

Treewidth & Planarity

Want to solve NP-hard problems faster

E.g. maximum independent set

Idea 1: Let's approximate solution

"Inapproximable": no $\frac{1}{n^\epsilon}$ -apx for some constant $\epsilon > 0$

Idea 2: Restrict the input graph somehow

Disallow "lower bound graphs"

This lecture: bounded treewidth graphs, planar graphs

Thm: can solve MaxIS in time $2^{O(k)} \text{poly}(n)$ on graphs of treewidth k

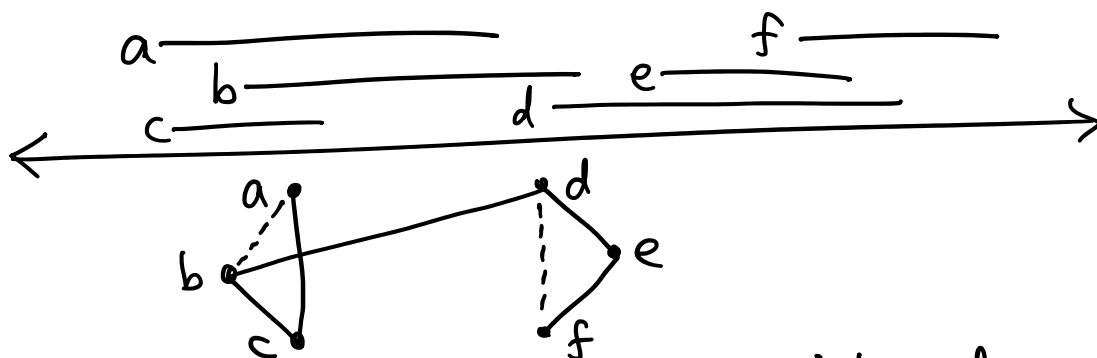
Thm: can $(1-\epsilon)$ -approximate MaxIS in time $2^{O(1/\epsilon)} \text{poly}(n)$ on planar graphs

(Exact is still NP-hard on planar graphs)

Pathwidth

- Simpler than treewidth, but contains all the main ideas

A pathwidth- k graph is constructed as follows:



- Every vertex in graph is an interval on the real line

$\cap \text{ of } k+1 \text{ intervals is inside } \leq k+1$

real line

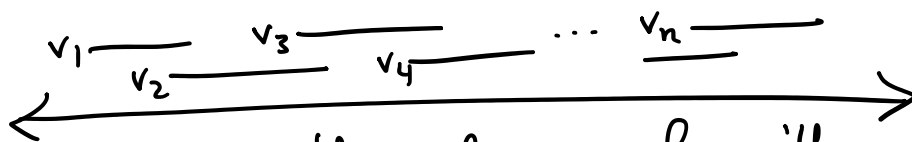
- Every point in real line is inside $\leq k+1$ intervals
- Can only add edge between vertices whose intervals intersect

Def: the class of pathwidth-k graphs is all graphs that can be constructed this way

Def: a graph has pathwidth k if k is the smallest integer s.t. it's in the class of pathwidth-k graphs

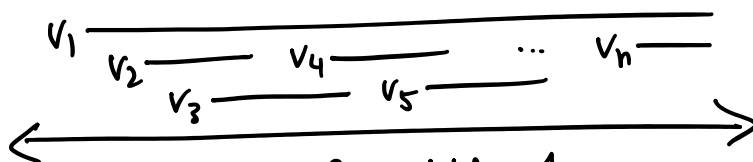
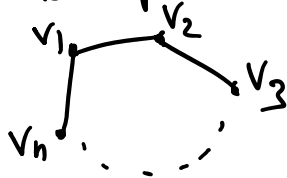
Obs: paths have pathwidth 1.

Pf: 

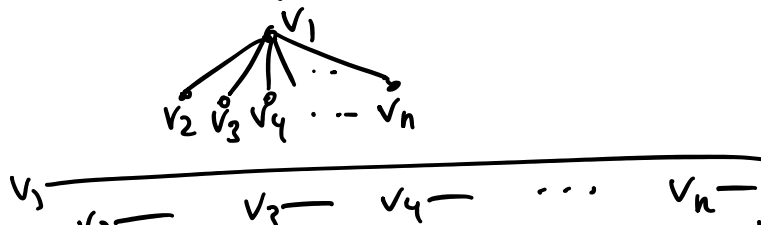


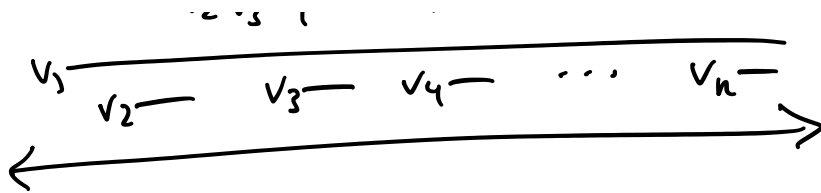
~~Obs: paths are the only graphs with pathwidth 1. (incorrect)~~
(This is why the "off by 1" in " $k+1$ " here)

Obs: cycle has pathwidth 2.



Obs: star has pathwidth 1.





Prop: A pathwidth- k graph does not contain a $(k+2)$ -clique.

Pf: Let G have a $(k+2)$ -clique K . Suppose for contradiction that G can be constructed as pathwidth k .

The intervals corresponding to vertices in K must pairwise intersect

Claim: if a collection of intervals pairwise intersect, then they contain a common endpoint.

Therefore, $\exists$ point in all $k+2$ intervals
This violates pathwidth- k requirement ~~$\neq$~~ .

"Discrete" view of pathwidth:

path: $(v_1) - (v_1, v_2) - (v_2) - (v_2, v_3) - \dots - (v_{n-1}, v_n) - (v_n)$

cycle: $(v_1) - (v_1, v_2) - (v_1, v_2, v_3) - (v_1, v_3) - (v_1, v_3, v_4) - \dots$

Def (Path decomposition).

A path decomposition of G is a sequence

$(X_1, X_2, \dots)$ of bags $X_i \subseteq V$ s.t.

① Every vertex in G belongs to at least one bag.

② For every edge in G , there is a bag containing

- ① Every vertex in G is in some bag.
- ② For each edge in G , there is a bag containing both of its endpoints.
- ③ For each vertex v , the set of bags containing v forms a connected interval in the sequence.

Equivalently: If $u \in X_i$ and $u \in X_k$ for some $i \leq k$, then $u \in X_j$ for all $j \in [i, k]$.

The width of the path decomp is the maximum size of a bag X_i .

Def: the pathwidth of G is the smallest w s.t. exists path decomp of width w , minus 1.

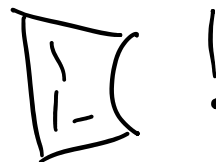
Thm: Can solve MaxIS in time $2^{O(k)} \text{poly}(n)$, given a pathwidth- k decomposition.

Algo: DP with the following insight:



Suppose computed MaxIS inside here.

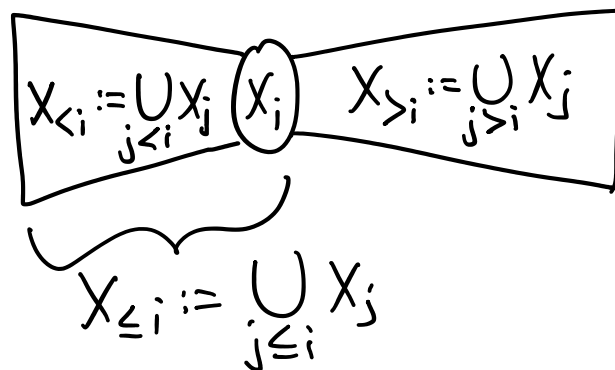
Idea: can "forget" all of



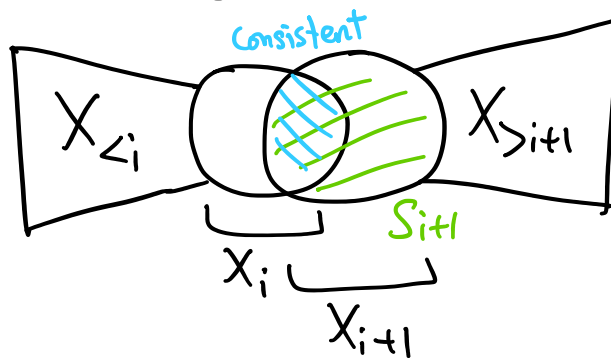
DP only on states $\bigcirc$, which are the bags X_i

bags X_i

$DP(X_i, S_i)$ $S_i \subseteq X_i$ independent set in $G[X_i]$



Want: $DP(X_i, S_i)$ is size of max IS
 S in $G[X_{\leq i}]$ s.t. $S \cap X_i = S_i$



$$DP(X_{i+1}, S_{i+1}) = \max_{\substack{S_i \text{ consistent} \\ \text{with } S_{i+1}}} \left(DP(X_i, S_i) + |S_{i+1} \setminus X_i| \right)$$

$$S_i \cap (X_i \cap X_{i+1}) = S_{i+1} \cap (X_i \cap X_{i+1})$$

Final answer: $\max_{S_r} DP(X_r, S_r)$

Runtime: $4^k \text{poly}(n)$

Can solve many other graph problems on bounded pathwidth

Tree width

Treewidth

Generalization of pathwidth, similar DP

Def (Tree decomposition)

A tree decomposition of G is a tree with bags $X_i \subseteq V$ as vertices, s.t.

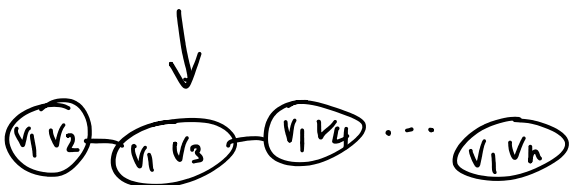
- ① (same)
- ② (same)
- ③ " " forms a (connected) (sub)tree in the tree

Equivalently: " " $u \in X_j$ for all j in the $X_i - X_k$ path in the tree

The width is (same)

Def: the treewidth of G is (same, except tree decomp)

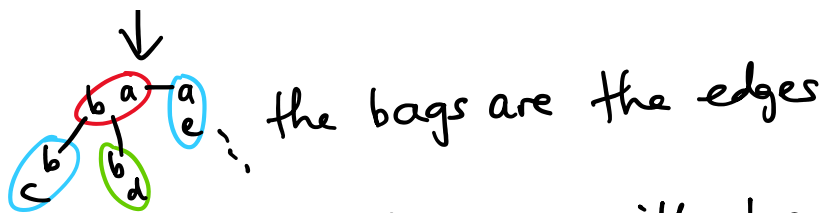
Obs: star has treewidth 1



Obs: trees have treewidth 1



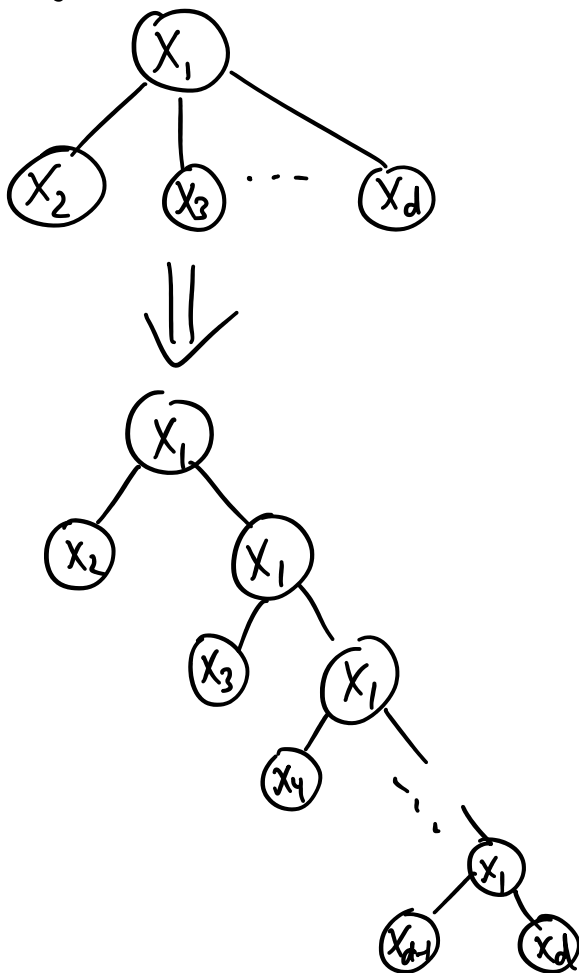
$a-b, b-c, c-d, d-e$ are the edges



Obs: trees are the only graphs with treewidth 1.

Claim: Given tree decomp, can convert it to a tree decomp of the same width, whose tree has max degree 3.

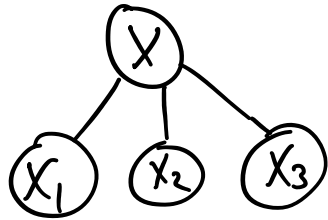
Pf: High-degree X_1 :



Thm: Can solve MaxIS in time $2^{O(k)} \text{poly}(n)$, given a $tw-k$ decomp.

Algo: Root the tree decomp. Same DP, except now have



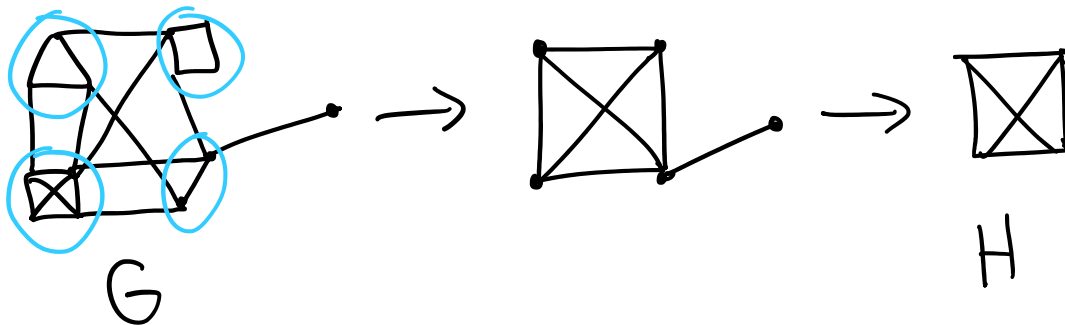


$$DP(X, S) = \max_{\substack{S_1 \subseteq X_1 \\ S_2 \subseteq X_2 \\ S_3 \subseteq X_3 \\ \left. \vphantom{\begin{matrix} S_1 \subseteq X_1 \\ S_2 \subseteq X_2 \\ S_3 \subseteq X_3 \end{matrix}} \right\} \text{consistent with } S}} \left(\sum_{i=1}^3 (DP(X_i, S_i) - |S_i \cap X|) + |S| \right)$$

Runtime: $8^k \text{ poly}(n)$

Graph Minors

- H is a minor of G if it can be obtained from G by a series of vertex/edge deletions and edge contractions.



G has K_4 as a minor.

Claim: If G has $\text{treewidth} \leq k$, then any minor of G also has $\text{treewidth} \leq k$.

"The family of $\text{tw} \leq k$ graphs is minor-closed."

Pf: Start with tree decomp of G of width k , and update it alongside the series of

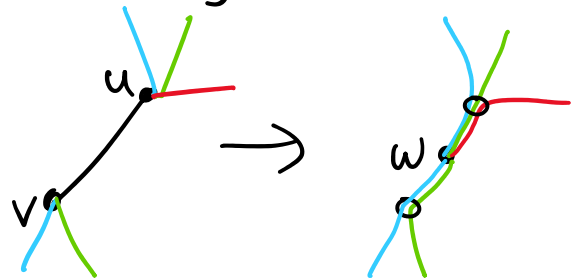
It is shown that we can maintain a tree decomposition and update it alongside the series of vertex/edge deletions and edge contractions, without increasing its width.

- ① Edge deletion: do nothing
- ② Vertex deletion: remove v from all bags containing v (empty bags are OK)
- ③ Edge contraction: Suppose contract edge uv into new vertex w . Replace all occurrences of u, v in tree decomp with w .

Claim: If G is planar, then any minor of G is also planar.

Pf: Start with a planar drawing of G ; modify it alongside.

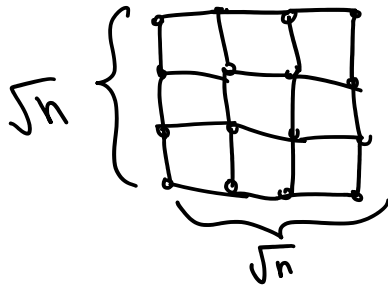
- ① Vertex/edge deletion: easy
- ② Edge contraction:



Thm [Kuratowski]: G is planar $\iff G$ does not contain K_5 or $K_{3,3}$ as a minor.

Thm [Eppstein]: If G is planar, then $\text{treewidth}(G) \leq O(\text{diameter}(G))$.

Thm: The grid on n vertices is planar, but has treewidth $\sqrt{n}$.



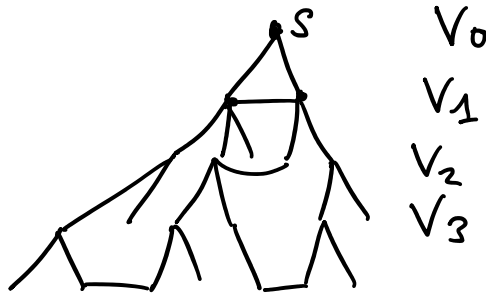
Baker's technique

Def (PTAS): a polynomial-time approximation scheme (PTAS) is an algorithm that, for any $\epsilon > 0$, can compute a $(1 \pm \epsilon)$ -apx to the problem in $\text{poly}(n)$ time

↑ can be $2^{O(1/\epsilon)} \text{poly}(n)$, or even $n^{O(1/\epsilon)}$

Thm: can $(1 - \epsilon)$ -approximate MaxIS in time $2^{O(1/\epsilon)} \text{poly}(n)$ on planar graphs

Algo: take a BFS from an arbitrary vertex s



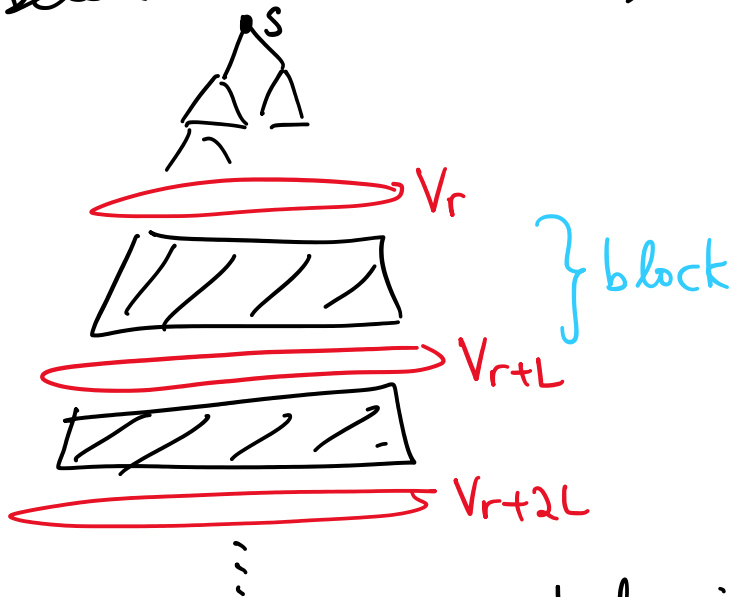
$V_i :=$ vertices on level i of BFS
 $=$ vertices distance i from s

$\cdot$ = vertices distance 1 from $\cup$

Obs: no edges between levels ≥ 2 apart
(e.g. V_i and V_{i+2})

Fix $L = 1/\epsilon$

- Pick a random offset $r \in \{0, 1, \dots, \Delta-1\}$
- Delete all vertices in $V_r, V_{r+L}, V_{r+2L}, \dots$



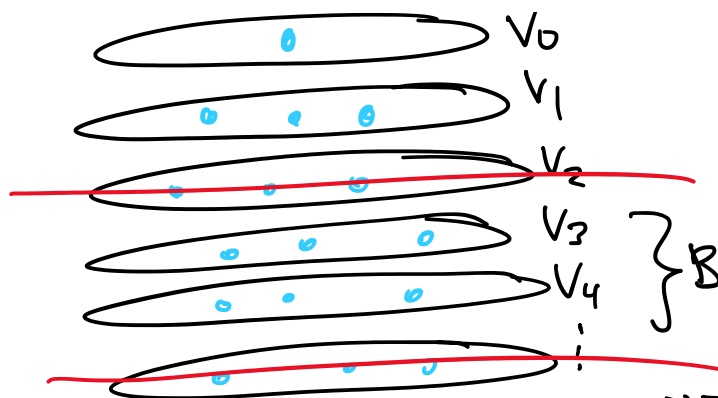
- These vertices will never belong in IS.
- $\mathbb{E}[\# \text{ vtes deleted}]$? Each vertex deleted w.p. $\frac{1}{L}$, so it's $\frac{n}{L} = \epsilon n$
- Define vertices between two consecutive deleted levels to be a block.

Claim: For each block B , $G[B]$ has treewidth $O(1/\epsilon)$.

- Assuming this, solve MaxIS exactly in each block in $2^{O(1/\epsilon)} \text{poly}(n)$ time

What is size of solution? Take OPT MaxIS





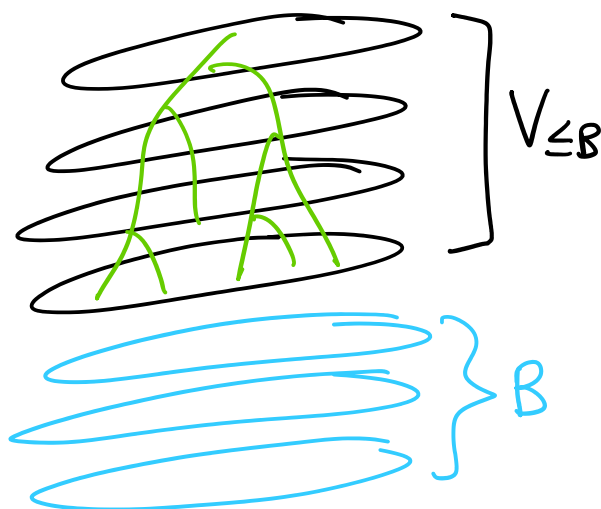
$$\mathbb{E}[\# \text{ vertices deleted in OPT}] = \frac{\text{OPT}}{L} = \varepsilon \cdot \text{OPT}$$

In each block B , $\text{MaxIS}(G[B]) \geq |\text{OPT} \cap B|$.

$$\sum_{\text{block } B} \text{MaxIS}(G[B]) \geq \sum_{\text{block } B} |\text{OPT} \cap B| = \# \text{ vts not deleted in OPT} = (1 - \varepsilon) \text{OPT in expectation.}$$

$$\Rightarrow \mathbb{E}[\text{solution size}] \geq (1 - \varepsilon) \text{OPT.}$$

→ Pf of Claim: Take a block B .



Can't directly apply diameter-treewidth thm to B ; B may have large diameter (may even be disconnected!)

Start with planar graph G .

- Delete all vertices outside $V_{\leq B}$ and B .

- Delete all vertices outside $V \leq \epsilon$ and B .
Since planar graphs are minor-closed,
remaining graph still planar.
- Contract $V \leq \epsilon$. (Contract edges one-by-one;
since $V \leq \epsilon$ is connected, we end up with
a single vertex.) Still planar
- This graph has diameter $\leq O(1/\epsilon)$!
- Apply diameter-treewidth thm: graph
has treewidth $O(1/\epsilon)$
- Delete contracted vertex; since
treewidth- $O(1/\epsilon)$ graphs are minor-closed,
still have a treewidth $O(1/\epsilon)$ graph.
This graph is our desired $G[V \leq \epsilon]$.