Design and Analysis of Algorithms (VII)

Treewidth

Guoqiang Li
School of Software

Shanghai Jiao Tong University

Polynomial Time on Trees

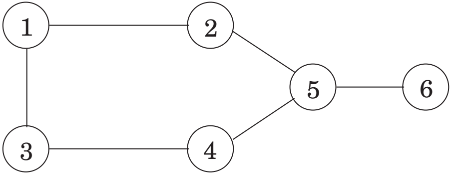
Shanghai Jiao Tong University

It is well known that many NP-hard problems can be solved in polynomial time on trees, i.e., Independent Set, Dominating Set, 3-Colorability, etc.

Independent Sets on Trees

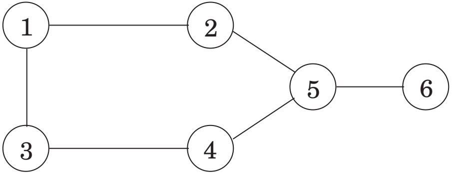
Independent Set

A subset of nodes $S \subseteq V$ is an independent set of graph $G = (V, E)$ if there are no edges between them.



Independent Set

A subset of nodes $S \subseteq V$ is an independent set of graph $G = (V, E)$ if there are no edges between them.



Finding the largest independent set in a graph is believed to be intractable.

However, when the graph happens to be a tree, the problem can be solved in linear time, using dynamic programming.

The Subproblems

$I(u) =$ size of largest independent set of subtree hanging from u.

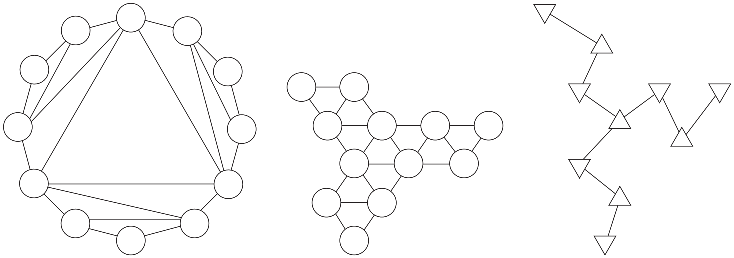
The Subproblems

$I(u)$ = size of largest independent set of subtree hanging from u.

$$I(u) = \max\{1 + \sum_{\text{grandchildren } w \text{ of } u} I(w), \sum_{\text{children } w \text{ of } u} I(w)\}$$

Independent Sets on Trees

An Example



Tree Decompositions of Graphs

SHANGHAI JIAO TONG UNIVERSITY

Definition

Let $\mathcal{G}$ be a graph. A tree decomposition of $\mathcal{G}$ is a tuple $(\mathcal{T}, (B_t)_{t\in V(\mathcal{T})})$, where $\mathcal{T}$ is a tree and B_t the bag at t such that the following conditions are satisfied:

T1 For every $v \in V(\mathcal{G})$ the set

$$T_v := \{t \in V(\mathcal{T}) \mid v \in B_t\}$$

is nonempty and connected in $\mathcal{T}$, i.e., $\mathcal{T}[T_v]$ is a subtree of $\mathcal{T}$.

T2 For every $e \in E(\mathcal{G})$ there exists a $t \in V(\mathcal{T})$ such that $e \subseteq B_t$.

Tree Decomposition of Graphs, Examples

The complete graphs $\mathcal{K}_n$ for $n \in \mathbb{N}$.

Tree Decomposition of Graphs, Examples

The complete graphs $\mathcal{K}_n$ for $n \in \mathbb{N}$.

The trees.

Tree Decomposition of Graphs, Examples

The complete graphs $\mathcal{K}_n$ for $n \in \mathbb{N}$.

The trees.

The grids $\mathcal{G}_{n \times n}$ for $n \in \mathbb{N}$.

Treewidth

The width of a tree decomposition $(\mathcal{T}, (B_t)_{t\in V(\mathcal{T})})$ is

$$width(\mathcal{T}, (B_t)_{t\in V(\mathcal{T})}) := \max\{|B_t| - 1 \mid t \in V(\mathcal{T})\}$$

The treewidth of $\mathcal{G}$ is

$$tw(\mathcal{G}) := \min\{width(\mathcal{T}, (B_t)_{t\in V(\mathcal{T})}) \mid (\mathcal{T}, (B_t)_{t\in V(\mathcal{T})}) \text{ is a tree decomposition of } \mathcal{G}\}$$

Treewidth, Examples

$tw(\mathcal{K}_n) = n - 1$ for the complete graphs $\mathcal{K}_n$.

Treewidth, Examples

$tw(\mathcal{K}_n) = n - 1$ for the complete graphs $\mathcal{K}_n$.

$tw(\mathcal{T}) = 1$ for every tree $\mathcal{K}$ of size at least 2.

Treewidth, Examples

SHANGHAI JIAO TONG UNIVERSITY

$tw(\mathcal{K}_n) = n - 1$ for the complete graphs $\mathcal{K}_n$.

$tw(\mathcal{T}) = 1$ for every tree $\mathcal{K}$ of size at least 2.

$tw(\mathcal{G}_{n \times n}) = n$ for every grid $\mathcal{G}_{n \times n}$.

Smooth Tree Decomposition

Definition

A tree decomposition $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ is smooth if for every $(t, t') \in E(\mathcal{T})$ we have

$$|B_t \backslash B_{t'}| = |B_{t'} \backslash B_t| = 1$$

Theorem

Every tree decomposition can be efficiently transferred to a smooth one of the same width.

Theorem

Every graph $\mathcal{G}$ has a smooth tree decomposition of width $tw(\mathcal{G})$.

Make Tree Decomposition Smooth

SHANGHAI JIAO TONG UNIVERSITY

Let $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ be a tree decomposition of width w.

Make Tree Decomposition Smooth

Let $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ be a tree decomposition of width w.

1. Make bags equal size: We choose a node $r \in V(\mathcal{T})$ with $|B_r| = w + 1$ as the root. Let t be a child of r with $|B_t| \leq w$. Clearly

$$|B_r \setminus B_t| + |B_t| \geq w + 1$$

Make Tree Decomposition Smooth

Let $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ be a tree decomposition of width w.

1. Make bags equal size: We choose a node $r \in V(\mathcal{T})$ with $|B_r| = w + 1$ as the root. Let t be a child of r with $|B_t| \leq w$. Clearly
$$|B_r \backslash B_t| + |B_t| \geq w + 1$$
We add $w + 1 - |B_t|$ vertices in $B_r \backslash B_t$ to B_t. After repeating this procedure recursively from the root to leaves, every bag has size $w + 1$.

Make Tree Decomposition Smooth

Let $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ be a tree decomposition of width w.

1. Make bags equal size: We choose a node $r \in V(\mathcal{T})$ with $|B_r| = w + 1$ as the root. Let t be a child of r with $|B_t| \leq w$. Clearly
$$|B_r \backslash B_t| + |B_t| \geq w + 1$$
We add $w + 1 - |B_t|$ vertices in $B_r \backslash B_t$ to B_t. After repeating this procedure recursively from the root to leaves, every bag has size $w + 1$.
2. Remove repetition: If there is an edge $(t, t') \in E(\mathcal{T})$ with $B_t = B_{t'}$, then we merge t' with t.

Make Tree Decomposition Smooth

Let $(\mathcal{T}, (B_t)_{t\in V(\mathcal{T})})$ be a tree decomposition of width w.

1. Make bags equal size: We choose a node $r \in V(\mathcal{T})$ with $|B_r| = w+1$ as the root. Let t be a child of r with $|B_t| \le w$. Clearly
$$|B_r \backslash B_t| + |B_t| \ge w+1$$
We add $w+1-|B_t|$ vertices in $B_r \backslash B_t$ to B_t. After repeating this procedure recursively from the root to leaves, every bag has size $w+1$.
2. Remove repetition: If there is an edge $(t,t') \in E(\mathcal{T})$ with $B_t = B_{t'}$, then we merge t' with t.
3. Interpolation: Let $(t,t') \in E(\mathcal{T})$ with $|B_t \backslash B_{t'}| < w$, i.e.,
$$B_t \backslash B_{t'} = \{u_1, \dots u_\ell\} \text{ and } B_{t'} \backslash B_t = \{v_1, \dots, v_\ell\}$$
for some $\ell > 2$ and pairwise distinct $u_1, \dots u_\ell$ and $v_1, \dots, v_\ell$. We insert new nodes $t_1, \dots t_{\ell-1}$ between t and t' with
$$B_{t_i} := B_t \cap B_{t'} \cup \{v_1, \dots, v_i, u_{i+1}, \dots, u_\ell\}$$
for every $i \in [\ell-1]$.

The Size of Smooth Tree Decompositions

Theorem

For every smooth tree decomposition $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ *of* $\mathcal{G}$ *we have*

$$|V(\mathcal{T})| \leq |V(\mathcal{G})|$$

The Size of Smooth Tree Decompositions

Theorem

For every smooth tree decomposition $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ *of* $\mathcal{G}$ *we have*

$$|V(\mathcal{T})| \leq |V(\mathcal{G})|$$

Theorem

$$|E(\mathcal{G})| \leq tw(\mathcal{G}) \cdot |V(\mathcal{G})|$$

$tw(\mathcal{K}_n) = n - 1$

$tw(\mathcal{K}_n) = n - 1$

$tw(\mathcal{K}_n) \leq n - 1$:

$tw(\mathcal{K}_n) = n - 1$

$tw(\mathcal{K}_n) \leq n - 1$: Take a tree decomposition with a singleton tree.

$tw(\mathcal{K}_n) = n - 1$

Shanghai Jiao Tong University

$tw(\mathcal{K}_n) \leq n - 1$: Take a tree decomposition with a singleton tree.

$tw(\mathcal{K}_n) \geq n - 1$:

$tw(\mathcal{K}_n) = n - 1$

$tw(\mathcal{K}_n) \leq n - 1$: Take a tree decomposition with a singleton tree.

$tw(\mathcal{K}_n) \geq n - 1$: Let $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ be a smooth tree decomposition of $\mathcal{K}_n$ of width $tw(\mathcal{K}_n)$. We show that there exists a B_t with $|B_t| = n$.

$tw(\mathcal{K}_n) = n - 1$

$tw(\mathcal{K}_n) \leq n - 1$: Take a tree decomposition with a singleton tree.

$tw(\mathcal{K}_n) \geq n - 1$: Let $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ be a smooth tree decomposition of $\mathcal{K}_n$ of width $tw(\mathcal{K}_n)$. We show that there exists a B_t with $|B_t| = n$.

Trivial if $|V(\mathcal{T})| = 1$. Otherwise choose a leaf t and let t' be its parent in $\mathcal{T}$.

$tw(\mathcal{K}_n) = n - 1$

SHANGHAI JIAO TONG UNIVERSITY

$tw(\mathcal{K}_n) \leq n-1$: Take a tree decomposition with a singleton tree.

$tw(\mathcal{K}_n) \geq n-1$: Let $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ be a smooth tree decomposition of $\mathcal{K}_n$ of width $tw(\mathcal{K}_n)$. We show that there exists a B_t with $|B_t| = n$.

Trivial if $|V(\mathcal{T})| = 1$. Otherwise choose a leaf t and let t' be its parent in $\mathcal{T}$.

By the smoothness

$$B_t \backslash B_{t'} = \{v\} \text{ for some } v \in V(\mathcal{K}_n)$$

$tw(\mathcal{K}_n) = n - 1$

Shanghai Jiao Tong University

$tw(\mathcal{K}_n) \leq n - 1$: Take a tree decomposition with a singleton tree.

$tw(\mathcal{K}_n) \geq n - 1$: Let $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ be a smooth tree decomposition of $\mathcal{K}_n$ of width $tw(\mathcal{K}_n)$. We show that there exists a B_t with $|B_t| = n$.

Trivial if $|V(\mathcal{T})| = 1$. Otherwise choose a leaf t and let t' be its parent in $\mathcal{T}$.

By the smoothness

$$B_t \backslash B_{t'} = \{v\} \text{ for some } v \in V(\mathcal{K}_n)$$

Since v is adjacent to every other vertex in $\mathcal{K}_n$, we see that $B_t = V(\mathcal{K}_n)$

Helly Property for Trees

Theorem

Let $\mathcal{T}$ be a tree and $\mathcal{T}_1, \ldots, \mathcal{T}_n$ subtrees of $\mathcal{T}$ such that

$$V(\mathcal{T}_i) \cap V(\mathcal{T}_j) \neq \emptyset$$

for every $i, j \in [n]$. Then

$$\bigcap_{i \in [n]} V(\mathcal{T}_i) \neq \emptyset$$

Proof

We prove by induction on the size of $\mathcal{T}$.

Proof

We prove by induction on the size of $\mathcal{T}$.

Basic step $|V(\mathcal{T})|$: Trivial.

Proof

We prove by induction on the size of $\mathcal{T}$.

Basic step $|V(\mathcal{T})|$: Trivial.

Hypothesis step

Proof

We prove by induction on the size of $\mathcal{T}$.

Basic step $|V(\mathcal{T})|$: Trivial.

Hypothesis step Let t be a leaf of $\mathcal{T}$. If $t \in V(\mathcal{T}_i)$ for every $i \in [n]$, then we are done.

Proof

We prove by induction on the size of $\mathcal{T}$.

Basic step $|V(\mathcal{T})|$: Trivial.

Hypothesis step Let t be a leaf of $\mathcal{T}$. If $t \in V(\mathcal{T}_i)$ for every $i \in [n]$, then we are done.

Now assume $t \notin V(\mathcal{T}_i)$ for some $i \in [n]$.

Proof

SHANGHAI JIAO TONG UNIVERSITY

We prove by induction on the size of $\mathcal{T}$.

Basic step $|V(\mathcal{T})|$: Trivial.

Hypothesis step Let t be a leaf of $\mathcal{T}$. If $t \in V(\mathcal{T}_i)$ for every $i \in [n]$, then we are done.

Now assume $t \notin V(\mathcal{T}_i)$ for some $i \in [n]$. Consider

$$\mathcal{T} \backslash \{t\}; \mathcal{T}_1 \backslash \{t\}, \ldots, \mathcal{T}_n \backslash \{t\}$$

Proof

We prove by induction on the size of $\mathcal{T}$.

Basic step $|V(\mathcal{T})|$: Trivial.

Hypothesis step Let t be a leaf of $\mathcal{T}$. If $t \in V(\mathcal{T}_i)$ for every $i \in [n]$, then we are done.

Now assume $t \notin V(\mathcal{T}_i)$ for some $i \in [n]$. Consider

$$\mathcal{T} \setminus \{t\}; \mathcal{T}_1 \setminus \{t\}, \ldots, \mathcal{T}_n \setminus \{t\}$$

Then

- every $\mathcal{T}_i \setminus \{t\}$ is a (nonempty) subtree of $\mathcal{T} \setminus \{t\}$.
- $V(\mathcal{T}_i \setminus \{t\}) \cap V(\mathcal{T}_j \setminus \{t\}) \neq \emptyset$ for every $i, j \in [n]$.

The result follows from the induction hypothesis.

$tw(\mathcal{K}_n) = n - 1$, Again

SHANGHAI JIAO TONG UNIVERSITY

Theorem

Let $\mathcal{G} = (V, E)$ *be a graph and* $S \subseteq V$ *a clique. Then for every tree decomposition* $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ *there is a node* $t \in V(\mathcal{T})$ *with* $S \subseteq B_t$ *.*

$tw(\mathcal{K}_n) = n - 1$, Again

Shanghai Jiao Tong University

Theorem

Let $\mathcal{G} = (V, E)$ *be a graph and* $S \subseteq V$ *a clique. Then for every tree decomposition* $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ *there is a node* $t \in V(\mathcal{T})$ *with* $S \subseteq B_t$ *.*

Proof.

$tw(\mathcal{K}_n) = n - 1$, Again

Theorem

Let $\mathcal{G} = (V, E)$ be a graph and $S \subseteq V$ a clique. Then for every tree decomposition $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ there is a node $t \in V(\mathcal{T})$ with $S \subseteq B_t$.

Proof.

For every $v \in V$ recall

$$T_v := \{t \in V(\mathcal{T}) \mid v \in B_t\}$$

induces a subtree $\mathcal{T}_v := T[\mathcal{T}_v]$ of $\mathcal{T}$.

$tw(\mathcal{K}_n) = n - 1$, Again

Theorem

Let $\mathcal{G} = (V, E)$ be a graph and $S \subseteq V$ a clique. Then for every tree decomposition $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ there is a node $t \in V(\mathcal{T})$ with $S \subseteq B_t$.

Proof.

For every $v \in V$ recall

$$T_v := \{t \in V(\mathcal{T}) \mid v \in B_t\}$$

induces a subtree $\mathcal{T}_v := T[T_v]$ of $\mathcal{T}$.

Clearly for every $u, v \in S$, we have

$$V(\mathcal{T}_u) \cap V(\mathcal{T}_v) = T_u \cap T_v \neq \emptyset$$

$tw(\mathcal{K}_n) = n - 1$, Again

Shanghai Jiao Tong University

Theorem

Let $\mathcal{G} = (V, E)$ be a graph and $S \subseteq V$ a clique. Then for every tree decomposition $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ there is a node $t \in V(\mathcal{T})$ with $S \subseteq B_t$.

Proof.

For every $v \in V$ recall

$$T_v := \{t \in V(\mathcal{T}) \mid v \in B_t\}$$

induces a subtree $\mathcal{T}_v := T[T_v]$ of $\mathcal{T}$.

Clearly for every $u, v \in S$, we have

$$V(\mathcal{T}_u) \cap V(\mathcal{T}_v) = T_u \cap T_v \neq \emptyset$$

since there is an edge (u, v) in $\mathcal{G}$. The result follows from Helly property.

Computing the Treewidth

Theorem (Bodlaender, 1996)

The problem

> TREEWIDTH
>
> INPUT: A graph $\mathcal{G}$ and a number $k \in \mathbb{N}$.
>
> PROBLEM: Decide whether $tw(\mathcal{G}) \leq k$ and if so output a tree decomposition of $\mathcal{G}$ with width $\leq k$.

can be computed in time

$$2^{k^{O(1)}} \cdot ||\mathcal{G}||$$

Computing the Treewidth

Corollary

For every $k \in \mathbb{N}$ *there is a linear time algorithm which on every graph* $\mathcal{G}$ *either outputs a tree decomposition of* $\mathcal{G}$ *of width* $\leq k$ *or reports that* $tw(\mathcal{G}) > k$.

Independent Sets via Tree Decompositions

Independent Sets via Tree Decompositions (1)

Let $\mathcal{G}$ be a graph and $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ a smooth tree decomposition of $\mathcal{G}$. And let $w := tw(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$.

Independent Sets via Tree Decompositions (1)

Let $\mathcal{G}$ be a graph and $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ a smooth tree decomposition of $\mathcal{G}$. And let $w := tw(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$.

We fix an arbitrary node $r \in V(\mathcal{T})$ as the root of $\mathcal{T}$.

Let $t \in V(\mathcal{T})$. We define $\mathcal{G}_t$ as the induced subgraph of $\mathcal{G}$ on vertices in B_t.

Furthermore, $\mathcal{G}_{\leq t}$ is the induced subgraph of $\mathcal{G}$ on vertices in

$$B_t \cup \bigcup_{\text{descendants } t' \text{ of } t} B_{t'}$$

Independent Sets via Tree Decompositions (2)

Shanghai Jiao Tong University

By dynamic programming we compute for every $t \in V(\mathcal{T})$ and every $X \subseteq B_t$ independent in $\mathcal{G}_t$

$$I(t,X) := \text{size of a largest independent set } I \text{ of } \mathcal{G}_{\leq t} \text{ with } I \cap B_t = X$$

$$= |X| + \sum_{\text{children } t' \text{ of } t} \max\{I(t',X') - |X' \cap X| \mid X' \cap B_t = X\ capB_{t'}\}$$

Independent Sets via Tree Decompositions (2)

Shanghai Jiao Tong University

By dynamic programming we compute for every $t \in V(\mathcal{T})$ and every $X \subseteq B_t$ independent in $\mathcal{G}_t$

$$I(t, X) \quad := \text{ size of a largest independent set } I \text{ of } \mathcal{G}_{\le t} \text{ with } I \cap B_t = X$$

$$= |X| + \sum_{\text{children } t' \text{ of } t} \max\{I(t', X') - |X' \cap X| \mid X' \cap B_t = X\ capB_{t'}\}$$

Note there are at most

$$|V(\mathcal{T})| \cdot 2^{w+1} \le |V(\mathcal{G})| \cdot 2^{w+1}$$

many $I(t, X)$.

Independent Sets via Tree Decompositions (3)

Shanghai Jiao Tong University

Theorem

For every $k \in \mathbb{N}$ there is a linear time algorithm which on every graph $\mathcal{G}$ with $tw(\mathcal{G}) \leq k$ outputs a largest independent set in $\mathcal{G}$.

Partial k-Trees

Definition (k-trees)

Let $k \in \mathbb{N}$. Then the set of k-trees is defined as follows.

K1 A complete graph $\mathcal{K}_{k+1}$ is a k-tree.

K2 Let $\mathcal{G}$ be a graph and $v \in V$ such that

- $\mathcal{N}^G[v]$ is isomorphic to $\mathcal{K}_{k+1}$, where $\mathcal{N}^G[v]$ is the induced subgraph of $\mathcal{G}$ on

$$\mathcal{N}^G[v] := \{u \in V(\mathcal{G}) \mid (u, v) \in E(\mathcal{G})\} \cup \{v\}$$

- $\mathcal{G}[V(\mathcal{G}) \backslash \{v\}]$ is a k-tree.

Then $\mathcal{G}$ is a k-tree.

k-Trees and Partial k-Trees

Shanghai Jiao Tong University

Definition (k-trees)

Let $k \in \mathbb{N}$. Then the set of k-trees is defined as follows.

K1 A complete graph $\mathcal{K}_{k+1}$ is a k-tree.

K2 Let $\mathcal{G}$ be a graph and $v \in V$ such that

- $\mathcal{N}^G[v]$ is isomorphic to $\mathcal{K}_{k+1}$, where $\mathcal{N}^G[v]$ is the induced subgraph of $\mathcal{G}$ on

$$\mathcal{N}^G[v] := \{u \in V(\mathcal{G}) \mid (u, v) \in E(\mathcal{G})\} \cup \{v\}$$

- $\mathcal{G}[V(\mathcal{G}) \backslash \{v\}]$ is a k-tree.

Then $\mathcal{G}$ is a k-tree.

Definition (partial k-tree)

A graph is a partial k-tree if it is a subgraph of a k-tree.

Partial k-Trees and Bounded Treewidth

SHANGHAI JIAO TONG UNIVERSITY

Theorem

A graph $\mathcal{G}$ is a partial k-tree if and only if $tw(G) \leq k$.

Partial k-Trees and Bounded Treewidth

Theorem

A graph $\mathcal{G}$ is a partial k-tree if and only if $tw(G) \leq k$.

Lemma

Let $\mathcal{G}$ be a subgraph of $\mathcal{H}$, i.e., $V(\mathcal{G}) \subseteq V(\mathcal{H})$ and $E(\mathcal{G}) \subseteq E(\mathcal{H})$. Then $tw(\mathcal{G}) \leq tw(\mathcal{H})$.

Partial k-Trees and Bounded Treewidth

Shanghai Jiao Tong University

Theorem

A graph $\mathcal{G}$ is a partial k-tree if and only if $tw(G) \leq k$.

Lemma

Let $\mathcal{G}$ be a subgraph of $\mathcal{H}$, i.e., $V(\mathcal{G}) \subseteq V(\mathcal{H})$ and $E(\mathcal{G}) \subseteq E(\mathcal{H})$. Then $tw(\mathcal{G}) \leq tw(\mathcal{H})$.

Theorem

1. *Every graph of treewidth $\leq k$ is a partial k-tree.*
2. *Every k-tree has a tree decomposition of width $\leq k$.*

Proof

Let $\mathcal{G}$ be a graph with $tw(\mathcal{G}) \leq k$. Moreover, let $\mathscr{T} = (\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ be a smooth tree decomposition of $\mathcal{G}$ of width k.

Proof

Let $\mathcal{G}$ be a graph with $tw(\mathcal{G}) \leq k$. Moreover, let $\mathscr{T} = (\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ be a smooth tree decomposition of $\mathcal{G}$ of width k.

From $\mathcal{T}$ we define a graph $\mathcal{H}_{\mathscr{T}}$ by induction on $V(\mathcal{T})$ such that

(H1) $\mathcal{H}_{\mathscr{T}}$ is a k-tree.

(H2) $\mathcal{H}_{\mathscr{T}}[B_t]$ is isomorphic to $\mathcal{K}_{k+1}$ for every $t \in V(\mathcal{T})$.

(H3) $\mathcal{G} \subseteq \mathcal{H}_{\mathscr{T}}$.

Proof

Let $\mathcal{G}$ be a graph with $tw(\mathcal{G}) \leq k$. Moreover, let $\mathscr{T} = (\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ be a smooth tree decomposition of $\mathcal{G}$ of width k.

From $\mathcal{T}$ we define a graph $\mathcal{H}_{\mathscr{T}}$ by induction on $V(\mathcal{T})$ such that

(H1) $\mathcal{H}_{\mathscr{T}}$ is a k-tree.

(H2) $\mathcal{H}_{\mathscr{T}}[B_t]$ is isomorphic to $\mathcal{K}_{k+1}$ for every $t \in V(\mathcal{T})$.

(H3) $\mathcal{G} \subseteq \mathcal{H}_{\mathscr{T}}$.

If $|V(\mathcal{T})| = 1$, then $\mathcal{H}_{\mathscr{T}}$ is $\mathcal{K}_{k+1}$, and we are done.

Proof

Let $\mathcal{G}$ be a graph with $tw(\mathcal{G}) \leq k$. Moreover, let $\mathscr{T} = (\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ be a smooth tree decomposition of $\mathcal{G}$ of width k.

From $\mathcal{T}$ we define a graph $\mathcal{H}_{\mathscr{T}}$ by induction on $V(\mathcal{T})$ such that

(H1) $\mathcal{H}_{\mathscr{T}}$ is a k-tree.

(H2) $\mathcal{H}_{\mathscr{T}}[B_t]$ is isomorphic to $\mathcal{K}_{k+1}$ for every $t \in V(\mathcal{T})$.

(H3) $\mathcal{G} \subseteq \mathcal{H}_{\mathscr{T}}$.

If $|V(\mathcal{T})| = 1$, then $\mathcal{H}_{\mathscr{T}}$ is $\mathcal{K}_{k+1}$, and we are done.

Otherwise, choose a leaf t and let t' be its parent in $\mathcal{T}$. Therefore, $B_t \backslash B_{t'} = \{v\}$ for some $v \in V(\mathcal{K}_n)$. Then $\mathscr{T}' := (\mathcal{T} \backslash \{t\}, (B_t)_{t \in V(\mathcal{T} \backslash \{t\})})$ is a smooth tree decomposition of the graph $\mathcal{G} \backslash \{v\}$.

Proof

Let $\mathcal{G}$ be a graph with $tw(\mathcal{G}) \leq k$. Moreover, let $\mathscr{T} = (\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ be a smooth tree decomposition of $\mathcal{G}$ of width k.

From $\mathcal{T}$ we define a graph $\mathcal{H}_{\mathscr{T}}$ by induction on $V(\mathcal{T})$ such that

(H1) $\mathcal{H}_{\mathscr{T}}$ is a k-tree.

(H2) $\mathcal{H}_{\mathscr{T}}[B_t]$ is isomorphic to $\mathcal{K}_{k+1}$ for every $t \in V(\mathcal{T})$.

(H3) $\mathcal{G} \subseteq \mathcal{H}_{\mathscr{T}}$.

If $|V(\mathcal{T})| = 1$, then $\mathcal{H}_{\mathscr{T}}$ is $\mathcal{K}_{k+1}$, and we are done.

Otherwise, choose a leaf t and let t' be its parent in $\mathcal{T}$. Therefore, $B_t \backslash B_{t'} = \{v\}$ for some $v \in V(\mathcal{K}_n)$. Then $\mathscr{T}' := (\mathcal{T} \backslash \{t\}, (B_t)_{t \in V(\mathcal{T} \backslash \{t\})})$ is a smooth tree decomposition of the graph $\mathcal{G} \backslash \{v\}$.

By (H2) of the induction hypothesis, $\mathcal{H}_{\mathscr{T}'}[B_t \cap B_{t'}]$ is isomorphic to $\mathcal{K}_k$. Then from $\mathcal{H}_{\mathscr{T}'}$, we obtain $\mathcal{H}_{\mathscr{T}}$ by adding the vertex v and the edges (v, u) for every $u \in B_t \cap B_{t'}$.

Proof (Con't)

Let $\mathcal{H}$ be a k-tree. We show that $tw(\mathcal{H}) \leq k$ by induction on the construction of $\mathcal{H}$.

Proof (Con't)

Let $\mathcal{H}$ be a k-tree. We show that $tw(\mathcal{H}) \leq k$ by induction on the construction of $\mathcal{H}$.

If $\mathcal{H}$ is isomorphic to $\mathcal{K}_{k+1}$, i.e., (K1), then we are done.

Proof (Con't)

Let $\mathcal{H}$ be a k-tree. We show that $tw(\mathcal{H}) \leq k$ by induction on the construction of $\mathcal{H}$.

If $\mathcal{H}$ is isomorphic to $\mathcal{K}_{k+1}$, i.e., (K1), then we are done.

Otherwise by (K2) let $v \in V(\mathcal{H})$ satisfy that $\mathcal{H} \backslash \{v\}$ is a k-tree and $\mathcal{N}^{\mathcal{G}}[v]$ is isomorphic to $\mathcal{K}_{k+1}$.

Proof (Con't)

Let $\mathcal{H}$ be a k-tree. We show that $tw(\mathcal{H}) \le k$ by induction on the construction of $\mathcal{H}$.

If $\mathcal{H}$ is isomorphic to $\mathcal{K}_{k+1}$, i.e., (K1), then we are done.

Otherwise by (K2) let $v \in V(\mathcal{H})$ satisfy that $\mathcal{H} \backslash \{v\}$ is a k-tree and $\mathcal{N}^{\mathcal{G}}[v]$ is isomorphic to $\mathcal{K}_{k+1}$.

By induction hypothesis, there is a tree decomposition $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ of $\mathcal{H} \backslash \{v\}$ of width k. As

$$\mathcal{N}^{\mathcal{H}}(v) := \{u \in V(\mathcal{H}) \mid (u, v) \in E(\mathcal{H})$$

is a clique in $\mathcal{H} \backslash \{v\}$, by Helly property, there is a B_t with $\mathcal{N}^{\mathcal{H}}(v) \subseteq B_t$.

Proof (Con't)

Let $\mathcal{H}$ be a k-tree. We show that $tw(\mathcal{H}) \leq k$ by induction on the construction of $\mathcal{H}$.

If $\mathcal{H}$ is isomorphic to $\mathcal{K}_{k+1}$, i.e., (K1), then we are done.

Otherwise by (K2) let $v \in V(\mathcal{H})$ satisfy that $\mathcal{H} \backslash \{v\}$ is a k-tree and $\mathcal{N}^{\mathcal{G}}[v]$ is isomorphic to $\mathcal{K}_{k+1}$.

By induction hypothesis, there is a tree decomposition $(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})})$ of $\mathcal{H} \backslash \{v\}$ of width k. As

$$\mathcal{N}^{\mathcal{H}}(v) := \{u \in V(\mathcal{H}) \mid (u, v) \in E(\mathcal{H})$$

is a clique in $\mathcal{H} \backslash \{v\}$, by Helly property, there is a B_t with $\mathcal{N}^{\mathcal{H}}(v) \subseteq B_t$.

We add a new leaf t' adjacent to t and set $B_{t'} := \mathcal{N}^{\mathcal{H}}[v]$.