Graphs & Algorithms: Advanced Topics

Treewidth

Uli Wagner

ETH Zürich
Dynamic Programming on Trees

Example

**MaxIndependentSet** in a tree $T = (V, E)$ rooted at $r \in V$.

For $v \in V$ let $T_v$ denote the subtree rooted at $v$.

$$f^+(v) := \max\{|I| : I \subseteq V(T_v) \text{ independent}, v \in I\}$$

$$f^-(v) := \max\{|I| : I \subseteq V(T_v) \text{ independent}, v \notin I\}$$

**Algorithm** Compute maximum independent set in $T$ (time $O(|V|)$)

Traverse $T$ in post order starting from $r$. $v :=$ current vertex

- If $v$ is a leaf, let $f^+(v) = 1$ and $f^-(v) = 0$.
- Else let $x_1, \ldots, x_k$ be the children of $v$

$$f^+(v) := 1 + \sum_{i=1}^{k} f^-(x_i), \quad f^-(v) := \sum_{i=1}^{k} \max\{f^+(x_i), f^-(x_i)\}$$

Return $\max\{f^+(r), f^-(r)\}$.
Tree Decompositions

Definition
A tree decomposition for a graph \( G = (V, E) \) is a pair
\[
( \{ X_i \mid i \in I \}, \quad T = (I, F) )
\]
such that
\[
\begin{align*}
\top & \quad \bigcup_{i \in I} X_i = V \quad (\text{bags cover vertices}); \\
\top & \quad \{ u, v \} \in E \implies \{ u, v \} \subseteq X_i \text{ for some } i \in I \quad (\text{bags cover edges}); \\
\top & \quad \text{for all } v \in V \text{ the set } I_v = \{ i \in I \mid v \in X_i \} \text{ is connected in } T \quad (\text{tree property})
\end{align*}
\]
The width of a tree decomposition is \( \max_{i \in I} |X_i| - 1 \).
The treewidth of \( G \) is the minimum width of a tree decomposition for \( G \), denoted \( \text{tw}(G) \).

Example. Trees have treewidth 1.
Basic Observations

Observation
For any graph $G = (V, E)$ a single bag containing $V$ forms a tree decomposition of width $n - 1$.

Goal
Tree decompositions of small width, certify that the graph is in some way “tree-like”

Observation
If $H \subseteq G$ is a subgraph then $\text{tw}(H) \leq \text{tw}(G)$

Proposition
If a graph $G = (V, E)$ has two components $A$ and $B$ with $A \cup B = V$ then $\text{tw}(G) = \max\{\text{tw}(A), \text{tw}(B)\}$. 
Treewidth of cliques and grids

Lemma
Let \( \{X_i \mid i \in I\}, T = (I, F) \) be a tree decomposition for \( G \). For any clique \( G[W], W \subseteq V(G) \), there is an \( i \in I \) such that \( W \subseteq X_i \).

Proof. Root \( T \) arbitrarily. For \( w \in W \) choose a “bag label” \( r_w \in l_w \) of minimal depth (distance from the root). Then the bag with the label \( \{r_w \mid w \in W\} \) with maximum depth contains \( W \). \( \square \)

Corollary
The treewidth of \( K_n \) is \( n - 1 \).

Example
The \( n \times n \)-grid on \( \{(i, j) \mid 1 \leq i, j \leq n\} \) has treewidth \( \leq n \):
Consider the path on
\[
X_n(i-1)+j = \{(i, k) \mid j \leq k \leq n\} \cup \{(i + 1, k) \mid 1 \leq k \leq j\},
\]
\( 1 \leq i \leq n - 1, \ 1 \leq j \leq n. \)
How many vertices are needed in $T$?

**Definition**
A tree decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ of width $k$ is **smooth** if
- $|X_i| = k + 1$ for all $i \in I$;
- $|X_i \cap X_j| = k$ for all $\{i, j\} \in F$.

**Proposition**
For any graph with treewidth $k$ there exists a smooth tree decomposition of width $k$.

**Exercise**

**Lemma**
If $(X, T = (I, F))$ is a smooth tree decomposition of width $k$ for $G = (V, E)$ then $|I| = |V| - k$. In particular, $n(T) \leq n(G)$.

**Proof.** Induction on $|I|$. Consider a leaf $i \Rightarrow$ unique vertex $v \in X_i$ that does not belong to any other $X_j, j \neq i$. Removing $i$ from $T$ yields smooth tree decomposition of $G[V \setminus \{v\}]$. 

\[\square\]
Number of edges

Lemma
A graph $G = (V, E)$ of treewidth at most $k$ has at most $k|V| - \binom{k+1}{2}$ edges.

Proof. Induction on $|V|$. Base case is $|V| = k + 1$. Consider a smooth tree decomposition $(\{X_i | i \in I\}, T = (I, F))$ for $G$ and a leaf $i$ of $T$. Then there is a unique vertex $v \in X_i$ that does not belong to any other $X_j$, $j \neq i$. Clearly $\deg_G(v) \leq k$. Removing $i$ from $T$ yields a tree decomposition for $G[V \setminus \{v\}]$. □

Corollary
A graph has treewidth 1 if and only if it is a forest (and contains at least one edge).
Lemma
Let \( \{X_i \mid i \in I\}, T = (I, F) \) be a tree decomposition for a connected graph \( G = (V, E) \) such that \( X_i \not\subseteq X_j \) for \( i \neq j \). Then

a) \( X_i \cap X_j \) is a vertex cut in \( G \) for any \( \{i, j\} \in F \), i.e.,
\[ G - (X_i \cap X_j) \] is disconnected.

b) \( X_i \) is a vertex cut in \( G \) for any \( i \in I \) that is not a leaf in \( T \).

Remark
It is possible to adapt any tree decomposition in \( O(|I|) \) time to fulfill the non-containment condition without changing its width. (If there is a containment, there is also one between adjacent bags.)
Treewidth and Separators

Theorem

Tree decomposition \( \{X_i \mid i \in I\}, T = (I, F) \) of width \( k \) for \( G \) \( \Rightarrow \) can compute \((k + 1, \frac{1}{2})\)-separator for \( G \) in \( O(|I|) \) time.

Proof. Root \( T \) arbitrarily and define a weight function \( w \) on \( I \) by \( w(i) := |X_i \setminus X_{\text{parent}(i)}| \). Each \( v \in V \) is counted exactly once (bags containing \( v \) are connected) \( \Rightarrow \sum_{i \in I} w(i) = |V| \).

By the Separator Theorem for (weighted) trees we obtain a \((1, \frac{1}{2})\)-separator \( s \) for \( T \).

Removing \( X_s \) disconnects \( G \) because

- any \( v \in V \setminus X_s \) can appear in at most one subtree (otw, it would also appear in \( X_s \) by connectivity);
- each subtree defines at least one component (no edge between subtrees);
- each subtree (and hence component) consists of at most \( \frac{n}{2} \) vertices.
Dynamic Programming on Graphs of Treewidth at most $k$

**Given:** $G = (V, E)$ and smooth tree decomposition $(\{X_i | i \in I\}, T = (I, F))$ of width $\leq k$ for $G$

**Algorithm** to compute a maximum independent set in $G$ in time $O(k^2 4^k |V|)$

Pick an arbitrary root $r \in I$ and for $i \in I$ let $V_i = \bigcup_{j \in T_i} X_j$, where $T_i$ denotes the subtree rooted at $i$.

For $U \subseteq X_i$ let $f^U(i)$ be the size of a maximum independent subset of $V_i$ whose intersection with $X_i$ is exactly $U$.

- Traverse $T$ starting from $r$ top-down. Let $i :=$ current vertex.
- If $i$ is a leaf, for every $U \subseteq X_i$, we have $f^U(i) = |U|$ if $U$ is independent in $G$ and $f^U(i) = -\infty$, otherwise.
- Else let $c_1, \ldots, c_\ell$ be the children of $i$. We have

$$f^U(i) = |U| + \sum_{j=1}^{\ell} \max \left\{ f^W(c_j) - |U \cap W| \right\}$$

$W \subseteq X_{c_j}$ and $W \cap X_i = U \cap X_{c_j}$ and $W \cup U$ independent

*(Convention: $\max(\emptyset) = -\infty$).*
Treewidth and Minors

Proposition

Graphs of treewidth at most $k$ are closed under taking minors.

Proof. Removal of edges and isolated vertices are trivial. When contracting an edge \{u, v\}, replace all occurrences of $u$ in any bag by $v$.

By the Robertson-Seymour Graph Minor Theorem there is hence a finite set of forbidden minors. But they are not known, except for small $k$.

- $k = 0$: $K_2$.
- $k = 1$: $K_3$.
- $k = 2$: $K_4$.
- $k = 3$: $K_5$, $K_{2,2,2}$, \[\text{\includegraphics[width=2cm]{k3}}\]
- $k = 4$: more than 75...
Computing treewidth

Theorem (Arnborg, Corneil, Proskurowski ’87)

For given $G$ and variable $k$, it is NP-complete to decide whether $\text{tw}(G) \leq k$.

Theorem (Bodlaender ’96)

For any fixed $k \in \mathbb{N}$ there exists a linear time algorithm to test whether a given graph has treewidth at most $k$ and—if so—output a corresponding tree decomposition.
(The running time is exponential in $k^3$.)

Open Problem

- Is there a (polynomial-time) constant-factor approximation algorithm for treewidth? Is there a PTAS?
  (Known: $O(\sqrt{\log n})$-approximation algorithm for vertex separators [Feige, Hajiaghayi and Lee 2008], implies the same for treewidth.)
- Can the treewidth be computed in polynomial time for planar graphs?
Not everything is easy for bounded treewidth...

**Theorem (Nishizeki, Vygen, Zhou ’01)**

*Edge-disjoint paths is NP-complete for graphs of treewidth 2.*

(Trivial for trees and polynomial for outerplanar graphs.)

**Input:** a graph $G = (V, E)$ and pairs $\{s_i, t_i\} \in \binom{V}{2}$, $1 \leq i \leq k$

**Goal:** find $k$ edge-disjoint paths $P_i$ in $G$ such that $P_i$ connects $s_i$ and $t_i$.

**Theorem (McDiarmid & Reed ’01)**

*Weighted coloring is NP-hard for graphs of treewidth 3.*

(Trivial for bipartite graphs, hence for forests.)

**Input:** a graph $G = (V, E)$ and a weight function $w : E \rightarrow \mathbb{N}$

**Goal:** a weighted $k$-coloring, i.e., a function $c : V \rightarrow [k]$ such that $|c(u) - c(v)| \geq w(e)$ for all $\{u, v\} \in E$. 
Cops and robber

In the omniscient cops and robber game, $k$ cops each occupy a vertex of a graph in which a robber moves around, trying to escape capture. The robber moves along edges “at infinite speed”, the cops move “by helicopter”.

Definition

Given a graph $G = (V, E)$ and $k \in \mathbb{N}$, a position in the $k$ cops and robber game on $G$ is a pair $(C, r)$, where

- $C \in \binom{V}{k}$ (location of cops) and
- $r$ is a vertex in some component of $G \setminus C$ (location of robber).

- In Round 0, the cops choose $C_0 \in \binom{V}{k}$ and then the robber chooses $r_0 \in V \setminus C_0$ arbitrarily.
- In Round $i$, $i > 0$, the cops choose $C_i \in \binom{V}{k}$ and then the robber chooses a vertex $r_i \in V \setminus C_i$ such that there is a path between $r_i$ and $r_{i-1}$ in $G \setminus (C_i \cap C_{i-1})$.
- The cops win if after some finite number of rounds the robber has no vertex to choose.
Cops, Robber, and Treewidth

Theorem (Seymour & Thomas '93)

If a graph $G$ has treewidth at most $k$ then $k + 1$ omniscient cops can catch a robber on $G$.

Proof. Suppose $n(G) > k + 1$ and let ($\{X_i \mid i \in I\}$, $T = (I, F)$) be a smooth tree decomposition of width $\leq k$ for $G$.

- Pick an arbitrary root $a \in I$ and for $i \in I$ let $V_i = \bigcup_{j \in T_i} X_j$, where $T_i$ denotes the subtree rooted at $i$.
- In the first round choose $C_0 = X_a$.
- In Round $j$, we suppose $C_{j-1} = X_b$ for some $b \in I$ and $r_{j-1} \in V_b \setminus X_b$. Let $c$ be the child of $b$ for which $r_{j-1} \in V_c$. Observe that $X_b \cap X_c$ is a $k$-cut in $G$. Thus choosing $C_j = X_c$ confines the robber to $V_c \setminus X_c$.
- After a finite number of steps, we arrive at a leaf $\ell$ of $T$ for which $V_\ell \setminus X_\ell = \emptyset$. Thus, the robber has nowhere to go. □

Remark

The converse also holds but the proof is much more involved.
Cops and robber on the grid

Proposition

On the $n \times n$-grid $n - 1$ omniscient cops cannot catch a robber.

Proof. Whichever positions the $n - 1$ cops occupy, there are always a cop-free row and a cop-free column.

Claim. The robber can always move to the intersection of a cop-free row with a cop-free column.

Initially, this is clear. Suppose that at some point one or more cops enter the free row and/or free column where the robber is located. Then the robber can move along the previously free row to the to-be free column and within this column to the to-be free row.

Proposition

On the $n \times n$-grid $n$ omniscient cops cannot catch a robber, for $n \geq 2$. → Exercise.

Corollary

The $n \times n$-grid has treewidth $n$. ⇒ There are planar graphs on $n$ vertices whose treewidth is $\Omega(\sqrt{n})$. 
Partial $k$-trees

Definition
A $k$-tree is a graph formed from a $k$-clique by iteratively joining a new vertex to some $k$-clique.

In other words, a graph is a $k$-tree $\iff$ there is an order $\pi = (v_1, v_2, \ldots, v_n)$ of its vertices such that the neighbors of $v_i$ preceding it in $\pi$ form a $\min\{i - 1, k\}$-clique, for all $1 \leq i \leq n$.

Observation

a) 1-trees are exactly trees.

b) A $k$-tree on $n \geq k$ vertices has $kn - \binom{k+1}{2}$ edges.

Definition
A graph is a partial $k$-tree if it is a subgraph of a $k$-tree.
Partial k-trees and treewidth

Theorem
A graph G is partial k-tree $\iff$ G has treewidth at most k.

Proof (‘$\Leftarrow$’): Let $\{X_i \mid i \in I\}$, $T = (I, F)$ be a smooth tree decomposition of width $\leq k$ for G. Add all edges inside bags to G.

Claim. The resulting graph $H$ is a k-tree.

Induction on $|I|$:

- $|I| = 1$: $H$ is $K_{k+1}$, a k-tree.
- Otherwise, let $i \in I$ be a leaf of $T$. Then there is a $v \in X_i$ that does not occur in any $X_j$, $j \in I \setminus \{i\}$. Removal of $i$ from $I$ results in a tree decomposition of width $\leq k$ for $H' = H \setminus v$.
- By induction $H'$ is a k-tree. Adding $v$ to $H'$ and connect it to the $k$-clique $X_i \setminus \{v\}$, we get $H$, which hence is a k-tree.
Partial $k$-trees and treewidth

**Theorem**

A graph $G$ is partial $k$-tree $\iff$ $G$ has treewidth at most $k$.

**Proof** ("$\Rightarrow$") Let $H$ be a $k$-tree containing $G$ and $
\pi = (v_1, \ldots, v_n)$ a vertex order for $H$ such that the neighbors of $v_i$
preceding it in $\pi$ form a min\{i − 1, $k$\}-clique, for all $1 \leq i \leq n$.
Build a tree decomposition of width $k$ for $V_i = \{v_1, \ldots, v_i\}$
inductively such that for every $j$, $1 \leq j \leq i$, there is a bag that
contains $\{v_j\} \cup \Pi_j$, where $\Pi_j = V_j \cap N_H(v_j)$.

- $i \leq k + 1$: A single bag for $V_i$ suffices.
- Otw, let $\ell = \max\{1 \leq \ell < i \mid v_\ell \in \Pi_j\}$. By the induction
  hypothesis there is a tree decomposition of width $k$ for $V_{i-1}$
in which one bag $X_a$ contains $\{v_\ell\} \cup \Pi_\ell$.
- Create a new node $b$, make it adjacent to $a$ only, and set
  $X_b = \{v_i\} \cup \Pi_i$. (Note that $\Pi_i \subseteq \{v_\ell\} \cup \Pi_\ell$ because $\Pi_i$ is a
  clique.)
Grids, minors, and treewidth

Theorem (Alon, Seymour, Thomas '90)
For any fixed graph \( H \), every graph \( G \) that does not contain \( H \) as a minor has treewidth at most \( n(H)^{3/2} \sqrt{n(G)} \).

Corollary
A planar graph on \( n \) vertices has treewidth \( O(\sqrt{n}) \).

Theorem ([Robertson, Seymour, Thomas ’94)
Every graph of treewidth larger than \( 20^{2k^5} \) has a \( k \times k \)-grid as a minor.
On the other hand, there are graphs of treewidth \( \Omega(k^2 \log k) \) that do not have a \( k \times k \)-grid as a minor.