FEDOR V. FOMIN

Lectures on treewidth

The Parameterized
Complexity Summer School

1-3 September 2017
Vienna, Austria

UNIVERSITAS BERGENSIS

Why treewidth?

Very general idea in science: large structures can be understood by breaking them into small pieces

Why treewidth?

Very general idea in science: large structures can be understood by breaking them into small pieces

In Computer Science: divide and conquer; dynamic programming

Why treewidth?

Very general idea in science: large structures can be understood by breaking them into small pieces

In Computer Science: divide and conquer; dynamic programming

In Graph Algorithms: Exploiting small separators

Why treewidth?

Very convenient to decompose a graph via small separations + Obstacles for decompositions = Powerful tool

Plan

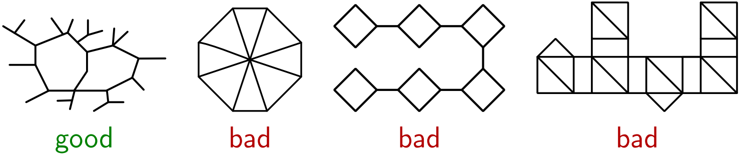
- Treewidth
- Applications on planar graphs

Plan

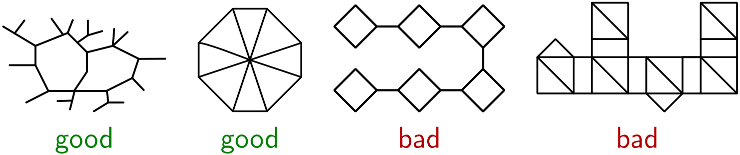
- Treewidth
- Applications on planar graphs
- FPT, bidimensionality, kernelization

What is a tree-like graph?

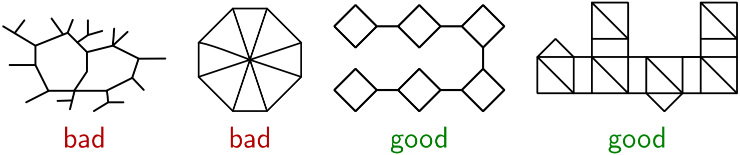
1. Number of cycles is bounded.



2. Removing a bounded number of vertices makes it acyclic.



3. Bounded-size parts connected in a tree-like way.



Tree Decomposition: canonical definition

A *tree decomposition* of a graph G is a pair $\mathcal{T} = (T, \chi)$, where T is a tree and mapping χ assigns to every node t of T a vertex subset X_t (called a bag) such that

Tree Decomposition: canonical definition

A *tree decomposition* of a graph G is a pair $\mathcal{T} = (T, \chi)$, where T is a tree and mapping χ assigns to every node t of T a vertex subset X_t (called a bag) such that

(T1) $\bigcup_{t \in V(T)} X_t = V(G)$.

(T2) For every $vw \in E(G)$, there exists a node t of T such that bag $\chi(t) = X_t$ contains both v and w.

(T3) For every $v \in V(G)$, the set $\chi^{-1}(v)$, i.e. the set of nodes $T_v = \{t \in V(T) \mid v \in X_t\}$ forms a connected subgraph (subtree) of T.

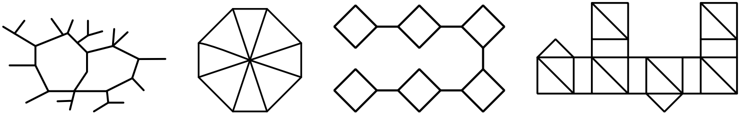
The *width* of tree decomposition $\mathcal{T} = (T, \chi)$ equals $\max_{t \in V(T)} |X_t| - 1$, i.e the maximum size of its bag minus one. The *treewidth* of a graph G is the minimum width of a tree decomposition of G.

Treewidth Applications

- Graph Minors
- Parameterized Algorithms
- Exact Algorithms
- Approximation Schemes
- Kernelization
- Databases
- CSP's
- Bayesian Networks
- AI
- ...

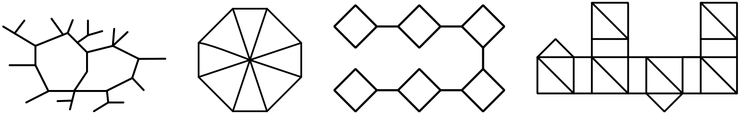
Exercise: What are the widths of these graphs?

1. Number of cycles is bounded.



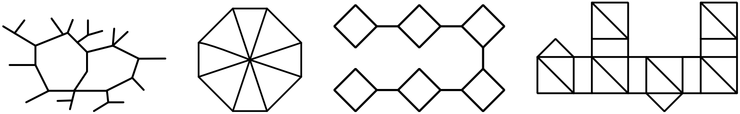
good bad bad bad

2. Removing a bounded number of vertices makes it acyclic.



good good bad bad

3. Bounded-size parts connected in a tree-like way.



bad bad good good

Treewidth

- Discovered and rediscovered many times: Halin 1976, Bertelé and Brioschi, 1972
- In the 80's as a part of Robertson and Seymour's Graph Minors project.
- Arnborg and Proskurowski: algorithms

Fact

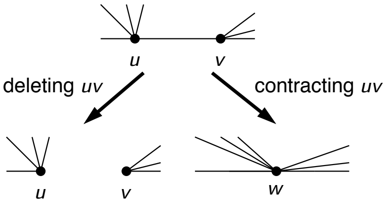
Independent Set, Dominating Set, q-Coloring, Max-Cut, Odd Cycle Transversal, Hamiltonian Cycle, Partition into Triangles, Feedback Vertex Set, Vertex Disjoint Cycle Packing and million other problems are FPT parameterized by the treewidth.

Meta-theorem for treewidth DP

While arguments for each of the problems are different, there are a lot of things in common...

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.

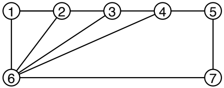


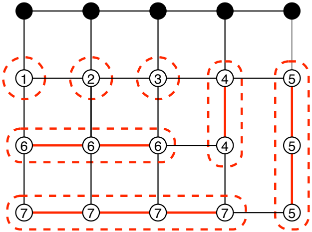
Example: A triangle is a minor of a graph G if and only if G has a cycle (i.e., it is not a forest).

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 $\phi(v)$.





Minor closed properties

Definition: A set $\mathcal{G}$ of graphs is **minor closed** if whenever $G \in \mathcal{G}$ and $H \leq G$, then $H \in \mathcal{G}$ as well.

Examples of minor closed properties:

planar graphs
acyclic graphs (forests)
graphs having no cycle longer than k
empty graphs

Examples of not minor closed properties:

complete graphs
regular graphs
bipartite graphs

Applications of treewidth

In parameterized algorithms various modifications of WIN/WIN approach: either treewidth is small, and we solve the problem, or something good happens

- Finding a path of length $\geq k$ is FPT because every graph with treewidth k contains a k-path

Applications of treewidth

In parameterized algorithms various modifications of WIN/WIN approach: either treewidth is small, and we solve the problem, or something good happens

- Finding a path of length $\geq k$ is FPT because every graph with treewidth k contains a k-path
- Feedback vertex set is FPT because if the treewidth is more than k, the answer is NO.

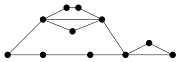
Applications of treewidth

In parameterized algorithms various modifications of WIN/WIN approach: either treewidth is small, and we solve the problem, or something good happens

- Finding a path of length $\geq k$ is FPT because every graph with treewidth k contains a k-path
- Feedback vertex set is FPT because if the treewidth is more than k, the answer is NO.
- Disjoint Path problem is FPT because if the treewidth is $\geq f(k)$, then the graph contains irrelevant vertex (non-trivial arguments)

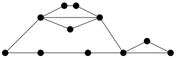
Properties of treewidth

Fact: treewidth ≤ 2 if and only if graph is subgraph of a series-parallel graph

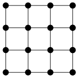


Properties of treewidth

Fact: treewidth ≤ 2 if and only if graph is subgraph of a series-parallel graph

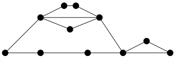


Fact: For every $k \geq 2$, the treewidth of the $k \times k$ grid is exactly k.

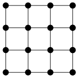


Properties of treewidth

Fact: treewidth ≤ 2 if and only if graph is subgraph of a series-parallel graph



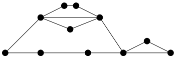
Fact: For every $k \geq 2$, the treewidth of the $k \times k$ grid is exactly k.



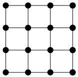
Fact: Treewidth does not increase if we delete edges, delete vertices, or contract edges.

$\Longrightarrow$ If F is a minor of G, then the treewidth of F is at most the treewidth of G.

Properties of treewidth



Fact: treewidth ≤ 2 if and only if graph is subgraph of a series-parallel graph



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

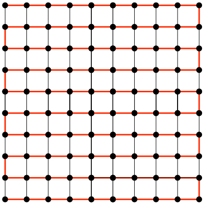
Fact: Treewidth does not increase if we delete edges, delete vertices, or contract edges.

$\Longrightarrow$ If F is a minor of G, then the treewidth of F is at most the treewidth of G.

The treewidth of the k-clique is $k - 1$.

Obstruction to Treewidth

Extremely useful obstructions to small treewidth are grid-minors. Let t be a positive integer. The $t \times t$*-grid* $\boxplus_t$ is a graph with vertex set $\{(x, y) \mid x, y \in \{1, 2, \ldots, t\}\}$. Thus $\boxplus_t$ has exactly t^2 vertices. Two different vertices (x, y) and (x', y') are adjacent if and only if $|x - x'| + |y - y'| \leq 1$.



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

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.

Excluded Grid Theorem

Theorem (Excluded Grid Theorem, Robertson, Seymour and Thomas, 1994)

If the treewidth of G *is at least* $k^{4t^2(t+2)}$*, then* G *has* $\boxplus_t$ *as a minor.*

Excluded Grid Theorem

Theorem (Excluded Grid Theorem, Robertson, Seymour and Thomas, 1994)

If the treewidth of G is at least $k^{4t^2(t+2)}$, then G has $\boxplus_t$ as a minor.

It was open for many years whether a polynomial relationship could be established between the treewidth of a graph G and the size of its largest grid minor.

Excluded Grid Theorem

Theorem (Excluded Grid Theorem, Robertson, Seymour and Thomas, 1994)

If the treewidth of G is at least $k^{4t^2(t+2)}$, then G has $\boxplus_t$ as a minor.

It was open for many years whether a polynomial relationship could be established between the treewidth of a graph G and the size of its largest grid minor.

Theorem (Excluded Grid Theorem, Chekuri and Chuzhoy, 2013, Chuzhoy 2016)

Let $t \geq 0$ be an integer. There exists a universal constant c, such that every graph of treewidth at least $c \cdot t^{19}$ contains $\boxplus_t$ as a minor.

Excluded Grid Theorem A : Planar Graph

Our set of treewidth applications on planar graphs 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

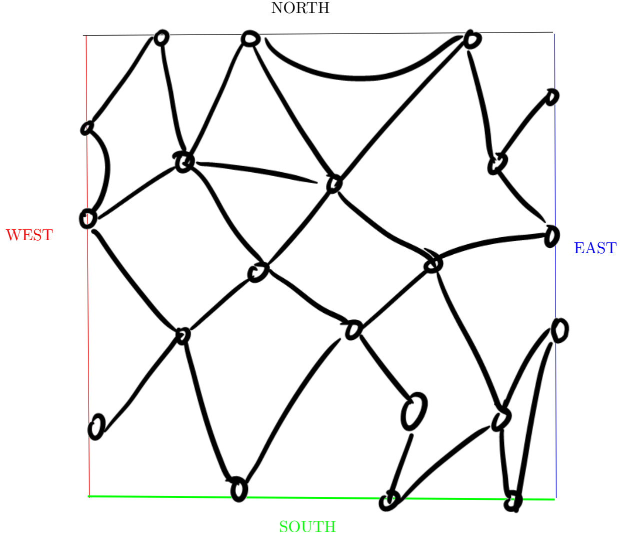
The proof is based on Menger's Theorem

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.

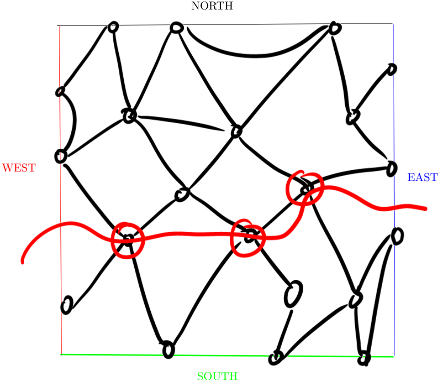
Grid Theorem: Sketch of the proof

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



Grid Theorem: Sketch of the proof

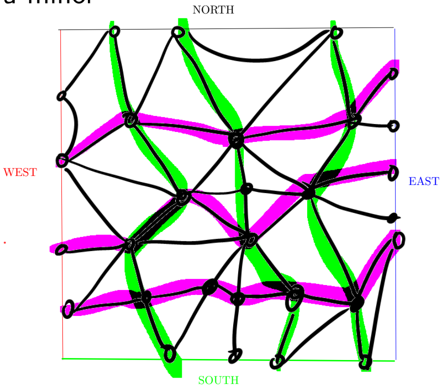
Either East can be separated from West, or South from North by



removing at most ℓ vertices

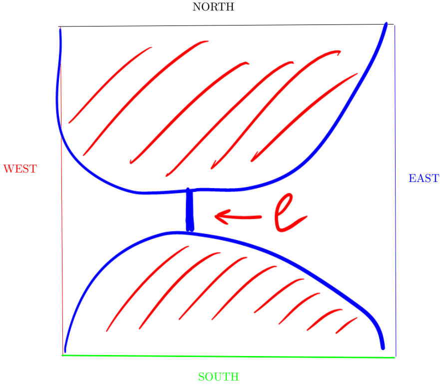
Grid Theorem: Sketch of the proof

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.



Grid Theorem: Sketch of the proof

"At this point we have reached a degree of handwaving so exuberant, one may fear we are about to fly away. Surprisingly, this handwaving has a completely formal theorem behind it."

(Ryan Williams 2011, SIGACT News)

Excluded Grid Theorem: Planar Graphs

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

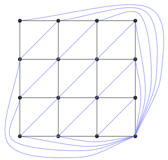


Figure: A triangulated grid Γ_4.

Excluded Grid Theorem: Planar Graphs

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

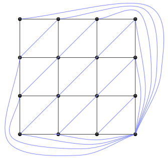


Figure: A triangulated grid Γ_4.

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 Graphs

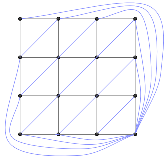
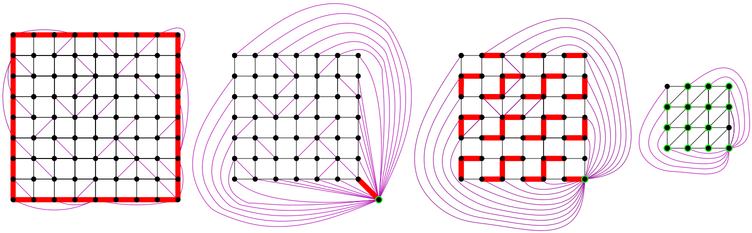


Figure: A triangulated grid Γ_4.

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 results 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 $\mathbf{tw}(G_v^r) \leq 18(r+1)$.

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 $\textbf{tw}(G_v^r) \leq 18(r+1)$.

Proof.

Hint: use contraction-grid theorem. □

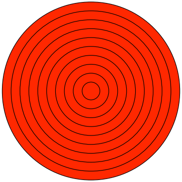
$18(r+1)$ in the above lemma can be made $3r+1$.

Locally bounded treewidth

Lemma

Let v be a vertex of a planar graph G and let L_i, be the vertices of G at distance i, $0 \leq i \leq n$, from v. Then for any $i, j \geq 0$, the treewidth of the subgraph $G_{i,i+j}$ induced by vertices in $L_i \cup L_{i+1} \cup \cdots \cup L_{i+j}$ does not exceed $3j + 1$.

Proof.



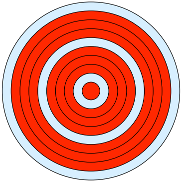
□

Locally bounded treewidth

Lemma

Let v be a vertex of a planar graph G and let L_i, be the vertices of G at distance i, $0 \leq i \leq n$, from v. Then for any $i, j \geq 0$, the treewidth of the subgraph $G_{i,i+j}$ induced by vertices in $L_i \cup L_{i+1} \cup \cdots \cup L_{i+j}$ does not exceed $3j + 1$.

Proof.



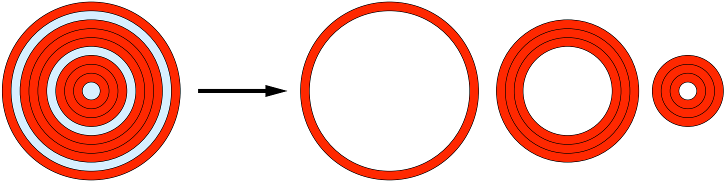
□

Locally bounded treewidth

Lemma

Let v be a vertex of a planar graph G and let L_i, be the vertices of G at distance i, $0 \leq i \leq n$, from v. Then for any $i, j \geq 0$, the treewidth of the subgraph $G_{i,i+j}$ induced by vertices in $L_i \cup L_{i+1} \cup \cdots \cup L_{i+j}$ does not exceed $3j + 1$.

Proof.



□

Intuition

The idea behind the shifting technique is as follows:

- Pick a vertex v of planar graph G and run breadth-first search (BFS) from v.
- For any $i, j \geq 0$, the treewidth of the subgraph $G_{i,i+j}$ induced by vertices in levels $i, i+1, \ldots, i+j$ of BFS does not exceed $3j+1$.
- Now for an appropriate choice of parameters, we can find a "shift" of "windows", i.e. a disjoint set of a small number of consecutive levels of BFS, "covering" the solution. Because every window is of small treewidth, we can employ the dynamic programing or the power of Courcelle's theorem to solve the problem.

Intuition

The idea behind the shifting technique is as follows:

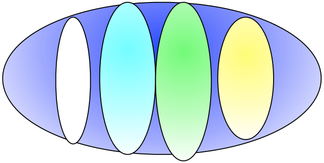
- Pick a vertex v of planar graph G and run breadth-first search (BFS) from v.
- For any $i, j \geq 0$, the treewidth of the subgraph $G_{i,i+j}$ induced by vertices in levels $i, i+1, \dots, i+j$ of BFS does not exceed $3j+1$.
- Now for an appropriate choice of parameters, we can find a "shift" of "windows", i.e. a disjoint set of a small number of consecutive levels of BFS, "covering" the solution. Because every window is of small treewidth, we can employ the dynamic programing or the power of Courcelle's theorem to solve the problem.

We will see two examples.

Useful viewpoint

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 polynomial time.

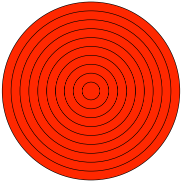


Useful viewpoint

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 polynomial time.

Proof.



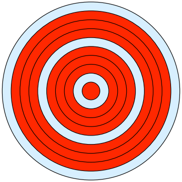
□

Useful viewpoint

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 polynomial time.

Proof.



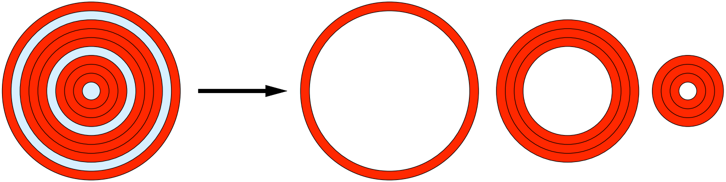
□

Useful viewpoint

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 polynomial time.

Proof.



□

Example 1: Subgraph Isomorphism

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

Example 1: Subgraph Isomorphism

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

$\mathbf{MSO_2}$ formula of size $k^{O(1)}$ for Subgraph Isomorphism exists.

Example 1: Subgraph Isomorphism

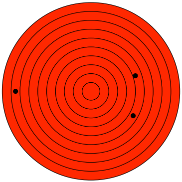
Subgraph Isomorphism: given graphs H and G, find a copy of H in G as subgraph. Parameter $k := |V(H)|$.

$\mathbf{MSO}_2$ formula of size $k^{O(1)}$ for Subgraph Isomorphism exists.

Courcelle's Theorem implies that we have $f(k,t) \cdot n$ time algorithm for Subgraph Isomorphism on graphs of treewidth t.

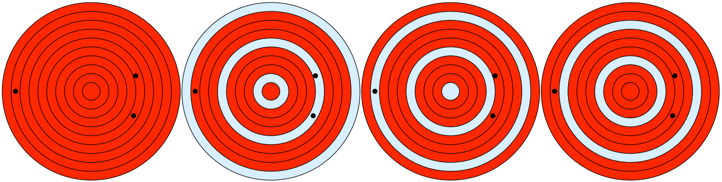
Example 1: Subgraph Isomorphism

- Partition the vertex set of G into $k+1$ sets $S_0 \cup \cdots \cup S_k$ such that for every $i \in \{0, \ldots, k\}$, graph $G - S_i$ is of treewidth at most $3k+1$.



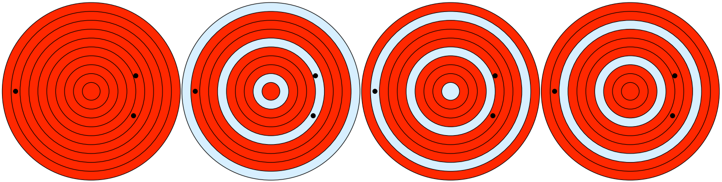
Example 1: Subgraph Isomorphism

- Partition the vertex set of G into $k+1$ sets $S_0 \cup \cdots \cup S_k$ such that for every $i \in \{0, \ldots, k\}$, graph $G - S_i$ is of treewidth at most $3k+1$.
- For every k-vertex subset X of G, there is $i \in \{0, \ldots, k\}$ such that $X \cap S_i = \emptyset$. Therefore, if G contains H as a subgraph, then for at least one value of i, $G - S_i$ also contains H.



Example 1: Subgraph Isomorphism

- Partition the vertex set of G into $k+1$ sets $S_0 \cup \cdots \cup S_k$ such that for every $i \in \{0, \ldots, k\}$, graph $G - S_i$ is of treewidth at most $3k+1$.
- For every k-vertex subset X of G, there is $i \in \{0, \ldots, k\}$ such that $X \cap S_i = \emptyset$. Therefore, if G contains H as a subgraph, then for at least one value of i, $G - S_i$ also contains H.
- It means that by trying each of the graphs $G - S_i$ for each $i \in \{0, \ldots, k\}$, we find a copy of H in G if there is one.



Example: Subgraph Isomorphism

Theorem

SUBGRAPH ISOMORPHISM *on planar graphs is FPT parameterized by* $|V(H)|$.

Example: Subgraph Isomorphism

For every $k \geq 1$ every planar graph G has a family $\mathcal{F}$ of polynomial size such that

- The treewidth of every graph in the family is $\mathcal{O}(k)$, and
- Every k-vertex subgraph H of G is a subgraph of some graph in this family.

Note: For Subgraph Isomorphism it would be much more interesting to have a k-graph covering family $\mathcal{F}$ of size $2^{o(k)}n^{\mathcal{O}(1)}$ of graphs of treewidth $o(k)$.

Further reading: Fedor V. Fomin, Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michal Pilipczuk, Saket Saurabh: Subexponential parameterized algorithms for planar and apex-minor-free graphs via low treewidth pattern covering. CoRR abs/1604.05999 (2016)

Shifting technique: history

- Originated as a tool for obtaining PTAS. The basic idea due to Baker (1994)
- Eppstein: the notion of local treewidth (1995)
- Grohe: extending to H-minor-free graphs (2003)
- Demaine, Hajiaghayi, and Kawarabayashi contractions on H-minor-free graphs (2005).

Bidimensionality

Bidimensionality

Subexponential algorithms, EPTAS, kernels on planar, bounded genus, H-minor free 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.

Lipton-Tarjan Theorem

Corollary

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

Vertex Cover on planar graphs. Just three questions

Does a planar graph contains a vertex cover of size at most k?

- Vertex Cover has a kernel with at most $2k$ vertices which is an induced subgraph of the input graph. Thus when the input graph is planar we obtain in polynomial time an equivalent planar instance of size at most $2k$.

Vertex Cover on planar graphs. Just three questions

Does a planar graph contains a vertex cover of size at most k?

- Vertex Cover has a kernel with at most $2k$ vertices which is an induced subgraph of the input graph. Thus when the input graph is planar we obtain in polynomial time an equivalent planar instance of size at most $2k$.
- Find a tree decomposition of width $t = \mathcal{O}(\sqrt{k})$ of the kernel.
- Dynamic programming solves Vertex Cover in time $2^t n^{\mathcal{O}(1)} = 2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$

Other problems on Planar Graphs

What about other problems like INDEPENDENT SET, FEEDBACK VERTEX SET, DOMINATING SET or k-PATH?

Other problems on Planar Graphs

What about other problems like Independent Set, Feedback Vertex Set, Dominating Set or k-path?

- For most of the problems, obtaining a kernel is not that easy, and
- For some like k-Path, we know that no polynomial kernel exists (of course unless ....)

Concrete plan

Solve Dominating Set in time $\mathcal{O}(2^{\mathcal{O}(\sqrt{k})}n)$ on planar graphs.

Vertex Cover, one more try

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

Vertex Cover, one more try

(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, one more try

(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) Given a tree decomposition of width t of G, how fast can we solve Vertex Cover ?

Vertex Cover, one more try

(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) Given a tree decomposition of width t of G, how fast can we solve Vertex Cover ? In time $2^t \cdot t^{\mathcal{O}(1)} \cdot n$.

(iii) Is Vertex Cover minor-closed?

Vertex Cover, one more try

(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) Given a tree decomposition of width t of G, how fast can we solve Vertex Cover ? In time $2^t \cdot t^{\mathcal{O}(1)} \cdot n$.

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

Vertex Cover, one more try

(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) Given a tree decomposition of width t of G, how fast can we solve Vertex Cover ? In time $2^t \cdot t^{\mathcal{O}(1)} \cdot n$.

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

$(i) + (ii) + (iii)$ give $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$-time algorithm for Vertex Cover on planar graphs.

Vertex Cover, one more try

$(i) + (ii) + (iii)$ gives $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$-time algorithm for Vertex Cover on planar graphs.

(i) Compute the treewidth of G. If it is more than $c\sqrt{k}$—say NO. (It contains $\boxplus_{2\sqrt{k}}$ as a minor...)

Vertex Cover, one more try

$(i) + (ii) + (iii)$ gives $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$-time algorithm for VERTEX COVER on planar graphs.

(i) Compute the treewidth of G. If it is more than $c\sqrt{k}$—say NO. (It contains $\boxplus_{2\sqrt{k}}$ as a minor...)

(ii) If the treewidth is less than $c\sqrt{k}$, do DP.

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

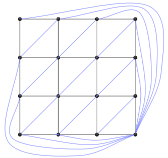


Figure: A triangulated grid Γ_4.

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:

($P1$) The size of any solution in Γ_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 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

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 $N[S] = V(G)$.

Lets try to formalize

Definition

For function ϕ, we define *vertex-subset problem* Π as a parameterized problem, where input is a graph G and an integer k, the parameter is k.

For *maximization* problem, the task is to decide whether there is a set $S \subseteq V(G)$ such that $|S| \geq k$ and $\phi(G, S) = \mathbf{true}$. Similarly, for *minimization* problem, we are looking for a set $S \subseteq V(G)$ such that $|S| \leq k$ and $\phi(G, S) = \mathbf{true}$.

Optimization problem

Definition

For a vertex-subset minimization problem Π,

$$OPT_{\Pi}(G) = \min\{k \mid (G, k) \in \Pi\}.$$

If there is no k such that $(G, k) \in \Pi$, we put $OPT_{\Pi}(G) = +\infty$.
For a vertex-subset maximization problem Π,

$$OPT_{\Pi}(G) = \max\{k \mid (G, k) \in \Pi\}.$$

If no k such that $(G, k) \in \Pi$ exists, then $OPT_{\Pi}(G) = -\infty$.

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, $\mathbf{tw}(G) \leq \alpha_{\Pi} \cdot \sqrt{OPT_{\Pi}(G)}$. 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.