Gridology

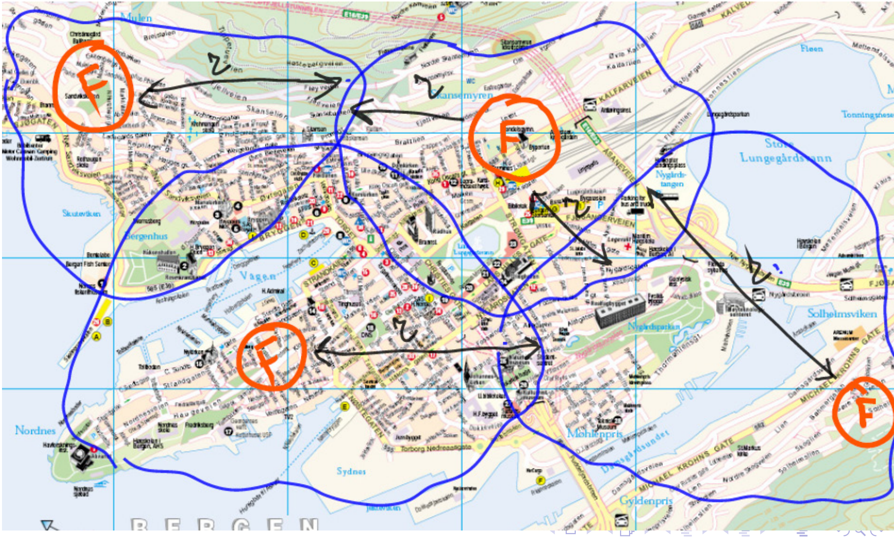
Fedor V. Fomin

Department of Informatics, University of Bergen, Norway

AGAPE 2009

Example: The city of Bergen

How to place k fire stations such that every building is within r city blocks from the nearest fire station?



How to place k fire stations?

- Some simplifications: Bergen is a planar graph and $r = 1$.

How to place k fire stations?

- Some simplifications: Bergen is a planar graph and $r = 1$.
- There is a linear kernel $O(k)$ for dominating set on planar graph, so $2^{O(k)}n^{O(1)}$ algorithm is possible

How to place k fire stations?

- Some simplifications: Bergen is a planar graph and $r = 1$.
- There is a linear kernel $O(k)$ for dominating set on planar graph, so $2^{O(k)}n^{O(1)}$ algorithm is possible
- We show how to get *subexponential* $2^{\sqrt{k}}n^{O(1)}$ algorithms.

How to place k fire stations?

- Some simplifications: Bergen is a planar graph and $r = 1$.
- There is a linear kernel $O(k)$ for dominating set on planar graph, so $2^{O(k)}n^{O(1)}$ algorithm is possible
- We show how to get *subexponential* $2^{\sqrt{k}}n^{O(1)}$ algorithms.
- The idea works even when Bergen has more complicated structure, like embedded on a surface of bounded genus, or excluding some fixed graph as a minor; it works for every fixed $r \geq 1$, and for many other problems

Outline of the tutorial

- Framework for parameterized algorithms: combinatorial bounds + dynamic programming
- Combinatorial bounds via Graph Minor theorems
  - Bidimensionality
- Dynamic programming which uses graph structure
  - Catalan structures

Graph Minors

The framework exploits the structure of graph classes that exclude some graph as a minor

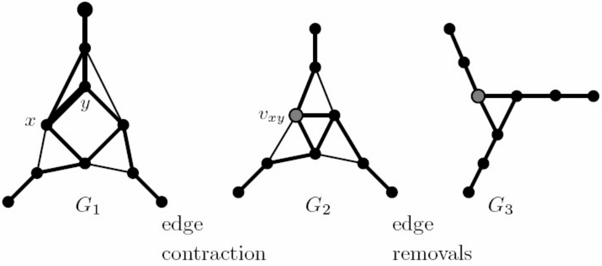
Minors and contractions

H is a contraction of G ($H \leq_c G$) if H occurs from G after applying a series of edge contractions.

H is a minor of G ($H \leq_m G$) if H is the contraction of some subgraph of G.

Notice: $\leq_m$ and $\leq_c$ are partial relations on graphs

Minors and contractions



$G_3 \preceq G_2 \preceq G_1$, $G_2 \preceq_c G_1$ but also $G_3 \not\preceq_c G_2$ and $G_3 \not\preceq_c G_1$

Minors and contractions

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

A graph G is *H-minor-free* when it does not contain H as a minor.

A graph class $\mathcal{G}$ is *H-minor-free* (or, excludes H as a minor) when all its members are H-minor-free.

Examples of H-minor-free classes

- Forests: K_3
- Outerplanar Graphs: $K_{2,3}$, K_4
- Planar Graphs: $K_{3,3}$, K_5
- Link-free Graphs: 7 graphs (X-Y transformations of K_6)
- Graphs of the projective plane: 103 graphs

Graph Minor theorem

Robertson & Seymour (1986–2004):

Theorem (Graphs Minor Theorem)

Graphs are well-quasi-ordered by the minor relation $\leq_m$.

- Consequence: every minor closed graph class $\mathcal{G}$ has a finite set of minimal excluded minors.

Graph Minor theorem

Graphs Minor Theorem is not used in our tutorial. However,we need tools created by Roberston-Seymour in order to prof this theorem.

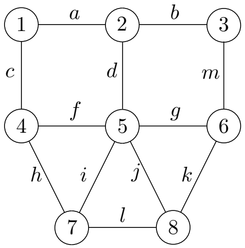
Main tool: Branch Decompositions

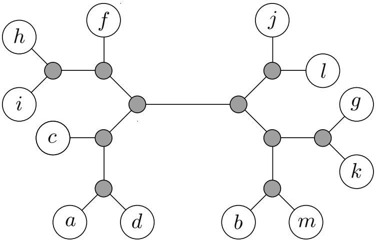
Definition

A branch decomposition of a graph $G = (V, E)$ is a tuple (T, μ) where

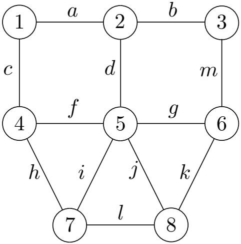
- T is a tree with degree 3 for all internal nodes.
- μ is a bijection between the leaves of T and $E(G)$.

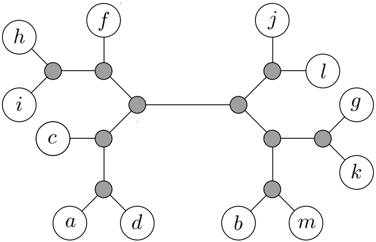
Example of Branch Decomposition



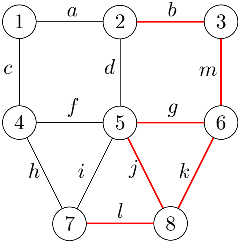


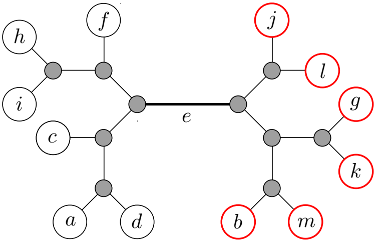
Edge $e \in T$ partitions the edge set of G in A_e and B_e



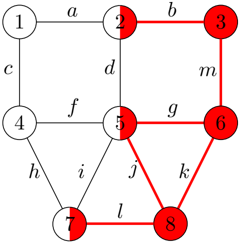


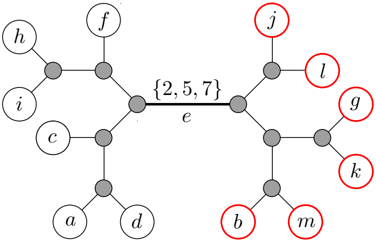
Edge $e \in T$ partitions the edge set of G in A_e and B_e





Middle set $\mathrm{mid}(e) = V(A_e) \cap V(B_e)$





Branchwidth

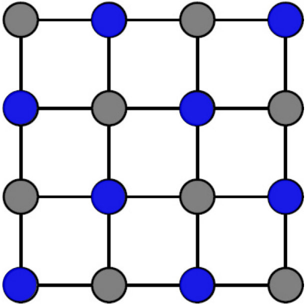
- The *width* of a branch decomposition is $\max_{e \in T} |\mathrm{mid}(e)|$.
- The *branchwidth* of a graph G is the minimum width over all branch decompositions of G.

Exercises

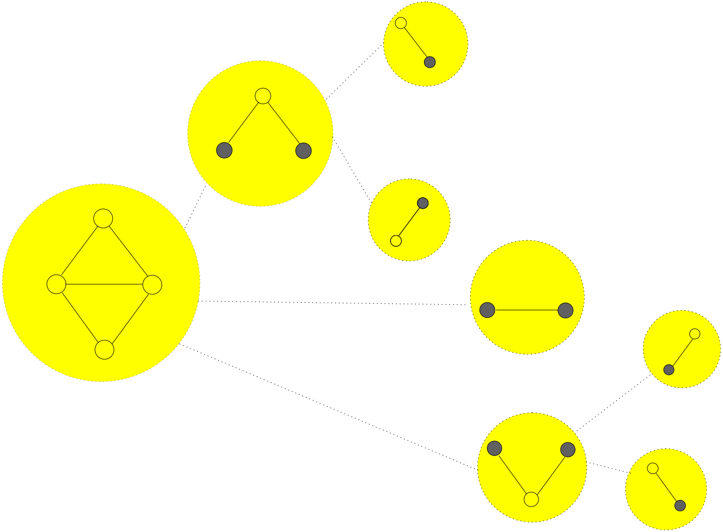
- What is the branchwidth of a tree?
- Complete graph on n vertices?
- $(\ell \times \ell)$-grid?

Vertex Cover

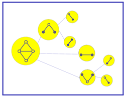
A *vertex cover* C of a graph G, $vc(G)$, is a set of vertices such that every edge of G has at least one endpoint in C.



Dynamic programming: Vertex Cover



Dynamic programming: Vertex Cover



Main idea—dynamic programming.

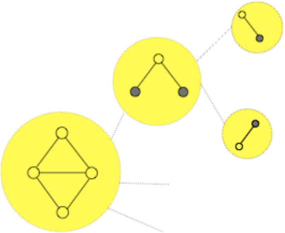
- Start from leaves, compute all possible vertex covers of each edge
- We have two branches Left and Right, and middle set M of vertices separating Left and Right. For every possible assignment A of VC for vertices M, compute

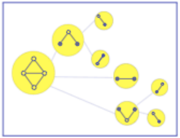
$$VALUE(\text{ Left}, A) + VALUE(\text{Right}, A) - VALUE(A)$$

Dynamic programming: Vertex Cover

v	u	VC
0	0	∞
0	1	1
1	0	1
1	1	2

v	w	VC
0	0	∞
0	1	1
1	0	1
1	1	2

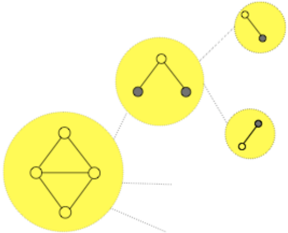


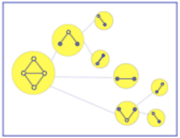


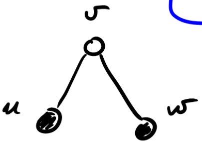
Dynamic programming: Vertex Cover

v	u	VC
0	0	∞
0	1	1
1	0	1
1	1	2

v	w	VC
0	0	∞
0	1	1
1	0	1
1	1	2

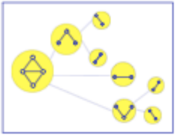




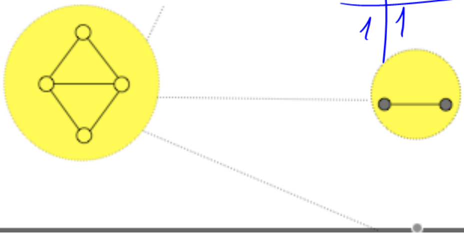


u	w	VC	
0	0	1	($v=1$)
0	1	2	($v=1$)
1	0	2	($v=1$)
1	1	2	($v=0$)

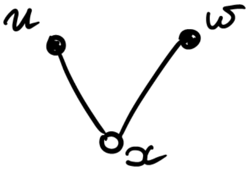
Dynamic programming: Vertex Cover



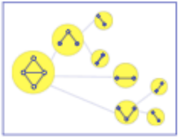
u	w	VC
0	0	∞
0	1	1
1	0	1
1	1	2

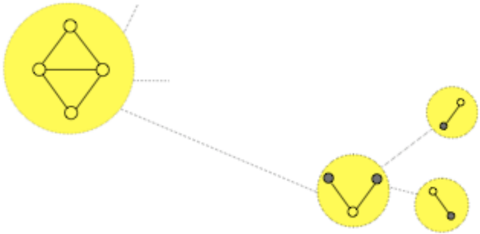


Dynamic programming: Vertex Cover

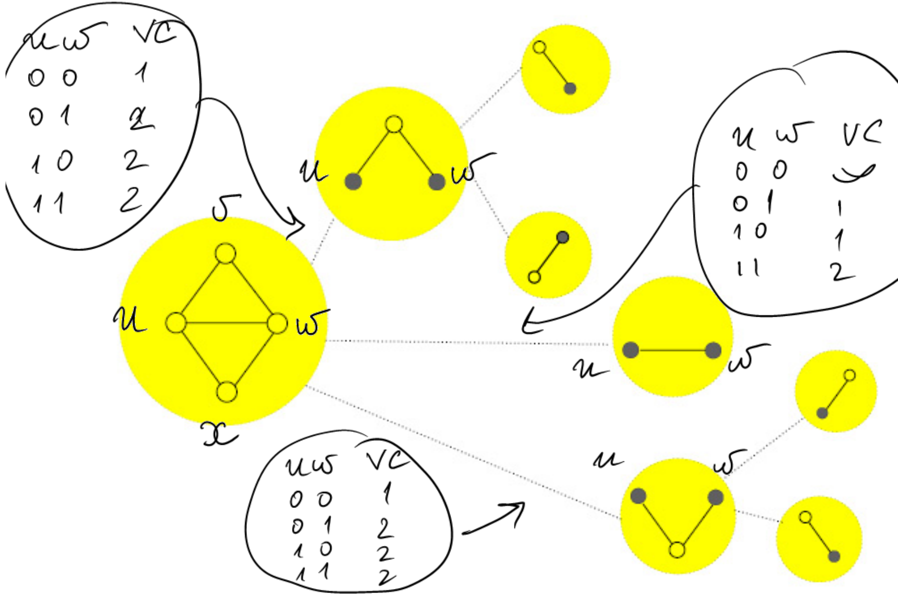


u	w	VC
0	0	1
0	1	2
1	0	2
1	1	2

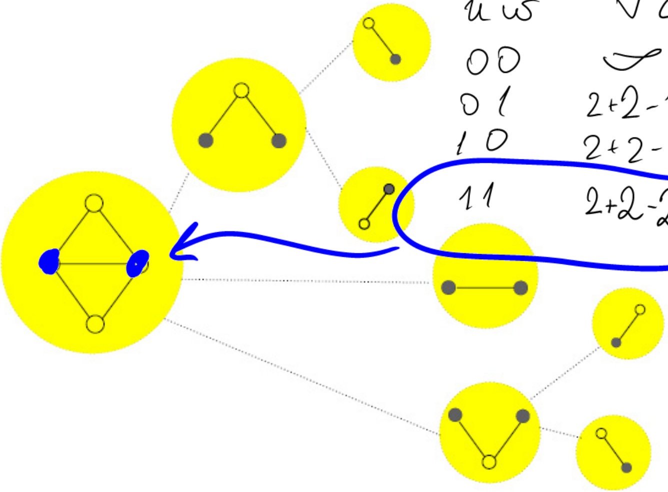




Dynamic programming: Vertex Cover



Dynamic programming: Vertex Cover



u w	VC
0 0	∞
0 1	$2+2-1=3$
1 0	$2+2-1=3$
1 1	$2+2-2=2$

Dynamic programming: Vertex Cover

Let $\ell = \mathbf{bw}(G)$ and $m = |E(G)|$.

- Running time: size of every table for middle set is $O(2^\ell)$.
- To compute a new table: $O(2^{2\ell})$
- Number of steps $O(m)$
- Total running time: $O(2^{2\ell}m)$.

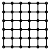
Dynamic programming: Vertex Cover

Exercise

Try to improve the running time, say to $O(2^{1.5\ell}m)$.

Grid Theorem

Theorem (Robertson, Seymour & Thomas, 1994)

Let $\ell \geq 1$ be an integer. Every planar graph of branchwidth $\geq 4\ell$ contains $_{\ell}$ *as a minor.*

Grid Theorem: Sketch of the proof

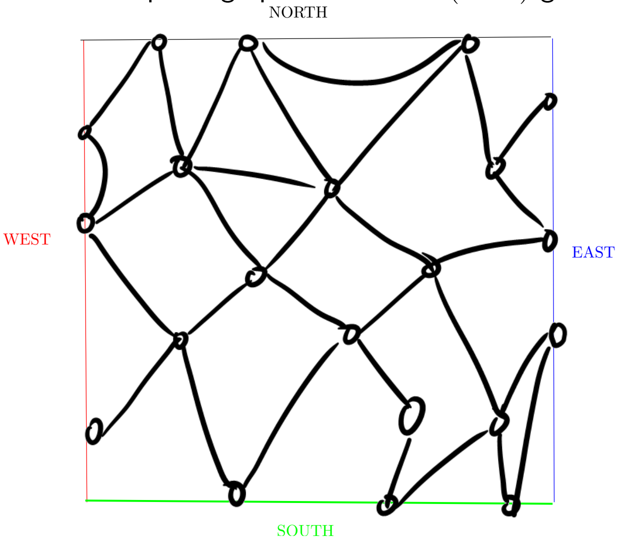
The proof is based on Menger's Theorem

Theorem (Menger 1927)

Let G be a finite undirected graph and x and y two nonadjacent vertices. The size of the minimum vertex cut for x and y (the minimum number of vertices whose removal disconnects x and y) is equal to the maximum number of pairwise vertex-disjoint paths from x to y.

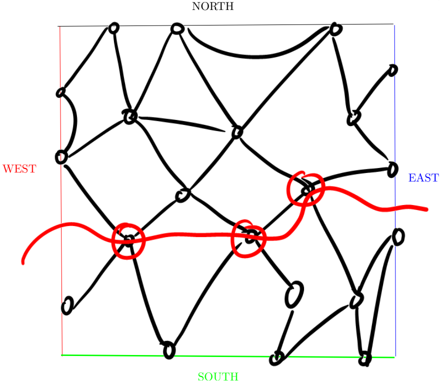
Grid Theorem: Sketch of the proof

Let G be a plane graph that has no $(\ell \times \ell)$-grid as a minor.



Grid Theorem: Sketch of the proof

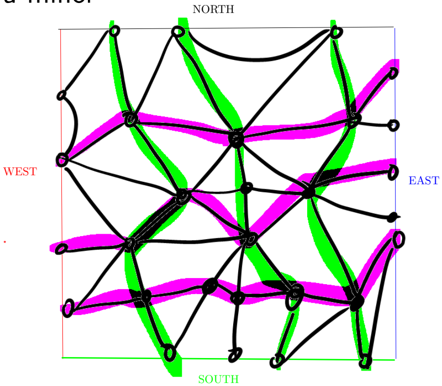
Either East can be separated from West, or South from North by



removing at most ℓ vertices

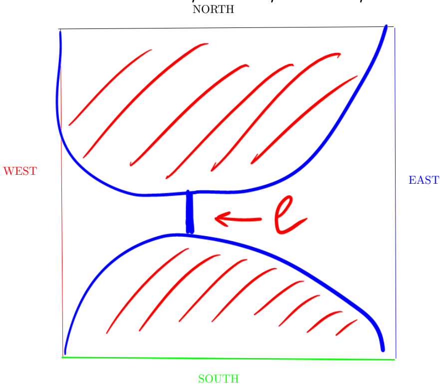
Grid Theorem: Sketch of the proof

Otherwise by making use of Menger we can construct $\ell \times \ell$ grid as a minor



Grid Theorem: Sketch of the proof

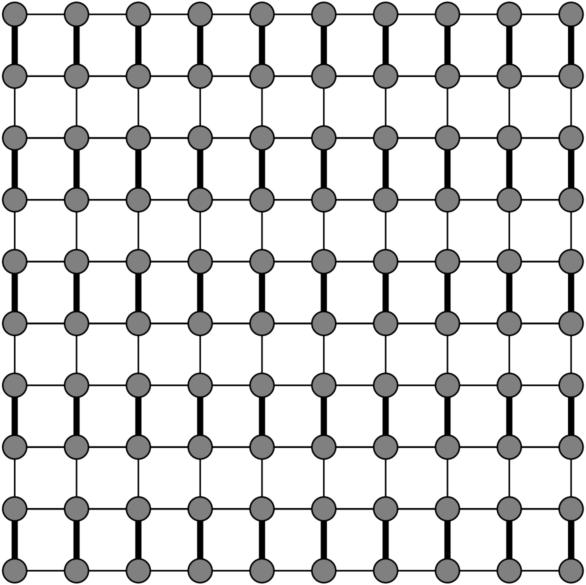
Partition the edges. Every time the middle set contains only vertices of East, West, South, and North, at most 4ℓ in total.



How to compute branchwidth

- NP-hard in general (Seymour-Thomas, Combinatorica 1994)
- On planar graphs can be computed in time $O(n^3)$ (Seymour-Thomas, Combinatorica 1994 and Gu-Tamaki, ICALP 2005)
- RST grid theorem provides 4-approximation.

We know enough to solve Vertex Cover!



$\mathbf{vc}(H_{r,r}) \geq \frac{r^2}{2}$

We know enough to solve Vertex Cover!

Let G be a planar graph of

branchwidth $\geq \ell$

We know enough to solve Vertex Cover!

Let G be a planar graph of branchwidth $\geq \ell$ $\implies$ G contains an $(\ell/4 \times \ell/4)$-grid H as a minor

We know enough to solve Vertex Cover!

Let G be a planar graph of branchwidth $\geq \ell$ $\implies$ G contains an $(\ell/4 \times \ell/4)$-grid H as a minor

The size of any vertex cover of H is at least $\ell^2/32$. Since H is a minor of G, the size of any vertex cover of G is at least $\ell^2/32$.

We know enough to solve Vertex Cover!

Let G be a planar graph of branchwidth $\geq \ell$ $\Longrightarrow$ G contains an $(\ell/4 \times \ell/4)$-grid H as a minor

The size of any vertex cover of H is at least $\ell^2/32$. Since H is a minor of G, the size of any vertex cover of G is at least $\ell^2/32$.

WIN/WIN

If $k < \ell^2/32$, we say "NO"

If $k \geq \ell^2/32$, then we do DP in time $O(2^{2\ell}m) = O(2^{O(\sqrt{k})}m)$.

Challenges to discuss

- How to generalize the idea to work for other parameters?
- Does not work for Dominating Set. Why?
- Is planarity essential?
- Dynamic programming. Does MSOL helps here?

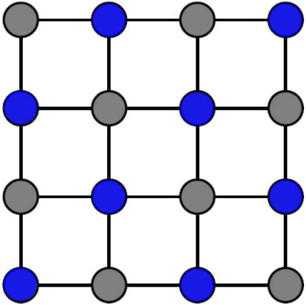
Parameters

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, $P(G) \leq k$ (for minimization) and $P(G) \geq k$ (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, $H \preceq G$ / $H \preceq_c G$ implies that $P(H) \leq P(G)$.

Examples of parameters: k-Vertex Cover

A *vertex cover* C of a graph G, $vc(G)$, is a set of vertices such that every edge of G has at least one endpoint in C. The k-Vertex Cover problem is to decide, given a graph G and a positive integer k, whether G has a vertex cover of size k.

k-Vertex Cover

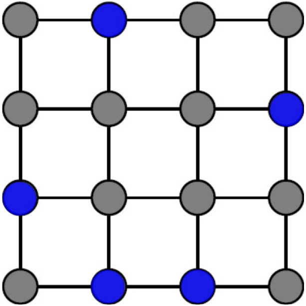


k-Vertex Cover is closed under taking minors.

Examples of parameters: k-Dominating set

A *dominating* set D of a graph G is a set of vertices such that every vertex outside D is adjacent to a vertex of D. The k-Dominating Set problem is to decide, given a graph G and a positive integer k, whether G has a dominating set of size k.

k-Dominating set



k-Dominating set is not closed under taking minors. However, it is closed under contraction of edges.

(Not exactly related to this tutorial but worth to be mentioned)

By Robertson-Seymour theory, every minor closed parameter problem is FPT.

Dynamic programming and Grid Theorem

Meta conditions

(A) For every graph $G \in \mathcal{G}$, $\mathbf{bw}(G) \leq \alpha \cdot \sqrt{P(G)} + O(1)$

(B) For every graph $G \in \mathcal{G}$ and given a branch decomposition (T, μ) of G, the value of $P(G)$ can be computed in $f(\mathbf{bw}(T, \mu)) \cdot n^{O(1)}$ steps.

Algorithm

(A) For every graph $G \in \mathcal{G}$, $\mathbf{bw}(G) \le \alpha \cdot \sqrt{P(G)} + O(1)$

(B) For every graph $G \in \mathcal{G}$ and given a branch decomposition (T, μ) of G, the value of $P(G)$ can be computed in $f(\mathbf{bw}(T, \mu)) \cdot n^{O(1)}$ steps.

If $\mathbf{bw}(T, \mu) > \alpha \cdot \sqrt{k}$, then by (A) the answer is clear

Else, by (B), $P(G)$ can be computed in $f(\alpha \cdot \sqrt{k}) \cdot n^{O(1)}$ steps.

When $f(k) = 2^{O(k)}$, the running time is $2^{O(\sqrt{k})} \cdot n^{O(1)}$

This tutorial:

(A) For every graph $G \in \mathcal{G}$, $\mathbf{bw}(G) \leq \alpha \cdot \sqrt{P(G)} + O(1)$

(B) For every graph $G \in \mathcal{G}$ and given a branch decomposition (T, μ) of G, the value of $P(G)$ can be computed in $f(\mathbf{bw}(T, \mu)) \cdot n^{O(1)}$ steps

- How to prove (A)
- How to do (B)

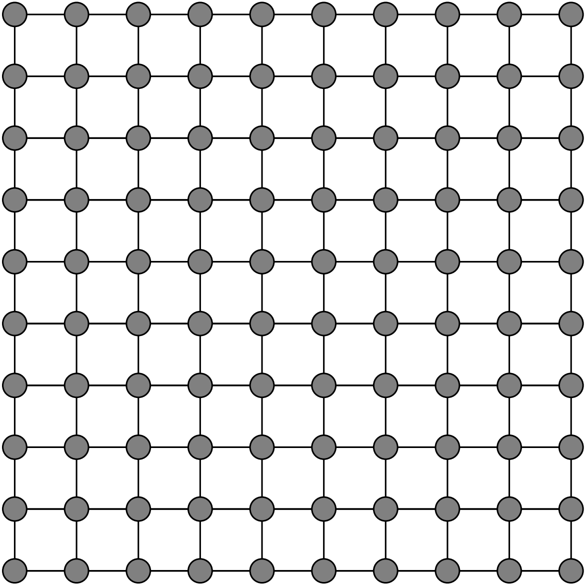
Combinatorial bounds:
Bidimensionality and excluding a grid as a minor

Reminder

Theorem (Robertson, Seymour & Thomas, 1994)

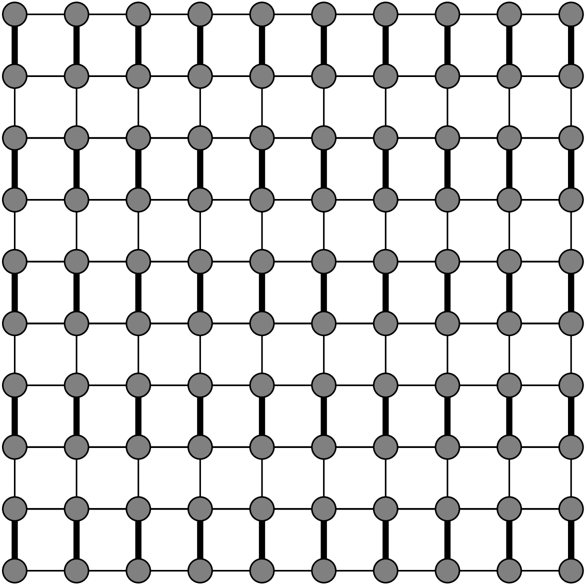
Let $\ell \geq 1$ be an integer. Every planar graph of branchwidth $\geq \ell$ contains an $(\ell/4 \times \ell/4)$-grid as a minor.

Planar k-Vertex Cover



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

Planar k-Vertex Cover



$\mathbf{vc}(H_{r,r}) \geq \frac{r^2}{2}$

PLANAR k-VERTEX COVER

Let G be a planar graph of

branchwidth $\geq \ell$

PLANAR k-VERTEX COVER

Let G be a planar graph of branchwidth $\geq \ell$ $\Longrightarrow$ G contains an $(\ell/4 \times \ell/4)$-grid H as a minor

Planar k-Vertex Cover

Let G be a planar graph of branchwidth $\geq \ell$ $\Longrightarrow$ G contains an $(\ell/4 \times \ell/4)$-grid H as a minor

The size of any vertex cover of H is at least $\ell^2/32$. Since H is a minor of G, the size of any vertex cover of G is at least $\ell^2/32$.

Planar k-Vertex Cover

Let G be a planar graph of branchwidth $\geq \ell$ $\Longrightarrow$ G contains an $(\ell/4 \times \ell/4)$-grid H as a minor

The size of any vertex cover of H is at least $\ell^2/32$. Since H is a minor of G, the size of any vertex cover of G is at least $\ell^2/32$.

Conclusion: Property (A) holds for $\alpha = 4\sqrt{2}$, i.e. $\mathbf{bw}(G) \leq 4\sqrt{2}\sqrt{\mathbf{vc}(G)}$.

Planar k-Vertex Cover

Dorn, 2006: given a branch decomposition of G of width ℓ, the minimum vertex cover of G can be computed in time $f(\ell)n = 2^{\frac{\omega}{2}\ell}n$, where ω is the fast matrix multiplication constant.

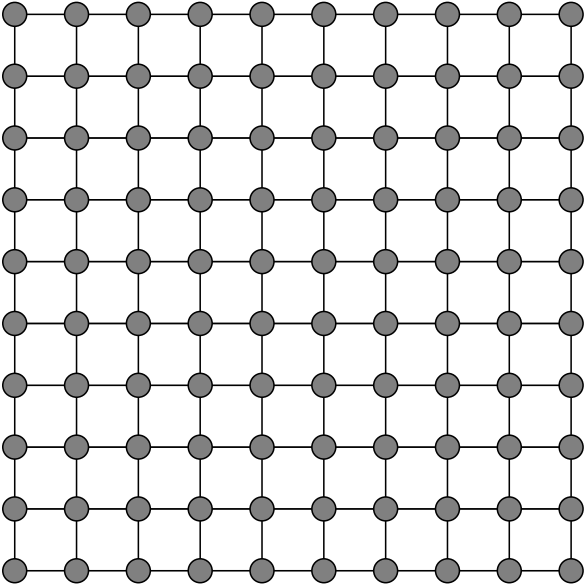
Planar k-Vertex Cover: Putting things together

- Use Seymour-Thomas algorithm to compute a branchwidth of a planar graph G in time $O(n^3)$
- If $\mathbf{bw}(G) \geq \frac{4\sqrt{k}}{\sqrt{2}}$, then G has no vertex cover of size k
- Otherwise, compute vertex cover in time $O(2^{\frac{2\omega\sqrt{k}}{\sqrt{2}}} n) = O(2^{3.56\sqrt{k}} n)$
- Total running time $O(n^3 + 2^{3.56\sqrt{k}} n)$

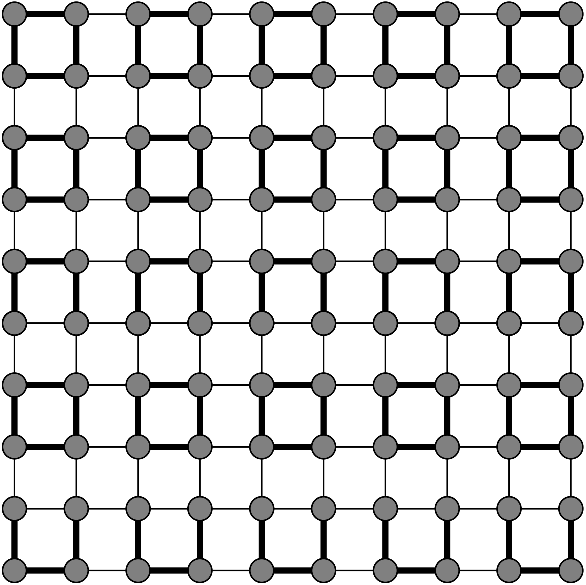
Planar k-Vertex Cover: Kernelization never hurts

- Find a kernel of size $O(k)$ in time $n^{3/2}$ (use Fellows et al. crown decomposition method)
- Use Seymour-Thomas algorithm to compute a branchwidth of the reduced planar graph G in time $O(k^3)$
- If $\mathbf{bw}(G) \geq \frac{4\sqrt{k}}{\sqrt{2}}$, then G has no vertex cover of size k
- Otherwise, compute vertex cover in time $O(2^{\frac{2\omega\sqrt{k}}{\sqrt{2}}} k) = O(2^{3.56\sqrt{k}} k)$
- Total running time $O(n^{3/2} + 2^{3.56\sqrt{k}} k)$

k-Feedback Vertex Set



k-Feedback Vertex Set



$\mathbf{fvc}(H_{r,r}) \geq \frac{r^2}{4}$

k-Feedback Vertex Set

- If $\mathbf{bw}(G) \geq r$, then $G \geq_m H_{\frac{r}{4},\frac{r}{4}}$
- $\mathbf{fvs}$ is minor-closed, therefore $\mathbf{fvs}(G) \geq \mathbf{fvs}(H_{\frac{r}{4},\frac{r}{4}}) \geq \frac{r^2}{64}$

we have that $\mathbf{bw}(G) \leq 8 \cdot \sqrt{\mathbf{fvs}(G)}$

k-Feedback Vertex Set

- If $\mathbf{bw}(G) \geq r$, then $G \geq_m H_{\frac{r}{4},\frac{r}{4}}$
- $\mathbf{fvs}$ is minor-closed, therefore $\mathbf{fvs}(G) \geq \mathbf{fvs}(H_{\frac{r}{4},\frac{r}{4}}) \geq \frac{r^2}{64}$

we have that $\mathbf{bw}(G) \leq 8 \cdot \sqrt{\mathbf{fvs}(G)}$

therefore, for p-Vertex Feedback Set, $f(k) = O(\sqrt{k})$

k-Feedback Vertex Set

- If $\mathbf{bw}(G) \geq r$, then $G \geq_m H_{\frac{r}{4},\frac{r}{4}}$
- $\mathbf{fvs}$ is minor-closed, therefore $\mathbf{fvs}(G) \geq \mathbf{fvs}(H_{\frac{r}{4},\frac{r}{4}}) \geq \frac{r^2}{64}$

we have that $\mathbf{bw}(G) \leq 8 \cdot \sqrt{\mathbf{fvs}(G)}$

therefore, for p-Vertex Feedback Set, $f(k) = O(\sqrt{k})$

Conclusion:

p-Vertex Feedback Set has a $2^{O(\log k \cdot \sqrt{k})} \cdot O(n)$ step algorithm.

PLANAR k-DOMINATING SET

Can we proceed by the same arguments with PLANAR k-DOMINATING SET?

Planar k-Dominating Set

Can we proceed by the same arguments with Planar k-Dominating Set?

Oops! Here is a problem! Dominating set is not minor closed!

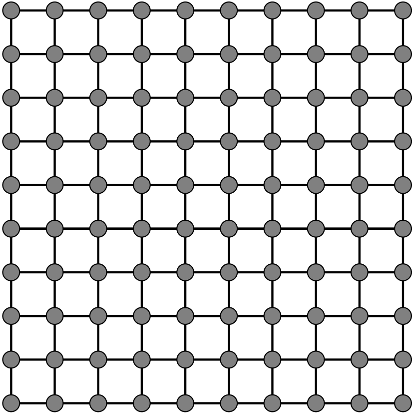
PLANAR k-DOMINATING SET

Can we proceed by the same arguments with PLANAR k-DOMINATING SET?

Oops! Here is a problem! Dominating set is not minor closed!

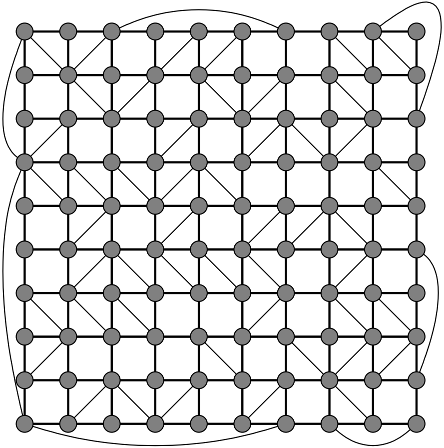
However, dominating set is closed under contraction

PLANAR k-DOMINATING SET



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

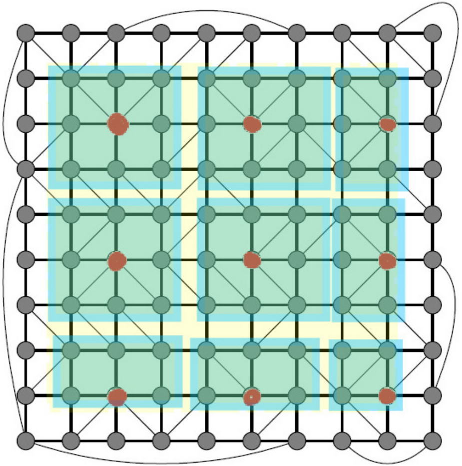
PLANAR k-DOMINATING SET



a partial triangulation of

$H_{10,10}$

PLANAR k-DOMINATING SET



Every inner vertex of p.t. grid $\tilde{H}_{r,r}$ dominates at most 9 vertices. Thus $ds(\tilde{H}_{r,r}) \geq \frac{(r-2)^2}{9}$.

Planar k-Dominating Set

- By RST-Theorem, a planar graph G of branchwidth $\geq \ell$ can be contracted to a partially triangulated $(\ell/4 \times \ell/4)$-grid
- Since dominating set is closed under contraction, we can make the following

Conclusion: Property (A) holds for $\alpha = 12$, i.e.
$\mathbf{bw}(G) \leq 12\sqrt{\mathbf{ds}(G)}$.

Planar k-Dominating Set

- By RST-Theorem, a planar graph G of branchwidth $\geq \ell$ can be contracted to a partially triangulated $(\ell/4 \times \ell/4)$-grid
- Since dominating set is closed under contraction, we conclude that Planar k-Dominating Set also satisfies property (A) with $\alpha = 12$.
- Dorn, 2006, show that for k-Dominating Set in (B), one can choose $f(\ell) = 3^{\frac{\omega}{2}\ell}$, where ω is the fast matrix multiplication constant.

Planar k-Dominating Set

- By RST-Theorem, a planar graph G of branchwidth $\geq \ell$ can be contracted to a partially triangulated $(\ell/4 \times \ell/4)$-grid
- Since dominating set is closed under contraction, we conclude that Planar k-Dominating Set also satisfies property (A) with $\alpha = 12$.
- Dorn, 2006, show that for k-Dominating Set in (B), one can choose $f(\ell) = 3^{\frac{\omega}{2}\ell}$, where ω is the fast matrix multiplication constant.
- Conclusion: Planar k-Dominating Set can be solved in time $O(n^3 + 2^{22.6\sqrt{k}}n)$

Bidimensionality: The main idea

If the graph parameter is closed under taking minors or contractions, the only thing needed for the proof branchwidth/parameter bound is to understand how this parameter behaves on a (partially triangulated) grid.

Bidimensionality: Demaine, FF, Hajiaghayi, Thilikos, 2005

Definition

A parameter P is *minor bidimensional with density* δ if

1. P is closed under taking of minors, and
2. for the $(r \times r)$-grid R, $P(R) = (\delta r)^2 + o((\delta r)^2)$.

Bidimensionality: Demaine, FF, Hajiaghayi, Thilikos, 2005

Definition

A parameter P is called *contraction bidimensional with density* δ if

1. P is closed under contractions,
2. for any partially triangulated $(r \times r)$-grid R,

   $P(R) = (\delta_R r)^2 + o((\delta_R r)^2)$, and
3. δ is the smallest δ_R among all paritally triangulated

   $(r \times r)$-grids.

Bidimensionality

Lemma

If P is a bidimensional parameter with density δ then P satisfies property (A) for $\alpha = 4/\delta$, on planar graphs.

Proof.

Let R be an $(r \times r)$-grid.

$P(R) \geq (\delta_R r)^2$.

If G contains R as a minor, then $\mathbf{bw}(G) \leq 4r \leq 4/\delta\sqrt{P(G)}$. $\square$

Examples of bidimensional problems

Vertex cover

Dominating Set

Independent Set

(k, r)-center

Feedback Vertex Set

Minimum Maximal Matching

Planar Graph TSP

Longest Path ...

How to extend bidimensionality to more general graph classes?

- We need excluding grid theorems (sufficient for minor closed parameters)
- For contraction closed parameters we have to be more careful

Bounded genus graphs: Demaine, FF, Hajiaghayi, Thilikos, 2005

Theorem

If G is a graph of genus at most γ with branchwidth more than r, then G contains a $(r/4(\gamma+1) \times r/4(\gamma+1))$-grid as a minor.

Can we go further?

What about more general graph classes?

- How to define bidimensionality for non-planar graphs?

The grid-minor-excluding theorem gives linear bounds for H-minor free graphs:

Theorem (Demaine & Hajiaghayi, 2008)

There is a function $\phi : \mathbb{N} \to \mathbb{N}$ such that for every graph G excluding a fixed h-vertex graph H as a minor the following holds:

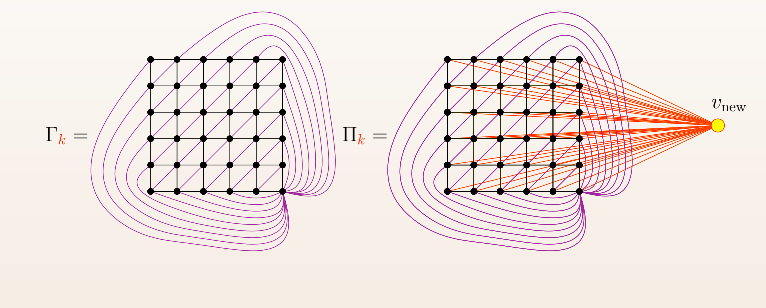
- *if* $\mathbf{bw}(G) \geq \phi(h) \cdot k$ *then* $\boxplus_k \leq_m G$.

For every minor-closed graph class a minor-closed parameter $\mathbf{p}$ is bidimensional if

$$\mathbf{p}(\boxplus_k) = \Omega(k^2)$$

What about contraction-closed parameters?

We define the following two pattern graphs Γ_k and Π_k:



$\Pi_k = \Gamma_k +$ a new vertex v_{new}, connected to all the vertices in $V(\Gamma_k)$.

The grid-minor-excluding theorem gives linear bounds for H-minor free graphs:

Theorem (Fomin, Golovach, & Thilikos, 2009)

There is a function $\phi : \mathbb{N} \to \mathbb{N}$ *such that for every graph* G *excluding a fixed* h*-vertex graph* H *as contraction the following holds:*

- *if* $\mathbf{bw}(G) \geq \phi(h) \cdot k$ *then either* $\Gamma_k \leq_c G$, *or* $\Pi_k \leq_c G$.

For contraction-closed graph class a contraction-closed parameter

$\mathbf{p}$ is bidimensional if

$$\mathbf{p}(\Gamma_k) = \Omega(k^2) \text{ and } \mathbf{p}(\Pi_k) = \Omega(k^2).$$

Limits of the bounded branchwidth WIN/WIN technique

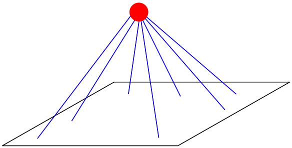
As for each contraction-closed parameter $\mathbf{p}$ that we know, it holds that $\mathbf{p}(\Pi_k) = O(1)$ for all k,

Bidimensionality can be defined for apex-minor free graphs

(*apex graphs* are exactly the minors of Π_k)

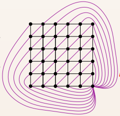
H^* is an *apex graph* if

$\exists v \in V(H^*)$: $H^* - v$ is planar

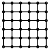


Therefore for every apex-minor free graph class

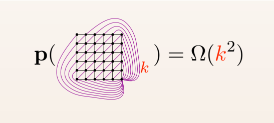
a contraction-closed parameter $\mathbf{p}$ is bidimensional if

$\mathbf{p}($  $_{k}) = \Omega(k^2)$

Conclusion

Minor bidimensional: minor- closed and $\mathbf{p}($  $_{k}) = \Omega(k^2)$

Contraction-bidimensional: contraction-closed and

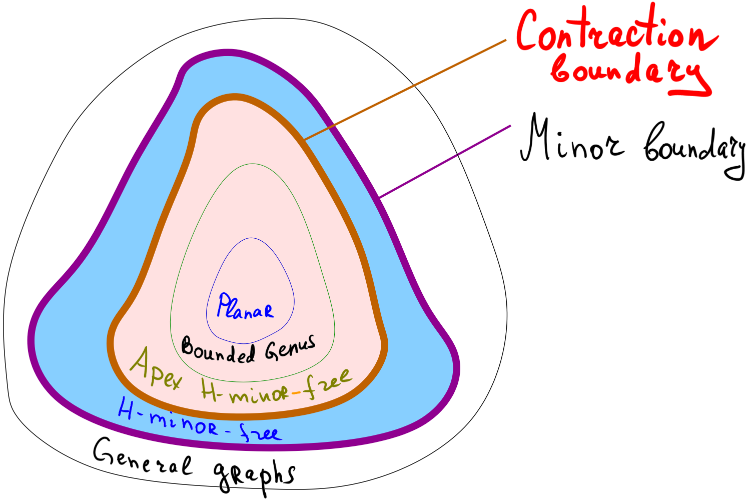


Theorem (Bidimensionality meta-algorithm)

Let $\mathbf{p}$ be a minor (resp. contraction)-bidimensional parameter that is computable in time $2^{O(\mathbf{bw}(G))} \cdot n^{O(1)}$.

Then, deciding $\mathbf{p}(G) \leq k$ for general (resp. apex) minor-free graphs can be done (optimally) in time $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

Limits of the bidimensionality



Remark

Bidimensionality cannot be used to obtain subexponential algorithms for contraction closed parameterized problems on H-minor free graphs.

For some problems, like k-DOMINATING SET, it is still possible to design subexponential algorithms on H-minor free graphs.

The main idea here is to use decomposition theorem of Robertson-Seymour about decomposing an H-minor free graph into pieces of apex-minor-free graphs, apply bidimensionality for each piece, and do dynamic programming over the whole decomposition.

Grids for other problems

EXAMPLE I: t-spanners (ICALP 2008, Dragan, FF, Golovach)

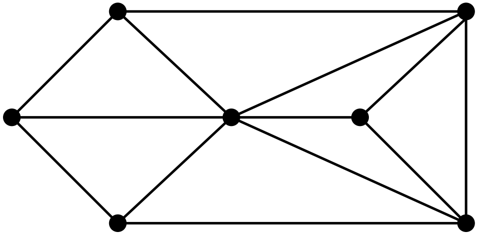
t-spanners

Definition (t-spanner)

Let t be a positive integer. A subgraph S of G, such that $V(S) = V(G)$, is called a t-spanner, if $\mathrm{dist}_S(u, v) \leq t \cdot \mathrm{dist}_G(u, v)$ for every pair of vertices u and v. The parameter t is called the *stretch factor* of S.

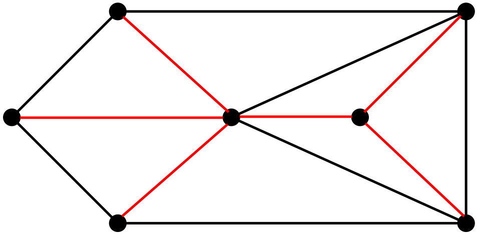
Examples of spanners

3 and 2-spanners



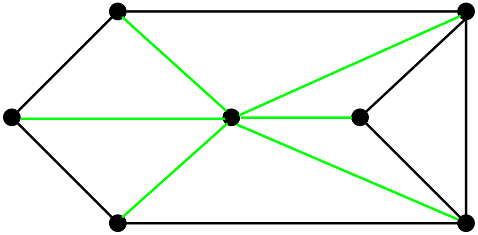
Examples of spanners

3 and 2-spanners



Examples of spanners

3 and 2-spanners



Spanners of bounded branchwidth

Problem (k-Branchwidth t-spanner)

Instance: A connected graph G and positive integers k and t.

Question: Is there a t-spanner of G of branchwidth at most k?

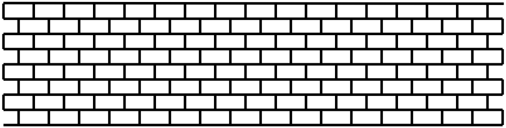
Planar graphs

Theorem (Bounds for planar graphs)

Let G be a planar graph of branchwidth k and let S be a t-spanner of G. Then the branchwidth of S is $\Omega(k/t)$.

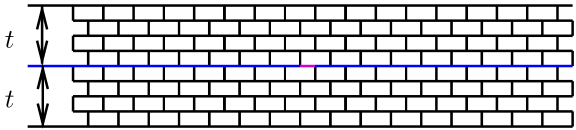
Sketch of the proof

Walls and grids



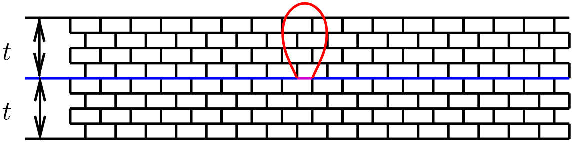
Sketch of the proof

Walls and grids



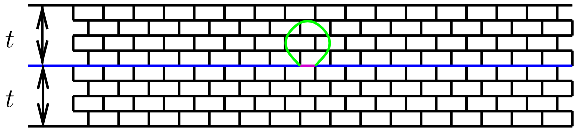
Sketch of the proof

Walls and grids



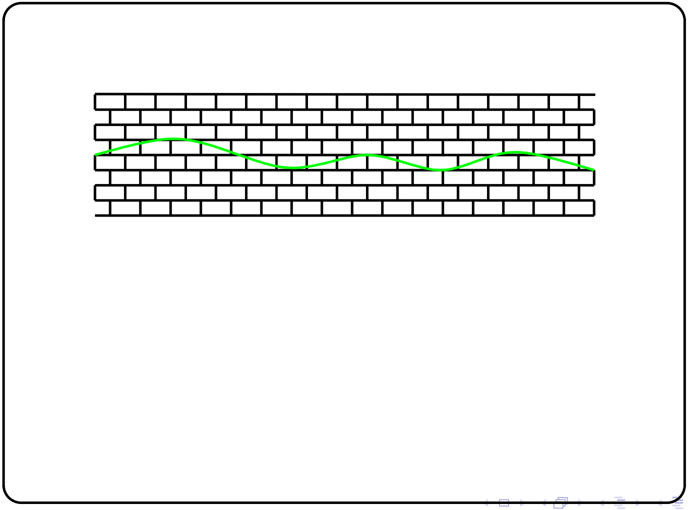
Sketch of the proof

Walls and grids



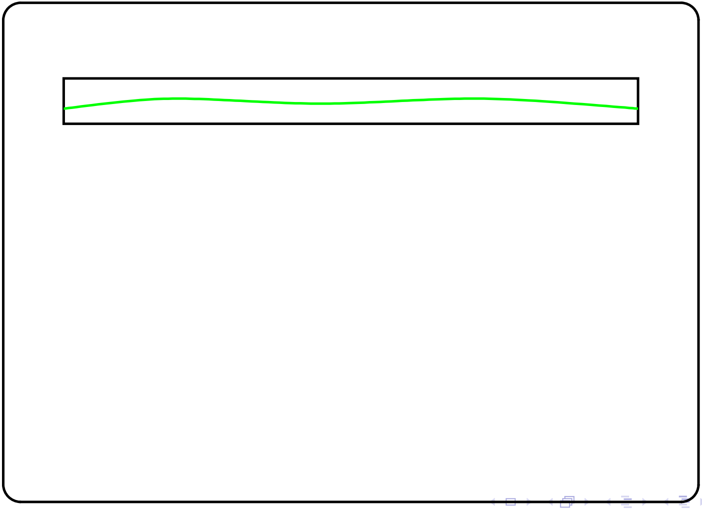
Sketch of the proof

Walls and grids



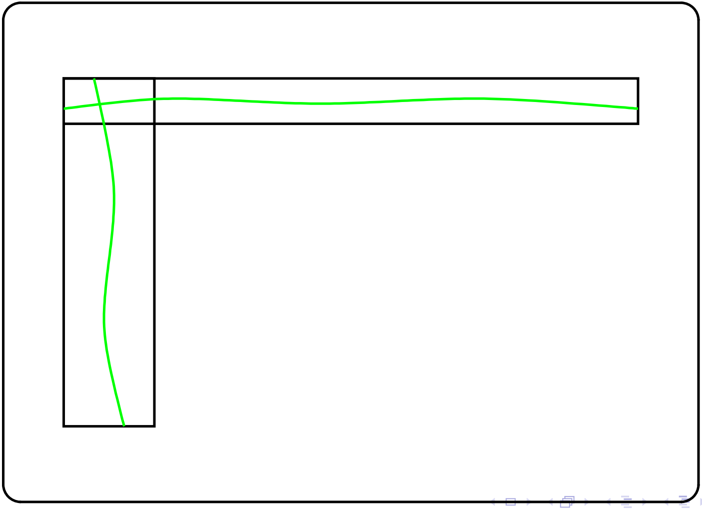
Sketch of the proof

Walls and grids



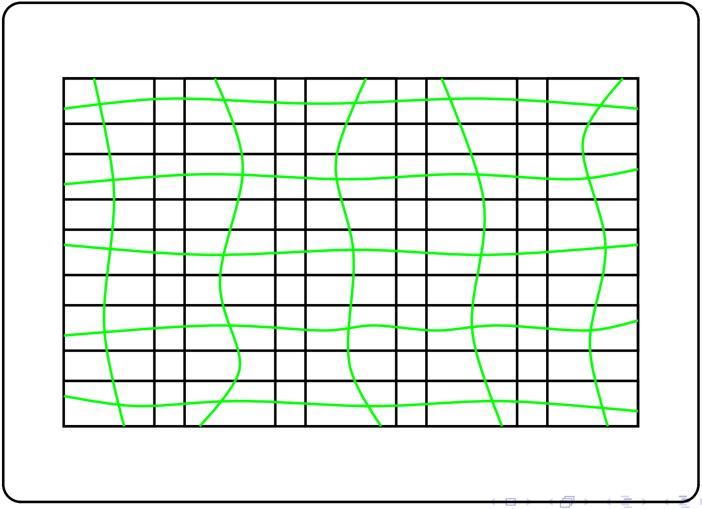
Sketch of the proof

Walls and grids



Sketch of the proof

Walls and grids



Algorithmic consequences

Theorem (Dragan, FF, Golovach, 2008)

Deciding if a planar graph G has a t-spanner of treewidth at most k is solvable in time $O(f(k,t) \cdot n)$.

Theorem (Dragan, FF, Golovach, 2008)

Let H be a fixed apex graph. For every fixed k and t, the existence of a t-spanner of treewidth at most k in an H-minor-free graph G can be decided in linear time.

Another example

Induced cycle spanning a specified set of vertices

(SODA 2009, KOBAYASHI and KAWARABAYASHI)

Induced cycle

Problem (Induced Cycle Problem)

Instance: Planar graph G and and a subset $S \subseteq V(G)$ of terminal vertices of size k.

Question: Is there an induced cycle in G containing all terminal vertices S?

Parameter k.

Algorithm sketch

- If there is a vertex which is far from each of the terminals—just remove it, it does not change the solution. (Far here means that there are $22k + 2$ nested disjoint cycles around v.)
- If every vertex is "close" to each of the terminals, then the branchwidth of the graph $O(k^{3/2})$. To prove this, one has to look at the grid!

Bidimensional theory: Conclusion

If $\mathbf{P}$ is a parameter that

(A) is minor (contraction) bidimensional

(B) can be computed in $f(\mathbf{bw}(G)) \cdot n^{O(1)}$ steps.

then there is a $f(O(\sqrt{k})) \cdot n^{O(1)}$ step algorithm for checking whether $\mathbf{P}(G) \leq k$ for H (apex) -minor free graphs.

We now fix our attention to property (B) and function f.

Dynamic programming and Catalan structures

Dynamic programming for branch decompositions

- We root the tree T of the branch decomposition (T, τ),
- We define a partial solution for each cut-set of an edge e of T
- We compute all partial solutions bottom-up (using the partial solutions corresponding to the children edges).

This can be done in $O(f(\ell) \cdot n)$ if we have a branch decomposition of width at most ℓ.

Dynamic programming for branch decompositions

- We root the tree T of the branch decomposition (T, τ),
- We define a partial solution for each cut-set of an edge e of T
- We compute all partial solutions bottom-up (using the partial solutions corresponding to the children edges).

This can be done in $O(f(\ell) \cdot n)$ if we have a branch decomposition of width at most ℓ.

$f(\ell)$ depends on the number of partial solutions we have to compute for each edge of T.

- To find a good bound for $f(\ell)$ is important!

For many problems, $2^{O(\mathbf{bw}(G))} \cdot n^{O(1)}$ step algorithms exist.

Dynamic programming on graphs with small branchwidth gives

such algorithms for problems like

Vertex Cover,

Dominating Set, or

Edge Dominating Set, (and others...)

However: There are (many) problems where no general $2^{O(\mathbf{bw}(G))} \cdot n^{O(1)}$ step algorithm is known.

However: There are (many) problems where no general $2^{O(\mathbf{bw}(G))} \cdot n^{O(1)}$ step algorithm is known.

Such problems are

Longest Path, Longest Cycle, Connected Dominating Set, Feedback Vertex Set, Hamiltonian Cycle, Max Leaf Tree and Graph Metric TSP

However: There are (many) problems where no general $2^{O(\mathbf{bw}(G))} \cdot n^{O(1)}$ step algorithm is known.

Such problems are

Longest Path, Longest Cycle, Connected Dominating Set, Feedback Vertex Set, Hamiltonian Cycle, Max Leaf Tree and Graph Metric TSP

For the natural parameterizations of these problems, no $2^{O(\sqrt{k})} \cdot n^{O(1)}$ step FPT-algorithm follows by just using bidimensionality theory and dynamic programming.

Example: k-Longest Path

The k-Longest Path problem is to decide, given a graph G and a positive integer k, whether G contains a path of length k.

This problem is closed under the operation of taking minor.

Example: k-Longest Path

k-Longest Path has a

$2^{O(\sqrt{k} \cdot \log k)} \cdot n^{O(1)}$ step algorithm.

Because

Example: k-Longest Path

k-Longest Path has a

$2^{O(\sqrt{k} \cdot \log k)} \cdot n^{O(1)}$ step algorithm.

Because

(A) The parameter is minor bidimensional

Example: k-Longest Path

k-Longest Path has a

$2^{O(\sqrt{k} \cdot \log k)} \cdot n^{O(1)}$ step algorithm.

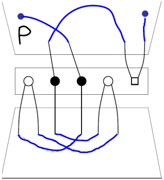
Because

(A) The parameter is minor bidimensional

(B) to find a longest path in a graph G takes

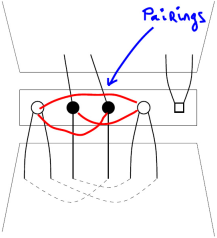
$2^{O(\mathbf{bw}(G) \cdot \log\ \mathbf{bw}(G))} \cdot n$ steps

Why $\log \mathbf{bw}(G)$?



Let P be a path in G. An edge e of a branch decomposition T splits G into G_e and $G \setminus G_e$.

Why log bw(G)?



Let P be a path in G. An edge e of a branch decomposition T splits G into G_e and $G \backslash G_e$.

- The restriction of a P to G_e is a collection $\mathcal{P}$ of internally disjoint paths in G_e with ends in $\mathrm{mid}(e)$.
- Each $\mathcal{P}$ corresponds to some pairing (a disjoint set of paths in the clique formed from $\mathrm{mid}(e)$)
- For a set S, let $\texttt{pairs}(S)$ be the set of all pairings of S

Therefore, the complexity of dynamic programming depends on $|\texttt{pairs}(\mathrm{mid}(e))|$, which is $\Omega(\mathbf{bw}!)$.

This obstacle does not allow to break $2^{O(\mathbf{bw}(G)\cdot\log\,\mathbf{bw}(G))}\cdot n$ barrier.

Therefore, the complexity of dynamic programming depends on $|\texttt{pairs}(\mathrm{mid}(e))|$, which is $\Omega(\mathbf{bw}!)$.

This obstacle does not allow to break $2^{O(\mathbf{bw}(G)\cdot\log\,\mathbf{bw}(G))}\cdot n$ barrier.

- Problem: The local info in dynamic programming is too big!

Therefore, the complexity of dynamic programming depends on $|\texttt{pairs}(\text{mid}(e))|$, which is $\Omega(\mathbf{bw}!)$.

This obstacle does not allow to break $2^{O(\mathbf{bw}(G)\cdot\log\ \mathbf{bw}(G))}\cdot n$ barrier.

- Problem: The local info in dynamic programming is too big!
- Issue: The same problem appears in many dynamic programming algorithms!

Therefore, the complexity of dynamic programming depends on $|\mathtt{pairs}(\mathrm{mid}(e))|$, which is $\Omega(\mathbf{bw}!)$.

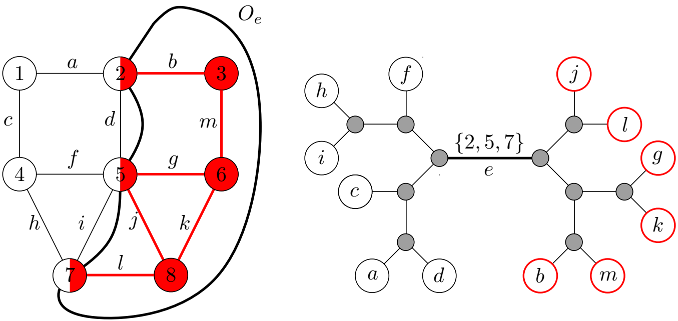
This obstacle does not allow to break $2^{O(\mathbf{bw}(G)\cdot\log\ \mathbf{bw}(G))}\cdot n$ barrier.

- Problem: The local info in dynamic programming is too big!
- Issue: The same problem appears in many dynamic programming algorithms!
- Idea: as long as we care about sparse graph classes, we can take their structure into consideration!

Sphere-cut decomposition

Let G be a planar graph embedded on the sphere $\mathcal{S}_0$

A *sphere-cut decomposition* of G is a branch decomposition (T, τ) where for every $e \in E(T)$, the vertices in $\mathrm{mid}(e)$ are the vertices in a Jordan curve of $\mathcal{S}_0$ that meets no edges of G.



Seymour-Thomas 1994, Dorn-Penninkx-Bodlaender-FF 2005

Theorem

Every planar graph G of branchwidth ℓ has a sphere-cut decomposition of width ℓ. This decomposition can be constructed in $O(n^3)$ steps.

For doing dynamic programming on a sphere cut decomposition (T, τ) of width ℓ we define, for every $e \in E(T)$ the set $\texttt{pairs}(\mathrm{mid}(e))$ be the set of all pairings of $\mathrm{mid}(e)$

For doing dynamic programming on a sphere cut decomposition (T, τ) of width ℓ we define, for every $e \in E(T)$ the set $\texttt{pairs}(\mathrm{mid}(e))$ be the set of all pairings of $\mathrm{mid}(e)$

The "usual" bound on the size of $\texttt{pairs}(\mathrm{mid}(e))$ is $2^{O(\ell \cdot \log \ell)}$

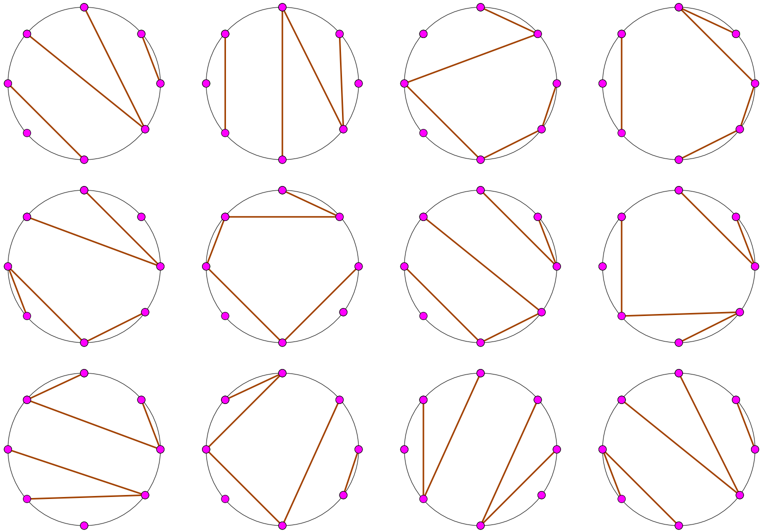
For doing dynamic programming on a sphere cut decomposition (T, τ) of width ℓ we define, for every $e \in E(T)$ the set $\texttt{pairs}(\mathrm{mid}(e))$ be the set of all pairings of $\mathrm{mid}(e)$

The "usual" bound on the size of $\texttt{pairs}(\mathrm{mid}(e))$ is $2^{O(\ell \cdot \log \ell)}$

However, we now have that

1: the vertices of $\mathrm{mid}(e)$ lay on the boundary of a disk and

2: the pairings cannot be crossing because of planarity.



It follows that $\texttt{pairs}(\mathrm{mid}(e)) = O(C(|\mathrm{mid}(e)|)) = O(C(\ell))$

Where $C(\ell)$ is the ℓ-th *Catalan Number*.

It is known that $C(\ell) \sim \frac{4^\ell}{\ell^{3/2}\sqrt{\pi}} = 2^{O(\ell)}$

It follows that $\texttt{pairs}(\mathrm{mid}(e)) = O(C(|\mathrm{mid}(e)|)) = O(C(\ell))$

Where $C(\ell)$ is the ℓ-th *Catalan Number*.

It is known that $C(\ell) \sim \frac{4^\ell}{\ell^{3/2}\sqrt{\pi}} = 2^{O(\ell)}$

Therefore: dynamic programming for finding the longest path of a planar graph G on a sphere cut decompositions of G with width $\leq \ell$ takes $O(2^{O(\ell)} \cdot n)$ steps.

It follows that $\texttt{pairs}(\mathrm{mid}(e)) = O(C(|\mathrm{mid}(e)|)) = O(C(\ell))$

Where $C(\ell)$ is the ℓ-th *Catalan Number*.

It is known that $C(\ell) \sim \frac{4^\ell}{\ell^{3/2}\sqrt{\pi}} = 2^{O(\ell)}$

Therefore: dynamic programming for finding the longest path of a planar graph G on a sphere cut decompositions of G with width $\leq \ell$ takes $O(2^{O(\ell)} \cdot n)$ steps.

Conclusion: [by bidimensionality]

Planar k-Longest Path can be solved in $O(2^{O(\sqrt{k})} \cdot n + n^3)$ steps

- The same holds for several other problems where an analogue of $\texttt{pairs}(\mathrm{mid}(e))$ can be defined for controlling the size of the tables in dynamic programming.

- The same holds for several other problems where an analogue of $\texttt{pairs}(\mathrm{mid}(e))$ can be defined for controlling the size of the tables in dynamic programming.
- Like that one can design $2^{O(\sqrt{k})} \cdot n^{O(1)}$ step algorithms for parameterized planar versions of CYCLE COVER, PATH COVER, LONGEST CYCLE, CONNECTED DOMINATING SET, FEEDBACK VERTEX SET, HAMILTONIAN CYCLE, GRAPH METRIC TSP, MAX LEAF TREE, etc.

How to use Catalan structure in non-planar graphs?

We say that branch decomposition (T, τ) of width ℓ has the *Catalan Structure* for k-LONGEST PATH if

$$\forall_{e \in E(T)} \ \texttt{pairs}(\mathrm{mid}(e)) = 2^{O(\ell)}$$

How to use Catalan structure in non-planar graphs?

We say that branch decomposition (T, τ) of width ℓ has the *Catalan Structure* for k-Longest Path if

$$\forall_{e \in E(T)} \ \texttt{pairs}(\mathrm{mid}(e)) = 2^{O(\ell)}$$

- We have seen that, for planar graphs, one can construct a branch decomposition with the Catalan structure for the k-Longest Path problem.

Dorn-FF-Thilikos 2008

Theorem

For any H-minor free graph class $\mathcal{G}$ there is a constant c_H (depending only on H) such that the following holds: For every graph $G \in \mathcal{G}$ and any positive integer w, it is possible to construct a $c_H \cdot n^{O(1)}$-step algorithm that outputs either

1. *a correct report that $\mathbf{bw}(G) > w$ or*
2. *a branch decomposition (T, τ) with the Catalan structure and of width $c_H \cdot w$.*

Consequences:

- For H-minor free graphs, one can construct an algorithm that solves the k-Longest Path problem in $2^{O(\sqrt{k})} \cdot n^{O(1)}$ steps.

Consequences:

- For H-minor free graphs, one can construct an algorithm that solves the k-Longest Path problem in $2^{O(\sqrt{k})} \cdot n^{O(1)}$ steps.
- Using the same result one can also solve, for H-minor free graphs, in $2^{O(\sqrt{k})} \cdot n^{O(1)}$ steps, the the standard parameterization of Longest Cycle, and Cycle/Path Cover, parameterized either by the total length of the cycles/paths or the number of the cycles/paths.

By applying modifications it is possible to define an analogue of Catalan Structure property for other problems like FEEDBACK VERTEX SET, CONNECTED DOMINATING SET, and MAX LEAF TREE

Proof idea: again Graph Minors

[Robertson and Seymour – GM 16]: *any H -minor free graph can roughly be obtained by identifying in a tree-like way small cliques of a collection of components that are almost embeddable on bounded genus surfaces.*

Proof idea: again Graph Minors

[Robertson and Seymour – GM 16]: *any H -minor free graph can roughly be obtained by identifying in a tree-like way small cliques of a collection of components that are almost embeddable on bounded genus surfaces.*

- Proof idea: We construct an "almost"-planarizing with certain topological properties, able to reduce the high genus "almost"-embeddings to planar ones where the planarizing vertices are "almost"-cyclically arranged in the plain.

In the plane, we use sphere cut decompositions, that permit to encode collections of paths that may pass through a separator as non crossing pairings of the vertices of a cycle.

In the plane, we use sphere cut decompositions, that permit to encode collections of paths that may pass through a separator as non crossing pairings of the vertices of a cycle.

- This provides the so-called Catalan structure of the decomposition and permits us to suitably bound the ways a path may cross its separators.

In the plane, we use sphere cut decompositions, that permit to encode collections of paths that may pass through a separator as non crossing pairings of the vertices of a cycle.

- This provides the so-called Catalan structure of the decomposition and permits us to suitably bound the ways a path may cross its separators.

- This decomposition is used to build a decomposition on the initial almost embeddible graph (following the tree-like way these components are linked together).

Open problems I

Lower bounds on dynamic programming over branchwidth. Is it possible to prove (up to some conjecture in complexity theory) that Longest Path on graphs of branchwidth ℓ cannot be solved in $2^{o(\ell \log \ell)} m$?

Open problems I

Lower bounds on dynamic programming over branchwidth. Is it possible to prove (up to some conjecture in complexity theory) that Longest Path on graphs of branchwidth ℓ cannot be solved in $2^{o(\ell \log \ell)} m$?

Can Vertex Cover be solved faster than $2^{\ell} n^{O(1)}$?

Open problems II

When applying our technique on different problems we define, for each one of them, an appropriate analogue of `pairs` and prove that it also satisfies the Catalan structure property (i.e. is bounded by $2^{O(|\operatorname{mid}(e)|)}$).

Open problems II

When applying our technique on different problems we define, for each one of them, an appropriate analogue of `pairs` and prove that it also satisfies the Catalan structure property (i.e. is bounded by $2^{O(|\operatorname{mid}(e)|)}$).

- It is challenging to find a classification criterion (logical or combinatorial) for the problems that are amenable to this approach.

Open problems III

Sufficient condition: Bidimensionality (plus fast dynamic programming) yields subexponential parameterized algorithm.

Open problems III

Sufficient condition: Bidimensionality (plus fast dynamic programming) yields subexponential parameterized algorithm.

What are the necessary conditions?

Remark: Every problem on planar graphs for which we know subexponential parameterized algorithm is either bidimensional, or can be reduced to a bidimensional problem.

Open problems IV

Branchwidth: Polynomial time algorithm for graphs of bounded genus? H-minor free graphs?

Further reading. Subexponential algorithms and bidimensionality

- J. Alber, H. L. Bodlaender, H. Fernau, T. Kloks, and R. Niedermeier, *Fixed parameter algorithms for dominating set and related problems on planar graphs*, Algorithmica, 33 (2002), pp. 461–493.
- E. D. Demaine, F. V. Fomin, M. Hajiaghayi, and D. M. Thilikos, *Subexponential parameterized algorithms on graphs of bounded genus and H-minor-free graphs*, Journal of the ACM, 52 (2005), pp. 866–893.

Further reading. Catalan structures and dynamic programming

- F. Dorn, E. Penninkx, H. Bodlaender, and F. V. Fomin, *Efficient exact algorithms on planar graphs: Exploiting sphere cut branch decompositions*, Proceedings of the 13th Annual European Symposium on Algorithms (ESA 2005), vol. 3669 of LNCS, Springer, 2005, pp. 95–106.
- F. Dorn, F. V. Fomin, and D. M. Thilikos, *Catalan structures and dynamic programming on H-minor-free graphs*, in Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2008), ACM-SIAM, pp. 631–640.

Further reading. Surveys

E. Demaine and M. Hajiaghayi, *The bidimensionality theory and its algorithmic applications*, The Computer Journal, (2007), pp. 332–337.

F. Dorn, F. V. Fomin, and D. M. Thilikos, *Subexponential parameterized algorithms*, Computer Science Review, 2 (2008), pp. 29–39.