FEDOR V. FOMIN

Part II. Treewidth applications

UNIVERSITAS BERGENSIS

Properties of treewidth

Fact: The treewidth of $G_1 \cup G_2$ is the maximum of $\mathbf{tw}(G_1)$ and $\mathbf{tw}(G_2)$.

Properties of treewidth

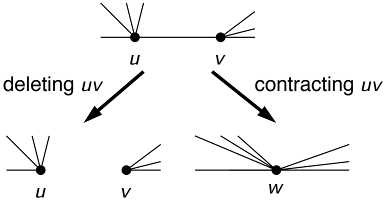
Fact: The treewidth of the complete graph K_k is $k-1$.

Properties of treewidth

Fact: Treewidth does not increase if we delete edges or delete vertices

Graph Minors

Definition: Graph H is a **minor** G ($H \leq G$) if H can be obtained from G by deleting edges, deleting vertices, and contracting edges.

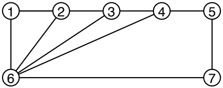


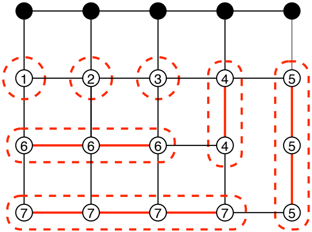
Example: A triangle is a minor of a graph G if and only if G has a cycle (i.e., it is not a forest).

Graph minors

Equivalent definition: Graph H is a **minor** of G if there is a mapping ϕ that maps each vertex of H to a connected subset of G such that

- $\phi(u)$ and $\phi(v)$ are disjoint if $u \neq v$, and
- if $uv \in E(G)$, then there is an edge between $\phi(u)$ and $\phi(v)$.





Properties of treewidth

Fact: Treewidth does not increase if we delete edges, delete vertices or contract edges.

Hence, if $H \leq G$ then $\textbf{tw}(H) \leq \textbf{tw}(G)$.

If G contains a complete graph K_{k+1} as a minor, then $\textbf{tw}(G) \geq k$.

Properties of treewidth

Fact: If G contains a complete graph K_{k+1} as a minor, then $\mathbf{tw}(G) \geq k$.

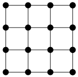
Properties of treewidth

Fact: If G contains a complete graph K_{k+1} as a minor, then $\mathbf{tw}(G) \geq k$.

If the treewidth of a graph is large, does it contain a large clique as a minor?

Properties of treewidth

Fact: For every $k \geq 2$, the treewidth of the $k \times k$ grid is exactly k.



Graph Minors

If a graph contains large grid as a minor, its treewidth is also large.

Graph Minors

If a graph contains large grid as a minor, its treewidth is also large.

What is much more surprising, is that the converse is also true: every graph of large treewidth contains a large grid as a minor.

Neil Robertson

Paul Seymour

Excluded Grid Theorem A : Planar Graph

Our set of treewidth applications is based on the following

Theorem (Planar Excluded Grid Theorem, Robertson, Seymour and Thomas; Guo and Tamaki)

Let $t \geq 0$ be an integer. Every planar graph G of treewidth at least $\frac{9}{2}t$, contains $\boxplus_t$ as a minor. Furthermore, there exists a polynomial-time algorithm that for a given planar graph G either outputs a tree decomposition of G of width $\frac{9}{2}t$ or constructs a minor model of $\boxplus_t$ in G.

Grid Theorem: Sketch of the proof

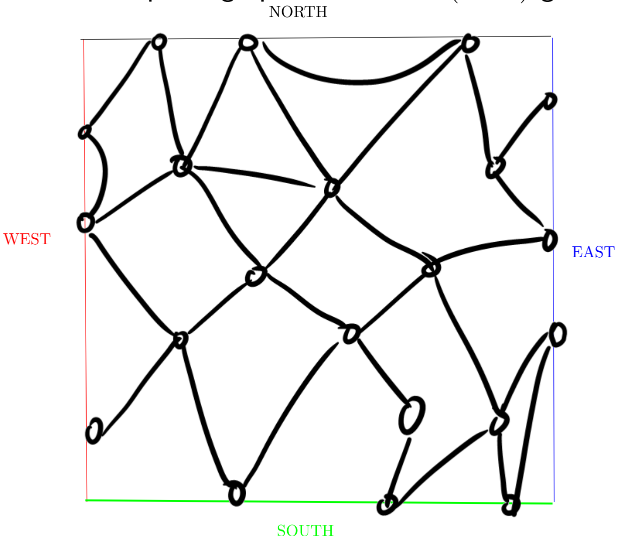
The proof is based on Menger's Theorem

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.

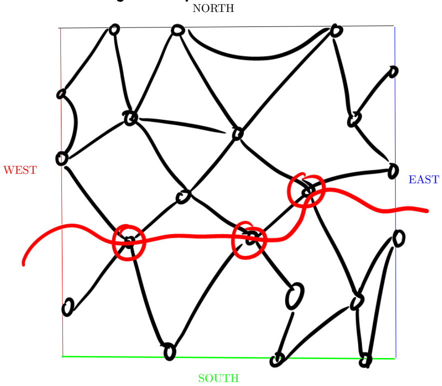
Grid Theorem: Sketch of the proof

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



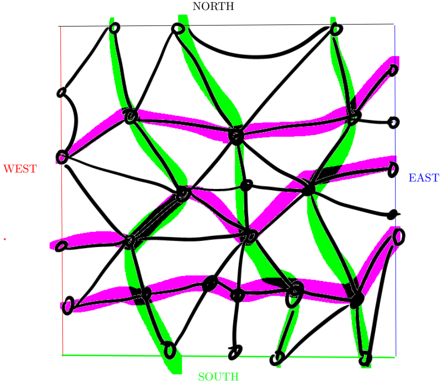
Grid Theorem: Sketch of the proof

If East cannot be separated from West, and South from North by removing at most ℓ vertices, then by Menger's theorem there are ℓ vertex disjoint paths from South to North and from East to West.



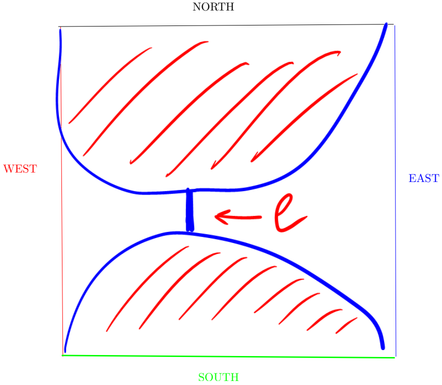
Grid Theorem: Sketch of the proof

We can construct $\Omega(\ell) \times \Omega(\ell)$ grid minor-model from disjoint paths



Grid Theorem: Sketch of the proof

If East can be separated from West, or South from North by ℓ vertices, we can proceed recursively by constructing a tree decomposition of width $O(\ell)$...



Excluded Grid Theorem: Planar Graphs

One more Excluded Grid Theorem, this time not for minors but just for edge contractions.

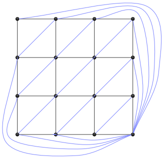


Figure : A triangulated grid Γ_4.

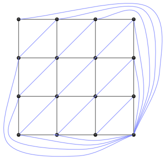
For an integer $t > 0$ the graph Γ_t is obtained from the grid $\boxplus_t$ by adding for every $1 \leq x, y \leq t - 1$, the edge $(x, y), (x + 1, y + 1)$, and making the vertex (t, t) adjacent to all vertices with $x \in \{1, t\}$ and $y \in \{1, t\}$.

Excluded Grid Theorem: Planar Graph

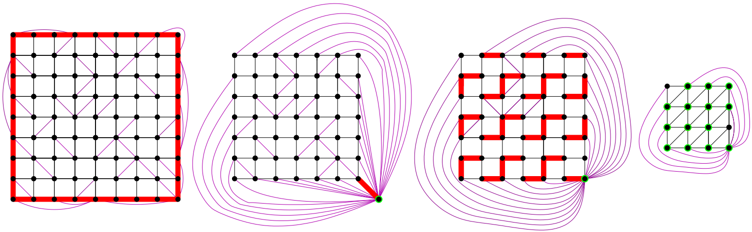
One more Excluded Grid Theorem, this time not for minors but just for edge contractions.

Theorem

For any connected planar graph G and integer $t \geq 0$, if **tw**$(G) \geq 9(t+1)$, *then G contains Γ_t as a contraction. Furthermore there exists a polynomial-time algorithm that given G either outputs a tree decomposition of G of width $9(t+1)$ or a set of edges whose contraction result in Γ_t.*



Proof sketch



Shifting Techniques

Locally bounded treewidth

For vertex v of a graph G and integer $r \geq 1$, we denote by G_v^r the subgraph of G induced by vertices within distance r from v in G.

Locally bounded treewidth

For vertex v of a graph G and integer $r \geq 1$, we denote by G_v^r the subgraph of G induced by vertices within distance r from v in G.

Lemma

Let G be a planar graph, $v \in V(G)$ and $r \geq 1$. Then **tw**$(G_v^r) \leq 18(r+1)$.

Proof.

Hint: use contraction-grid theorem. □

Locally bounded treewidth

For vertex v of a graph G and integer $r \geq 1$, we denote by G_v^r the subgraph of G induced by vertices within distance r from v in G.

Lemma

Let G be a planar graph, $v \in V(G)$ and $r \geq 1$. Then $\mathbf{tw}(G_v^r) \leq 18(r+1)$.

Proof.

Hint: use contraction-grid theorem. □

$18(r+1)$ in the above lemma can be made $3r+1$.

Locally bounded treewidth

Lemma

Let v be a vertex of a planar graph G and let $L = L_i \cup L_{i+1} \cup \cdots \cup L_{i+j}$ be j consecutive levels of BFS run from v. Then $\mathbf{tw}(L) \leq 3j + 1$.

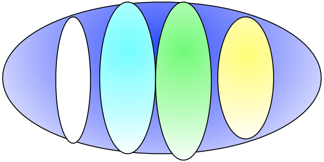
Proof.

□

Useful viewpoint

Lemma (Coloring Lemma)

Let G be a planar graph and k be an integer, $1 \leq k \leq |V(G)|$. Then the vertex set of G can be partitioned into k sets such that any $k-1$ of the sets induces a graph of treewidth at most $3k-2$. Moreover, such a partition can be found in linear time.



Useful viewpoint

Lemma (Coloring Lemma)

Let G be a planar graph and k be an integer, $1 \leq k \leq |V(G)|$. Then the vertex set of G can be partitioned into k sets such that any $k-1$ of the sets induce a graph of treewidth at most $3k-2$. Moreover, such a partition can be found in linear time.

Proof.

□

Example 1: FPT algorithm for Subgraph Isomorphism

Subgraph Isomorphism: given graphs H and G, find a copy of H in G as subgraph. Parameter $k := |V(H)|$.

Example 1: FPT algorithm for Subgraph Isomorphism

SUBGRAPH ISOMORPHISM: given graphs H and G, find a copy of H in G as subgraph. Parameter $k := |V(H)|$.

SUBGRAPH ISOMORPHISM can be solved on planar graphs in time $n^{O(k)}$ by brute-force plus graph isomorphism test.

Example 1: FPT algorithm for Subgraph Isomorphism

Subgraph Isomorphism: given graphs H and G, find a copy of H in G as subgraph. Parameter $k := |V(H)|$.

Subgraph Isomorphism can be solved on planar graphs in time $n^{O(k)}$ by brute-force plus graph isomorphism test.

Fact: There is $f(k,t) \cdot n$ time algorithm for Subgraph Isomorphism on graphs of treewidth t.

Example 1: FPT algorithm for Subgraph Isomorphism

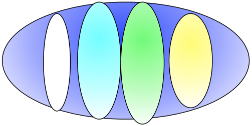
Subgraph Isomorphism: given graphs H and G, find a copy of H in G as subgraph. Parameter $k := |V(H)|$.

Subgraph Isomorphism can be solved on planar graphs in time $n^{O(k)}$ by brute-force plus graph isomorphism test.

Fact: There is $f(k,t) \cdot n$ time algorithm for Subgraph Isomorphism on graphs of treewidth t.

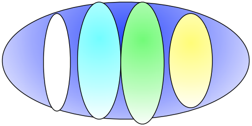
We want: $f(k) \cdot n$ time algorithm for Subgraph Isomorphism on planar graphs

Algorithm for SI



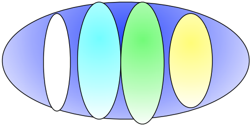
- Use Coloring Lemma with $k+1$ colors: $V(G) = X_1 \cup X_2 \cup \cdots \cup X_{k+1}$. For every $1 \leq i \leq k+1$, $\mathbf{tw}(G - X_i) \leq 3k$.

Algorithm for SI



- Use Coloring Lemma with $k+1$ colors: $V(G) = X_1 \cup X_2 \cup \cdots \cup X_{k+1}$. For every $1 \leq i \leq k+1$, **tw**$(G - X_i) \leq 3k$.
- If G contains k-vertex graph H as a subgraph, there is a color X_i such that $V(H) \cap X_i = \emptyset$.

Algorithm for SI



- Use Coloring Lemma with $k+1$ colors: $V(G) = X_1 \cup X_2 \cup \dots \cup X_{k+1}$. For every $1 \le i \le k+1$, **tw**$(G - X_i) \le 3k$.
- If G contains k-vertex graph H as a subgraph, there is a color X_i such that $V(H) \cap X_i = \emptyset$.
- For each $1 \le i \le k$, solve Subgraph Isomorphism for $G - X_i$ and H.

Example 2: PTAS for Independent Set

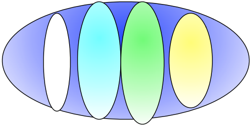
Independent Set: given graph G, find a maximum independent set in G.

Example 2: PTAS for Independent Set

INDEPENDENT SET: given graph G, find a maximum independent set in G.

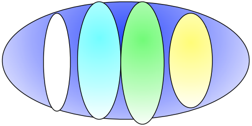
We want: An algorithm that for any $k \geq 1$ finds in time $O(2^{O(k)}n)$ an independent set of size at least $(1 - 1/k)OPT$ on planar graphs. In other words, an Efficient Polynomial Time Approximation Scheme (EPTAS) on planar graphs.

Algorithm for IS



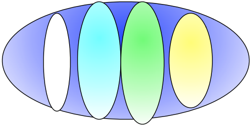
- Use Coloring Lemma with k colors:
  $V(G) = X_1 \cup X_2 \cup \cdots \cup X_k$. For every $1 \leq i \leq k$,
  $\mathbf{tw}(G - X_i) \leq 3k$.

Algorithm for IS



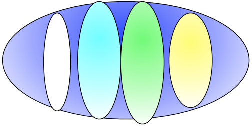
- Use Coloring Lemma with k colors: $V(G) = X_1 \cup X_2 \cup \cdots \cup X_k$. For every $1 \leq i \leq k$, $\mathbf{tw}(G - X_i) \leq 3k$.
- Let I be a maximum independent set in G. Then there is a color X_i such that $|I \cap X_i| \leq |I|/k$.

Algorithm for IS



- Use Coloring Lemma with k colors: $V(G) = X_1 \cup X_2 \cup \cdots \cup X_k$. For every $1 \leq i \leq k$, $\mathbf{tw}(G - X_i) \leq 3k$.
- Let I be a maximum independent set in G. Then there is a color X_i such that $|I \cap X_i| \leq |I|/k$.
- For each $1 \leq i \leq k$, solve Independent Set for $G - X_i$.

Algorithm for IS



- Use Coloring Lemma with k colors:
  $V(G) = X_1 \cup X_2 \cup \cdots \cup X_k$. For every $1 \leq i \leq k$,
  $\mathbf{tw}(G - X_i) \leq 3k$.
- Let I be a maximum independent set in G. Then there is a color X_i such that $|I \cap X_i| \leq |I|/k$.
- For each $1 \leq i \leq k$, solve Independent Set for $G - X_i$.
- The size of the maximum set we found is at least $|I| - |I|/k$.

Example 2: PTAS for Independent Set

Independent Set: given graph G, find a maximum independent set in G.

Example 2: PTAS for Independent Set

INDEPENDENT SET: given graph G, find a maximum independent set in G.

We want: An algorithm that for any $k \geq 1$ finds in time $O(2^{O(k)}n)$ an independent set of size at least $(1 - 1/k)OPT$ on planar graphs. In other words, an Efficient Polynomial Time Approximation Scheme (EPTAS) on planar graphs.

Example 3: Subexponential parameterized algorithm for Vertex Cover

Vertex Cover: given graph G and integer k. Decide whether G contains a vertex cover of size at most k.

Example 3: Subexponential parameterized algorithm for VERTEX COVER

VERTEX COVER: given graph G and integer k. Decide whether G contains a vertex cover of size at most k.

We want: An algorithm that solves time VERTEX COVER in time $O(2^{O(\sqrt{k})}n)$ on planar graphs.

Reminder: Grid Theorem

Theorem (Planar Excluded Grid Theorem)

Let $t \geq 0$ be an integer. Every planar graph G of treewidth at least $\frac{9}{2}t$, contains $\boxplus_t$ as a minor. Furthermore, there exists a polynomial-time algorithm that for a given planar graph G either outputs a tree decomposition of G of width $\frac{9}{2}t$ or constructs a minor model of $\boxplus_t$ in G.

Subexponential treewidth

Theorem

The treewidth of an n-vertex planar graph is $\mathcal{O}(\sqrt{n})$

Proof.

□

Subexponential treewidth: refinement

We want to prove

Theorem

If a planar graph G contains a vertex cover of size k, then the treewidth of G is $\mathcal{O}(\sqrt{k})$

Subexponential treewidth: refinement

We want to prove

Theorem

If a planar graph G contains a vertex cover of size k, then the treewidth of G is $\mathcal{O}(\sqrt{k})$

If we prove it, we are done with $O(2^{O(\sqrt{k})}n)$-time algorithm. Why?

Subexponential treewidth: refinement

We want to prove

Theorem

If a planar graph G contains a vertex cover of size k, then the treewidth of G is $\mathcal{O}(\sqrt{k})$

If we prove it, we are done with $O(2^{O(\sqrt{k})}n)$-time algorithm. Why?
Given a tree decomposition of width t of G, we solve Vertex Cover
In time $2^t \cdot t^{\mathcal{O}(1)} \cdot n$.

Vertex Cover

Some questions to ask

(i) How large can be the vertex cover of $\boxplus_t$?

Vertex Cover

Some questions to ask

(i) How large can be the vertex cover of $\boxplus_t$? $\boxplus_t$ contains a matching of size $t^2/2$, and thus vertex cover of $\boxplus_t$ is at least $t^2/2$.

Vertex Cover

Some questions to ask

(i) How large can be the vertex cover of $\boxplus_t$? $\boxplus_t$ contains a matching of size $t^2/2$, and thus vertex cover of $\boxplus_t$ is at least $t^2/2$.

(ii) Is Vertex Cover minor-closed?

Vertex Cover

Some questions to ask

(i) How large can be the vertex cover of $\boxplus_t$? $\boxplus_t$ contains a matching of size $t^2/2$, and thus vertex cover of $\boxplus_t$ is at least $t^2/2$.

(ii) Is Vertex Cover minor-closed?YES!

Vertex Cover

Some questions to ask

(i) How large can be the vertex cover of $\boxplus_t$? $\boxplus_t$ contains a matching of size $t^2/2$, and thus vertex cover of $\boxplus_t$ is at least $t^2/2$.

(ii) Is Vertex Cover minor-closed?YES!

$(i) + (ii)$ give

Theorem

If a planar graph G contains a vertex cover of size k, then the treewidth of G is $\mathcal{O}(\sqrt{k})$

What is special in Vertex Cover?

Theorem

If a planar graph G contains a feedback vertex set of size k, then the treewidth of G is $\mathcal{O}(\sqrt{k})$

What is special in Vertex Cover?

Theorem

If a planar graph G contains a feedback vertex set of size k, then the treewidth of G is $\mathcal{O}(\sqrt{k})$

Theorem

If the treewidth of a planar graph$G$ is more than $c \cdot \sqrt{k}$ for some c, then G contains a path on k vertices

What is special in Vertex Cover?

Same strategy should work for any problem if

- $(P1)$ The size of any solution in $\boxplus_t$ is of order $\Omega(t^2)$.
- $(P2)$ On graphs of treewidth t, the problem is solvable in time $2^{\mathcal{O}(t)} \cdot n^{\mathcal{O}(1)}$.
- $(P3)$ The problem is minor-closed, i.e. if G has a solution of size k, then every minor of G also has a solution of size k.

This settles Feedback Vertex Set and k-path. Why not Dominating Set?

Reminder: Contracting to a grid

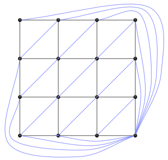


Figure : A triangulated grid Γ_4.

Theorem

For any connected planar graph G and integer $t \geq 0$, if **tw**$(G) \geq 9(t+1)$*, then G contains Γ_t as a contraction. Furthermore there exists a polynomial-time algorithm that given G either outputs a tree decomposition of G of width $9(t+1)$ or a set of edges whose contraction result in Γ_t.*

Strategy for Dominating Set

Same strategy should work for any problem with:

(i) The size of any solution in Γ_t is of order $\Omega(t^2)$.

(ii) The problem is contraction-closed, i.e. if G has a solution of size k, then every minor of G also has a solution of size k.

This settles Dominating Set

Theorem

If a planar graph G contains a dominating set of size k, then the treewidth of G is $\mathcal{O}(\sqrt{k})$

Lets try to formalize

Restrict to vertex-subset problems.
Let ϕ be a computable function which takes as an input graph G, a set $S \subseteq V(G)$ and outputs **true** or **false**.
For an example, for Dominating Set: $\phi(G, S) = \mathbf{true}$ if and only if $N[S] = V(G)$.

Bidimensionality

Definition (Bidimensional problem)

A vertex subset problem Π is *bidimensional* if it is contraction-closed, and there exists a constant $c > 0$ such that $OPT_{\Pi}(\Gamma_k) \geq ck^2$.

Bidimensionality

Definition (Bidimensional problem)

A vertex subset problem Π is *bidimensional* if it is contraction-closed, and there exists a constant $c > 0$ such that $OPT_{\Pi}(\Gamma_k) \geq ck^2$.

Vertex Cover, Independent Set, Feedback Vertex Set, Induced Matching, Cycle Packing, Scattered Set for fixed value of d, k-Path, k-cycle, Dominating Set, Connected Dominating Set, Cycle Packing, r-Center...

Bidimensionality

Definition (Bidimensional problem)

A vertex subset problem Π is *bidimensional* if it is contraction-closed, and there exists a constant $c > 0$ such that $OPT_{\Pi}(\Gamma_k) \geq ck^2$.

Lemma (Parameter-Treewidth Bound)

Let Π be a bidimensional problem. Then there exists a constant α_{Π} such that for any connected planar graph G, $\textbf{tw}(G) \leq \alpha_{\Pi} \cdot \sqrt{OPT_{\Pi}(G)}$. Furthermore, there exists a polynomial time algorithm that for a given G constructs a tree decomposition of G of width at most $\alpha_{\Pi} \cdot \sqrt{OPT_{\Pi}(G)}$.

Bidimensionality: Summing up

Theorem

Let Π be a bidimensional problem such that there exists an algorithm for Π with running time $2^{O(t)}n^{O(1)}$ when a tree decomposition of the input graph G of width t is given. Then Π is solvable in time $2^{O(\sqrt{k})}n^{O(1)}$ on connected planar graphs.

Bidimensionality: Remarks

- Polynomial dependence on n can be turned into linear, so all bidimensionality based algorithms run in time $2^{\mathcal{O}(\sqrt{k})}n$.

Bidimensionality: Remarks

- Polynomial dependence on n can be turned into linear, so all bidimensionality based algorithms run in time $2^{\mathcal{O}(\sqrt{k})}n$.
- Is it possible to obtain $2^{o(\sqrt{k})}n^{\mathcal{O}(1)}$ running time for problems on planar graphs? (NO, unless ETH fails)

Bidimensionality: Remarks

- Polynomial dependence on n can be turned into linear, so all bidimensionality based algorithms run in time $2^{\mathcal{O}(\sqrt{k})}n$.
- Is it possible to obtain $2^{o(\sqrt{k})}n^{\mathcal{O}(1)}$ running time for problems on planar graphs? (NO, unless ETH fails)
- Planarity is used only to exclude a grid. Thus all the arguments extend to classes of graphs with a similar property.

Bidimensionality: Remarks

- Polynomial dependence on n can be turned into linear, so all bidimensionality based algorithms run in time $2^{\mathcal{O}(\sqrt{k})}n$.
- Is it possible to obtain $2^{o(\sqrt{k})}n^{\mathcal{O}(1)}$ running time for problems on planar graphs? (NO, unless ETH fails)
- Planarity is used only to exclude a grid. Thus all the arguments extend to classes of graphs with a similar property.
- Bidimensionality+Separability+$\mathbf{MSO}_2$ brings to Linear kernelization on apex-minor-free graphs. For minor-closed problems to minor-free graphs.

Something to take home

- What works on trees (usually) works on graphs of small treewidth
- Excluding a grid is often helpful and can bring to various WIN/WIN scenarios