Introduction + Bidimensionality Theory

Ignasi Sau

CNRS, LIRMM, Montpellier (France)

Many thanks to Dimitrios M. Thilikos!!

14èmes JCALM. October 2013. *UPC, Barcelona, Catalunya*

Outline of the talk

1. Introduction, part II
   - Treewidth
   - Dynamic programming on tree decompositions
   - Structure of H-minor-free graphs
   - Some algorithmic issues
   - A few words on other containment relations

2. Bidimensionality
   - Some ingredients
   - An illustrative example
   - Meta-algorithms
   - Further extensions

Next section is...

1. Introduction, part II
   - Treewidth
   - Dynamic programming on tree decompositions
   - Structure of H-minor-free graphs
   - Some algorithmic issues
   - A few words on other containment relations
2. Bidimensionality
   - Some ingredients
   - An illustrative example
   - Meta-algorithms
   - Further extensions

Parameterized complexity in one slide

- Idea given an NP-hard problem, fix one parameter of the input to see if the problem gets more "tractable".

  Example: the size of a Vertex Cover.

- Given a (NP-hard) problem with input of size n and a parameter k, a fixed-parameter tractable (FPT) algorithm runs in

$$f(k) \cdot n^{\mathcal{O}(1)}, \quad \text{for some function } f.$$

  Examples: k-Vertex Cover, k-Longest Path.

Parameterized complexity in one slide

- Idea given an NP-hard problem, fix one parameter of the input to see if the problem gets more "tractable".

  Example: the size of a VERTEX COVER.

- Given a (NP-hard) problem with input of size n and a parameter k, a fixed-parameter tractable (FPT) algorithm runs in

$$f(k) \cdot n^{\mathcal{O}(1)}, \quad \text{for some function } f.$$

  Examples: k-VERTEX COVER, k-LONGEST PATH.

- Single-exponential FPT algorithm: $2^{O(k)} \cdot n^{O(1)}$

Parameterized complexity in one slide

- Idea given an NP-hard problem, fix one parameter of the input to see if the problem gets more "tractable".

  Example: the size of a Vertex Cover.

- Given a (NP-hard) problem with input of size n and a parameter k, a fixed-parameter tractable (FPT) algorithm runs in

$$f(k) \cdot n^{\mathcal{O}(1)}, \quad \text{for some function } f.$$

  Examples: k-Vertex Cover, k-Longest Path.

- Single-exponential FPT algorithm: $2^{O(k)} \cdot n^{O(1)}$

- Subexponential FPT algorithm: $2^{o(k)} \cdot n^{O(1)}$

Next subsection is...

1. Introduction, part II
   - Treewidth
   - Dynamic programming on tree decompositions
   - Structure of H-minor-free graphs
   - Some algorithmic issues
   - A few words on other containment relations

2. Bidimensionality
   - Some ingredients
   - An illustrative example
   - Meta-algorithms
   - Further extensions

Tree decompositions and treewidth

A *tree decomposition* of a graph G is a pair $D = (T, \mathcal{X})$ such that T is a tree and $\mathcal{X} = \{X_t \mid t \in V(T)\}$ is a collection of subsets of $V(G)$ such that:

(each $X_t \in \mathcal{X}$ corresponds to a vertex $t \in V(T)$: we call X_t *node* of D)

Tree decompositions and treewidth

A *tree decomposition* of a graph G is a pair $D = (T, \mathcal{X})$ such that T is a tree and $\mathcal{X} = \{X_t \mid t \in V(T)\}$ is a collection of subsets of $V(G)$ such that:

(each $X_t \in \mathcal{X}$ corresponds to a vertex $t \in V(T)$: we call X_t *node* of D)

- Any vertex $v \in V(G)$ and the endpoints of any edge $e \in E(G)$ belong to some node X_t of D; and
- For any $v \in V(G)$, the set $\{t \in V(T) \mid v \in X_t\}$ is a subtree of T.

Tree decompositions and treewidth

A *tree decomposition* of a graph G is a pair $D = (T, \mathcal{X})$ such that T is a tree and $\mathcal{X} = \{X_t \mid t \in V(T)\}$ is a collection of subsets of $V(G)$ such that:

(each $X_t \in \mathcal{X}$ corresponds to a vertex $t \in V(T)$: we call X_t *node* of D)

- Any vertex $v \in V(G)$ and the endpoints of any edge $e \in E(G)$ belong to some node X_t of D; and
- For any $v \in V(G)$, the set $\{t \in V(T) \mid v \in X_t\}$ is a subtree of T.

The *width* of a tree decomposition is $\max\{|X_t| \mid t \in V(T)\} - 1$.

Tree decompositions and treewidth

A *tree decomposition* of a graph G is a pair $D = (T, \mathcal{X})$ such that T is a tree and $\mathcal{X} = \{X_t \mid t \in V(T)\}$ is a collection of subsets of $V(G)$ such that:

(each $X_t \in \mathcal{X}$ corresponds to a vertex $t \in V(T)$: we call X_t *node* of D)

- Any vertex $v \in V(G)$ and the endpoints of any edge $e \in E(G)$ belong to some node X_t of D; and
- For any $v \in V(G)$, the set $\{t \in V(T) \mid v \in X_t\}$ is a subtree of T.

The *width* of a tree decomposition is $\max\{|X_t| \mid t \in V(T)\} - 1$.

The *treewidth* of a graph G, denoted $\mathrm{tw}(G)$, is the minimum width over all tree decompositions of G.

Tree decompositions and treewidth

A *tree decomposition* of a graph G is a pair $D = (T, \mathcal{X})$ such that T is a tree and $\mathcal{X} = \{X_t \mid t \in V(T)\}$ is a collection of subsets of $V(G)$ such that:

(each $X_t \in \mathcal{X}$ corresponds to a vertex $t \in V(T)$: we call X_t *node* of D)

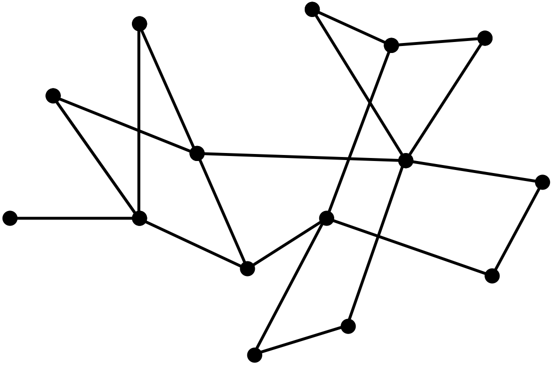
- Any vertex $v \in V(G)$ and the endpoints of any edge $e \in E(G)$ belong to some node X_t of D; and
- For any $v \in V(G)$, the set $\{t \in V(T) \mid v \in X_t\}$ is a subtree of T.

The *width* of a tree decomposition is $\max\{|X_t| \mid t \in V(T)\} - 1$.

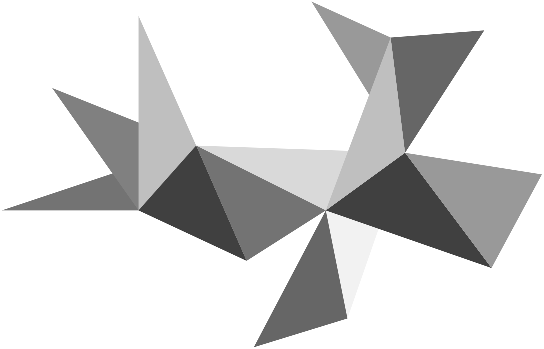
The *treewidth* of a graph G, denoted $\mathrm{tw}(G)$, is the minimum width over all tree decompositions of G.

Invariant that measures the topological complexity of a graph.

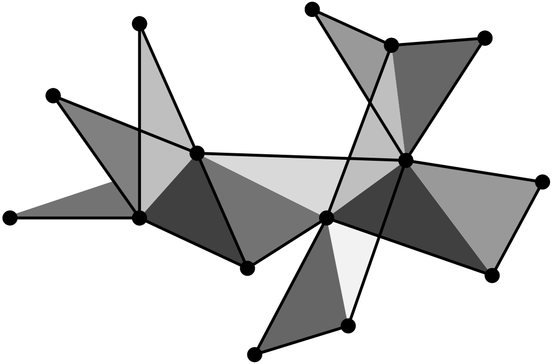
Tree decompositions: example



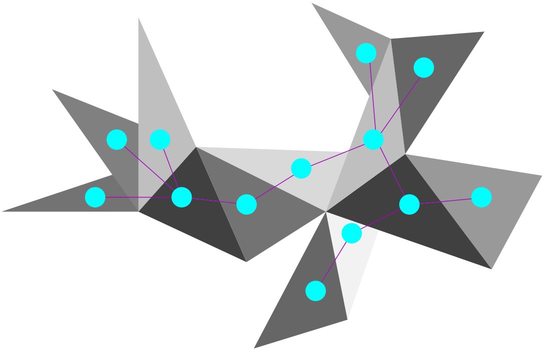
Tree decompositions: example



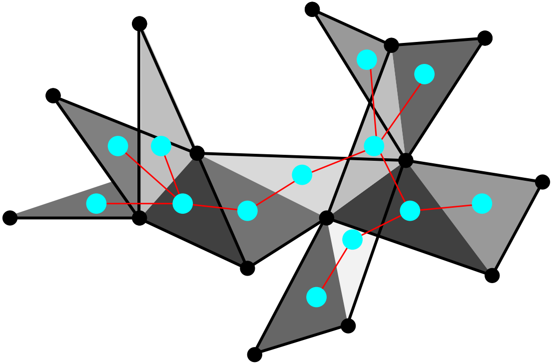
Tree decompositions: example



Tree decompositions: example



Tree decompositions: example



About computing treewidth: a global picture

1. Computing treewidth is NP-hard [Arnborg, Corneil, Proskurowski. 1987]

About computing treewidth: a global picture

1. Computing treewidth is NP-hard [Arnborg, Corneil, Proskurowski. 1987]
2. Approximation algorithms on general graphs
   - $O(\textsc{Opt} \cdot \sqrt{\textsc{Opt}})$-approximation. [Feige, Hajiaghayi, Lee. 2008]

About computing treewidth: a global picture

1. Computing treewidth is NP-hard [Arnborg, Corneil, Proskurowski. 1987]
2. Approximation algorithms on general graphs
   - $O(\textsc{Opt} \cdot \sqrt{\textsc{Opt}})$-approximation. [Feige, Hajiaghayi, Lee. 2008]
3. (Exact) FPT algorithms on general graphs ($\text{tw} \leqslant k$?)

About computing treewidth: a global picture

1. Computing treewidth is NP-hard [Arnborg, Corneil, Proskurowski. 1987]

2. Approximation algorithms on general graphs
   - $O(\textsc{Opt} \cdot \sqrt{\textsc{Opt}})$-approximation. [Feige, Hajiaghayi, Lee. 2008]

3. (Exact) FPT algorithms on general graphs ($\mathrm{tw} \leqslant k$?)
   - In time $O(n^{k+2})$. [Arnborg, Corneil, Proskurowski. 1987]

About computing treewidth: a global picture

1. Computing treewidth is NP-hard [Arnborg, Corneil, Proskurowski. 1987]
2. Approximation algorithms on general graphs
   - $O(\textsc{Opt} \cdot \sqrt{\textsc{Opt}})$-approximation. [Feige, Hajiaghayi, Lee. 2008]
3. (Exact) FPT algorithms on general graphs ($\mathrm{tw} \leqslant k$?)
   - In time $O(n^{k+2})$. [Arnborg, Corneil, Proskurowski. 1987]
   - In time $k^{O(k^3)} \cdot n$. [Bodlaender. 1996]

About computing treewidth: a global picture

1. Computing treewidth is NP-hard [Arnborg, Corneil, Proskurowski. 1987]
2. Approximation algorithms on general graphs
   - $O(\textsc{Opt} \cdot \sqrt{\textsc{Opt}})$-approximation. [Feige, Hajiaghayi, Lee. 2008]
3. (Exact) FPT algorithms on general graphs ($\mathrm{tw} \leqslant k$?)
   - In time $O(n^{k+2})$. [Arnborg, Corneil, Proskurowski. 1987]
   - In time $k^{O(k^3)} \cdot n$. [Bodlaender. 1996]
4. FPT approximation algorithms on general graphs
   - 4.5-approximation in time $2^{O(k)} \cdot n^2$. [Amir. 2010]

About computing treewidth: a global picture

1. Computing treewidth is NP-hard — [Arnborg, Corneil, Proskurowski. 1987]

2. Approximation algorithms on general graphs
   - $O(\textsc{Opt} \cdot \sqrt{\textsc{Opt}})$-approximation. [Feige, Hajiaghayi, Lee. 2008]

3. (Exact) FPT algorithms on general graphs ($\text{tw} \leqslant k$?)
   - In time $O(n^{k+2})$. [Arnborg, Corneil, Proskurowski. 1987]
   - In time $k^{O(k^3)} \cdot n$. [Bodlaender. 1996]

4. FPT approximation algorithms on general graphs
   - 4.5-approximation in time $2^{O(k)} \cdot n^2$. [Amir. 2010]
   - 5-approx. in $2^{O(k)} \cdot n$. [Bodlaender, Drange, Dregi, Fomin, Lokshtanov, Pilipczuk. 2013]

About computing treewidth: a global picture

1. Computing treewidth is NP-hard — [Arnborg, Corneil, Proskurowski. 1987]
2. Approximation algorithms on general graphs
   - $O(\text{OPT} \cdot \sqrt{\text{OPT}})$-approximation. — [Feige, Hajiaghayi, Lee. 2008]
3. (Exact) FPT algorithms on general graphs ($\text{tw} \leqslant k$?)
   - In time $O(n^{k+2})$. — [Arnborg, Corneil, Proskurowski. 1987]
   - In time $k^{O(k^3)} \cdot n$. — [Bodlaender. 1996]
4. FPT approximation algorithms on general graphs
   - 4.5-approximation in time $2^{O(k)} \cdot n^2$. — [Amir. 2010]
   - 5-approx. in $2^{O(k)} \cdot n$. — [Bodlaender, Drange, Dregi, Fomin, Lokshtanov, Pilipczuk. 2013]
5. Approximation algorithms on sparse graphs
   - 3/2-approximation on planar graphs. — [Seymour and Thomas. 1994]

About computing treewidth: a global picture

1. Computing treewidth is NP-hard [Arnborg, Corneil, Proskurowski. 1987]

2. Approximation algorithms on general graphs
   - $O(\textsc{Opt} \cdot \sqrt{\textsc{Opt}})$-approximation. [Feige, Hajiaghayi, Lee. 2008]

3. (Exact) FPT algorithms on general graphs ($\mathrm{tw} \leqslant k$?)
   - In time $O(n^{k+2})$. [Arnborg, Corneil, Proskurowski. 1987]
   - In time $k^{O(k^3)} \cdot n$. [Bodlaender. 1996]

4. FPT approximation algorithms on general graphs
   - 4.5-approximation in time $2^{O(k)} \cdot n^2$. [Amir. 2010]
   - 5-approx. in $2^{O(k)} \cdot n$. [Bodlaender, Drange, Dregi, Fomin, Lokshtanov, Pilipczuk. 2013]

5. Approximation algorithms on sparse graphs
   - $3/2$-approximation on planar graphs. [Seymour and Thomas. 1994]
   - c_H-approximation on H-minor-free graphs. [Demaine and Hajiaghayi. 2008]

About computing treewidth: a global picture

1. Computing treewidth is NP-hard [Arnborg, Corneil, Proskurowski. 1987]
2. Approximation algorithms on general graphs
   - $O(\text{OPT} \cdot \sqrt{\text{OPT}})$-approximation. [Feige, Hajiaghayi, Lee. 2008]
3. (Exact) FPT algorithms on general graphs ($\text{tw} \leqslant k$?)
   - In time $O(n^{k+2})$. [Arnborg, Corneil, Proskurowski. 1987]
   - In time $k^{O(k^3)} \cdot n$. [Bodlaender. 1996]
4. FPT approximation algorithms on general graphs
   - 4.5-approximation in time $2^{O(k)} \cdot n^2$. [Amir. 2010]
   - 5-approx. in $2^{O(k)} \cdot n$. [Bodlaender, Drange, Dregi, Fomin, Lokshtanov, Pilipczuk. 2013]
5. Approximation algorithms on sparse graphs
   - $3/2$-approximation on planar graphs. [Seymour and Thomas. 1994]
   - c_H-approximation on H-minor-free graphs. [Demaine and Hajiaghayi. 2008]
   - Open Compute the treewidth of a planar graph.

Next subsection is...

1. Introduction, part II
    - Treewidth
    - Dynamic programming on tree decompositions
    - Structure of H-minor-free graphs
    - Some algorithmic issues
    - A few words on other containment relations
2. Bidimensionality
    - Some ingredients
    - An illustrative example
    - Meta-algorithms
    - Further extensions

Example: MAX. INDEPENDENT SET

- For each node X_i, let $G_i = G[X_i]$, and let G'_i be the subgraph of G induced by node X_i and its descendants.

Example: Max. Independent Set

- For each node X_i, let $G_i = G[X_i]$, and let G_i' be the subgraph of G induced by node X_i and its descendants.
- Tables of dynamic programming (DP) at each node X_i: (S_i, a_i), where S_i is an independent set in G_i, and a_i is the maximum size of an independent set in G_i' which coincides with S_i in X_i.

Example: Max. Independent Set

- For each node X_i, let $G_i = G[X_i]$, and let G_i' be the subgraph of G induced by node X_i and its descendants.
- Tables of dynamic programming (DP) at each node X_i: (S_i, a_i), where S_i is an independent set in G_i, and a_i is the maximum size of an independent set in G_i' which coincides with S_i in X_i.
- The tables (S_i, a_i) at node X_i can be computed from the tables of its children as follows. For each S_i independent set of G_i:
  - If X_j is a child of X_i, an independent set S_j of G_j is feasible for S_i if $X_j \cap S_i = X_i \cap S_j$.
  - For each children X_j of X_i, let S_j be feasible for S_i s.t. (S_j, a_j) is defined and a_j is maximized.
  - We set $a_i := |S_i| + \sum_j (a_j - |S_i \cap S_j|)$.

Example: MAX. INDEPENDENT SET

- For each node X_i, let $G_i = G[X_i]$, and let G'_i be the subgraph of G induced by node X_i and its descendants.
- Tables of dynamic programming (DP) at each node X_i: (S_i, a_i), where S_i is an independent set in G_i, and a_i is the maximum size of an independent set in G'_i which coincides with S_i in X_i.
- The tables (S_i, a_i) at node X_i can be computed from the tables of its children as follows. For each S_i independent set of G_i:
  - If X_j is a child of X_i, an independent set S_j of G_j is feasible for S_i if $X_j \cap S_i = X_i \cap S_j$.
  - For each children X_j of X_i, let S_j be feasible for S_i s.t. (S_j, a_j) is defined and a_j is maximized.
  - We set $a_i := |S_i| + \sum_j (a_j - |S_i \cap S_j|)$.
- Running time for each node X_i: $O(2^{2|X_i|})$.
- Overall running time: $2^{O(\mathrm{tw}(G))} \cdot n$.

Courcelle's Theorem - algorithmic importance of treewidth

What we have seen with Max. Independent Set can be generalized to a wide family of problems:

Theorem (Courcelle. 1988)

Graph problems expressible in Monadic Second Order Logic (MSOL) can be solved in time $f(k) \cdot n^{O(1)}$ in graphs with n vertices and $\mathrm{tw} \leqslant k$.

Courcelle's Theorem - algorithmic importance of treewidth

What we have seen with MAX. INDEPENDENT SET can be generalized to a wide family of problems:

Theorem (Courcelle. 1988)

Graph problems expressible in Monadic Second Order Logic (MSOL) can be solved in time $f(k) \cdot n^{O(1)}$ in graphs with n vertices and $\mathrm{tw} \leqslant k$.

- In other words, all these problems are fixed -parameter tractable (FPT) when parameterized by the treewidth of their input graphs.

Courcelle's Theorem - algorithmic importance of treewidth

What we have seen with MAX. INDEPENDENT SET can be generalized to a wide family of problems:

Theorem (Courcelle. 1988)

Graph problems expressible in Monadic Second Order Logic (MSOL) can be solved in time $f(k) \cdot n^{O(1)}$ in graphs with n vertices and $\mathrm{tw} \leqslant k$.

- In other words, all these problems are fixed -parameter tractable (FPT) when parameterized by the treewidth of their input graphs.
- Running time (tight):

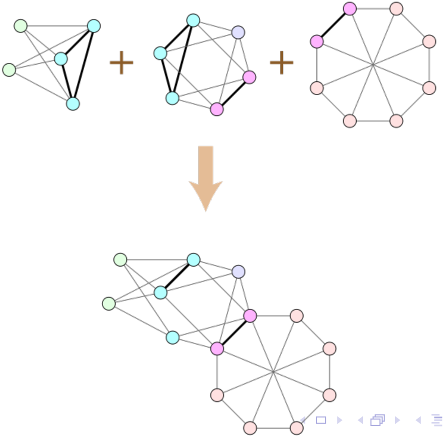
$$2^{2^{2^{\cdots 2^k}}} \cdot n^{O(1)}.$$

Next subsection is...

1. Introduction, part II
   - Treewidth
   - Dynamic programming on tree decompositions
   - Structure of H-minor-free graphs
   - Some algorithmic issues
   - A few words on other containment relations
2. Bidimensionality
   - Some ingredients
   - An illustrative example
   - Meta-algorithms
   - Further extensions

Structure of H-minor-free graphs: roughly "bidimensional"

- Some (simplified) preliminaries:
  - *h*-clique-sum of two graphs G_1 and G_2:
    choose cliques $K_1 \subseteq G_1$ and $K_2 \subseteq G_2$ with $|V(K_1)| = |V(K_2)| = h$, identify them, and possibly remove some edges of that clique.



Structure of H-minor-free graphs: roughly "bidimensional"

- Some (simplified) preliminaries:
  - *h*-clique-sum of two graphs G_1 and G_2:
    choose cliques $K_1 \subseteq G_1$ and $K_2 \subseteq G_2$ with $|V(K_1)| = |V(K_2)| = h$, identify them, and possibly remove some edges of that clique.
  - Apex in an embedded graph:
    add a vertex with any neighbors in the embedded graph.

Structure of H-minor-free graphs: roughly "bidimensional"

- Some (simplified) preliminaries:
  - h-clique-sum of two graphs G_1 and G_2:
    choose cliques $K_1 \subseteq G_1$ and $K_2 \subseteq G_2$ with $|V(K_1)| = |V(K_2)| = h$,
    identify them, and possibly remove some edges of that clique.
  - Apex in an embedded graph:
    add a vertex with any neighbors in the embedded graph.
  - Vortex of depth h in an embedded graph:
    paste a graph of pathwidth at most h in a face of the embedding.

Structure of H-minor-free graphs: roughly "bidimensional"

- Some (simplified) preliminaries:
  - h-clique-sum of two graphs G_1 and G_2:
    choose cliques $K_1 \subseteq G_1$ and $K_2 \subseteq G_2$ with $|V(K_1)| = |V(K_2)| = h$,
    identify them, and possibly remove some edges of that clique.
  - Apex in an embedded graph:
    add a vertex with any neighbors in the embedded graph.
  - Vortex of depth h in an embedded graph:
    paste a graph of pathwidth at most h in a face of the embedding.

- Structure Theorem [Robertson and Seymour]:

  Fix a graph H.

Structure of H-minor-free graphs: roughly "bidimensional"

- Some (simplified) preliminaries:
  - h-clique-sum of two graphs G_1 and G_2: choose cliques $K_1 \subseteq G_1$ and $K_2 \subseteq G_2$ with $|V(K_1)| = |V(K_2)| = h$, identify them, and possibly remove some edges of that clique.
  - Apex in an embedded graph: add a vertex with any neighbors in the embedded graph.
  - Vortex of depth h in an embedded graph: paste a graph of pathwidth at most h in a face of the embedding.
- Structure Theorem [Robertson and Seymour]:

Fix a graph H. There exists a constant $h = f(\lvert V(H)\rvert)$ such that any H-minor-free graph G can be decomposed (in a tree-like way) into h-clique-sums from h-almost-embeddable graphs:

Structure of H-minor-free graphs: roughly "bidimensional"

- Some (simplified) preliminaries:
  - h-clique-sum of two graphs G_1 and G_2:
    choose cliques $K_1 \subseteq G_1$ and $K_2 \subseteq G_2$ with $|V(K_1)| = |V(K_2)| = h$, identify them, and possibly remove some edges of that clique.
  - Apex in an embedded graph:
    add a vertex with any neighbors in the embedded graph.
  - Vortex of depth h in an embedded graph:
    paste a graph of pathwidth at most h in a face of the embedding.

- Structure Theorem [Robertson and Seymour]:

  Fix a graph H. There exists a constant $h = f(|V(H)|)$ such that any H-minor-free graph G can be decomposed (in a tree-like way) into h-clique-sums from h-almost-embeddable graphs:

  obtained from graphs of genus at most h by adding at most h apices and at most h vortices of depth at most h.

1 Introduction, part II
- Treewidth
- Dynamic programming on tree decompositions
- Structure of H-minor-free graphs
- Some algorithmic issues
- A few words on other containment relations

2 Bidimensionality
- Some ingredients
- An illustrative example
- Meta-algorithms
- Further extensions

Some algorithmic issues of graph minors

★ Minor testing

Theorem (Robertson and Seymour)

Fix a graph H. Given an n-vertex graph G, one can test whether $H \preceq_m G$ in time $O(n^3)$.

Some algorithmic issues of graph minors

★ Minor testing

Theorem (Robertson and Seymour)

Fix a graph H. Given an n-vertex graph G, one can test whether $H \preceq_m G$ in time $O(n^3)$.

- Improved algorithm running in time $O(n^2)$. [Kawarabayashi, Kobayashi, Reed. 2012]

Some algorithmic issues of graph minors

★ Minor testing

Theorem (Robertson and Seymour)

Fix a graph H. Given an n-vertex graph G, one can test whether $H \preceq_m G$ in time $O(n^3)$.

- Improved algorithm running in time $O(n^2)$. [Kawarabayashi, Kobayashi, Reed. 2012]
- Even faster algorithms have been claimed... [Reed ??]

Some algorithmic issues of graph minors

★ Minor testing

Theorem (Robertson and Seymour)

Fix a graph H. Given an n-vertex graph G, one can test whether $H \preceq_m G$ in time $O(n^3)$.

- Improved algorithm running in time $O(n^2)$. [Kawarabayashi, Kobayashi, Reed. 2012]
- Even faster algorithms have been claimed... [Reed ??]

★ Membership in a minor-closed graph class

Theorem (Robertson and Seymour)

Let $\mathcal{G}$ be a minor-closed graph class. Given an n-vertex graph G, one can test whether $G \in \mathcal{G}$ in time $O(n^2)$.

Some algorithmic issues of graph minors

★ Minor testing

Theorem (Robertson and Seymour)

Fix a graph H. Given an n-vertex graph G, one can test whether $H \preceq_m G$ in time $O(n^3)$.

- Improved algorithm running in time $O(n^2)$. [Kawarabayashi, Kobayashi, Reed. 2012]
- Even faster algorithms have been claimed... [Reed ??]

★ Membership in a minor-closed graph class

Theorem (Robertson and Seymour)

Let $\mathcal{G}$ be a minor-closed graph class. Given an n-vertex graph G, one can test whether $G \in \mathcal{G}$ in time $O(n^2)$.

Proof. By the Graph Minors Theorem, there exists a finite list of minimal excluded minors $\{F_1, \ldots, F_\ell\}$ for the class $\mathcal{G}$. Then we can do minor testing and check whether $F_i \preceq_m G$ in time $O(n^2)$ for each $1 \leqslant i \leqslant \ell$.

Some algorithmic issues of graph minors (II)

- Let us recall the Structure Theorem of H-minor-free graphs:

  Fix a graph H. There exists a constant $h = f(|V(H)|)$ such that any H-minor-free graph G can be decomposed (in a tree-like way) into h-clique-sums from h-almost-embeddable graphs.

Some algorithmic issues of graph minors (II)

- Let us recall the Structure Theorem of H-minor-free graphs:

  Fix a graph H. There exists a constant $h = f(|V(H)|)$ such that any H-minor-free graph G can be decomposed (in a tree-like way) into h-clique-sums from h-almost-embeddable graphs.

- Using this tree-like structure, a number of hard optimization problems can be solved efficiently in H-minor-free graphs using DP.

Some algorithmic issues of graph minors (II)

- Let us recall the Structure Theorem of H-minor-free graphs:

  Fix a graph H. There exists a constant $h = f(|V(H)|)$ such that any H-minor-free graph G can be decomposed (in a tree-like way) into h-clique-sums from h-almost-embeddable graphs.

- Using this tree-like structure, a number of hard optimization problems can be solved efficiently in H-minor-free graphs using DP.

- But... what about the constant $h = f(|V(H)|)$??

Some algorithmic issues of graph minors (II)

- Let us recall the Structure Theorem of H-minor-free graphs:

  > Fix a graph H. There exists a constant $h = f(|V(H)|)$ such that any H-minor-free graph G can be decomposed (in a tree-like way) into h-clique-sums from h-almost-embeddable graphs.

- Using this tree-like structure, a number of hard optimization problems can be solved efficiently in H-minor-free graphs using DP.
- But... what about the constant $\boxed{h = f(|V(H)|)}$??
- Roughly the same constant appears in the Minor Testing algorithms.
- The proofs say "there exists a constant h such that..."

Some algorithmic issues of graph minors (II)

- Let us recall the Structure Theorem of H-minor-free graphs:

  > Fix a graph H. There exists a constant $h = f(|V(H)|)$ such that any H-minor-free graph G can be decomposed (in a tree-like way) into h-clique-sums from h-almost-embeddable graphs.

- Using this tree-like structure, a number of hard optimization problems can be solved efficiently in H-minor-free graphs using DP.
- But... what about the constant $h = f(|V(H)|)$??
- Roughly the same constant appears in the Minor Testing algorithms.
- The proofs say "there exists a constant h such that..."

★ Very recently, this constant has been made explicit and "reasonable"!

[Geelen, Huynh, and Richter. 2013]

[Mazoit. 2013]

Next subsection is...

1. Introduction, part II
   - Treewidth
   - Dynamic programming on tree decompositions
   - Structure of H-minor-free graphs
   - Some algorithmic issues
   - A few words on other containment relations

2. Bidimensionality
   - Some ingredients
   - An illustrative example
   - Meta-algorithms
   - Further extensions

★ Minor: $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

★ Minor: $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ Minor: $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ Contraction minor: $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

★ Minor: $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ Contraction minor: $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation?

★ Minor: $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ Contraction minor: $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO!

★ Minor: $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ Contraction minor: $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO!
2. Contraction Minor Testing is FPT when param. by $|V(H)|$?

★ Minor: $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. MINOR TESTING is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ Contraction minor: $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO!
2. CONTRACTION MINOR TESTING is FPT when param. by $|V(H)|$? NO! NP-hard already for $|V(H)| \leqslant 4$. [Brouwer and Veldman. 1987]

★ Minor: $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. MINOR TESTING is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ Contraction minor: $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO!
2. CONTRACTION MINOR TESTING is FPT when param. by $|V(H)|$? NO! NP-hard already for $|V(H)| \leqslant 4$. [Brouwer and Veldman. 1987]
3. Nice structure?

★ Minor: $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ Contraction minor: $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO!
2. Contraction Minor Testing is FPT when param. by $|V(H)|$?
   NO! NP-hard already for $|V(H)| \leqslant 4$. [Brouwer and Veldman. 1987]
3. Nice structure? Not really: They contain cliques, chordal graphs...

★ Minor: $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ Contraction minor: $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO!
2. Contraction Minor Testing is FPT when param. by $|V(H)|$?
   NO! NP-hard already for $|V(H)| \leqslant 4$. [Brouwer and Veldman. 1987]
3. Nice structure? Not really: They contain cliques, chordal graphs...

★ Topological minor: $H \preceq_{tp} G$ if H can be obtained from a subgraph of G by contracting edges with at least one endpoint of degree $\leqslant 2$.

★ **Minor:** $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. MINOR TESTING is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ **Contraction minor:** $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO!
2. CONTRACTION MINOR TESTING is FPT when param. by $|V(H)|$? NO! NP-hard already for $|V(H)| \leqslant 4$. [Brouwer and Veldman. 1987]
3. Nice structure? Not really: They contain cliques, chordal graphs...

★ **Topological minor:** $H \preceq_{tp} G$ if H can be obtained from a subgraph of G by contracting edges with at least one endpoint of degree $\leqslant 2$.

1. Graphs are WQO w.r.t. the topological minor relation?

★ **Minor:** $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. MINOR TESTING is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ **Contraction minor:** $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO!
2. CONTRACTION MINOR TESTING is FPT when param. by $|V(H)|$?
   NO! NP-hard already for $|V(H)| \leqslant 4$. [Brouwer and Veldman. 1987]
3. Nice structure? Not really: They contain cliques, chordal graphs...

★ **Topological minor:** $H \preceq_{tp} G$ if H can be obtained from a subgraph of G by contracting edges with at least one endpoint of degree $\leqslant 2$.

1. Graphs are WQO w.r.t. the topological minor relation? NO!

★ Minor: $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ Contraction minor: $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO!
2. Contraction Minor Testing is FPT when param. by $|V(H)|$? NO! NP-hard already for $|V(H)| \leqslant 4$. [Brouwer and Veldman. 1987]
3. Nice structure? Not really: They contain cliques, chordal graphs...

★ Topological minor: $H \preceq_{tp} G$ if H can be obtained from a subgraph of G by contracting edges with at least one endpoint of degree $\leqslant 2$.

1. Graphs are WQO w.r.t. the topological minor relation? NO!
2. Topological Minor Testing is FPT when param. by $|V(H)|$?

★ Minor: $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ Contraction minor: $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO!
2. Contraction Minor Testing is FPT when param. by $|V(H)|$? NO! NP-hard already for $|V(H)| \leqslant 4$. [Brouwer and Veldman. 1987]
3. Nice structure? Not really: They contain cliques, chordal graphs...

★ Topological minor: $H \preceq_{tp} G$ if H can be obtained from a subgraph of G by contracting edges with at least one endpoint of degree $\leqslant 2$.

1. Graphs are WQO w.r.t. the topological minor relation? NO!
2. Topological Minor Testing is FPT when param. by $|V(H)|$? YES! [Grohe, Kawarabayashi, Marx, Wollan. 2011]

★ **Minor:** $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ **Contraction minor:** $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO!
2. Contraction Minor Testing is FPT when param. by $|V(H)|$?
   NO! NP-hard already for $|V(H)| \leqslant 4$. [Brouwer and Veldman. 1987]
3. Nice structure? Not really: They contain cliques, chordal graphs...

★ **Topological minor:** $H \preceq_{tp} G$ if H can be obtained from a subgraph of G by contracting edges with at least one endpoint of degree $\leqslant 2$.

1. Graphs are WQO w.r.t. the topological minor relation? NO!
2. Topological Minor Testing is FPT when param. by $|V(H)|$?
   YES! [Grohe, Kawarabayashi, Marx, Wollan. 2011]
3. Nice structure?

★ Minor: $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

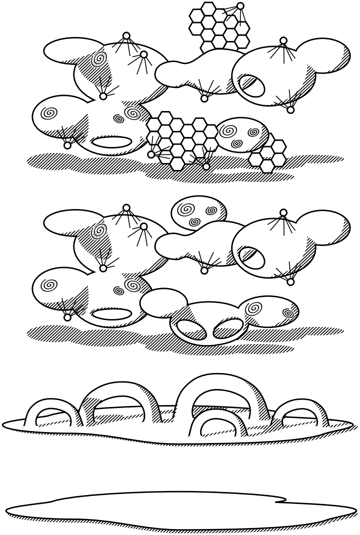
★ Contraction minor: $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO!
2. Contraction Minor Testing is FPT when param. by $|V(H)|$?
   NO! NP-hard already for $|V(H)| \leqslant 4$. [Brouwer and Veldman. 1987]
3. Nice structure? Not really: They contain cliques, chordal graphs...

★ Topological minor: $H \preceq_{tp} G$ if H can be obtained from a subgraph of G by contracting edges with at least one endpoint of degree $\leqslant 2$.

1. Graphs are WQO w.r.t. the topological minor relation? NO!
2. Topological Minor Testing is FPT when param. by $|V(H)|$?
   YES! [Grohe, Kawarabayashi, Marx, Wollan. 2011]
3. Nice structure? YES! [Grohe and Marx. 2012]

Structure of sparse graphs - a nice picture by Felix Reidl



H-topological-minor-free — treewidth-bounding

$\cup$

H-minor-free — bidimensional, separation property

$\cup$

bounded genus — quasi-compact

$\cup$

planar — “distance-property”

Next section is...

1 Introduction, part II
- Treewidth
- Dynamic programming on tree decompositions
- Structure of H-minor-free graphs
- Some algorithmic issues
- A few words on other containment relations

2 Bidimensionality
- Some ingredients
- An illustrative example
- Meta-algorithms
- Further extensions

Next subsection is...

1. Introduction, part II
   - Treewidth
   - Dynamic programming on tree decompositions
   - Structure of *H*-minor-free graphs
   - Some algorithmic issues
   - A few words on other containment relations

2. Bidimensionality
   - Some ingredients
   - An illustrative example
   - Meta-algorithms
   - Further extensions

A few representative problems

VERTEX COVER
Input: A graph $G = (V, E)$ and a positive integers k.
Parameter: k.
Question: Does there exist a subset $C \subseteq V$ of size at most k such that $G[V \setminus C]$ is an independent set?

A few representative problems

VERTEX COVER
Input: A graph $G = (V, E)$ and a positive integers k.
Parameter: k.
Question: Does there exist a subset $C \subseteq V$ of size at most k such that $G[V \setminus C]$ is an independent set?

LONG PATH
Input: A graph $G = (V, E)$ and a positive integers k.
Parameter: k.
Question: Does there exist a path P in G of length at least k?

A few representative problems (II)

FEEDBACK VERTEX SET
Input: A graph $G = (V, E)$ and a positive integers k.
Parameter: k.
Question: Does there exist a subset $F \subseteq V$ of size at most k such that for $G[V \setminus F]$ is a forest?

A few representative problems (II)

FEEDBACK VERTEX SET
Input: A graph $G = (V, E)$ and a positive integers k.
Parameter: k.
Question: Does there exist a subset $F \subseteq V$ of size at most k such that for $G[V \setminus F]$ is a forest?

DOMINATING SET
Input: A graph $G = (V, E)$ and a positive integers k.
Parameter: k.
Question: Does there exist a subset $D \subseteq V$ of size at most k such that for all $v \in V$, $N[v] \cap D \neq \emptyset$?

Minor closed parameters

- A graph class $\mathcal{G}$ is *minor (contraction) closed* if any minor (contraction) of a graph in $\mathcal{G}$ is also in $\mathcal{G}$.

Minor closed parameters

- A graph class $\mathcal{G}$ is *minor (contraction) closed* if any minor (contraction) of a graph in $\mathcal{G}$ is also in $\mathcal{G}$.

- A parameter P is any function mapping graphs to nonnegative integers.

Minor closed parameters

- A graph class $\mathcal{G}$ is *minor (contraction) closed* if any minor (contraction) of a graph in $\mathcal{G}$ is also in $\mathcal{G}$.
- A parameter P is any function mapping graphs to nonnegative integers.
- The *parameterized problem associated with* P asks, for some fixed k, whether for a given graph G, $P(G) \leqslant k$ (for minimization) or $P(G) \geqslant k$ (for maximization problem).

Minor closed parameters

- A graph class $\mathcal{G}$ is *minor (contraction) closed* if any minor (contraction) of a graph in $\mathcal{G}$ is also in $\mathcal{G}$.
- A parameter P is any function mapping graphs to nonnegative integers.
- The *parameterized problem associated with* P asks, for some fixed k, whether for a given graph G, $P(G) \leqslant k$ (for minimization) or $P(G) \geqslant k$ (for maximization problem).
- We say that a parameter P is *closed under taking of minors/contractions* (or, briefly, *minor/contraction closed*) if for every graph H, $H \preceq_m G$ / $H \preceq_{cm} G$ implies that $P(H) \leqslant P(G)$.

Examples of minor/contraction closed parameters

- Minor closed parameters:

  VERTEX COVER, FEEDBACK VERTEX SET, LONG PATH, TREEWIDTH, ...

Examples of minor/contraction closed parameters

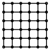
- Minor closed parameters:

  Vertex Cover, Feedback Vertex Set, Long Path, Treewidth, ...

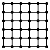
- Contraction closed parameters:

  Dominating Set, Connected Vertex Cover, r-Dominating Set, ...

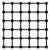
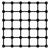
Grid Exclusion Theorem

- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid:  $_{\ell}$

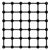
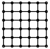
Grid Exclusion Theorem

- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid:  ℓ We have $\mathrm{tw}\left(H_{\ell,\ell}\right) = \ell$.

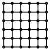
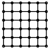
Grid Exclusion Theorem

- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid:  $_\ell$ We have $\mathrm{tw}\,(H_{\ell,\ell}) = \ell$.
- As TREEWIDTH is minor closed, if  $_\ell \preceq_m G$, then $\mathrm{tw}(G) \geqslant \mathrm{tw}(H_{\ell,\ell}) = \ell$.

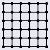
Grid Exclusion Theorem

- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid: $_\ell$ We have $\text{tw}\,(H_{\ell,\ell}) = \ell$.
- As TREEWIDTH is minor closed, if $_\ell \preceq_m G$, then $\text{tw}(G) \geqslant \text{tw}(H_{\ell,\ell}) = \ell$.
  Does the reverse implication hold?

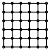
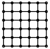
Grid Exclusion Theorem

- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid: $_\ell$ We have $\text{tw}(H_{\ell,\ell}) = \ell$.
- As TREEWIDTH is minor closed, if $_\ell \preceq_m G$, then $\text{tw}(G) \geqslant \text{tw}(H_{\ell,\ell}) = \ell$.
  Does the reverse implication hold?

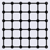
Theorem (Robertson and Seymour. 1986)

For every integer $\ell > 0$, there is an integer $c(\ell)$ such that every graph of treewidth $\geqslant c(\ell)$ contains $_\ell$ *as a minor.*

Grid Exclusion Theorem

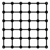
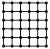
- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid:  $_\ell$ We have $\text{tw}(H_{\ell,\ell}) = \ell$.
- As TREEWIDTH is minor closed, if  $_\ell \preceq_m G$, then $\text{tw}(G) \geqslant \text{tw}(H_{\ell,\ell}) = \ell$.
  Does the reverse implication hold?

Theorem (Robertson and Seymour. 1986)

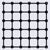
For every integer $\ell > 0$, there is an integer $c(\ell)$ such that every graph of treewidth $\geqslant c(\ell)$ contains  $_\ell$ as a minor.

- Smallest possible function $c(\ell)$?

Grid Exclusion Theorem

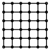
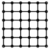
- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid:  $_\ell$ We have $\mathrm{tw}\,(H_{\ell,\ell}) = \ell$.
- As TREEWIDTH is minor closed, if  $_\ell \preceq_m G$, then
  $\mathrm{tw}(G) \geqslant \mathrm{tw}(H_{\ell,\ell}) = \ell$.
  Does the reverse implication hold?

Theorem (Robertson and Seymour. 1986)

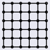
For every integer $\ell > 0$, there is an integer $c(\ell)$ such that every graph of treewidth $\geqslant c(\ell)$ contains  $_\ell$ *as a minor.*

- Smallest possible function $c(\ell)$? $\Omega(\ell^2 \log \ell) \leqslant c(\ell) \leqslant 20^{2^{\ell^5}}$

Grid Exclusion Theorem

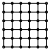
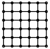
- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid:  $_\ell$ We have $\mathrm{tw}(H_{\ell,\ell}) = \ell$.
- As TREEWIDTH is minor closed, if  $_\ell \preceq_m G$, then
  $\mathrm{tw}(G) \geqslant \mathrm{tw}(H_{\ell,\ell}) = \ell$.
  Does the reverse implication hold?

Theorem (Robertson and Seymour. 1986)

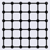
For every integer $\ell > 0$, there is an integer $c(\ell)$ such that every graph of treewidth $\geqslant c(\ell)$ contains  $_\ell$ *as a minor.*

- Smallest possible function $c(\ell)$? $\Omega(\ell^2 \log \ell) \leqslant c(\ell) \leqslant 20^{2^{\ell^5}}$
- Some improvement: $c(\ell) = 2^{O(\ell \log \ell)}$. [Leaf and Seymour. 2012]

Grid Exclusion Theorem

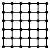
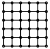
- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid:  $_\ell$ We have $\mathrm{tw}\,(H_{\ell,\ell}) = \ell$.
- As TREEWIDTH is minor closed, if  $_\ell \preceq_m G$, then $\mathrm{tw}(G) \geqslant \mathrm{tw}(H_{\ell,\ell}) = \ell$.
  Does the reverse implication hold?

Theorem (Robertson and Seymour. 1986)

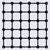
For every integer $\ell > 0$, there is an integer $c(\ell)$ such that every graph of treewidth $\geqslant c(\ell)$ contains  $_\ell$ *as a minor.*

- Smallest possible function $c(\ell)$? $\Omega(\ell^2 \log \ell) \leqslant c(\ell) \leqslant 20^{2\ell^5}$
- Some improvement: $c(\ell) = 2^{O(\ell \log \ell)}$. [Leaf and Seymour. 2012]
- Recent breakthrough: $c(\ell) = \mathrm{poly}(\ell)$. [Chekuri and Chuzhoy. 2013]

Grid Exclusion Theorem

- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid:  $_\ell$ We have $\mathrm{tw}\,(H_{\ell,\ell}) = \ell$.
- As Treewidth is minor closed, if  $_\ell \preceq_m G$, then $\mathrm{tw}(G) \geqslant \mathrm{tw}(H_{\ell,\ell}) = \ell$.
  Does the reverse implication hold?

Theorem (Robertson and Seymour. 1986)

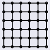
For every integer $\ell > 0$, there is an integer $c(\ell)$ such that every graph of treewidth $\geqslant c(\ell)$ contains  $_\ell$ *as a minor.*

- Smallest possible function $c(\ell)$? $\Omega(\ell^2 \log \ell) \leqslant c(\ell) \leqslant 20^{2\ell^5}$
- Some improvement: $c(\ell) = 2^{O(\ell \log \ell)}$. [Leaf and Seymour. 2012]
- Recent breakthrough: $c(\ell) = \mathrm{poly}(\ell)$. [Chekuri and Chuzhoy. 2013]

Important message grid-minors are the certificate of large treewidth.

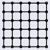
Grid Exclusion Theorems on sparse graphs

Theorem (Robertson, Seymour, Thomas. 1994)

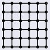
Every planar graph of treewidth $\geqslant 6 \cdot \ell$ *contains*  $_\ell$ *as a minor.*

Grid Exclusion Theorems on sparse graphs

Theorem (Robertson, Seymour, Thomas. 1994)

Every planar graph of treewidth $\geqslant 6 \cdot \ell$ contains  $_\ell$ *as a minor.*

Theorem (Demaine, Fomin, Hajiaghayi, Thilikos. 2005)

For every fixed g, there is a constant c_g such that every graph of genus g and of treewidth $\geqslant c_g \cdot \ell$ contains  $_\ell$ *as a minor.*

Grid Exclusion Theorems on sparse graphs

Theorem (Robertson, Seymour, Thomas. 1994)

Every planar graph of treewidth $\geqslant 6 \cdot \ell$ contains $\boxplus_\ell$ as a minor.

Theorem (Demaine, Fomin, Hajiaghayi, Thilikos. 2005)

For every fixed g, there is a constant c_g such that every graph of genus g and of treewidth $\geqslant c_g \cdot \ell$ contains $\boxplus_\ell$ as a minor.

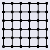
Theorem (Demaine and Hajiaghayi. 2008)

For every fixed graph H, there is a constant c_H such that every H-minor-free graph of treewidth $\geqslant c_H \cdot \ell$ contains $\boxplus_\ell$ as a minor.

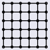
Best constant in the above theorem is by [Kawarabayashi and Kobayashi. 2012]

Grid Exclusion Theorems on sparse graphs

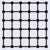
Theorem (Robertson, Seymour, Thomas. 1994)

Every planar graph of treewidth $\geqslant 6 \cdot \ell$ *contains*  $_\ell$ *as a minor.*

Theorem (Demaine, Fomin, Hajiaghayi, Thilikos. 2005)

For every fixed g*, there is a constant* c_g *such that every graph of genus* g *and of treewidth* $\geqslant c_g \cdot \ell$ *contains*  $_\ell$ *as a minor.*

Theorem (Demaine and Hajiaghayi. 2008)

For every fixed graph H*, there is a constant* c_H *such that every* H*-minor-free graph of treewidth* $\geqslant c_H \cdot \ell$ *contains*  $_\ell$ *as a minor.*

Best constant in the above theorem is by [Kawarabayashi and Kobayashi. 2012]

In sparse graphs: linear dependency between treewidth and grid-minors

How to use Grid Theorems algorithmically?

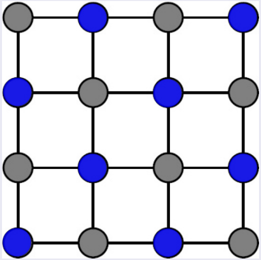
Next subsection is...

1. Introduction, part II
   - Treewidth
   - Dynamic programming on tree decompositions
   - Structure of H-minor-free graphs
   - Some algorithmic issues
   - A few words on other containment relations

2. Bidimensionality
   - Some ingredients
   - An illustrative example
   - Meta-algorithms
   - Further extensions

Example: FPT algorithm for Planar Vertex Cover

A vertex cover C of a graph G, $\mathbf{vc}(G)$, is a set of vertices such that every edge of G has at least one endpoint in C.



Example: FPT algorithm for Planar Vertex Cover

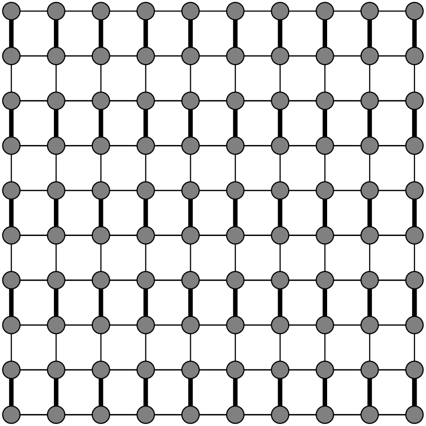
INPUT: Planar graph G on n vertices, and an integer k.

OUTPUT: Either a vertex cover of G of size $\leqslant k$, or a proof that G has no such a vertex cover.

RUNNING TIME: $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

Objective: subexponential FPT algorithm for PLANAR VERTEX COVER.

Example: FPT algorithm for Planar Vertex Cover



$\mathbf{vc}(H_{\ell,\ell}) \geqslant \frac{\ell^2}{2}$

Example: FPT algorithm for Planar Vertex Cover

Let G be a planar graph of treewidth $\geqslant 6 \cdot \ell$

Example: FPT algorithm for Planar Vertex Cover

Let G be a planar graph of treewidth $\geqslant 6 \cdot \ell$ $\Longrightarrow$ G contains the $(\ell \times \ell)$-grid $H_{\ell,\ell}$ as a minor

Example: FPT algorithm for Planar Vertex Cover

Let G be a planar graph of treewidth $\geqslant 6 \cdot \ell$ $\Longrightarrow$ G contains the $(\ell \times \ell)$-grid $H_{\ell,\ell}$ as a minor

- The size of any vertex cover of $H_{\ell,\ell}$ is at least $\ell^2/2$.
- Recall that VERTEX COVER is a minor closed parameter.
- Since $H_{\ell,\ell} \preceq_m G$, it holds that $\mathbf{vc}(G) \geqslant \mathbf{vc}(H_{\ell,\ell}) \geqslant \ell^2/2$.

We are already very close to an algorithm...

Recall:

- k is the parameter of the problem.
- We have that $\mathrm{tw}(G) = 6 \cdot \ell$ and ℓ is the size of a grid-minor of G.
- Therefore, $\mathbf{vc}(G) \geqslant \ell^2/2$.

We are already very close to an algorithm...

Recall:

- k is the parameter of the problem.
- We have that $\mathrm{tw}(G) = 6 \cdot \ell$ and ℓ is the size of a grid-minor of G.
- Therefore, $\mathbf{vc}(G) \geqslant \ell^2/2$.

WIN/WIN approach:

- If $k < \ell^2/2$, we can safely answer "NO".

We are already very close to an algorithm...

Recall:

- k is the parameter of the problem.
- We have that $\mathrm{tw}(G) = 6 \cdot \ell$ and ℓ is the size of a grid-minor of G.
- Therefore, $\mathbf{vc}(G) \geqslant \ell^2/2$.

WIN/WIN approach:

- If $k < \ell^2/2$, we can safely answer "NO".
- If $k \geqslant \ell^2/2$, then $\mathrm{tw}(G) = O(\ell) = O(\sqrt{k})$,

We are already very close to an algorithm...

Recall:

- k is the parameter of the problem.
- We have that $\mathrm{tw}(G) = 6 \cdot \ell$ and ℓ is the size of a grid-minor of G.
- Therefore, $\mathbf{vc}(G) \geqslant \ell^2/2$.

WIN/WIN approach:

- If $k < \ell^2/2$, we can safely answer "NO".
- If $k \geqslant \ell^2/2$, then $\mathrm{tw}(G) = O(\ell) = O(\sqrt{k})$, and we can solve the problem by standard DP in time $2^{O(\mathrm{tw}(G))} \cdot n^{O(1)}$

We are already very close to an algorithm...

Recall:

- k is the parameter of the problem.
- We have that $\mathrm{tw}(G) = 6 \cdot \ell$ and ℓ is the size of a grid-minor of G.
- Therefore, $\mathbf{vc}(G) \geqslant \ell^2/2$.

WIN/WIN approach:

- If $k < \ell^2/2$, we can safely answer "NO".
- If $k \geqslant \ell^2/2$, then $\mathrm{tw}(G) = O(\ell) = O(\sqrt{k})$, and we can solve the problem by standard DP in time $2^{O(\mathrm{tw}(G))} \cdot n^{O(1)} = 2^{O(\sqrt{k})} \cdot n^{O(1)}$.

We are already very close to an algorithm...

Recall:

- k is the parameter of the problem.
- We have that $\mathrm{tw}(G) = 6 \cdot \ell$ and ℓ is the size of a grid-minor of G.
- Therefore, $\mathbf{vc}(G) \geqslant \ell^2/2$.

WIN/WIN approach:

- If $k < \ell^2/2$, we can safely answer "NO".
- If $k \geqslant \ell^2/2$, then $\mathrm{tw}(G) = O(\ell) = O(\sqrt{k})$, and we can solve the problem by standard DP in time $2^{O(\mathrm{tw}(G))} \cdot n^{O(1)} = 2^{O(\sqrt{k})} \cdot n^{O(1)}$.

This gives a subexponential FPT algorithm!

Was Vertex Cover really just an example...?

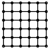
What is so special in VERTEX COVER?

Where did we use planarity?

Was Vertex Cover really just an example...?

What is so special in VERTEX COVER?

★ Nothing special! It is just a minor bidimensional parameter:

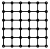
minor-closed $+$ **vc**( $_k$) $= \Omega(k^2)$.

Where did we use planarity?

Was Vertex Cover really just an example...?

What is so special in VERTEX COVER?

★ Nothing special! It is just a minor bidimensional parameter:

minor-closed $+$ **vc**( $_k$) $= \Omega(k^2)$.

Where did we use planarity?

★ Only the linear Grid Theorem!

Arguments go through up to H-minor-free graphs.

Next subsection is...

1. Introduction, part II
   - Treewidth
   - Dynamic programming on tree decompositions
   - Structure of *H*-minor-free graphs
   - Some algorithmic issues
   - A few words on other containment relations

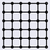
2. Bidimensionality
   - Some ingredients
   - An illustrative example
   - Meta-algorithms
   - Further extensions

Minor Bidimensionality:

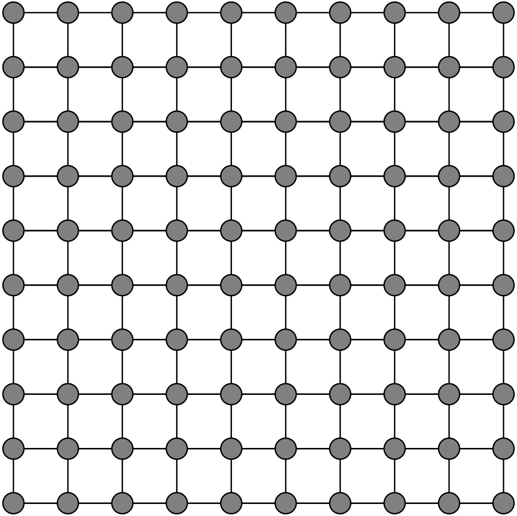
[Demaine, Fomin, Hajiaghayi, Thilikos. 2005]

Definition

A parameter **p** is *minor bidimensional* if

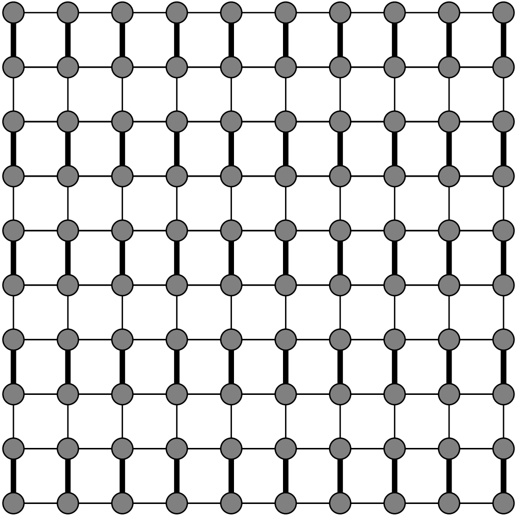
1. **p** is closed under taking of minors (minor closed), and
2. **p** ( $_k$) $= \Omega(k^2)$.

Vertex Cover of a Grid



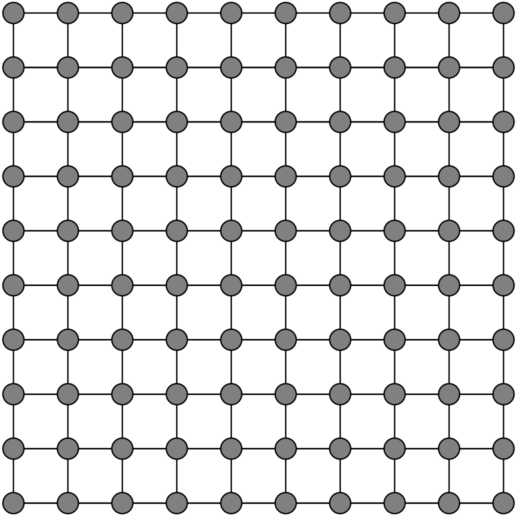
$H_{r,r}$ for $r = 10$

Vertex Cover of a Grid

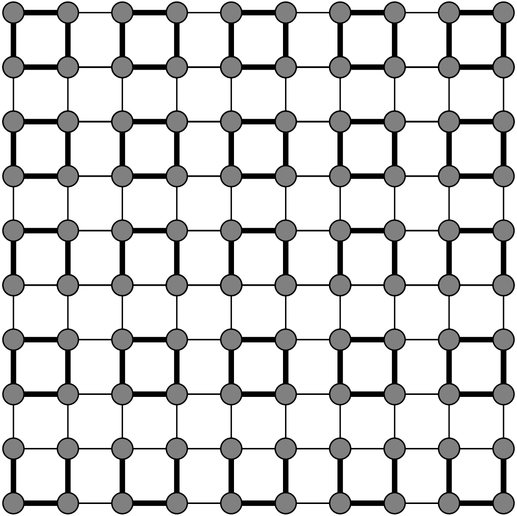


$\mathbf{vc}(H_{r,r}) \geqslant r^2/2$

Feedback Vertex Set of a Grid



Feedback Vertex Set of a Grid



$\mathbf{fvs}(H_{r,r}) \geqslant r^2/4$

How to obtain subexponential algorithms for BP?

- First we must restrict ourselves to special graph classes, like planar or H-minor-free graphs.

How to obtain subexponential algorithms for BP?

- First we must restrict ourselves to special graph classes, like planar or H-minor-free graphs.
  - Show that if the graph has large treewidth $(> c\sqrt{k})$ then it has a $(\sqrt{k} \times \sqrt{k})$-grid as a minor, and hence the answer to the problem is YES (or NO) immediately.

How to obtain subexponential algorithms for BP?

- First we must restrict ourselves to special graph classes, like planar or H-minor-free graphs.
  - Show that if the graph has large treewidth ($> c\sqrt{k}$) then it has a ($\sqrt{k} \times \sqrt{k}$)-grid as a minor, and hence the answer to the problem is YES (or NO) immediately.
  - Otherwise, the treewidth is bounded by $c\sqrt{k}$, and hence we can use a dynamic programming (DP) algorithm on graphs of bounded treewidth.

How to obtain subexponential algorithms for BP?

- First we must restrict ourselves to special graph classes, like planar or H-minor-free graphs.
  - Show that if the graph has large treewidth $(> c\sqrt{k})$ then it has a $(\sqrt{k} \times \sqrt{k})$-grid as a minor, and hence the answer to the problem is YES (or NO) immediately.
  - Otherwise, the treewidth is bounded by $c\sqrt{k}$, and hence we can use a dynamic programming (DP) algorithm on graphs of bounded treewidth.
- If we have a DP algorithm for bounded treewidth running in time c^t or t^t, then it implies $2^{O(\sqrt{k})}$ or $2^{O(\sqrt{k}\log k)}$ algorithm.

Piecing everything together

Theorem

Let G be an H-minor-free graph, and let $\mathbf{p}$ be a minor bidimensional graph parameter computable in time $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$.
Then deciding "$\mathbf{p}(G) = k$" can be done in time $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

Piecing everything together

Theorem

Let G be an H-minor-free graph, and let $\mathbf{p}$ be a minor bidimensional graph parameter computable in time $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$.
Then deciding "$\mathbf{p}(G) = k$" can be done in time $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

1. Compute (or approximate) $\mathbf{tw}(G)$.
2. If $\mathbf{tw}(G) = \Omega(\sqrt{k})$, then answer NO.
3. Otherwise $\mathbf{tw}(G) = O(\sqrt{k})$, and we solve the problem by DP.

Piecing everything together

Theorem

Let G be an H-minor-free graph, and let $\mathbf{p}$ be a minor bidimensional graph parameter computable in time $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$.
Then deciding "$\mathbf{p}(G) = k$" can be done in time $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

1. Compute (or approximate) $\mathbf{tw}(G)$.
   We can use a fast FPT algorithm or a constant-factor approx.
2. If $\mathbf{tw}(G) = \Omega(\sqrt{k})$, then answer NO.
3. Otherwise $\mathbf{tw}(G) = O(\sqrt{k})$, and we solve the problem by DP.

Piecing everything together

Theorem

Let G be an H-minor-free graph, and let $\mathbf{p}$ be a minor bidimensional graph parameter computable in time $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$.
Then deciding "$\mathbf{p}(G) = k$" can be done in time $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

1. Compute (or approximate) $\mathbf{tw}(G)$.
   We can use a fast FPT algorithm or a constant-factor approx.
2. If $\mathbf{tw}(G) = \Omega(\sqrt{k})$, then answer NO.
   This follows because of the linear Grid Exclusion Theorems.
3. Otherwise $\mathbf{tw}(G) = O(\sqrt{k})$, and we solve the problem by DP.

Piecing everything together

Theorem

Let G be an H-minor-free graph, and let $\mathbf{p}$ be a minor bidimensional graph parameter computable in time $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$.
Then deciding "$\mathbf{p}(G) = k$" can be done in time $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

1. Compute (or approximate) $\mathbf{tw}(G)$.
   We can use a fast FPT algorithm or a constant-factor approx.
2. If $\mathbf{tw}(G) = \Omega(\sqrt{k})$, then answer NO.
   This follows because of the linear Grid Exclusion Theorems.
3. Otherwise $\mathbf{tw}(G) = O(\sqrt{k})$, and we solve the problem by DP.
   Doing DP in time $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$ is a whole area of research:

Piecing everything together

Theorem

Let G be an H-minor-free graph, and let $\mathbf{p}$ be a minor bidimensional graph parameter computable in time $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$.
Then deciding "$\mathbf{p}(G) = k$" can be done in time $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

1. Compute (or approximate) $\mathbf{tw}(G)$.
   We can use a fast FPT algorithm or a constant-factor approx.
2. If $\mathbf{tw}(G) = \Omega(\sqrt{k})$, then answer NO.
   This follows because of the linear Grid Exclusion Theorems.
3. Otherwise $\mathbf{tw}(G) = O(\sqrt{k})$, and we solve the problem by DP.
   Doing DP in time $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$ is a whole area of research:
   - Exploiting Catalan structures on sparse graphs. [Dorn *et al.* 2005-2008]

[Rué, S., Thilikos. 2010]

Piecing everything together

Theorem

Let G be an H-minor-free graph, and let $\mathbf{p}$ be a minor bidimensional graph parameter computable in time $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$.
Then deciding "$\mathbf{p}(G) = k$" can be done in time $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

1. Compute (or approximate) $\mathbf{tw}(G)$.
   We can use a fast FPT algorithm or a constant-factor approx.
2. If $\mathbf{tw}(G) = \Omega(\sqrt{k})$, then answer NO.
   This follows because of the linear Grid Exclusion Theorems.
3. Otherwise $\mathbf{tw}(G) = O(\sqrt{k})$, and we solve the problem by DP.
   Doing DP in time $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$ is a whole area of research:
   - Exploiting Catalan structures on sparse graphs. [Dorn *et al.* 2005-2008] [Rué, S., Thilikos. 2010]
   - Randomized algorithms using Cut&Count. [Cygan *et al.* 2011]
   - Deterministic algorithms based on rank of matrices. [Boadlaender *et al.* 2012]
   - Deterministic algorithms based on matroids. [Fomin *et al.* 2013]

Minor Bidimensionality provides a meta-algorithm

- This result applies to all minor closed parameters:

  VERTEX COVER, FEEDBACK VERTEX SET, LONG PATH, CYCLE COVER, . . .

Minor Bidimensionality provides a meta-algorithm

- This result applies to all minor closed parameters:

  VERTEX COVER, FEEDBACK VERTEX SET, LONG PATH, CYCLE COVER, ...

- What about contraction closed parameters??

  DOMINATING SET, CONNECTED VERTEX COVER, r-DOMINATING SET, ...

Next subsection is...

1. Introduction, part II
   - Treewidth
   - Dynamic programming on tree decompositions
   - Structure of H-minor-free graphs
   - Some algorithmic issues
   - A few words on other containment relations

2. Bidimensionality
   - Some ingredients
   - An illustrative example
   - Meta-algorithms
   - Further extensions

Contraction bidimensionality

- Dominating Set is NOT minor closed, so we cannot use Grid Exclusion Theorems!!

Contraction bidimensionality

- DOMINATING SET is NOT minor closed, so we cannot use Grid Exclusion Theorems!!
- But it is contraction closed...

Contraction bidimensionality

- Dominating Set is NOT minor closed, so we cannot use Grid Exclusion Theorems!!
- But it is contraction closed...

Contraction Bidimensionality:

[Demaine, Fomin, Hajiaghayi, Thilikos. 2005]

Definition

A parameter **p** is *contraction bidimensional* if

1. **p** is closed under taking of contractions (contraction closed), and
2. for a "$(k \times k)$-grid-like graph" Γ, $\mathbf{p}(\Gamma) = \Omega(k^2)$.

Contraction bidimensionality

- DOMINATING SET is NOT minor closed,
  so we cannot use Grid Exclusion Theorems!!
- But it is contraction closed...

Contraction Bidimensionality:

[Demaine, Fomin, Hajiaghayi, Thilikos. 2005]

Definition

A parameter **p** is *contraction bidimensional* if

1. **p** is closed under taking of contractions (contraction closed), and
2. for a "$(k \times k)$-grid-like graph" Γ, $\mathbf{p}(\Gamma) = \Omega(k^2)$.

What is a $(k \times k)$-grid-like graph...?

Contraction bidimensionality: old setting

A "$(k \times k)$-grid-like graph" was different for each graph class:

Contraction bidimensionality: old setting

A "$(k \times k)$-grid-like graph" was different for each graph class:

★ For planar graphs this is a partially triangulated $(k \times k)$-grid.

[Demaine, Fomin, Hajiaghayi, Thilikos. 2006]

Contraction bidimensionality: old setting

A "$(k \times k)$-grid-like graph" was different for each graph class:

- ★ For planar graphs this is a partially triangulated $(k \times k)$-grid.

  [Demaine, Fomin, Hajiaghayi, Thilikos. 2006]

- ★ For graphs of Euler genus γ, this is a partially triangulated $(k \times k)$-grid with up to γ additional handles.

  [Demaine, Hajiaghayi, Thilikos. 2006]

Contraction bidimensionality: old setting

A "($k \times k$)-grid-like graph" was different for each graph class:

- ★ For planar graphs this is a partially triangulated ($k \times k$)-grid.

  [Demaine, Fomin, Hajiaghayi, Thilikos. 2006]

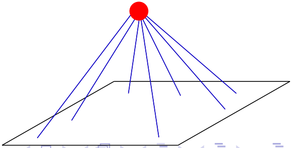
- ★ For graphs of Euler genus γ, this is a partially triangulated ($k \times k$)-grid with up to γ additional handles.

  [Demaine, Hajiaghayi, Thilikos. 2006]

- ★ For apex-minor-free graphs, this is a ($k \times k$)-augmented grid, i.e., partially triangulated grid augmented with additional edges such that each vertex is incident to $O(1)$ edges to non-boundary vertices of the grid.

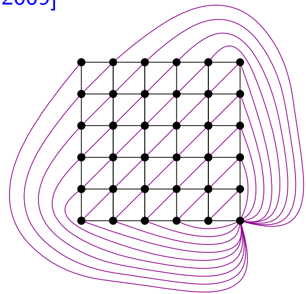
  [Demaine, Fomin, Hajiaghayi, Thilikos. 2005]

H is an *apex graph* if $\exists v \in V(H)$: $H - v$ is planar



Contraction bidimensionality: new definition

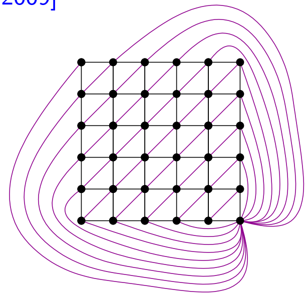
Finally, the right "($k \times k$)-grid-like graph" was found:
[Fomin, Golovach, Thilikos. 2009]



k

Contraction bidimensionality: new definition

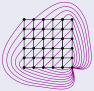
Finally, the right "$(k \times k)$-grid-like graph" was found:
[Fomin, Golovach, Thilikos. 2009]



k

Definition

A parameter **p** is *contraction bidimensional* if the following hold:

1. **p** is contraction closed, and
2. **p**( $_k$) $= \Omega(k^2)$.

Meta-algorithms for contraction bidimensional parameters

Theorem

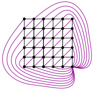
Let H be a fixed apex *graph, let G be an* H-minor free graph*, and let* $\mathbf{p}$ *be a* contraction bidimensional *parameter computable in* $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$. *Then deciding* $\mathbf{p}(G) = k$ *can be done in time* $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

Meta-algorithms for contraction bidimensional parameters

Theorem

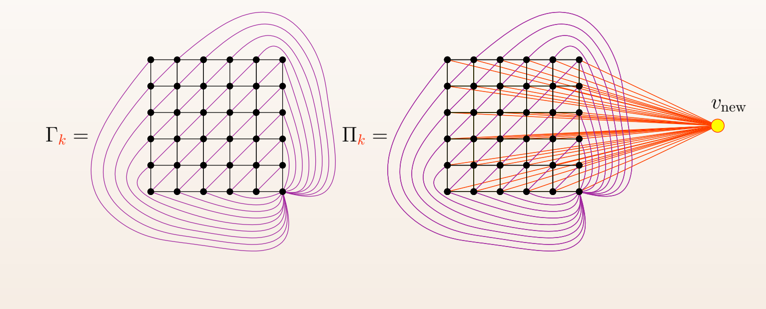
Let H be a fixed apex graph, let G be an H-minor free graph, and let $\mathbf{p}$ be a contraction bidimensional parameter computable in $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$. Then deciding $\mathbf{p}(G) = k$ can be done in time $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

As for minor bidimensionality, we need to prove that

- If $\mathbf{tw}(G) = \Omega(k)$ then G contains  $_k$ as a contraction.

Two important grid-like graphs

Two pattern graphs Γ_k and Π_k:



$\Pi_k = \Gamma_k +$ a new universal vertex v_{new}.

The "contraction-certificates" for large treewidth

Theorem (Fomin, Golovach, Thilikos. 2009)

For any integer $\ell > 0$, there is c_ℓ such that every connected graph of treewidth at least c_ℓ, contains K_ℓ, Γ_ℓ, or Π_ℓ as a contraction.

The "contraction-certificates" for large treewidth

Theorem (Fomin, Golovach, Thilikos. 2009)

For any integer $\ell > 0$, there is c_ℓ such that every connected graph of treewidth at least c_ℓ, contains K_ℓ, Γ_ℓ, or Π_ℓ as a contraction.

Theorem (Fomin, Golovach, Thilikos. 2009)

For every graph H, there is $c_H > 0$ such that every connected H-minor-free graph of treewidth at least $c_H \cdot \ell^2$ contains Γ_ℓ or Π_ℓ as a contraction.

The "contraction-certificates" for large treewidth

Theorem (Fomin, Golovach, Thilikos. 2009)

For any integer $\ell > 0$, there is c_ℓ such that every connected graph of treewidth at least c_ℓ, contains K_ℓ, Γ_ℓ, or Π_ℓ as a contraction.

Theorem (Fomin, Golovach, Thilikos. 2009)

For every graph H, there is $c_H > 0$ such that every connected H-minor-free graph of treewidth at least $c_H \cdot \ell^2$ contains Γ_ℓ or Π_ℓ as a contraction.

Theorem (Fomin, Golovach, Thilikos. 2009)

For every apex graph H, there is $c_H > 0$ such that every connected H-minor-free graph of treewidth at least $c_H \cdot \ell$ contains Γ_ℓ as a contraction.

Further applications of Bidimensionality

1. Bidimensionality + DP $\Rightarrow$ Subexponential FPT algorithms

   [Demaine, Fomin, Hajiaghayi, Thilikos. 2004-2005]
   [Fomin, Golovach, Thilikos. 2009]

Further applications of Bidimensionality

1. Bidimensionality + DP ⇒ Subexponential FPT algorithms

   [Demaine, Fomin, Hajiaghayi, Thilikos. 2004-2005]
   [Fomin, Golovach, Thilikos. 2009]

2. Bidimensionality + separation properties ⇒ (E)PTAS

   [Demaine and Hajiaghayi. 2005]
   [Fomin, Lokshtanov, Raman, Saurabh. 2011]

Further applications of Bidimensionality

1. Bidimensionality + DP ⇒ Subexponential FPT algorithms

   [Demaine, Fomin, Hajiaghayi, Thilikos. 2004-2005]
   [Fomin, Golovach, Thilikos. 2009]

2. Bidimensionality + separation properties ⇒ (E)PTAS

   [Demaine and Hajiaghayi. 2005]
   [Fomin, Lokshtanov, Raman, Saurabh. 2011]

3. Bidimensionality + separation properties ⇒ Kernelization

   [Fomin, Lokshtanov, Saurabh, Thilikos. 2009-2010]

Further applications of Bidimensionality

1. Bidimensionality + DP ⇒ Subexponential FPT algorithms

   [Demaine, Fomin, Hajiaghayi, Thilikos. 2004-2005]
   [Fomin, Golovach, Thilikos. 2009]

2. Bidimensionality + separation properties ⇒ (E)PTAS

   [Demaine and Hajiaghayi. 2005]
   [Fomin, Lokshtanov, Raman, Saurabh. 2011]

3. Bidimensionality + separation properties ⇒ Kernelization

   [Fomin, Lokshtanov, Saurabh, Thilikos. 2009-2010]

4. Bidimensionality + new Grid Theorems ⇒ Geometric graphs

   [Fomin, Lokshtanov, Saurabh. 2012]
   [Grigoriev, Koutsonas, Thilikos. 2013]

Gràcies!