Tree Decompositions and Tree-Width

CS 511

Iowa State University

December 6, 2010
Definition

A \textit{tree decomposition} of a graph $G = (V, E)$ consists of a tree $T$ and a subset $V_t \subseteq V$ for every node $t \in T$, such that the collection $\{V_t : t \in T\}$ satisfies:

- (Node coverage) For every $v \in V$, there is some node $t$ in $T$ such that $v \in V_t$.
- (Edge coverage) For every $e \in E$, there is some node $t$ in $T$ such that $V_t$ contains both endpoints of $e$.
- (Coherence) Let $t_1, t_2, t_3$ be three nodes in $T$ such that $t_2$ lies on the path between $t_1$ and $t_3$ in $T$. Then, if $v \in V$ belongs to both $V_{t_1}$ and $V_{t_3}$, $v$ must also belong to $V_{t_2}$. 
Tree-Width

Definition

The width of tree decomposition \((T, \{V_t : t \in T\})\) is

\[
\text{width}(T, \{V_t : t \in T\}) = \max_{t \in T} |V_t| - 1.
\]

Definition

The tree-width of \(G\), denoted \(\text{tw}(G)\), is the minimum width of a tree decomposition of \(G\).
Complexity of Tree-Width

Let $TW(k)$ be the class of graphs $G$ such that $tw(G) \leq k$.

**Tree-Width (Decision Version)**

**Input**: An undirected graph $G$ and an integer $k$.

**Question**: Is $G \in TW(k)$?

**Theorem**

*Tree-width (decision version) is NP-complete.*
Complexity of Tree-Width

Lemma

For every positive integer $k$, $TW(k)$ is minor closed.

Corollary (Tree-width is fixed-parameter tractable)

For every fixed $k$, the problem of determining whether or not $G \in TW(k)$ can be solved in $O(f(k) \cdot n^{O(1)})$ time.

- Corollary follows from Robertson & Seymour’s graph minor results.
  - $f(k)$ is superpolynomial, but depends only on $k$.
- Running time can be improved to $O(n)$ for each fixed $k$.
  - Simple $O(n)$ algorithms exist for $k \leq 4$. 
Notation

Let \((T, \{V_t : t \in T\})\) be a tree decomposition of \(G\). Then, if \(T'\) is a subgraph of \(T\), \(G_{T'}\) denotes the subgraph induced by the set \(\bigcup_{t \in T'} V_t\).
Theorem (Node Separation Property)

Suppose $T - t$ has components $T_1, \ldots, T_d$. Then, the subgraphs

$$G_{T_1} - V_t, G_{T_2} - V_t, \ldots, G_{T_d} - V_t$$

have no nodes in common, and there are no edges between them.
Theorem (Edge Separation Property)

Let $X$ and $Y$ be the two components of $T$ after the deletion of edge $(x, y)$. Then, deleting $V_x \cap V_y$ disconnects $G$ into two subgraphs $H_X = G_X - (V_x \cap V_y)$ and $H_Y = G_Y - (V_x \cap V_y)$. That is,

- $H_X$ and $H_Y$ share no nodes and
- there is no edge in $G$ with one endpoint in $H_X$ and the other in $H_Y$. 

\[ V_x \cap V_y \]
Definition

A tree decomposition \((T, \{V_t : t \in T\})\) of \(G\) is nonredundant if there is no edge \((x, y)\) in \(T\) such that \(V_x \subseteq V_y\).

Lemma

Any graph has a nonredundant tree decomposition.

Lemma

Any non-redundant tree decomposition of an \(n\)-node graph has at most \(n\) pieces.
Rooted tree decomposition

**Definition**

A rooted tree decomposition of $G$ is a tree decomposition $(T, \{V_t : t \in T\})$ of $G$ where some node $r$ in $T$ is declared to be the root.

Let $t$ be a node in a rooted tree decomposition. Then,

- $T_t$ is the subtree of $T$ rooted at $t$,
- $G_t$ is the subgraph of $G$ induced by the vertices in $\bigcup_{x \in T_t} V_x$. 

CS 511 (Iowa State University)  
Tree Decompositions and Tree-Width  
December 6, 2010  11 / 15
Subproblems

Definition

For each node $t$ in a rooted tree decomposition of $G$ and each independent set $U \subseteq V_t$, $\text{opt}_U(t)$ is the maximum weight of an independent set $S$ of $G_t$ such that $S \cap V_t = U$. 
Optimal Substructure

Let

- $t$ be a node in $T$ with children $t_1, \ldots, t_d$,
- $U$ be an independent set of $V_t$,
- $S$ be a maximum independent set in $G_t$ subject to $S \cap V_t = U$ (i.e., $w(S) = \text{opt}_U(t)$),
- $S_i$ be the intersection of $S$ with the nodes of $G_{T_i}$.

**Lemma (Optimal Substructure)**

$S_i$ is a maximum-weight independent set of $G_{T_i}$, subject to the constraint that $S_i \cap V_t = U \cap V_{t_i}$. 
Theorem (Dynamic Programming Recurrence Relation for MWIS)

$$\text{opt}_U(t) = w(U) + \sum_{i=1}^{d} \max \{ \text{opt}_{U_i}(t_i) - w(U_i \cap U) : U_i \subseteq V_{t_i} \text{ is independent and } U_i \cap V_t = U \cap V_{t_i} \}.$$
Theorem (Running time analysis)

Suppose we are given a vertex-weighted graph \( G \in \text{TW}(k) \) with \( n \) nodes along with a tree decomposition of width \( \leq k \) for \( G \). Then, we can find a maximum weight independent set in \( G \) in \( O(4^{k+1}kn) \) time.

Proof.

Traverse the tree decomposition bottom-up.

- For a leaf, use exhaustive enumeration \( \Rightarrow O(2^{k+1}) \) time.
- For an internal node \( t \), apply the recurrence relation.
  - Enumerate each of the \( O(2^{k+1}) \) subsets \( U \) of \( V_t \).
  - For each child \( t_i \) of \( t \), enumerate each of the \( O(2^{k+1}) \) subsets \( U_i \) of \( V_{t_i} \), checking that \( U_i \cap V_t = U \cap V_{t_i} \).

\[ \Rightarrow \quad \text{Time} = O\left( \frac{2^{k+1}}{\# U's} \times \frac{d}{\# \text{children}} \times \frac{2^{k+1}}{\# U_i's} \times \frac{k}{\text{checking } U_i} \right) = O(4^{k+1}kd). \]

- Total time \( = O(4^{k+1} k \sum_{t \in T} \text{degree}(t)) = O(4^{k+1} kn) \).