Lecture 17 - Treewidth

[scribe - slides by Marx, Bodlaender]

On trees, NP-hard prob often easy.

\[ \text{e.g. } \text{Max} [\text{weighted}] \text{- Indep. Set} \]

\[ \text{set of non-adjacent verts} \]

\[ \text{[How?]} \text{ Recursion/dynamic prog.} \]

Let \( MIM^+[v] = \max \text{ indep set in } T_v \text{ which includes } v \)

\[ MIM^-[v] = \max_{i=1}^n M^+[w_i] \]

If \( v \) has kids \( w_1, \ldots, w_k \),

\[ MIM^+[v] = 1 + \sum_{i=1}^k MIM^+[w_i] \]

\[ MIM^-[v] = \max_{i=1}^k MIM^+[w_i] \]

[leaves are trivial.

End up doing post-order traversal of tree. \( O(n) \) time.]

eg. 2: Give CSP inst, "primal graph" = undir (\( Q, \Psi \))

Sp's primal graph is a tree (\( \Psi \) arity \( \leq 2 \))

Then Satisfiability \( \in P \).

DP: Let \( E[v, a] = \{ \text{true if } \exists \text{ partial assign. to } T_v \text{ with } v \leftarrow a \} \)

[leaves trivial (true unless every constr. forbids). If you have \( \neq \)

for \( v \) is child, it's easy...]

Q: More gen. class of graphs? \( \exists \text{ in poly time?} \)
'70s: "Series parallel graphs":

Recursive def. —

Series parallel graphs have a kind of tree decom. II

Ex: MIS on S/P graphs in $O(n)$ time by D.P.

Hint

Rec: also works for subgraphs of S/P graphs

E.g.:

MIS $[s, t]$

$[s, t']$

$[s', t']$ $[s, t']$
1980s: several groups of researchers extended to even more "weakly tree-like" graphs, w/ some "tree decmp".

Definition of Tree-decomposition of $G = (V,E)$:

A tree $T$ of "bags" $X$ s.t.,

1. if $(u,v) \in E$ then $u,v$ together in some bag
2. $\forall v \in V$ the bags containing $v$ are connected in $T$ (form a subtree)

Do example of 6, d 1

Dr. will work w/ bags, want them small. Could just shave everything in 1 bag! 1

Width: $(\text{max bag size}) - 1$. 1 is so that trees will have tw.

E.g. has width 2.

Tree-width of $G$: least width of a tree decom. 1

Introduced in 80s by zillions of indep groups: Halin, A. Goldberg-Posukhovsky, Robertson-Seymour, ...

Also, 3 zillions of equivalent definitions...

Fact: Trees have tw 1. bags are edges.

$\text{tw}(G) = 1 \iff G$ is a forest (easy)

$\text{tw}(G) = 2 \iff G$ is subgraph of a series/par graph. (just as easy)

"trees: recursive gluing together of single vertices"

\[ \text{tw} k = \ldots k \text{ etc.} \]
deleting edges cannot incr. tw. [Same with contracting, actually.]

facts:

- Any clique in $G$ must be $\emptyset$ in a bag.
  $\Rightarrow tw(K_n) = n-1$.

def: $G$ is chordal if every cycle of len $\geq 4$ has a "chord." [Example graph: not chordal.]

- adding edges to get a chordal graph = triangulation.

fact: $tw(G) \leq k \Rightarrow G$ has triangulation w/ max clique size $\leq k+1$

[Example graph: Can you triangulate so no 4-cliques?]

"Chordal graphs are like maximal tree-width graphs." [Example:]

fact: $G$ chordal $\Rightarrow$ has tree-decomp w/ each bag a clique

- $\exists$ "perfect elim ordering" $v_0, v_1, \ldots, v_n$ s.t.
  - for each $v_x$, higher #d vts form a clique
  - start w/ $k+1$ clique, keep taking on clipped

ex: any tree-decomp easily transformed to a smooth one

- all bags have $k+1$ vts
- $n$-ring bags have size-$k$ intersection

[In example, just contract up junk on bottom left; sometimes you need to split edges]

rem: smooth tree-decomps have $n - tw(G)$ edges

[Clear from picture. Less than $n$, in particular.]
One more fun characterization...

(Seymour-Thomassen) \( Tu(G) = k \Rightarrow k+1 \) cops can win the Cops & Robber game.

Explain game in words. Robber has great speed, can walk along edges or orbit; fast. Cops can be on nodes or hovering in helicopters. Cops decide to land at node; robber can see them coming. Inc.

2 cops nec. & suff. on a tree is easy.

Illustrate 3 cops winning on eg. graph. Uses smoothness.

⇒) easy. (⇐) hard.

cor: \( M \times M \) grid has \( Tu > M-1 \) \( (M-1 \) cops can't win.

hint: \( \exists \) a cop-free row, col.

ex: \( Tu(M \times M \text{ grid}) = M \) \( \text{[need slightly better robber stat.]} \)

rem: shows that even a deg. 4 planar graph can have \( Tu \) in t.w.

\[\text{In a second we'll see that } 0(1) \text{ to } \equiv \text{fast algo for } NP \text{-hard problems. These typically require tree decomposition itself... how to find?}]\]

\([ACP'87]\): exact in time \( n^{k+o(1)} \).

\([Bo'96]\): \( 2^{O(k^3)} \n \) \( \text{horrible costs. Also, not great for } \alpha = O(\log n). \)

\([BDLP'13]\): \( k \) cops in time \( 2^{O(k^2)} \).

\([FHL'08]\): \( O(k \log k) \) in time poly \( n \).
Finally, algs on odd treewidth graphs.

Thm: (easy) If the primal graph of a CSP inst. has treewidth ≤ k,
satisfiability decidable in 2^O(kn) time.

E.g. 3Sat on Ω(k)-tw graphs — np. II

Thm: Also solvable in poly-time when tw(G) ≤ O(k):
- colorability
- Han-Cycle/TSP
- Clique,
- vtx-disj. paths
- [not edge]

Corollary: let F be formula in extended monadic 2nd-order logic,
like \[ \exists C \subseteq V \forall v \in C \exists u, u_1, u_2 \in C (u \neq u_1 \wedge \text{adj}(u_1, v) \wedge \text{adj}(u_2, v)) \]
quantify over sets of edges, and \text{"incident(e)" OK too}.

Then deciding F on G with tw(G) \leq k, G = (V, E) has a cycle,
in \[ 2^O(kF) \cdot O(n) \] time. 

Basic illustration:

- [Not strictly nec., but helpful]
- Convert to "nice" tree decomp.
- Rooted \( \pm \) binary tree.
- 4 types of node:
  - leaf
  - introduce
  - forget
  - join

- \[ \text{size 1} \]
- \[ \text{size 2} \]
- \[ \text{size 3} \]
- \[ \text{size 4} \]
Example.
For bag $X$, coloring $c : X \rightarrow \{R,G,B\}$

Let $E[X,c] = \{ \text{true if } c \text{ can be extended to a proper col. of }$
$\cup X \text{ in } X \text{ and descendant bags.} \}$

**D.P.** 3-Colorability

- Leaf (easy): $a \rightarrow R$, true $a \rightarrow G$ $a \rightarrow B$
- Intro nodes: Say $X = \text{introducing } v$
  $E[X,c] = \begin{cases} 
  E[X,c'] & \text{if } c(v) + c(u) \\
  \text{otherwise} & \text{else FALSE.}
  \end{cases}$

- Forget nodes: $E[X,c] = \text{true iff } E[Y,c'] \text{ true for } \geq 1 \text{ of the three exts } c' \text{ of } c$

- Join nodes: $E[X,c] = E[Y,c'] \lor E[Y',c]$

**TIME:** $O(3^k)$
Finally, although planar graphs can have large treewidth, can often upgrade treewidth algo to planar PTASs.\

Thm: \cite{Epstein} A planar graph w/ diam D has tw \leq 3D - 2.

Let k ≥ 2, \text{poly}(k) time can partition G's edges into \( k \) sets such that when any 1 set is
\rightarrow deleted \cite{BourdonneauBaker99}
\rightarrow contracted \cite{DAM02}
the result has tw \( O(k) \)
\rightarrow Set \( k = O(\varepsilon) \) \rightarrow "negligible."

\textbf{Corollary:} Can find \((-\varepsilon)\)-approx. max-indep-set in planar graph in time \( 2^{O(1/\varepsilon)} \cdot \text{poly}(m) \)
\text{in fact, in.}