETION

Input: A graph G and an integer k.

Parameter: k.

Question: Does G contain a set $S \subseteq V(G)$ with $|S| \leqslant k$ such that $G \setminus S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?

It is easy to see that, for every $k \geqslant 1$, the class of graphs

$$\mathcal{C}_k = \{G \mid (G, k) \text{ is a positive instance of } \mathcal{F}\text{-M-DELETION}\}$$

is minor-closed.

We parameterize by the size of the desired solution

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

It is easy to see that, for every $k \geqslant 1$, the class of graphs

$$\mathcal{C}_k = \{G \mid (G, k) \text{ is a positive instance of } \mathcal{F}\text{-M-DELETION}\}$$

is minor-closed.

Theorem (Robertson and Seymour. 1983-2004)

For every minor-closed graph class $\mathcal{C}$, deciding whether an n-vertex graph G belongs to $\mathcal{C}$ can be solved in time $f(\mathcal{C}) \cdot n^3$.

We parameterize by the size of the desired solution

$\mathcal{F}$-M-DELETION

Input: A graph G and an integer k.

Parameter: k.

Question: Does G contain a set $S \subseteq V(G)$ with $|S| \leqslant k$ such that $G \setminus S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?

It is easy to see that, for every $k \geqslant 1$, the class of graphs

$$\mathcal{C}_k = \{G \mid (G, k) \text{ is a positive instance of } \mathcal{F}\text{-M-DELETION}\}$$

is minor-closed.

Theorem (Robertson and Seymour. 1983-2004)

For every minor-closed graph class $\mathcal{C}$, deciding whether an n-vertex graph G belongs to $\mathcal{C}$ can be solved in time $f(\mathcal{C}) \cdot n^3$.

For every $k \geq 1$, there exists an FPT algorithm for $\mathcal{F}$-M-DELETION.

We parameterize by the size of the desired solution

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

It is easy to see that, for every $k \geqslant 1$, the class of graphs

$$\mathcal{C}_k = \{G \mid (G, k) \text{ is a positive instance of } \mathcal{F}\text{-M-Deletion}\}$$

is minor-closed.

Theorem (Robertson and Seymour. 1983-2004)

For every minor-closed graph class $\mathcal{C}$, deciding whether an n-vertex graph G belongs to $\mathcal{C}$ can be solved in time $f(\mathcal{C}) \cdot n^3$.

For every $k \geq 1$, there exists an FPT algorithm for $\mathcal{F}$-M-Deletion.

But... only existential, non-uniform, $f(\mathcal{C}_k)$ astronomical.

Can we do better?

- The function $f(\mathcal{C}_k)$ is constructible. [Adler, Grohe, Kreutzer. 2008]

Can we do better?

- The function $f(\mathcal{C}_k)$ is constructible. [Adler, Grohe, Kreutzer. 2008]
- If $\mathcal{F}$ contains a planar graph: $2^{\mathcal{O}_{\mathcal{F}}(k)} \cdot n^{\mathcal{O}(1)}$.

[Fomin, Lokshtanov, Misra, Saurabh. 2012]

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

Can we do better?

- The function $f(\mathcal{C}_k)$ is constructible. [Adler, Grohe, Kreutzer. 2008]
- If $\mathcal{F}$ contains a planar graph: $2^{\mathcal{O}_{\mathcal{F}}(k)} \cdot n^{\mathcal{O}(1)}$.

  [Fomin, Lokshtanov, Misra, Saurabh. 2012]

  [Kim, Langer, Paul , Reidl, Rossmanith, S., Sikdar. 2013]
- For some non-planar collections $\mathcal{F}$:
  - $\mathcal{F} = \{K_5, K_{3,3}\}$: $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$. [Jansen, Lokshtanov, Saurabh. 2014]

Can we do better?

- The function $f(\mathcal{C}_k)$ is constructible. [Adler, Grohe, Kreutzer. 2008]
- If $\mathcal{F}$ contains a planar graph: $2^{\mathcal{O}_{\mathcal{F}}(k)} \cdot n^{\mathcal{O}(1)}$.

  [Fomin, Lokshtanov, Misra, Saurabh. 2012]

  [Kim, Langer, Paul , Reidl, Rossmanith, S., Sikdar. 2013]
- For some non-planar collections $\mathcal{F}$:
  - $\mathcal{F} = \{K_5, K_{3,3}\}$: $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$. [Jansen, Lokshtanov, Saurabh. 2014]
  - Deletion to genus at most g: $2^{\mathcal{O}_g(k^2 \log k)} \cdot n^{\mathcal{O}(1)}$. [Kociumaka, Ma. Pilipczuk. 2019]

Can we do better?

- The function $f(\mathcal{C}_k)$ is constructible. [Adler, Grohe, Kreutzer. 2008]
- If $\mathcal{F}$ contains a planar graph: $2^{\mathcal{O}_{\mathcal{F}}(k)} \cdot n^{\mathcal{O}(1)}$.

  [Fomin, Lokshtanov, Misra, Saurabh. 2012]

  [Kim, Langer, Paul , Reidl, Rossmanith, S., Sikdar. 2013]
- For some non-planar collections $\mathcal{F}$:
  - $\mathcal{F} = \{K_5, K_{3,3}\}$: $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$. [Jansen, Lokshtanov, Saurabh. 2014]
  - Deletion to genus at most g: $2^{\mathcal{O}_g(k^2 \log k)} \cdot n^{\mathcal{O}(1)}$. [Kociumaka, Ma. Pilipczuk. 2019]
- For every $\mathcal{F}$, some enormous explicit function $f_{\mathcal{F}}(k)$ can be derived from an FPT algorithm for hitting topological minors:

  $$f_{\mathcal{F}}(k) \cdot n^{\mathcal{O}(1)}.$$

  [Fomin, Lokshtanov, Panolan, Saurabh, Zehavi. 2020]

Our results

Theorem (S., Stamoulis, Thilikos. 2020)

For all $\mathcal{F}$, the $\mathcal{F}$-M-DELETION problem can be solved in time $2^{\text{poly}(k)} \cdot n^3$.

Here, $\text{poly}(k)$ is a polynomial whose degree depends on $\mathcal{F}$.

Our results

Theorem (S., Stamoulis, Thilikos. 2020)

For all $\mathcal{F}$, the $\mathcal{F}$-M-Deletion problem can be solved in time $2^{\text{poly}(k)} \cdot n^3$.

Here, $\text{poly}(k)$ is a polynomial whose degree depends on $\mathcal{F}$.

Theorem (S., Stamoulis, Thilikos. 2020)

If $\mathcal{F}$ contains an apex graph, the $\mathcal{F}$-M-Deletion problem can be solved in time $2^{\text{poly}(k)} \cdot n^2$.

Again, $\text{poly}(k)$ is a polynomial whose degree depends on $\mathcal{F}$.

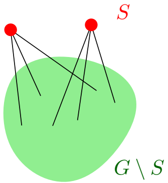
skip

General scheme of the algorithm:

[whole slide shamelessly borrowed from Giannos Stamoulis]

General scheme of the algorithm:

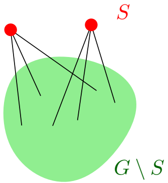
[whole slide shamelessly borrowed from Giannos Stamoulis]



Iterative compression: given solution S of size $k+1$, search solution of size k.

General scheme of the algorithm:

[whole slide shamelessly borrowed from Giannos Stamoulis]

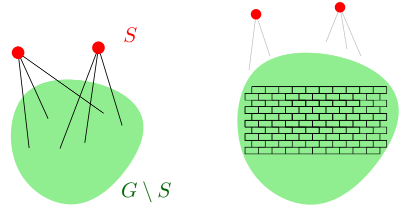


Iterative compression: given solution S of size $k+1$, search solution of size k.

If treewidth of $G \setminus S$ is "large enough" (as a polynomial function of k):

General scheme of the algorithm:

[whole slide shamelessly borrowed from Giannos Stamoulis]



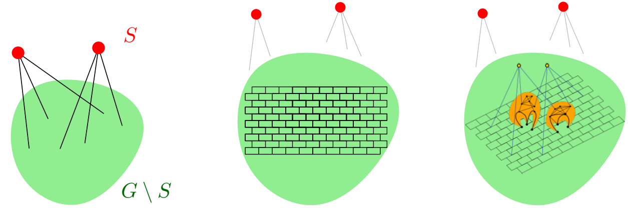
Iterative compression: given solution S of size $k+1$, search solution of size k.

If treewidth of $G \setminus S$ is "large enough" (as a polynomial function of k):

1. Find a "very very large" wall in $G \setminus S$.

General scheme of the algorithm:

[whole slide shamelessly borrowed from Giannos Stamoulis]



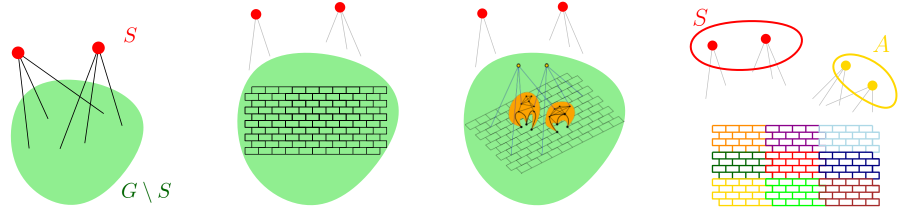
Iterative compression: given solution S of size $k+1$, search solution of size k.

If treewidth of $G \setminus S$ is "large enough" (as a polynomial function of k):

1. Find a "very very large" wall in $G \setminus S$.
2. Find a "very large" flat wall W of $G \setminus S$ with few apices A.

General scheme of the algorithm:

[whole slide shamelessly borrowed from Giannos Stamoulis]



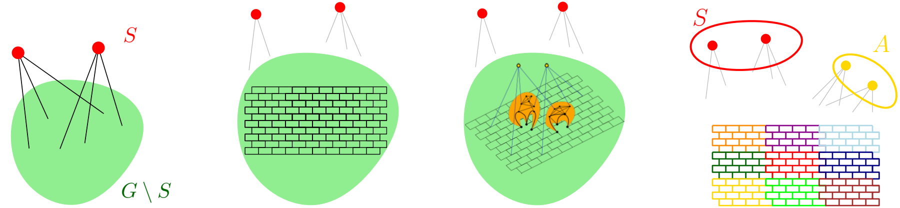
Iterative compression: given solution S of size $k+1$, search solution of size k.

If treewidth of $G \setminus S$ is "large enough" (as a polynomial function of k):

1. Find a "very very large" wall in $G \setminus S$.
2. Find a "very large" flat wall W of $G \setminus S$ with few apices A.
3. Find in W a packing of $\mathcal{O}_{\mathcal{F}}(k^4)$ disjoint "large" subwalls:

General scheme of the algorithm:

[whole slide shamelessly borrowed from Giannos Stamoulis]



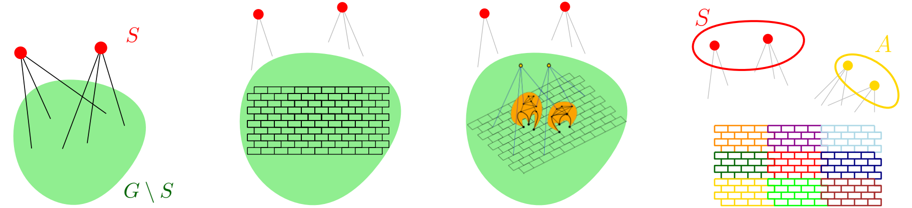
Iterative compression: given solution S of size $k+1$, search solution of size k.

If treewidth of $G \setminus S$ is "large enough" (as a polynomial function of k):

1. Find a "very very large" wall in $G \setminus S$.
2. Find a "very large" flat wall W of $G \setminus S$ with few apices A.
3. Find in W a packing of $\mathcal{O}_{\mathcal{F}}(k^4)$ disjoint "large" subwalls:
   - If every subwall has at least $|A|+1$ neighbors in $S \cup A$:

General scheme of the algorithm:

[whole slide shamelessly borrowed from Giannos Stamoulis]



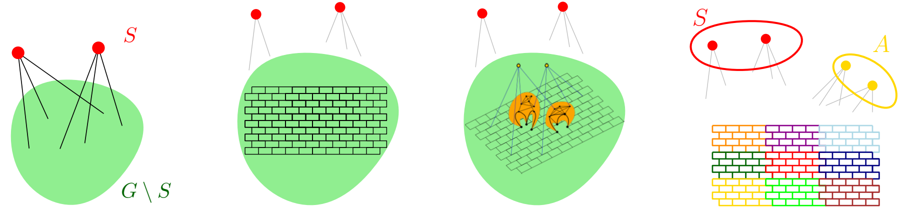
Iterative compression: given solution S of size $k+1$, search solution of size k.

If treewidth of $G \setminus S$ is "large enough" (as a polynomial function of k):

1. Find a "very very large" wall in $G \setminus S$.
2. Find a "very large" flat wall W of $G \setminus S$ with few apices A.
3. Find in W a packing of $\mathcal{O}_{\mathcal{F}}(k^4)$ disjoint "large" subwalls:
    - If every subwall has at least $|A|+1$ neighbors in $S \cup A$:
      Every solution intersects $S \cup A$ → we can branch!

General scheme of the algorithm:

[whole slide shamelessly borrowed from Giannos Stamoulis]



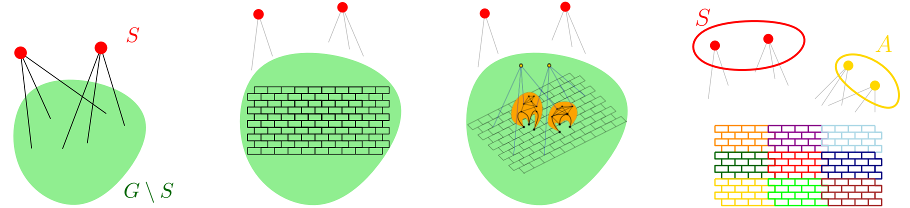
Iterative compression: given solution S of size $k+1$, search solution of size k.

If treewidth of $G \setminus S$ is "large enough" (as a polynomial function of k):

1. Find a "very very large" wall in $G \setminus S$.
2. Find a "very large" flat wall W of $G \setminus S$ with few apices A.
3. Find in W a packing of $\mathcal{O}_{\mathcal{F}}(k^4)$ disjoint "large" subwalls:
    - If every subwall has at least $|A|+1$ neighbors in $S \cup A$:
      Every solution intersects $S \cup A$ $\rightarrow$ we can branch!
    - If one of these subwalls has at most $|A|$ neighbors in $S \cup A$:

General scheme of the algorithm:

[whole slide shamelessly borrowed from Giannos Stamoulis]



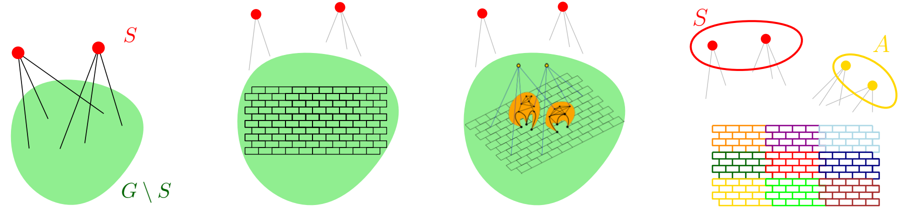
Iterative compression: given solution S of size $k+1$, search solution of size k.

If treewidth of $G \setminus S$ is "large enough" (as a polynomial function of k):

1. Find a "very very large" wall in $G \setminus S$.
2. Find a "very large" flat wall W of $G \setminus S$ with few apices A.
3. Find in W a packing of $\mathcal{O}_{\mathcal{F}}(k^4)$ disjoint "large" subwalls:
   - If every subwall has at least $|A|+1$ neighbors in $S \cup A$:
     Every solution intersects $S \cup A$ $\rightarrow$ we can branch!
   - If one of these subwalls has at most $|A|$ neighbors in $S \cup A$:
     Find an irrelevant vertex v inside this flat subwall.
     Update $G = G \setminus v$ and repeat.

General scheme of the algorithm:

[whole slide shamelessly borrowed from Giannos Stamoulis]



Iterative compression: given solution S of size $k+1$, search solution of size k.

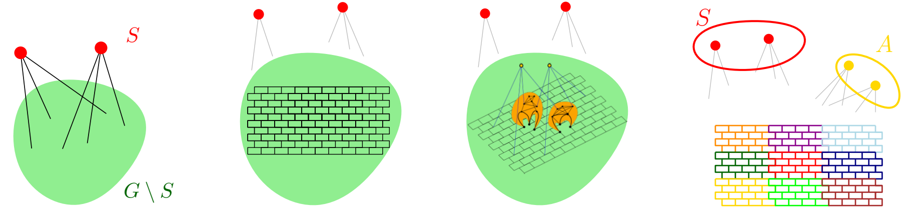
If treewidth of $G \setminus S$ is "large enough" (as a polynomial function of k):

1. Find a "very very large" wall in $G \setminus S$.
2. Find a "very large" flat wall W of $G \setminus S$ with few apices A.
3. Find in W a packing of $\mathcal{O}_{\mathcal{F}}(k^4)$ disjoint "large" subwalls:
   - If every subwall has at least $|A|+1$ neighbors in $S \cup A$:
     Every solution intersects $S \cup A$ $\rightarrow$ we can branch!
   - If one of these subwalls has at most $|A|$ neighbors in $S \cup A$:
     Find an irrelevant vertex v inside this flat subwall.
     Update $G = G \setminus v$ and repeat.

Thus, $\mathrm{tw}(G \setminus S) = k^{\mathcal{O}_{\mathcal{F}}(1)}$.

General scheme of the algorithm:

[whole slide shamelessly borrowed from Giannos Stamoulis]



Iterative compression: given solution S of size $k+1$, search solution of size k.

If treewidth of $G \setminus S$ is "large enough" (as a polynomial function of k):

1. Find a "very very large" wall in $G \setminus S$.
2. Find a "very large" flat wall W of $G \setminus S$ with few apices A.
3. Find in W a packing of $\mathcal{O}_{\mathcal{F}}(k^4)$ disjoint "large" subwalls:
   - If every subwall has at least $|A|+1$ neighbors in $S \cup A$:
     Every solution intersects $S \cup A$ $\rightarrow$ we can branch!
   - If one of these subwalls has at most $|A|$ neighbors in $S \cup A$:
     Find an irrelevant vertex v inside this flat subwall.
     Update $G = G \setminus v$ and repeat.

Thus, $\mathsf{tw}(G \setminus S) = k^{\mathcal{O}_{\mathcal{F}}(1)}$: our previous FPT algo gives $2^{k^{\mathcal{O}_{\mathcal{F}}(1)}} \cdot n^2$.

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

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

$\boxed{\text{With parameter tw}}$ Classify the asymptotic complexity of $\mathcal{F}$-M-Deletion for every family $\mathcal{F}$?

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

$\boxed{\text{With parameter tw}}$ Classify the asymptotic complexity of $\mathcal{F}$-M-Deletion for every family $\mathcal{F}$?

- We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).

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

$\boxed{\text{With parameter tw}}$ Classify the asymptotic complexity of $\mathcal{F}$-M-Deletion for every family $\mathcal{F}$?

- We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).
- Missing: When $|\mathcal{F}| \geq 2$ (connected): $2^{\Theta(\mathsf{tw})}$ or $2^{\Theta(\mathsf{tw} \cdot \log \mathsf{tw})}$?

$\boxed{\text{With parameter tw}}$ Classify the asymptotic complexity of $\mathcal{F}$-M-Deletion for every family $\mathcal{F}$?

- We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).
- Missing: When $|\mathcal{F}| \geq 2$ (connected): $2^{\Theta(\mathrm{tw})}$ or $2^{\Theta(\mathrm{tw} \cdot \log \mathrm{tw})}$?

We can also consider the $\boxed{\text{topological minor}}$ version:

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

$\boxed{\text{With parameter tw}}$ Classify the asymptotic complexity of $\mathcal{F}$-M-Deletion for every family $\mathcal{F}$?

- We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).
- Missing: When $|\mathcal{F}| \geq 2$ (connected): $2^{\Theta(\mathsf{tw})}$ or $2^{\Theta(\mathsf{tw} \cdot \log \mathsf{tw})}$?

We can also consider the $\boxed{\text{topological minor}}$ version:

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

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

$\boxed{\text{With parameter tw}}$ Classify the asymptotic complexity of $\mathcal{F}$-M-Deletion for every family $\mathcal{F}$?

- We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).
- Missing: When $|\mathcal{F}| \geq 2$ (connected): $2^{\Theta(\text{tw})}$ or $2^{\Theta(\text{tw} \cdot \log \text{tw})}$?

We can also consider 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 $2^{o(\text{tw}^2)} \cdot n^{\mathcal{O}(1)}$ under the ETH.

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

$\boxed{\text{With parameter tw}}$ Classify the asymptotic complexity of $\mathcal{F}$-M-DELETION for every family $\mathcal{F}$?

- We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).
- Missing: When $|\mathcal{F}| \geq 2$ (connected): $2^{\Theta(\text{tw})}$ or $2^{\Theta(\text{tw} \cdot \log \text{tw})}$?

We can also consider 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 $2^{o(\text{tw}^2)} \cdot n^{\mathcal{O}(1)}$ under the ETH.

$\boxed{\text{With parameter } k}$ We presented algorithm in time $2^{k^{\mathcal{O}_{\mathcal{F}}(1)}} \cdot n^3$.

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

$\boxed{\text{With parameter tw}}$ Classify the asymptotic complexity of $\mathcal{F}$-M-Deletion for every family $\mathcal{F}$?

- We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).
- Missing: When $|\mathcal{F}| \geq 2$ (connected): $2^{\Theta(\mathrm{tw})}$ or $2^{\Theta(\mathrm{tw} \cdot \log \mathrm{tw})}$?

We can also consider 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 $2^{o(\mathrm{tw}^2)} \cdot n^{\mathcal{O}(1)}$ under the ETH.

$\boxed{\text{With parameter } k}$ We presented algorithm in time $2^{k^{\mathcal{O}_{\mathcal{F}}(1)}} \cdot n^3$.

Is $2^{\mathcal{O}_{\mathcal{F}}(k^c)} \cdot n^{\mathcal{O}(1)}$ possible for some constant c?

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

$\boxed{\text{With parameter tw}}$ Classify the asymptotic complexity of $\mathcal{F}$-M-Deletion for every family $\mathcal{F}$?

- We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).
- Missing: When $|\mathcal{F}| \geq 2$ (connected): $2^{\Theta(\text{tw})}$ or $2^{\Theta(\text{tw} \cdot \log \text{tw})}$?

We can also consider 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 $2^{o(\text{tw}^2)} \cdot n^{\mathcal{O}(1)}$ under the ETH.

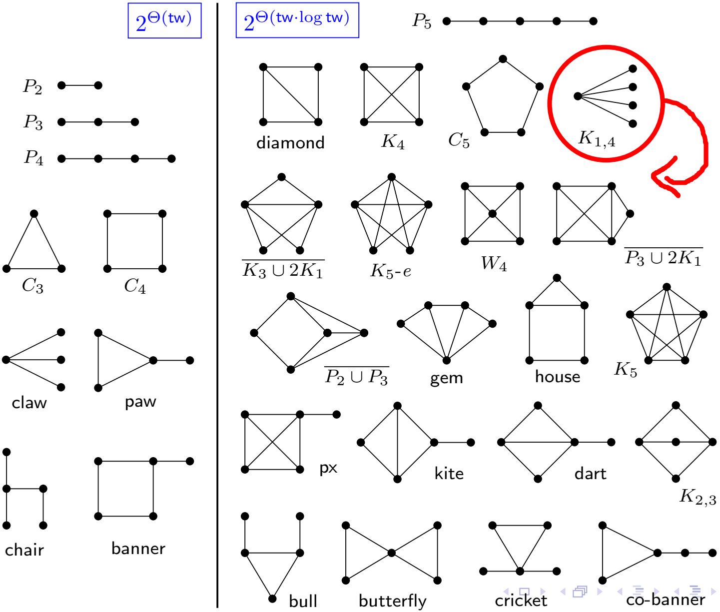
$\boxed{\text{With parameter } k}$ We presented algorithm in time $2^{k^{\mathcal{O}_{\mathcal{F}}(1)}} \cdot n^3$.

Is $2^{\mathcal{O}_{\mathcal{F}}(k^c)} \cdot n^{\mathcal{O}(1)}$ possible for some constant c?

Is the price of homogeneity unavoidable?

skip

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



Gràcies!

Toda raba!