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

Oliver Bendele and Dieter Rautenbach

Institute of Optimization and Operations Research, Ulm University, Germany
{o.liver.bendele,dieter.rautenbach}@uni-ulm.de

Abstract

The tree breadth $tb(G)$ of a connected graph $G$ is the smallest non-negative integer $\rho$ such that $G$ has a tree decomposition whose bags all have radius at most $\rho$. We show that, given a connected graph $G$ of order $n$ and size $m$, one can construct in time $O(m \log n)$ an additive tree $O(tb(G) \log n)$-spanner of $G$, that is, a spanning subtree $T$ of $G$ in which $d_T(u,v) \leq d_G(u,v) + O(tb(G) \log n)$ for every two vertices $u$ and $v$ of $G$. This improves earlier results of Dragan and Köhler (Algorithmica 69 (2014) 884-905), who obtained a multiplicative error of the same order, and of Dragan and Abu-Ata (Theoretical Computer Science 547 (2014) 1-17), who achieved the same additive error with a collection of $O(\log n)$ trees.

Keywords: additive tree spanner; multiplicative tree spanner; tree breadth; tree length

AMS subject classification: 05C05, 05C12, 05C85

1 Introduction

In the present paper we show how to construct in time $O(m \log n)$, for a given connected graph $G$ of order $n$ and size $m$, a tree spanner that approximates all distances up to some additive error of the form $O(\rho \log n)$, where $\rho$ is the so-called tree breadth of $G$ [8]. Our result improves a result of Dragan and Köhler [8] who show that one can construct in time $O(m \log n)$ a multiplicative tree $O(\rho \log n)$-spanner for a given graph $G$ as above, that is, we improve their multiplicative error to an additive one of the same order. Our result also improves a result by Dragan and Abu-Ata [6] who show how to efficiently construct $O(\log n)$ collective additive tree $O(\rho \log n)$-spanners for a given graph $G$ as above. Note that they obtain the same additive error bound but require several spanning trees that respect this bound only collectively, more precisely, for every pair of vertices, there is a tree in the collection that satisfies the distance condition for this specific pair. Not restricting the spanners to trees allows better guarantees; Dourisboure, Dragan, Gavoille, and Yan [3], for instance, showed that every graph $G$ as above has an additive $O(\rho n)$-spanner with $O(\rho n)$ edges. For more background on additive and multiplicative (collective) (tree) spanners please refer to [2,5,9,11] and the references therein.

Before we come to our results in Section 2, we collect some terminology and definitions. We consider finite, simple, and undirected graphs. Let $G$ be a connected graph. The vertex set,
edge set, order, and size of $G$ are denoted by $V(G)$, $E(G)$, $n(G)$, and $m(G)$, respectively. The
distance in $G$ between two vertices $u$ and $v$ of $G$ is denoted by $d_G(u, v)$. For a vertex $u$ of $G$
and a set $U$ of vertices of $G$, the distance in $G$ between $u$ and $U$ is
\[
d_G(u, U) = \min \{d_G(u, v) : v \in U\},
\]
and the radius $\text{rad}_G(U)$ of $U$ in $G$ is
\[
\min \{ \max \{d_G(u, v) : v \in U\} : u \in V(G)\},
\]
that is, it is the smallest radius of a ball around some vertex $u$ of $G$ that contains all of $U$.
Note that the vertex $u$ in the preceding minimum is not required to belong to $U$, and that all
distances are considered within $G$.

Let $H$ be a subgraph of $G$. For a non-negative integer $k$, the subgraph $H$ is $k$-additive if
\[
d_H(u, v) \leq d_G(u, v) + k
\]
for every two vertices $u$ and $v$ of $H$. If, additionally, the subgraph $H$ is spanning, that is, it
has the same vertex set as $G$, then $H$ is an additive $k$-spanner of $G$. Furthermore, if, again
additionally, the subgraph $H$ is a tree, then $H$ is an additive tree $k$-spanner of $G$. Replacing
the inequality (1) with
\[
d_H(u, v) \leq k \cdot d_G(u, v)
\]
yields the notions of a $k$-multiplicative subgraph, a multiplicative $k$-spanner, and a multiplicative
tree $k$-spanner of $G$, respectively.

For a tree $T$, let $B(T)$ be the set of vertices of $T$ of degree at least 3 in $T$, the so-called
branch vertices, and let $L(T)$ be the set of leaves of $T$.

A tree decomposition of $G$ is a pair $(T, (X_t)_{t \in V(T)})$, where $T$ is a tree and $X_t$ is a set of
vertices of $G$ for every vertex $t$ of $T$ such that
- for every vertex $u$ of $G$, the set $\{t \in V(T) : u \in X_t\}$ induces a non-empty subtree of $T$, and
- for every edge $uv$ of $G$, there is some vertex $t$ of $T$ such that $u$ and $v$ both belong to $X_t$.

The set $X_t$ is usually called the bag of $t$. The maximum radius
\[
\max \{\text{rad}_G(X_t) : t \in V(T)\}
\]
of a bag of the tree decomposition is the breadth of this decomposition, and the tree breadth
$\text{tb}(G)$ of $G$ is the minimum breadth of a tree decomposition of $G$. While the tree breadth
is an NP-hard parameter, one can construct in linear time, for a given connected graph $G$, a
tree decomposition of breadth at most $3\text{tb}(G)$, cf. also involving the related notion
of tree length.
2 Results

For a tree \( T \), let \( \rho(T) \) be the maximum depth of a perfect binary tree that is a topological minor of \( T \). In some sense \( \rho(T) \) quantifies how much \( T \) differs from a path.

Our main result is the following.

**Theorem 1.** Given a connected graph \( G \) of size \( m \) and a tree decomposition \((T, (X_t)_{t\in V(T)})\) of \( G \) of breadth \( \rho \), one can construct in time \( O(m \cdot \rho(T)) \) an additive tree \( 8\rho(2\rho(T)+1) \)-spanner of \( G \).

Some immediate consequences of Theorem 1 are the following.

**Corollary 2.** Given a connected graph \( G \) of order \( n \) and size \( m \), one can construct in time \( O(m \log n) \) an additive tree \( O(\text{tb}(G) \log n) \)-spanner of \( G \).

**Proof.** As observed towards the end of the introduction, given \( G \), one can construct in linear time a tree decomposition \((T, (X_t)_{t\in V(T)})\) of \( G \) of breadth at most \( 3\text{tb}(G) \). Possibly by contracting edges \( st \) of \( T \) with \( X_s \subseteq X_t \), we may assume that \( n(T) \leq n \). Since a perfect binary tree of depth \( b \) has \( 2^{b+1} - 1 \) vertices, it follows that \( 2^{\text{tb}(T)+1} - 1 \leq n(T) \leq n \), and, hence,

\[
\rho(T) \leq \log_2(n + 1) - 1.
\]

Applying Theorem 1 allows to construct in time \( O(m \cdot \rho(T)) = O(m \log n) \) an additive tree \( 24\text{tb}(G)(2 \log_2(n + 1) - 1) \)-spanner of \( G \).

**Corollary 3.** Given a connected graph \( G \) of order \( n \) and size \( m \) and a multiplicative tree \( k \)-spanner \( T \) of \( G \), one can construct in time \( O(mn) \) an additive tree \( O(k \log n) \)-spanner of \( G \).

**Proof.** For every vertex \( u \) of \( G \), let \( X_u \) be the set containing all vertices \( v \) of \( G \) with \( d_T(u, v) \leq \left\lceil \frac{\rho}{2} \right\rceil \). Since \( T \) is a multiplicative tree \( k \)-spanner, it follows easily that \((T, (X_t)_{t\in V(T)})\) is a tree decomposition of \( G \) of breadth at most \( \left\lceil \frac{\rho}{2} \right\rceil \), cf. also \( \text{[5]} \). Note that \((X_t)_{t\in V(T)}\) can be determined by \( n \) breadth first searches, each of which requires \( O(m) \) time. Applying Theorem 1 allows to construct in time \( O(m \cdot \rho(T)) = O(m \log n) \) an additive tree \( O(k \log n) \)-spanner of \( G \).

Note that if the tree \( T \) in Theorem 1 is a path, then we obtain an additive tree \( O(\rho) \)-spanner. Kratsch et al. \( \text{[11]} \) constructed a sequence of outerplanar chordal graphs \( G_1, G_2, \ldots \), which limit the extend to which Theorem 1 can be improved. The graph \( G_1 \) is a triangle, and, for every positive integer \( k \), the graph \( G_{k+1} \) arises from \( G_k \) by adding, for every edge \( uv \) of \( G_k \) that contains a vertex of degree 2 in \( G_k \), a new vertex \( w \) that is adjacent to \( u \) and \( v \); cf. Figure 1 for an illustration. It is easy to see \( n(G_k) = 3 \cdot 2^{k-1} \) and that \( \text{tb}(G_k) = 1 \) for every positive integer \( k \); in particular, we have \( k - 1 = \log_2 \left( \frac{n(G_k)}{3} \right) \). Now, Kratsch et al. showed that \( G_k \) admits no additive tree \((k - 1)\)-spanner, that is, the graph \( G_k \) admits no additive tree \( \text{tb}(G_k) \log_2 \left( \frac{n(G_k)}{3} \right) \)-spanner.
Our proof of Theorem 1 relies on four lemmas. The first is a simple consequence of elementary properties of breadth first search.

**Lemma 4.** Given a connected graph $G$ of size $m$, a subtree $S$ of $G$, and a set $U$ of vertices of $G$, one can construct in time $O(m)$ a subtree $S'$ of $G$ containing $S$ as well as all vertices from $U$ such that

(i) $d_{S'}(u, V(S)) = d_G(u, V(S))$ for every vertex $u$ in $U$, and

(ii) $L(S') \subseteq L(S) \cup U$.

**Proof.** The tree $S'$ with the desired properties can be obtained as follows:

- Construct the graph $G'$ from $G$ by contracting $S$ to a single vertex $r$.
- Construct a breadth first search tree $T$ of $G'$ rooted in $r$.
- Construct the graph $T'$ from $T$ by uncontracting $r$ back to $S$.
- Choose $S'$ as the minimal subtree of $T'$ that contains $S$ as well as all vertices from $U$.

Since $T$ is a breadth first search tree, property (i) follows. Furthermore, by construction, the set of leaves of $S'$ is contained in $L(S) \cup U$, that is, property (ii) follows. The running time follows easily from the running time of breadth first search; in fact, the contraction of $S$ to $r$ can be handled implicitly within a suitably adapted breadth first search.

The following lemma was inspired by Lemma 2.2 in [11]. It will be useful to complete the construction of our additive tree spanner starting from a suitable subtree.

**Lemma 5.** Given a connected graph $G$ of size $m$ and a $\rho$-additive subtree $S$ of $G$ such that $d_G(u, V(S)) \leq \rho'$ for every vertex $u$ of $G$, one can construct in time $O(m)$ an additive tree $(\rho + 4\rho')$-spanner of $G$.

**Proof.** Let $S'$ be the spanning tree of $G$ obtained by applying Lemma 4 to $G$, $S$, and $V(G) \setminus V(S)$ as the set $U$. We claim that $S'$ has the desired properties. Therefore, let $u$ and $v$ be any two vertices of $G$. Let $u'$ be the vertex of $S$ closest to $u$ within $S'$, and define $v'$ analogously. Clearly,
we have that $d_S(u, u') = d_G(u, u') \leq \rho'$, $d_S(v, v') = d_G(v, v') \leq \rho'$, and $d_S(u', v') = d_S(u', v') \leq d_G(u', v') + \rho$. By several applications of the triangle inequality, we obtain

$$d_S(u, v) = d_S(u, u') + d_S(u', v') + d_S(v', v) \leq \rho' + d_G(u', v') + \rho + \rho' \leq d_G(u', u) + d_G(u, v) + d_G(v, v') + \rho + 2\rho' \leq d_G(u, v) + \rho + 4\rho',$$

which completes the proof.

Our next lemma states that $pbt(T)$ can easily be determined for a given tree $T$, by constructing a suitable finite sequence

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

of nested trees. The construction of this sequence is also important for the proof of our main technical lemma, cf. Lemma 7 below. The sequence starts with $T_0$ equal to $T$. Now, suppose that $T_i$ has been defined for some non-negative integer $i$. If $B(T_i)$ is not empty, then let $T_{i+1}$ be the minimal subtree of $T_i$ that contains all vertices from $B(T_i)$, and continue the construction. Note that in this case

$$B(T_i) = B(T_{i+1}) \cup L(T_{i+1}).$$

Otherwise, if $B(T_i)$ is empty, then $T_i$ is a path of some length $\ell$. If $\ell \geq 3$, then let $T_{i+1}$ be the tree containing exactly one internal vertex of $T_i$ as its only vertex, and let $d(T) = i + 1$. Finally, if $\ell \leq 2$, then let $d(T) = i$. Once $d(T)$ has been defined, the construction of the sequence (2) terminates. See Figure 2 for an illustration.

![Figure 2: A sequence $T_0 \subset T_1 \subset T_2 \subset T_3$.](image)

**Lemma 6.** $pbt(T) = d(T)$ for every tree $T$.

**Proof.** The proof is by induction on $d(T)$. If $d(T) = 0$, the statement is trivial. Now, let $d(T) \geq 1$. The construction of (2) immediately implies

$$d(T) = d(T_1) + 1.$$
subdivision of a perfect binary tree, then one can first extend $S_1$ in such a way that all leaves of $S_1$ are also leaves of $T_1$, and then one can grow one further level to the subdivided binary tree by attaching two new paths to each leaf of $S_1$ using edges in $E(T) \setminus E(T_1)$. This implies \( \text{pbt}(T) \geq \text{pbt}(T_1) + 1 \). Conversely, if $S$ is a subtree of $T$ that is a subdivision of a perfect binary tree, then $S \cap T_1$ contains a subdivision of a perfect binary tree whose depth is one less, that is, we have $\text{pbt}(T_1) \geq \text{pbt}(T) - 1$. Altogether, by induction, we obtain

\[
\text{pbt}(T) = \text{pbt}(T_1) + 1 = d(T_1) + 1 = d(T),
\]

which completes the proof.

The following is our core technical lemma.

Lemma 7. Given a connected graph $G$ of size $m$ and a tree decomposition $(T, (X_t)_{t \in V(T)})$ of $G$ of breadth $\rho$, one can construct in time $O(m \cdot d(T))$ a $16\rho \cdot d(T)$-additive subtree $S$ of $G$ intersecting each bag of the given tree-decomposition.

Proof. Let the sequence $T_0 \supset T_1 \supset T_2 \supset \ldots \supset T_d$ be as in (2), and let $d = d(T)$. For $i$ from $d$ down to 0, we explain how to recursively construct a subtree $S_i$ of $G$ such that

(i) $S_i$ contains a vertex from bag $X_t$ for every vertex $t$ of $T_i$,

(ii) for every two distinct leaves $u$ and $v$ of $S_i$, there are two distinct vertices $s$ and $t$ of $T_i$ that belong to $B(T_i) \cup L(T_i)$ such that $u \in X_s$ and $v \in X_t$, and

(iii) $S_i$ is $16\rho(d - i)$-additive.

Note that $S_0$ is a subtree of $G$ with the desired properties.

First, we consider $i = d$. The tree $T_d$ has order at most 2, and, since $G$ is connected, there is a vertex $u$ of $G$ that belongs to all bags $X_t$ with $t \in V(T_d)$. Let $S_d$ be the subtree of $G$ containing only the vertex $u$. Since $S_d$ has order 1, and all vertices of $T_d$ are leaves, properties (ii) and (iii) are trivial for $S_d$, and property (i) follows from the choice of $u$. See Figure 3 for an illustration.

![Figure 3: Extending $S_i$ to $S_{i-1}$, and possible positions of the vertices $u$, $v$, $u^{(1)}$, and $v^{(1)}$ explained below.](image)

Now, suppose that $S_i$ has already been defined for some integer $i$ with $d \geq i > 0$. We explain how to construct $S_{i-1}$. Therefore, let $U$ be an inclusion-wise minimal set of vertices
intersecting every bag $X_i$ such that $t$ is a leaf of $T_{i-1}$ for which $S_i$ does not contain a vertex from $X_t$. Let $S_{i-1}$ arise by applying Lemma 3 to $G$, $S_i$ as $S$, and $U$. By construction, the subgraph $S_{i-1}$ of $G$ is connected and contains a vertex from every bag $X_i$ such that $t$ is a leaf of $T_{i-1}$. Since $G$ is connected, basic properties of tree decompositions imply that $S_{i-1}$ satisfies property (i), that is, the vertex set of $S_{i-1}$ intersects every bag of $T_{i-1}$.

Next, we verify property (ii) for $S_{i-1}$. Therefore, let $u$ and $v$ be two distinct leaves of $S_{i-1}$. If $u$ and $v$ are also leaves of $S_i$, then property (ii) for $S_{i-1}$ follows from property (ii) for $S_i$ using $B(T_i) \cup L(T_i) = B(T_{i-1})$. If $u$ is a leaf of $S_i$ and $v$ is not, then, by Lemma 3(ii), we have $v \in U$. By property (ii) for $S_i$, the vertex $u$ belongs to a bag $X_s$ such that $s \in B(T_i) \cup L(T_i) = B(T_{i-1})$, and, by the choice of $U$, the vertex $v$ belongs to a bag $X_t$ such that $t$ is a leaf of $T_{i-1}$ and $S_i$ contains no vertex from $X_t$. In particular, we have that $u \notin X_t$, which implies that $s$ and $t$ are distinct, that is, property (ii) holds also in this case. Finally, suppose that $u$ and $v$ are both leaves of $S_{i-1}$ but not of $S_i$. The choice of $U$ as minimal with respect to inclusion implies that property (ii) holds also in this final case. Note that $X_s$ is allowed to contain $v$ and that $X_t$ is allowed to contain $u$ in property (ii).

Finally, we verify the crucial property (iii) for $S_{i-1}$. Therefore, let $u$ and $v$ be two distinct vertices of $S_{i-1}$. It is easy to see that in order to verify that $S_{i-1}$ is $16\rho(d - (i - 1))$-additive, it suffices to consider the case where $u$ and $v$ are leaves of $S_{i-1}$. In fact, if (iii) is violated for $u$ and $v$, that is, we have $d_{S_{i-1}}(u, v) > d_G(u, v) + 16\rho(d - (i - 1))$, then the path in $S_{i-1}$ between $u$ and $v$ is contained in some path in $S_{i-1}$ between the two leaves $\hat{u}$ and $\hat{v}$ of $S_{i-1}$, and

$$d_{S_{i-1}}(\hat{u}, \hat{v}) = d_{S_{i-1}}(\hat{u}, u) + d_{S_{i-1}}(u, v) + d_{S_{i-1}}(v, \hat{v}) > d_G(\hat{u}, u) + d_G(u, v) + 16\rho(d - (i - 1)) + d_G(v, \hat{v}) \geq d_G(\hat{u}, \hat{v}) + 16\rho(d - (i - 1)),$$

that is, the two leaves $\hat{u}$ and $\hat{v}$ also violate (iii). Hence, we may assume that $u$ and $v$ are leaves of $S_{i-1}$. Let $P$ be a shortest path in $G$ between $u$ and $v$, and let $P_{i-1}$ be the path in $S_{i-1}$ between $u$ and $v$. Let $u^{(1)}$ be the vertex of $S_i$ that is closest within $S_{i-1}$ to $u$, and define $v^{(1)}$ analogously. See Figure 3 for an illustration. By Lemma 3(i), we have

$$d_G\left(u, u^{(1)}\right) = d_{S_{i-1}}\left(u, u^{(1)}\right) \text{ and } d_G\left(v, v^{(1)}\right) = d_{S_{i-1}}\left(v, v^{(1)}\right).$$

By (ii) for $S_{i-1}$, there are two distinct vertices $s$ and $t$ of $T_{i-1}$ that belong to $B(T_{i-1}) \cup L(T_{i-1})$ such that $u \in X_s$ and $v \in X_t$. Let $T'$ be the subgraph of $T_{i-1}$ that is induced by the set of all vertices $r$ of $T_{i-1}$ for which $S_i$ contains a vertex from the bag $X_r$. Since $S_i$ is connected, it follows from basic properties of tree decompositions that $T'$ is a subtree of $T_{i-1}$. Since $B(T_{i-1}) = B(T_i) \cup L(T_i) \subseteq V(T_i)$ and, by construction of $T_i$ from $T_{i-1}$, the path in $T_{i-1}$ between any two distinct leaves of $T_{i-1}$ contains a vertex of $T_i$, property (i) for $S_i$ implies that $T'$ contains a vertex from the path $Q$ in $T_{i-1}$ between $s$ and $t$. Let $s'$ be the vertex of $T'$ on $Q$ that is closest within $T_{i-1}$ to $s$. By the definition of $T'$, there is a vertex $u^{(2)}$ of $S_i$ that belongs
to $X_{s'}$. See Figure 4 for an illustration.

Figure 4: The path $Q$ in $T_{i-1}$ between $s$ and $t$, the subtree $T'$ of $T_{i-1}$ intersecting $Q$, and the vertices $s'$ and $t'$.

Basic properties of tree decompositions imply that $X_{s'}$ contains a vertex from the path $P$ as well as from the path $P_{i-1}$. Let $u^{(3)}$ be a vertex in $X_{s'} \cap V(P)$, and let $u^{(4)}$ be the first vertex on the path $P_{i-1}$, when traversed from $u$ towards $v$, that belongs to $X_{s'}$. See Figure 5 for an illustration.

Figure 5: The shortest paths $P$ in $G$ and $P_{i-1}$ in $S_{i-1}$ between $u$ and $v$, their intersection with the bags $X_{s'}$ and $X_r$, the vertices $u^{(4)}$ and $v^{(4)}$, and possible positions of $u^{(3)}$ and $v^{(3)}$.

Suppose, for a contradiction, that $u^{(1)}$ is distinct from $u^{(4)}$, and that $u^{(1)}$ lies closer to $u$ on $P_{i-1}$ than $u^{(4)}$. In this case, the choices of $u^{(1)}$ and $u^{(4)}$ imply that $u^{(1)}$ lies in some bag $X_r$ for a vertex $r$ of $T'$ distinct from $s'$, and that $u^{(1)}$ does not lie in $X_{s'}$. Since $s'$ separates $s$ from $r$ in $T_{i-1}$, basic properties of tree decompositions imply that $P_{i-1}$ contains a vertex from $X_{s'}$ that is strictly closer to $u$ than $u^{(4)}$, contradicting the choice of $u^{(4)}$. Hence, either $u^{(1)}$ equals $u^{(4)}$, or $u^{(4)}$ lies closer to $u$ on $P_{i-1}$ than $u^{(1)}$.

Since $u^{(2)}$, $u^{(3)}$, and $u^{(4)}$ all belong to the bag $X_{s'}$, which is of radius at most $\rho$, the pairwise distances of these three vertices within $G$ are at most $2\rho$. If $d_G(u^{(1)}, u^{(4)}) > 2\rho$, then connecting $u$ to $u^{(4)}$ via $P_{i-1}$, and connecting $u^{(4)}$ to $S_i$ via a shortest path in $G$, which is of length at most $2\rho$ in view of $u^{(2)}$, yields a contradiction to Lemma 4(i). Hence, we have

\[ d_G(u^{(1)}, u^{(4)}) \leq 2\rho, \]

and, thus, we obtain

\[ d_G(u^{(1)}, u^{(3)}) \leq d_G(u^{(1)}, u^{(4)}) + d_G(u^{(4)}, u^{(3)}) \leq 4\rho. \]
Now, let \( t' \) be the vertex of \( T' \) on \( Q \) that is closest within \( T_{i-1} \) to \( t \). See Figure 4 for an illustration. Clearly, the vertex \( t' \) lies on the subpath of \( Q \) between \( s' \) and \( t \). Since \( u^{(3)} \in X_{s'} \) and \( v \in X_t \), basic properties of tree decompositions imply that the subpath of \( P \) between \( u^{(3)} \) and \( v \) contains a vertex \( v^{(3)} \) of \( X_v \). See Figure 5 for an illustration. Choosing \( v^{(2)} \) and \( v^{(4)} \) in a symmetric way, and arguing similarly as above, we obtain

\[
d_G(v^{(1)}, v^{(3)}) \leq 4\rho.
\]

By property (iii) for \( S_i \), we have

\[
d_{S_i}(u^{(1)}, v^{(1)}) \leq d_G(u^{(1)}, v^{(1)}) + 16\rho(d - i).
\]

Note that the vertices \( u, u^{(3)}, v^{(3)}, \) and \( v \) appear in this order on \( P \). Altogether, by multiple applications of the triangle inequality, we obtain that

\[
d_{S_{i-1}}(u, v) = d_{S_{i-1}}(u, u^{(1)}) + d_{S_i}(u^{(1)}, v^{(1)}) + d_{S_{i-1}}(v^{(1)}, v)
\]
\[
= d_G(u, u^{(1)}) + d_{S_i}(u^{(1)}, v^{(1)}) + d_G(v^{(1)}, v)
\]
\[
\leq d_G(u, u^{(1)}) + d_G(u^{(1)}, v^{(1)}) + 16\rho(d - i) + d_G(v^{(1)}, v)
\]
\[
\leq d_G(u, u^{(3)}) + d_G(u^{(3)}, u^{(1)})
\]
\[
+ d_G(u^{(1)}, v^{(1)}) + d_G(u^{(3)}, v^{(3)}) + d_G(v^{(3)}, v^{(1)}) + 16\rho(d - i)
\]
\[
+ d_G(v^{(1)}, v^{(3)}) + d_G(v^{(3)}, v^{(1)}) + 16\rho(d - i)
\]
\[
\leq d_G(u, u^{(3)}) + 4\rho + 4\rho + d_G(u^{(3)}, v^{(3)}) + 4\rho + 16\rho(d - i) + 4\rho + d_G(v^{(3)}, v^{(1)})
\]
\[
= d_G(u, v) + 16\rho(d - (i - 1)),
\]

which completes the proof of property (iii) for \( S_{i-1} \).

We proceed to the running time of the described procedure. Clearly, the sequence as in (2) can be determined in time \( O(m \cdot d(T)) \), and the tree \( S_d \) can be obtained in time \( O(m(G)) \). By Lemma 4, given any tree \( S_i \) with \( i > 0 \), the tree \( S_{i-1} \) can be obtained in time \( O(m(G)) \). Altogether, the stated running time follows, which completes the proof.

Theorem 1 now follows immediately by combining Lemma 7 with Lemma 5, choosing \( \rho' \) equal to \( 2\rho \) for the latter. Note that, since the tree \( S \) produced by Lemma 7 intersects every bag of the tree decomposition, we have \( d_G(u, V(S)) \leq 2\rho \) for every vertex \( u \) of \( G \).

References

[1] M. Abu-Ata and F.F. Dragan, Metric tree-like structures in real-world networks: an empirical study, Networks 67 (2016) 49-68.

[2] L. Cai and D.G. Corneil, Tree Spanners, SIAM Journal on Discrete Mathematics 8 (1995) 359-387.
[3] V.D. Chepoi and F.F. Dragan, A note on distance approximating trees in graphs, European Journal of Combinatorics 21 (2000) 761-768.

[4] Y. Dourisboure and C. Gavoille, Tree-decompositions with bags of small diameter, Discrete Mathematics 307 (2007) 2008-2029.

[5] Y. Dourisboure, F.F. Dragan, C. Gavoille, and C. Yan, Spanners for bounded tree-length graphs, Theoretical Computer Science 383 (2007) 34-44.

[6] F.F. Dragan and M. Abu-Ata, Collective additive tree spanners of bounded tree-breadth graphs with generalizations and consequences, Theoretical Computer Science 547 (2014) 1-17.

[7] F.F. Dragan, D.G. Corneil, E. Köhler, and Y. Xiang, Collective additive tree spanners for circle graphs and polygonal graphs, Discrete Applied Mathematics 160 (2012) 1717-1729.

[8] F.F. Dragan and E. Köhler, An approximation algorithm for the tree t-spanner problem on unweighted graphs via generalized chordal graphs, Algorithmica 69 (2014) 884-905.

[9] F.F. Dragan, C. Yan, and D.G. Corneil, Collective tree spanners and routing in AT-free related graphs, Journal of Graph Algorithms and Applications 10 (2006) 97-122.

[10] G. Ducoffe, S. Legay, and N. Nisse, On the complexity of computing treebreadth, Algorithmica (2019) https://doi.org/10.1007/s00453-019-00657-7.

[11] D. Kratsch, H.-O. Le, H. Müller, E. Prisner, and D. Wagner, Additive tree spanners, SIAM Journal on Discrete Mathematics 17 (2002) 332-340.