6.889	Lecture 7	Oct. 3, 2011

Parameterized Complexity, Treewidth, Bidimensionality

Recall an independent set in a graph is a set of vertices with no edges between them.

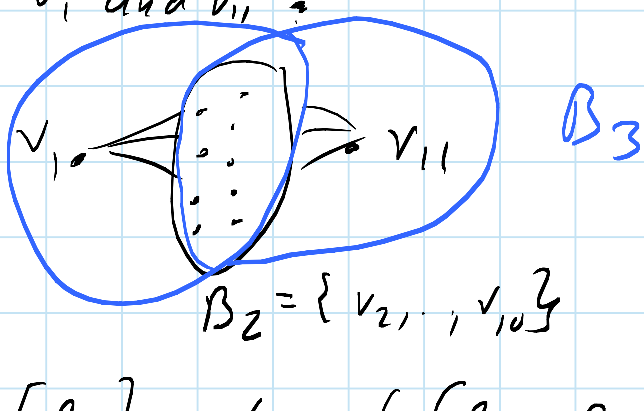
MIS: find a maximum independent set in a graph.

NP-complete and not approximable within a factor of $n^{1-\varepsilon}$ in general graphs.

When graph is small $\rightarrow$ easy: try all subsets!

Consider graph on 10 vertices $\rightarrow$ 2^{10} subsets ✓

Consider graph on 11 vertices $v_1, \ldots, v_{11}$ but no edge between v_1 and v_{11}:



$G_1 = G[B_1]$, $G_2 = G[B_1 \cup B_2]$, $G_3 = G[B_1 \cup B_2 \cup B_3] = G$

For $i=1,2,3$ and $S \subseteq B_i$, define

$$T_i(S) = \begin{cases} 0 & \text{if } S \text{ is not independent} \\ \text{MIS of } G_i \text{ given that } S \text{ must be in the solution and } B_i - S \text{ must } \underline{\text{not}} \text{ be in the solution.} \end{cases}$$

Hence,

$$T_1(S) = |S| \text{ or } 0$$

$$T_2(S) = \max\{T_1(S), T_1(S \cup \{v_1\})\} \rightsquigarrow \text{we only FORGET } v_1$$

$$T_3(S) = \begin{cases} 0 & \text{if } S \text{ is not independent} \\ T_2(S) & \text{if } v_{11} \notin S \\ 1 + T_2(S - \{v_{11}\}) & \text{if } v_{11} \in S \end{cases}$$

$\rightsquigarrow$ we only <u>INTRODUCE</u> v_{11}.

"Running time": $3 \cdot 2^{10}$ ← size of each bag

↑ # of bags

Consider any graph that we can obtain as follows:

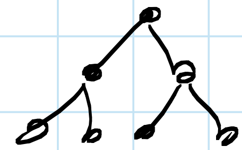
- start with a small <u>bag</u> of k vertices
- obtain next bag by either <u>forgetting</u> one vertex or <u>introducing</u> one vertex
- a forgotten vertex is never introduced again
- each bag has $\leq k$ vertices

→ we say the graph has **pathwidth** k

⇒ MIS can be solved in time $2^k \cdot O(n)$

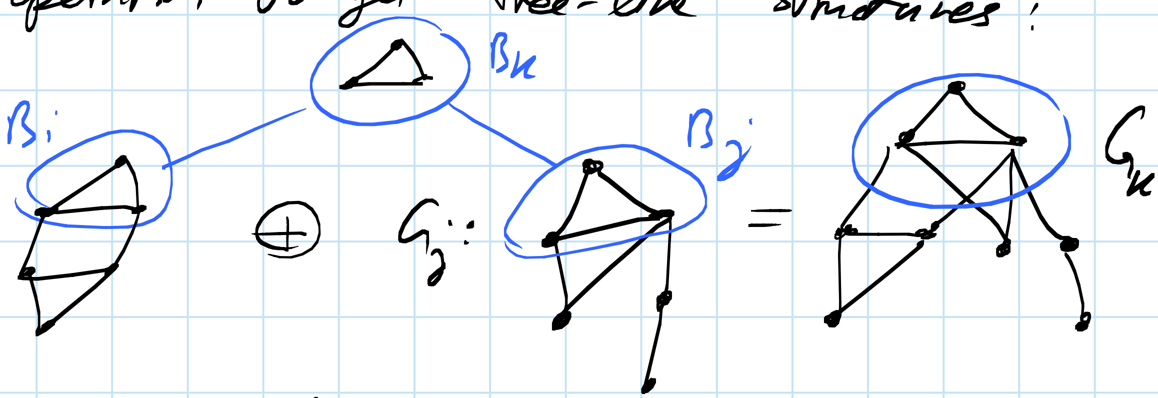
Small pathwidth → graph is "path-like" and hence easy

Consider full binary tree on n vertices



Pathwidth is $\sim \log n$ but trees are "easier" than that!

→ in addition to FORGET and INTRODUCE allow JOIN operation to get "tree-like" structures:



In the context of MIS:

$$T_k(S) = \begin{cases} 0 & \text{if } S \text{ not independent} \\ T_i(S) + T_j(S) - |S| \end{cases}$$

→ gives rise to the concept of **treewidth**

→ on graph of treewidth k, MIS can be solved in time $2^k \cdot O(n)$.

Definition: A tree decomposition of a graph $G=(V,E)$ is a pair $(\mathcal{T}, (B_t)_{t\in T})$ where $\mathcal{T}=(T,F)$ is a tree and $(B_t)_{t\in\mathcal{T}}$ is a family of subsets of V called bags, s.t.

(1) every vertex $v\in V$ appears in some bag B_t,
($\hookrightarrow$ every vertex must be introduced at least once)

(2) for every edge $uw\in V$ there is a $t\in T$ such that $u,w\in B_t$.
($\hookrightarrow$ we shall obtain the whole graph)

(3) for every vertex $v\in V$, the subgraph $\mathcal{T}_v$ of $\mathcal{T}$ on which v appears is connected, i.e.

$$\mathcal{T}_v := B^{-1}(v) := \{t\in\mathcal{T} \mid v\in B_t\} \text{ is connected.}$$

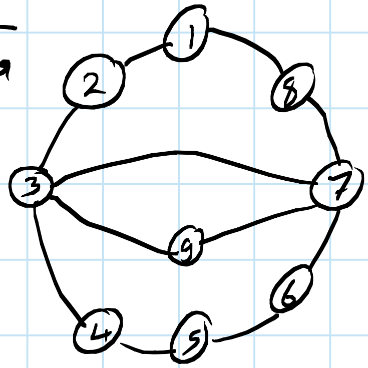
($\hookrightarrow$ every vertex must be forgotten exactly once)

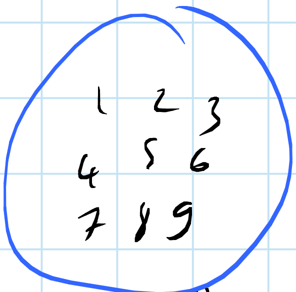
The width of the decomposition is $\max_{t\in T} |B_t| - 1$.

The treewidth of G is the width of a tree decomposition of G of minimum width.

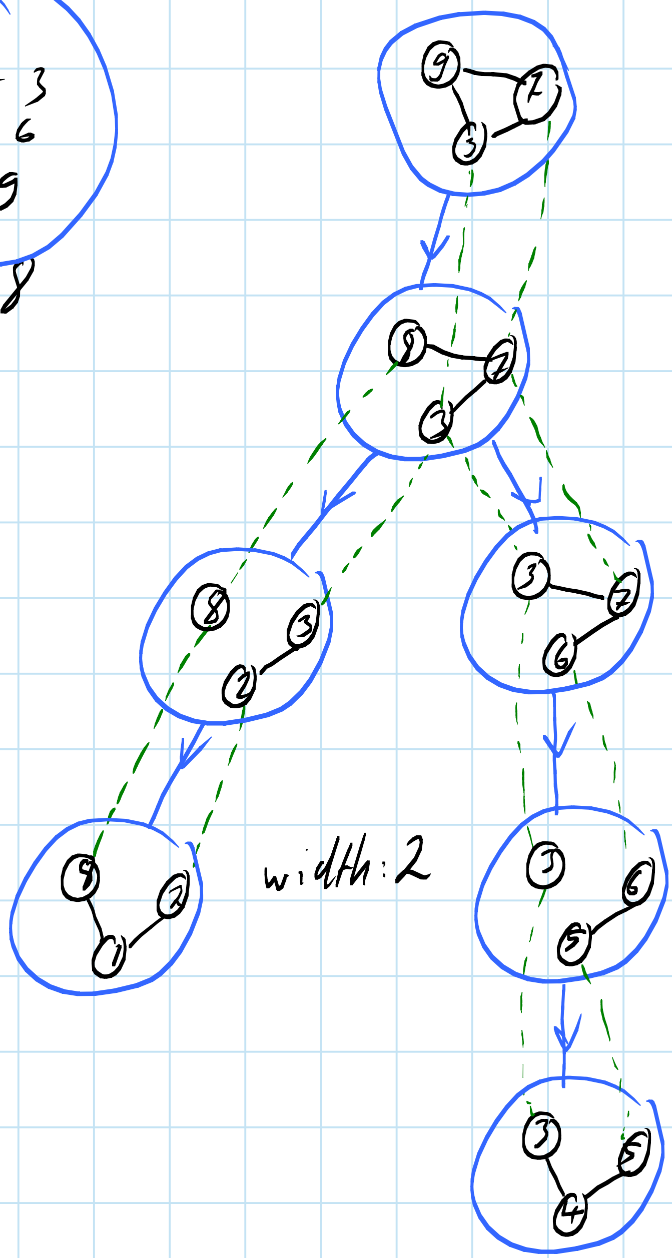
If $\mathcal{T}$ is a path we obtain a path decomposition and the analogous notion of pathwidth.

G

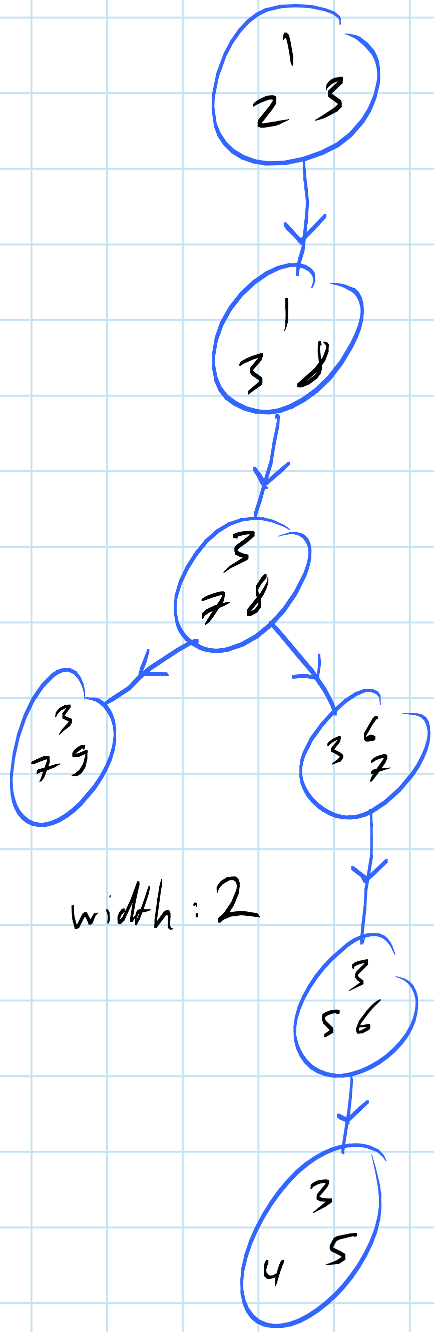




width = 8



width: 2



width: 2

contracting the dashed green lines results in G

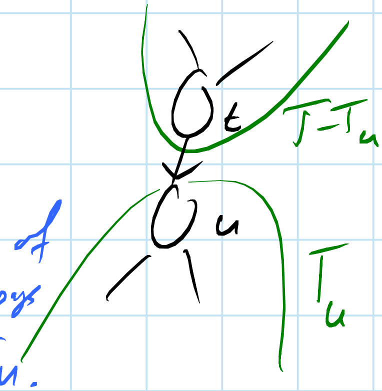
For a rooted tree T and $u \in T$ let T_u denote the subtree rooted at u.

Lemma: Let $(J, (B_t)_{t \in T})$ be a tree decomposition of G. Then for every edge (t,u) of J,

$$B_t \cap B_u$$

separates $B(T - T_u)$ from $B(T_u)$.

union of all bags of T_u.



→ every edge of J is a (small) separator of G. (and so is every bag)

Further facts (see exercises or references for proofs):

- G has treewidth 1 iff G is acyclic.
- a cycle has treewidth 2.
- the $k \times k$ grid has treewidth k.
- for every clique of G there is a bag that completely contains it. $\Rightarrow$ K_n has treewidth $n-1$.
- every k-connected graph has treewidth at least $k-1$.
- a tree decomposition is small if for $t \neq t' \in T$, $B_{t'} \not\subseteq B_t$. Every tree decomposition can be transformed to a small tree decomposition of the same width in linear time.
- every nonempty graph of treewidth at most w has a vertex of degree at most w.
- if $H \preccurlyeq G \Rightarrow tw(H) \leqslant tw(G)$, i.e. tw is minor-monotone.

Algorithmic aspects of treewidth (assume $tw(G)=k$):

- NP-hard; best known approximation $O(\sqrt{\log k})$ [FHL'08]
- tree decomposition of width k in time $2^{poly(k)} \cdot n$ [Bodlaender'96]
  → complicated
- tree decomposition of width $\leq 4k+1$ in time $2^{O(k)} \cdot n^2$
  → simple
- 1.5-approximable in planar graphs
- constant-factor approximable in H-minor-free graphs [FHL'08]
  → $O(h^2)$-approximation

small treewidth ⟷

- graph is "tree-like"
- small (balanced) separators "everywhere"
  → $tw(G)=k \Rightarrow$ balanced sep. of size $k+1$
  → planar graphs and H-minor-free graphs have treewidth $O(\sqrt{n})$
- many NP-hard problems easy.
  → often in time $f(k) \cdot O(n)$ → FPT

⇒ many NP-hard problems on H-minor-free graphs in time $2^{O(\sqrt{n})}$.

Parameterized Complexity Theory [Downey-Fellows '99]

[Flum-Grohe '06]

provides a framework for a refined analysis of hard algorithmic problems.

- classical complexity theory analyzes problem in terms of a resource, usually time or space, as a function of the size of the input.
- clean-cut theory but ignores structural information about the input $\longrightarrow$ often makes problems seem harder
- parameterized complexity theory is 2-dimensional:

$$(\underbrace{\text{input size}}_{n}, \underbrace{\text{parameter}}_{k})$$

Goal: address complexity issues when parameter rather small

Examples: Database Query:

size of database $\rightarrow n$

size of query $\rightarrow k$

Approximation Schemes:

input size $\rightarrow n$

approximation ratio $\rightarrow k = 1/\varepsilon$

Other common parameters: <u>treewidth</u>, degree, size of solution

Positive Theory: Fixed-Parameter-Tractability (FPT)

Algorithms with running time $f(k) \cdot n^{O(1)}$.

Example: Maximum independent set parameterized by the tree width is FPT.

Theorem [Courcelle '90]:
Any problem that can be described in the language of monadic second-order logic (MSO_2) is FPT when parameterized by the length of the formula plus the treewidth of the instance:

$$\text{Running time } f(\underbrace{|\varphi|}_{\text{formula}} + \underbrace{tw(G)}_{\text{treewidth}}) \cdot O(\underbrace{n}_{\text{problem size}})$$

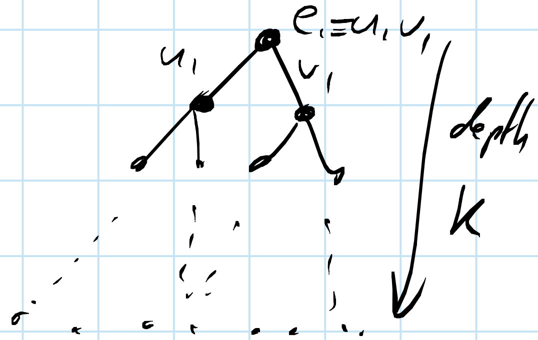
→ 3-colorability, vertex cover, hamiltonicity, dominating set, and many more are tractable on graphs of bounded treewidth.

Some methods:
- dynamic programming on tree-decompositions
- bounded search tree
- kernelization
- color coding
- iterative compression
- graph minors / well-quasi-ordering
- ...

very active research area

Simple example: A vertex cover in a graph G is a set of vertices of G such that every edge of G has an endpoint in it.
Given graph G and integer k, determine if G has a vertex cover of size at most k, where the parameter is k.
→ standard parameterization

```
VC(G,k):
{
  if G(k)=∅ return true;
  if k=0 return false;
  pick edge e=uv
     if VC(G-u,k-1) or
        VC(G-v,k-1) return true;
  return false;
}
```



$\Rightarrow$ Runtime $O(2^k \cdot n)$

Negative Theory

very active research area

- better algorithms have always been seeked
- main contribution of theory is a framework for intractability
- the class XP: $n^{f(k)}$ → bad
- the class $W[1]$: any param. problem reducible to k-clique
- the class $W[2]$: " " " to k-dominating set

⋮

$$FPT \subseteq W[1] \subseteq W[2] \subseteq \ldots \subseteq W[P] \subseteq XP$$

$\boxed{\text{Main assumption: } FPT \neq W[1]}$

analogous to $P \neq NP$ but stronger assumption

References

[Bod96] Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[Cou90] B. Courcelle. Graph rewriting: An algebraic and logic approach. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume 2, pages 194–242. Elsevier, 1990.

[DF99] Rodney G. Downey and Micheal R. Fellows. *Parameterized Complexity.* Springer-Verlag, 1999.

[Die05] Reinhard Diestel. *Graph Theory.* Springer, 2005.

[FG06] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory.* Springer, 2006. ISBN 3-54-029952-1.

[FHL08] Uriel Feige, MohammadTaghi Hajiaghayi, and James R. Lee. Improved approximation algorithms for minimum weight vertex separators. *SIAM J. Comput.*, 38(2):629–657, 2008.