Additive Tree $O(\rho \log n)$-Spanners from Tree Breadth ρ

Dieter Rautenbach

Universität Ulm

Additive Tree $O(\rho \log n)$-Spanners from Tree Breadth ρ

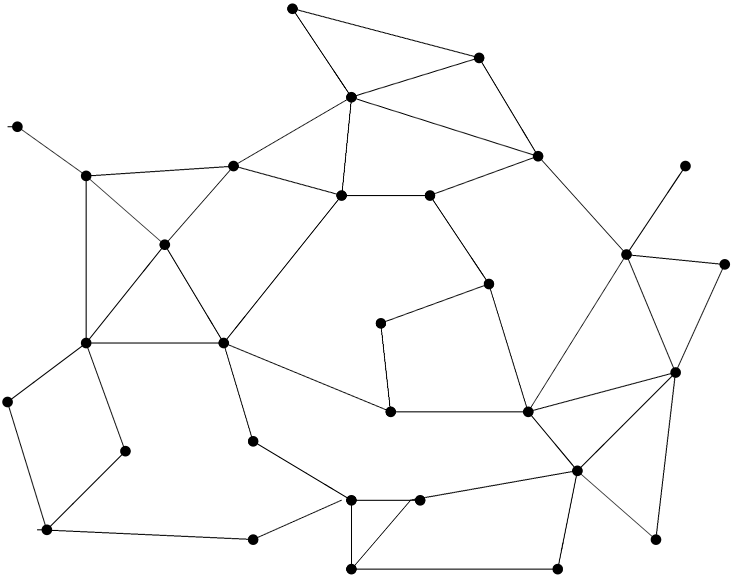
Dieter Rautenbach

Universität Ulm

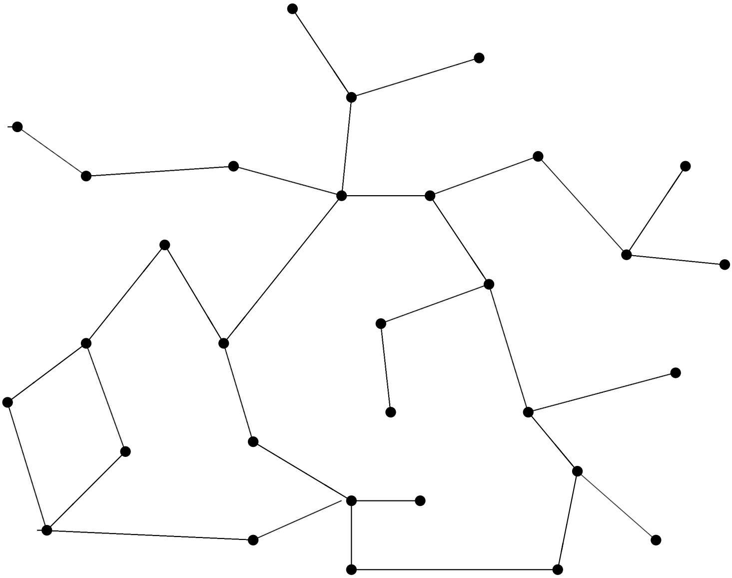
Joint with Oliver Bendele

Spanners

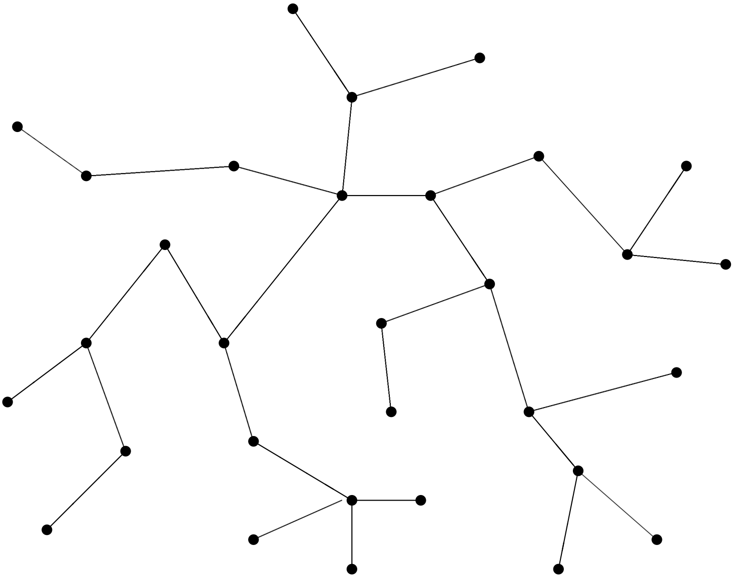
Spanners



Spanners



Spanners



Spanners

Definition (Additive tree k-spanner)

Spanners

Definition (Additive tree k-spanner)

Let H be a subgraph of a graph G.

Spanners

Definition (Additive tree k-spanner)

Let H be a subgraph of a graph G.

- H is k-additive if

$$d_H(u, v) \leq d_G(u, v) + k$$

  for every two vertices u and v of H.

Spanners

Definition (Additive tree k-spanner)

Let H be a subgraph of a graph G.

- H is k-additive if
$$d_H(u, v) \leq d_G(u, v) + k$$
for every two vertices u and v of H.
- H is an additive k-spanner if H is spanning and k-additive.

Spanners

Definition (Additive tree k-spanner)

Let H be a subgraph of a graph G.

- H is k-additive if

$$d_H(u, v) \leq d_G(u, v) + k$$

  for every two vertices u and v of H.
- H is an additive k-spanner if H is spanning and k-additive.
- H is an additive tree k-spanner if H is a tree and an additive k-spanner.

Spanners

Definition (Additive tree k-spanner)

Let H be a subgraph of a graph G.

- H is k-additive if
$$d_H(u, v) \leq d_G(u, v) + k$$
for every two vertices u and v of H.
- H is an additive k-spanner if H is spanning and k-additive.
- H is an additive tree k-spanner if H is a tree and an additive k-spanner.

Replacing
$$\text{“}d_H(u, v) \leq d_G(u, v) + k\text{”}$$
with
$$\text{“}d_H(u, v) \leq k \cdot d_G(u, v)\text{”}$$
yields the multiplicative versions.

Tree breadth

Tree breadth

For a set U of vertices of a graph G, the radius of U in G is

Tree breadth

For a set U of vertices of a graph G, the radius of U in G is

$$u$$

Tree breadth

For a set U of vertices of a graph G, the radius of U in G is

$$\max\left\{d_G(u, v) : v \in U\right\}$$

Tree breadth

For a set U of vertices of a graph G, the radius of U in G is

$$\min\Big\{\max\Big\{d_G(u,v) : v \in U\Big\} : u \in V(G)\Big\}.$$

Tree breadth

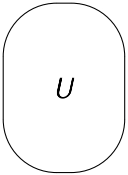
For a set U of vertices of a graph G, the radius of U in G is

$$\mathrm{rad}_G(U) = \min\Big\{ \max\Big\{ d_G(u,v) : v \in U \Big\} : u \in V(G) \Big\}.$$

Tree breadth

For a set U of vertices of a graph G, the radius of U in G is

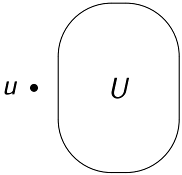
$$\mathrm{rad}_G(U) = \min\Big\{ \max\Big\{ d_G(u,v) : v \in U \Big\} : u \in V(G) \Big\}.$$



Tree breadth

For a set U of vertices of a graph G, the radius of U in G is

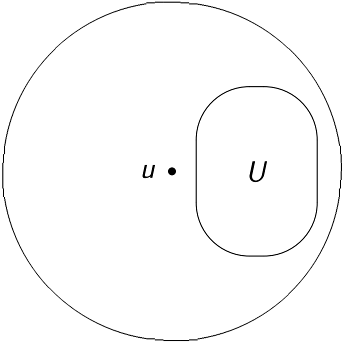
$$\mathrm{rad}_G(U) = \min\Big\{\max\Big\{d_G(u,v) : v \in U\Big\} : u \in V(G)\Big\}.$$



Tree breadth

For a set U of vertices of a graph G, the radius of U in G is

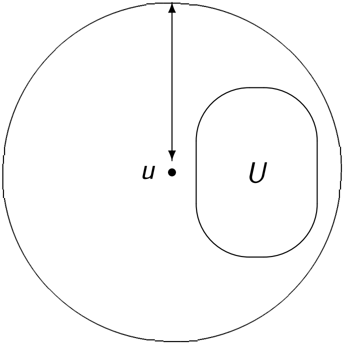
$$\mathrm{rad}_G(U) = \min\Big\{\max\Big\{d_G(u,v) : v \in U\Big\} : u \in V(G)\Big\}.$$



Tree breadth

For a set U of vertices of a graph G, the radius of U in G is

$$\mathrm{rad}_G(U) = \min\Big\{ \max\Big\{ d_G(u,v) : v \in U \Big\} : u \in V(G) \Big\}.$$



Tree breadth

Definition (Dragan and Köhler 2014)

Tree breadth

Definition (Dragan and Köhler 2014)

The breadth of a tree decomposition $\big(T, (X_t)_{t \in V(T)}\big)$ of a connected graph G is

$$\max \Big\{ \mathrm{rad}_G(X_t) : t \in V(T) \Big\}.$$

Tree breadth

Definition (Dragan and Köhler 2014)

The breadth of a tree decomposition $\big(T, (X_t)_{t \in V(T)}\big)$ of a connected graph G is

$$\max\Big\{\mathrm{rad}_G(X_t) : t \in V(T)\Big\}.$$

The tree breadth $\mathrm{tb}(G)$ of G is the minimum breadth of a tree decomposition of G.

Tree breadth

Definition (Dragan and Köhler 2014)

The breadth of a tree decomposition $\left(T, (X_t)_{t \in V(T)}\right)$ of a connected graph G is

$$\max \left\{ \mathrm{rad}_G(X_t) : t \in V(T) \right\}.$$

The tree breadth $\mathrm{tb}(G)$ of G is the minimum breadth of a tree decomposition of G.

- (Ducoffe, Legay, and Nisse 2019)
  Tree breadth is NP-hard.

Tree breadth

Definition (Dragan and Köhler 2014)

The breadth of a tree decomposition $\left(T, (X_t)_{t \in V(T)}\right)$ of a connected graph G is

$$\max\left\{\mathrm{rad}_G(X_t) : t \in V(T)\right\}.$$

The tree breadth $\mathrm{tb}(G)$ of G is the minimum breadth of a tree decomposition of G.

- (Ducoffe, Legay, and Nisse 2019)
  Tree breadth is NP-hard.
- (Dourisboure and Gavoille 2007, for tree length)
  A tree decomposition of breadth at most $6\mathrm{tb}(G) + 1$ can be found in linear time.

Tree breadth $\Rightarrow$ spanners

Tree breadth $\Rightarrow$ spanners

Let G be a connected graph of order n, size m, and tree breadth ρ.

Theorem (Dragan and Köhler 2014)

Given G as above, one can construct in time $O(m \log n)$ a multiplicative tree $O(\rho \log n)$-spanner of G.

Tree breadth ⇒ spanners

Let G be a connected graph of order n, size m, and tree breadth ρ.

Theorem (Dragan and Köhler 2014)

Given G as above, one can construct in time $O(m \log n)$ a multiplicative tree $O(\rho \log n)$-spanner of G.

Theorem (Dragan and Abu-Ata 2014)

Given G as above, one can efficiently construct a collection of $O(\log n)$ collective additive tree $O(\rho \log n)$-spanners,

Tree breadth $\Rightarrow$ spanners

Let G be a connected graph of order n, size m, and tree breadth ρ.

Theorem (Dragan and Köhler 2014)

Given G as above, one can construct in time $O(m \log n)$ a multiplicative tree $O(\rho \log n)$-spanner of G.

Theorem (Dragan and Abu-Ata 2014)

Given G as above, one can efficiently construct a collection of $O(\log n)$ collective additive tree $O(\rho \log n)$-spanners, that is, spanning trees $T_1, \ldots, T_{O(\log n)}$

Tree breadth ⇒ spanners

Let G be a connected graph of order n, size m, and tree breadth ρ.

Theorem (Dragan and Köhler 2014)

Given G as above, one can construct in time $O(m \log n)$ a multiplicative tree $O(\rho \log n)$-spanner of G.

Theorem (Dragan and Abu-Ata 2014)

Given G as above, one can efficiently construct a collection of $O(\log n)$ collective additive tree $O(\rho \log n)$-spanners, that is, spanning trees $T_1, \ldots, T_{O(\log n)}$ such that, for every two vertices u and v of G,

Tree breadth ⇒ spanners

Let G be a connected graph of order n, size m, and tree breadth ρ.

Theorem (Dragan and Köhler 2014)

Given G as above, one can construct in time $O(m \log n)$ a multiplicative tree $O(\rho \log n)$-spanner of G.

Theorem (Dragan and Abu-Ata 2014)

Given G as above, one can efficiently construct a collection of $O(\log n)$ collective additive tree $O(\rho \log n)$-spanners, that is, spanning trees $T_1, \ldots, T_{O(\log n)}$ such that, for every two vertices u and v of G, there is some tree T_i with

Tree breadth ⇒ spanners

Let G be a connected graph of order n, size m, and tree breadth ρ.

Theorem (Dragan and Köhler 2014)

Given G as above, one can construct in time $O(m \log n)$ a multiplicative tree $O(\rho \log n)$-spanner of G.

Theorem (Dragan and Abu-Ata 2014)

Given G as above, one can efficiently construct a collection of $O(\log n)$ collective additive tree $O(\rho \log n)$-spanners, that is, spanning trees $T_1, \ldots, T_{O(\log n)}$ such that, for every two vertices u and v of G, there is some tree T_i with

$$d_{T_i}(u, v) \leq d_G(u, v) + O(\rho \log n).$$

Tree breadth ⇒ spanners

Theorem (Bendele and R 2020)

Given G as above, one can construct in time $O(m \log n)$ an additive tree $O(\rho \log n)$-spanner of G.

Tree breadth ⇒ spanners

Theorem (Bendele and R 2020)

Given G as above, one can construct in time $O(m \log n)$ an additive tree $O(\rho \log n)$-spanner of G.

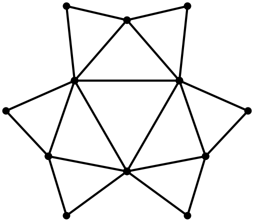


Figure: The graph G_3.

Tree breadth ⇒ spanners

Theorem (Bendele and R 2020)

Given G as above, one can construct in time $O(m \log n)$ an additive tree $O(\rho \log n)$-spanner of G.

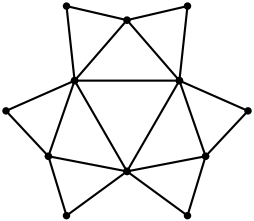


Figure: The graph G_3.

(Kratsch, Le, Müller, Prisner, and Wagner 2002)

G_k admits no additive tree $\mathrm{tb}(G_k) \log_2 \left(\frac{n(G_k)}{3}\right)$-spanner

Tree breadth ⇒ spanners

Theorem (Bendele and R 2020)

Given G as above, one can construct in time $O\big(m \log n\big)$ an additive tree $O\big(\rho \log n\big)$-spanner of G.

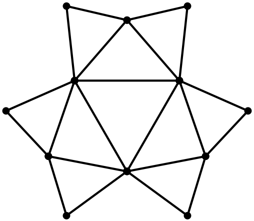


Figure: The graph G_3.

(Kratsch, Le, Müller, Prisner, and Wagner 2002)

G_k admits no additive tree $\mathrm{tb}(G_k) \log_2 \left(\frac{n(G_k)}{3}\right)$-spanner but $\mathrm{tb}(G_k) = 1$.

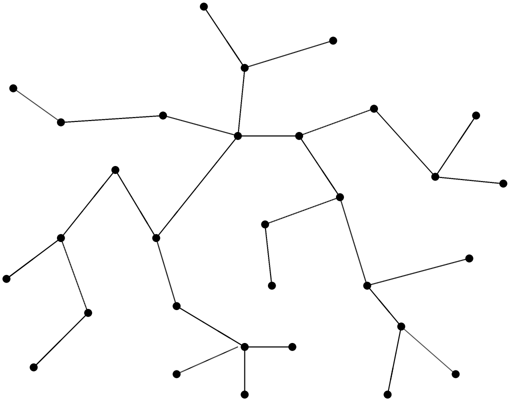
Tree breadth ⇒ spanners

Tree breadth ⇒ spanners

For a tree T, let $\mathrm{pbt}(T)$ be the maximum depth of a perfect binary tree that is a topological minor of T.

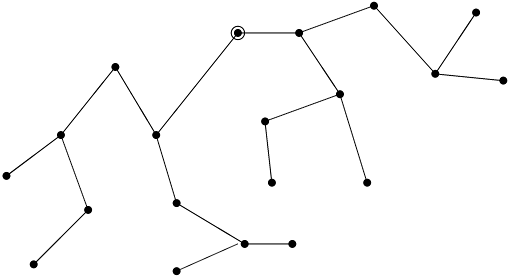
Tree breadth ⇒ spanners

For a tree T, let $\mathrm{pbt}(T)$ be the maximum depth of a perfect binary tree that is a topological minor of T.



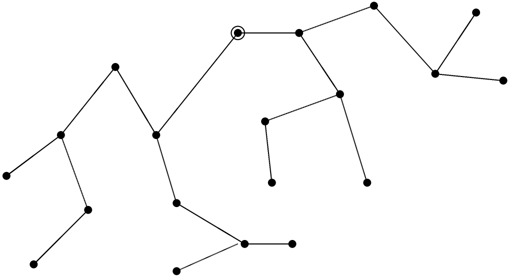
Tree breadth ⇒ spanners

For a tree T, let $\mathrm{pbt}(T)$ be the maximum depth of a perfect binary tree that is a topological minor of T.



Tree breadth ⇒ spanners

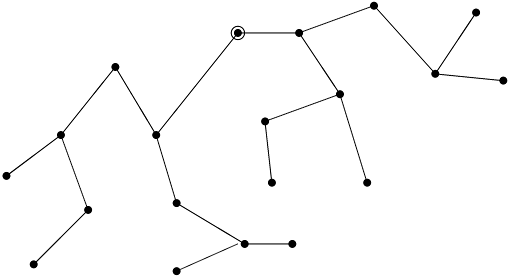
For a tree T, let $\mathrm{pbt}(T)$ be the maximum depth of a perfect binary tree that is a topological minor of T.



$\Rightarrow 3 \leq \mathrm{pbt}(T)$

Tree breadth ⇒ spanners

For a tree T, let $\mathrm{pbt}(T)$ be the maximum depth of a perfect binary tree that is a topological minor of T.



$\Rightarrow 3 \leq \mathrm{pbt}(T) \leq O(\log(n(T)))$.

Tree breadth $\Rightarrow$ spanners

Theorem (Bendele and R 2020)

Given G and tree decomposition $\big(T, (X_t)_{t \in V(T)}\big)$ of G of breadth ρ, one can construct in time $O\big(m \cdot \mathrm{pbt}(T)\big)$ an

additive tree $O\big(\rho \cdot \mathrm{pbt}(T)\big)$-spanner of G.

Tree breadth $\Rightarrow$ spanners

Theorem (Bendele and R 2020)

Given G and tree decomposition $\big(T, (X_t)_{t \in V(T)}\big)$ of G of breadth ρ, one can construct in time $O\big(m \cdot \mathrm{pbt}(T)\big)$ an

additive tree $O\big(\rho \cdot \mathrm{pbt}(T)\big)$-spanner of G.

Corollary (Bendele and R 2020)

Given G and given a

multiplicative tree k-spanner T of G,

one can construct in time $O\big(mn\big)$ an

additive tree $O\big(k \log n(G)\big)$-spanner of G.

Tree breadth $\Rightarrow$ spanners

- Allowing more edges leads to better spanners...

Tree breadth $\Rightarrow$ spanners

- Allowing more edges leads to better spanners...

  (Dourisboure, Dragan, Gavoille, and Yan 2007)
  G has an additive $O(\rho)$-spanner with $O(\rho n)$ edges.

Tree breadth ⇒ spanners

- Allowing more edges leads to better spanners...

  (Dourisboure, Dragan, Gavoille, and Yan 2007)
  G has an additive $O(\rho)$-spanner with $O(\rho n)$ edges.

- Good additive tree spanners for special graphs.

Tree breadth $\Rightarrow$ spanners

- Allowing more edges leads to better spanners...

  (Dourisboure, Dragan, Gavoille, and Yan 2007)
  G has an additive $O(\rho)$-spanner with $O(\rho n)$ edges.

- Good additive tree spanners for special graphs.

- ...

Proofs

For a tree T, let $L(T)$ be the set of leaves of T.

Proofs

For a tree T, let $L(T)$ be the set of leaves of T.

Lemma (Bendele and R 2020)

Given G, a subtree S of G, and a set U of vertices of G,

Proofs

For a tree T, let $L(T)$ be the set of leaves of T.

Lemma (Bendele and R 2020)

Given G, a subtree S of G, and a set U of vertices of G, one can construct in time $O(m)$ a subtree S' of G such that

Proofs

For a tree T, let $L(T)$ be the set of leaves of T.

Lemma (Bendele and R 2020)

Given G, a subtree S of G, and a set U of vertices of G, one can construct in time $O(m)$ a subtree S' of G such that

- $S \subseteq S'$ *and* $U \subseteq V(S')$,

Proofs

For a tree T, let $L(T)$ be the set of leaves of T.

Lemma (Bendele and R 2020)

Given G, a subtree S of G, and a set U of vertices of G, one can construct in time $O(m)$ a subtree S' of G such that

- $S \subseteq S'$ *and* $U \subseteq V(S')$,
- $d_{S'}(u, V(S)) = d_G(u, V(S))$ *for every vertex* u *in* U, *and*

Proofs

For a tree T, let $L(T)$ be the set of leaves of T.

Lemma (Bendele and R 2020)

Given G, a subtree S of G, and a set U of vertices of G, one can construct in time $O(m)$ a subtree S' of G such that

- $S \subseteq S'$ *and* $U \subseteq V(S')$,
- $d_{S'}(u, V(S)) = d_G(u, V(S))$ *for every vertex* u *in* U, *and*
- $L(S') \subseteq L(S) \cup U$.

Proofs

For a tree T, let $L(T)$ be the set of leaves of T.

Lemma (Bendele and R 2020)

Given G, a subtree S of G, and a set U of vertices of G, one can construct in time $O(m)$ a subtree S' of G such that

- $S \subseteq S'$ *and* $U \subseteq V(S')$,
- $d_{S'}(u, V(S)) = d_G(u, V(S))$ *for every vertex* u *in* U, *and*
- $L(S') \subseteq L(S) \cup U$.

Proof.

Contract S to r,

Proofs

For a tree T, let $L(T)$ be the set of leaves of T.

Lemma (Bendele and R 2020)

Given G, a subtree S of G, and a set U of vertices of G, one can construct in time $O(m)$ a subtree S' of G such that

- $S \subseteq S'$ *and* $U \subseteq V(S')$,
- $d_{S'}(u, V(S)) = d_G(u, V(S))$ *for every vertex u in U, and*
- $L(S') \subseteq L(S) \cup U$.

Proof.

Contract S to r, breadth first search from r,

Proofs

For a tree T, let $L(T)$ be the set of leaves of T.

Lemma (Bendele and R 2020)

Given G, a subtree S of G, and a set U of vertices of G, one can construct in time $O(m)$ a subtree S' of G such that

- $S \subseteq S'$ *and* $U \subseteq V(S')$,
- $d_{S'}(u, V(S)) = d_G(u, V(S))$ *for every vertex* u *in* U, *and*
- $L(S') \subseteq L(S) \cup U$.

Proof.

Contract S to r, breadth first search from r, uncontract S.

Proofs

For a tree T, let $L(T)$ be the set of leaves of T.

Lemma (Bendele and R 2020)

Given G, a subtree S of G, and a set U of vertices of G, one can construct in time $O(m)$ a subtree S' of G such that

- $S \subseteq S'$ *and* $U \subseteq V(S')$,
- $d_{S'}(u, V(S)) = d_G(u, V(S))$ *for every vertex u in U, and*
- $L(S') \subseteq L(S) \cup U$.

Proof.

Contract S to r, breadth first search from r, uncontract S. □

Proofs

Inspired by a lemma of Kratsch, Le, Müller, Prisner, and Wagner (2002):

Proofs

Inspired by a lemma of Kratsch, Le, Müller, Prisner, and Wagner (2002):

Lemma (Bendele and R 2020)

Given G and a ρ-additive subtree S of G such that

$$d_G(u, V(S)) \leq \rho' \text{ for every vertex } u \text{ of } G,$$

Proofs

Inspired by a lemma of Kratsch, Le, Müller, Prisner, and Wagner (2002):

Lemma (Bendele and R 2020)

Given G and a ρ-additive subtree S of G such that

$$d_G(u, V(S)) \leq \rho' \text{ for every vertex } u \text{ of } G,$$

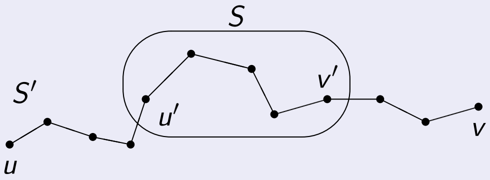
one can construct in time $O(m)$ an additive tree $(\rho + 4\rho')$-spanner of G.

Proofs

Proof.

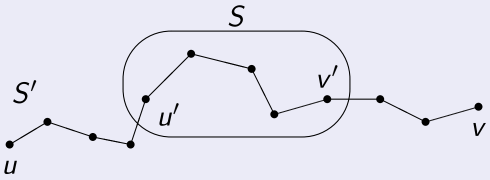
Proofs

Proof.



Proofs

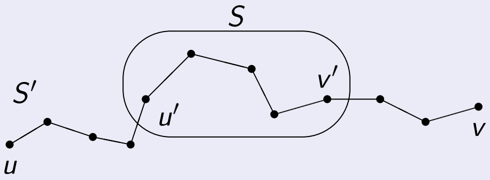
Proof.



$d_{S'}(u, v)$

Proofs

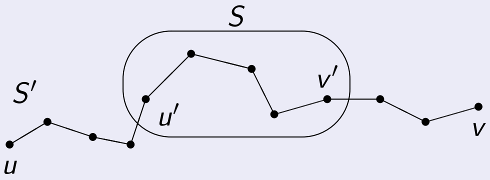
Proof.



$$d_{S'}(u, v) \quad = \quad d_{S'}(u, u') + d_S(u', v') + d_{S'}(v', v)$$

Proofs

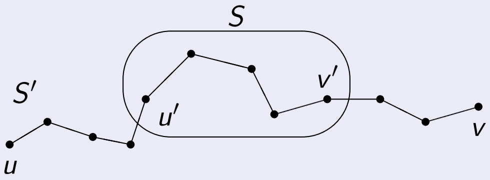
Proof.



$$\begin{aligned} d_{S'}(u,v) &= d_{S'}(u,u') + d_S(u',v') + d_{S'}(v',v) \\ &\leq \rho' \end{aligned}$$

Proofs

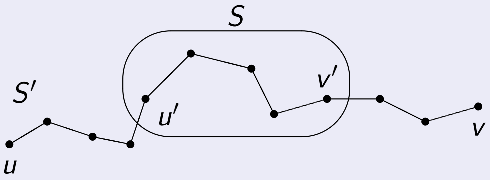
Proof.



$$\begin{aligned} d_{S'}(u,v) &= d_{S'}(u,u') + d_S(u',v') + d_{S'}(v',v) \\ &\leq \rho' + (d_G(u',v') + \rho) \end{aligned}$$

Proofs

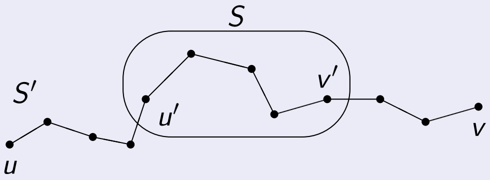
Proof.



$$
\begin{aligned}
d_{S'}(u, v) &= d_{S'}(u, u') + d_S(u', v') + d_{S'}(v', v) \\
&\leq \rho' + (d_G(u', v') + \rho) + \rho'
\end{aligned}
$$

Proofs

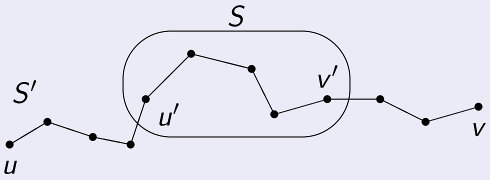
Proof.



$$
\begin{aligned}
d_{S'}(u,v) &= d_{S'}(u,u') + d_S(u',v') + d_{S'}(v',v) \\
&\leq \rho' + (d_G(u',v') + \rho) + \rho' \\
&\leq \rho + 2\rho'
\end{aligned}
$$

Proofs

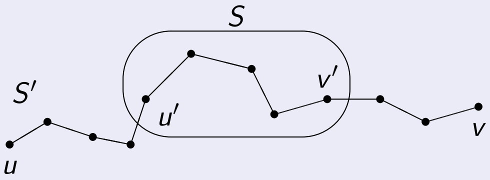
Proof.



$$
\begin{aligned}
d_{S'}(u,v) &= d_{S'}(u,u') + d_S(u',v') + d_{S'}(v',v) \\
&\leq \rho' + (d_G(u',v') + \rho) + \rho' \\
&\leq \rho + 2\rho' + d_G(u',u) + d_G(u,v) + d_G(v,v')
\end{aligned}
$$

Proofs

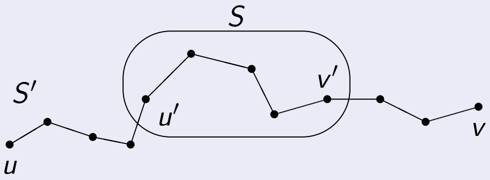
Proof.



$$
\begin{aligned}
d_{S'}(u, v) &= d_{S'}(u, u') + d_S(u', v') + d_{S'}(v', v) \\
&\leq \rho' + (d_G(u', v') + \rho) + \rho' \\
&\leq \rho + 2\rho' + d_G(u', u) + d_G(u, v) + d_G(v, v') \\
&\leq \rho + 2\rho'
\end{aligned}
$$

Proofs

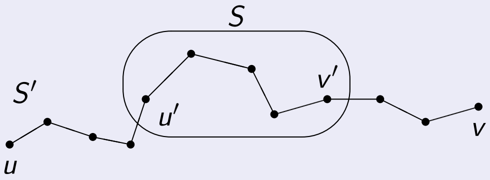
Proof.



$$
\begin{aligned}
d_{S'}(u,v) &= d_{S'}(u,u') + d_S(u',v') + d_{S'}(v',v) \\
&\leq \rho' + (d_G(u',v') + \rho) + \rho' \\
&\leq \rho + 2\rho' + d_G(u',u) + d_G(u,v) + d_G(v,v') \\
&\leq \rho + 2\rho' + \rho'
\end{aligned}
$$

Proofs

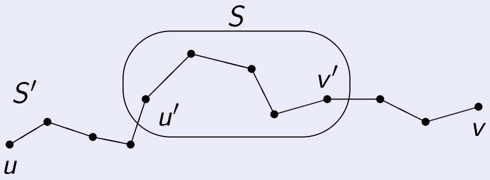
Proof.



$$
\begin{aligned}
d_{S'}(u,v) &= d_{S'}(u,u') + d_S(u',v') + d_{S'}(v',v) \\
&\leq \rho' + (d_G(u',v') + \rho) + \rho' \\
&\leq \rho + 2\rho' + d_G(u',u) + d_G(u,v) + d_G(v,v') \\
&\leq \rho + 2\rho' + \rho' + d_G(u,v)
\end{aligned}
$$

Proofs

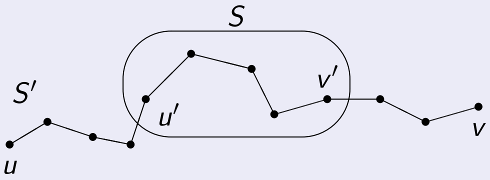
Proof.



$$
\begin{aligned}
d_{S'}(u,v) &= d_{S'}(u,u') + d_S(u',v') + d_{S'}(v',v) \\
&\le \rho' + (d_G(u',v') + \rho) + \rho' \\
&\le \rho + 2\rho' + d_G(u',u) + d_G(u,v) + d_G(v,v') \\
&\le \rho + 2\rho' + \rho' + d_G(u,v) + \rho'
\end{aligned}
$$

Proofs

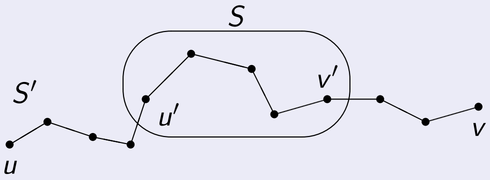
Proof.



$$
\begin{aligned}
d_{S'}(u,v) &= d_{S'}(u,u') + d_S(u',v') + d_{S'}(v',v) \\
&\le \rho' + (d_G(u',v') + \rho) + \rho' \\
&\le \rho + 2\rho' + d_G(u',u) + d_G(u,v) + d_G(v,v') \\
&\le \rho + 2\rho' + \rho' + d_G(u,v) + \rho' \\
&\le \rho + 4\rho' + d_G(u,v),
\end{aligned}
$$

Proofs

Proof.



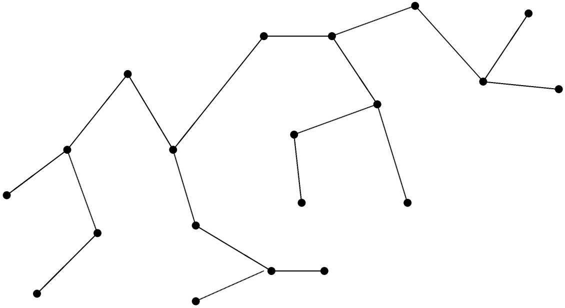
$$
\begin{aligned}
d_{S'}(u,v) &= d_{S'}(u,u') + d_S(u',v') + d_{S'}(v',v) \\
&\le \rho' + (d_G(u',v') + \rho) + \rho' \\
&\le \rho + 2\rho' + d_G(u',u) + d_G(u,v) + d_G(v,v') \\
&\le \rho + 2\rho' + \rho' + d_G(u,v) + \rho' \\
&\le \rho + 4\rho' + d_G(u,v),
\end{aligned}
$$

□

Proofs

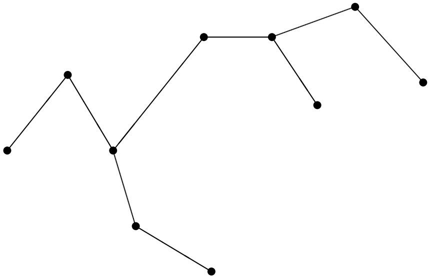
Proofs

Let T be a tree.



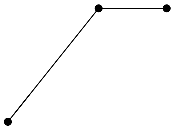
Proofs

Let T be a tree.



Proofs

Let T be a tree.



Proofs

Let T be a tree.

Proofs

Let T be a tree.

T_0

Proofs

Let T be a tree.

$T_0 \supset T_1$

Proofs

Let T be a tree.

$$T_0 \supset T_1 \supset T_2$$

Proofs

Let T be a tree.

- $$T_0 \supset T_1 \supset T_2 \supset \ldots \supset T_{d(T)},$$

Proofs

Let T be a tree.

$$T_0 \supset T_1 \supset T_2 \supset \ldots \supset T_{d(T)},$$

where

- $T_0 = T$.

Proofs

Let T be a tree.

•

$$T_0 \supset T_1 \supset T_2 \supset \ldots \supset T_{d(T)},$$

where

- $T_0 = T$.
- If $T_i \neq P_\ell$, then $T_{i+1} \subseteq T_i$ minimal with all branch vertices of T_i.

Proofs

Let T be a tree.

•

$$T_0 \supset T_1 \supset T_2 \supset \ldots \supset T_{d(T)},$$

where

- $T_0 = T$.
- If $T_i \neq P_\ell$, then $T_{i+1} \subseteq T_i$ minimal with all branch vertices of T_i.
- If $T_i = P_\ell$ for $\ell \geq 3$, then let $T_{i+1} \subseteq T_i$ have order 1.

Proofs

Let T be a tree.

$$T_0 \supset T_1 \supset T_2 \supset \ldots \supset T_{d(T)},$$

where

- $T_0 = T$.
- If $T_i \neq P_\ell$, then $T_{i+1} \subseteq T_i$ minimal with all branch vertices of T_i.
- If $T_i = P_\ell$ for $\ell \geq 3$, then let $T_{i+1} \subseteq T_i$ have order 1.
- If $T_i = P_\ell$ for $\ell \leq 2$, then terminate; $d(T) \leftarrow i$.

Proofs

Let T be a tree.

•

$$T_0 \supset T_1 \supset T_2 \supset \ldots \supset T_{d(T)},$$

where

- $T_0 = T$.
- If $T_i \neq P_\ell$, then $T_{i+1} \subseteq T_i$ minimal with all branch vertices of T_i.
- If $T_i = P_\ell$ for $\ell \geq 3$, then let $T_{i+1} \subseteq T_i$ have order 1.
- If $T_i = P_\ell$ for $\ell \leq 2$, then terminate; $d(T) \leftarrow i$.

Lemma (Bendele and R 2020)

$\mathrm{pbt}(T) = d(T)$ *for every tree* T.

Proofs

Lemma (Bendele and R 2020)

Given G and a tree decomposition $\left(T, (X_t)_{t \in V(T)}\right)$ of breadth ρ, one can construct in time $O(m \cdot \mathrm{d}(T))$ a $16\rho\mathrm{d}(T)$-additive subtree S of G

Proofs

Lemma (Bendele and R 2020)

Given G and a tree decomposition $\left(T, (X_t)_{t \in V(T)}\right)$ of breadth ρ, one can construct in time $O(m \cdot \mathrm{d}(T))$ a $16\rho\mathrm{d}(T)$-additive subtree S of G intersecting each bag of the given tree-decomposition.

Proofs

Lemma (Bendele and R 2020)

Given G and a tree decomposition $\left(T, (X_t)_{t \in V(T)}\right)$ of breadth ρ, one can construct in time $O(m \cdot \mathrm{d}(T))$ a $16\rho\mathrm{d}(T)$-additive subtree S of G intersecting each bag of the given tree-decomposition.

Proof.

$T_0 \supset T_1 \supset T_2 \supset \ldots \supset T_d$.

Proofs

Lemma (Bendele and R 2020)

Given G and a tree decomposition $\left(T, (X_t)_{t \in V(T)}\right)$ of breadth ρ, one can construct in time $O(m \cdot \mathrm{d}(T))$ a $16\rho\mathrm{d}(T)$-additive subtree S of G intersecting each bag of the given tree-decomposition.

Proof.

$T_0 \supset T_1 \supset T_2 \supset \ldots \supset T_d$. For i from d down to 0, construct a subtree S_i of G

Proofs

Lemma (Bendele and R 2020)

Given G and a tree decomposition $\left(T, (X_t)_{t \in V(T)}\right)$ of breadth ρ, one can construct in time $O(m \cdot \mathrm{d}(T))$ a $16\rho\mathrm{d}(T)$-additive subtree S of G intersecting each bag of the given tree-decomposition.

Proof.

$T_0 \supset T_1 \supset T_2 \supset \ldots \supset T_d$. For i from d down to 0, construct a subtree S_i of G such that

- S_i contains a vertex from bag X_t for every vertex t of T_i,

Proofs

Lemma (Bendele and R 2020)

Given G and a tree decomposition $\left(T, (X_t)_{t \in V(T)}\right)$ of breadth ρ, one can construct in time $O(m \cdot \mathrm{d}(T))$ a $16\rho\mathrm{d}(T)$-additive subtree S of G intersecting each bag of the given tree-decomposition.

Proof.

$T_0 \supset T_1 \supset T_2 \supset \ldots \supset T_d$. For i from d down to 0, construct a subtree S_i of G such that

- S_i contains a vertex from bag X_t for every vertex t of T_i,
- S_i is $16\rho(d - i)$-additive.

Proofs

Lemma (Bendele and R 2020)

Given G and a tree decomposition $\left(T, (X_t)_{t \in V(T)}\right)$ of breadth ρ, one can construct in time $O(m \cdot \mathrm{d}(T))$ a $16\rho\mathrm{d}(T)$-additive subtree S of G intersecting each bag of the given tree-decomposition.

Proof.

$T_0 \supset T_1 \supset T_2 \supset \ldots \supset T_d$. For i from d down to 0, construct a subtree S_i of G such that

- S_i contains a vertex from bag X_t for every vertex t of T_i,
- S_i is $16\rho(d - i)$-additive.

$n(S_d) \leq 2$.

Proofs

Lemma (Bendele and R 2020)

Given G and a tree decomposition $\left(T, (X_t)_{t \in V(T)}\right)$ of breadth ρ, one can construct in time $O(m \cdot \mathrm{d}(T))$ a $16\rho\mathrm{d}(T)$-additive subtree S of G intersecting each bag of the given tree-decomposition.

Proof.

$T_0 \supset T_1 \supset T_2 \supset \ldots \supset T_d$. For i from d down to 0, construct a subtree S_i of G such that

- S_i contains a vertex from bag X_t for every vertex t of T_i,
- S_i is $16\rho(d-i)$-additive.

$n(S_d) \leq 2$. $S \leftarrow S_0$.

Proofs

Lemma (Bendele and R 2020)

Given G and a tree decomposition $\left(T, (X_t)_{t \in V(T)}\right)$ of breadth ρ, one can construct in time $O(m \cdot \mathrm{d}(T))$ a $16\rho\mathrm{d}(T)$-additive subtree S of G intersecting each bag of the given tree-decomposition.

Proof.

$T_0 \supset T_1 \supset T_2 \supset \ldots \supset T_d$. For i from d down to 0, construct a subtree S_i of G such that

- S_i contains a vertex from bag X_t for every vertex t of T_i,
- S_i is $16\rho(d - i)$-additive.

$n(S_d) \leq 2$. $S \leftarrow S_0$. □

Thank you for the attention!