Collective Additive Tree Spanners of Bounded Tree-Breadth Graphs with Generalizations and Consequences

Feodor F. Dragan and Muad Abu-Ata

Department of Computer Science, Kent State University, Kent, OH 44242, USA
{dragan,mabuata}@cs.kent.edu

Abstract. In this paper, we study collective additive tree spanners for families of graphs enjoying special Robertson-Seymour's tree-decompositions, and demonstrate interesting consequences of obtained results. It is known that if a graph G has a multiplicative tree t-spanner, then G admits a Robertson-Seymour's tree-decomposition with bags of radius at most $\lceil t/2 \rceil$ in G. We use this to demonstrate that there is a polynomial time algorithm that, given an n-vertex graph G admitting a multiplicative tree t-spanner, constructs a system of at most $\log_2 n$ collective additive tree $O(t \log n)$-spanners of G. That is, with a slight increase in the number of trees and in the stretch, one can "turn" a multiplicative tree spanner into a small set of collective additive tree spanners. We extend this result by showing that, for every fixed k, there is a polynomial time algorithm that, given an n-vertex graph G admitting a multiplicative t-spanner with tree-width $k-1$, constructs a system of at most $k(1+\log_2 n)$ collective additive tree $O(t \log n)$-spanners of G.

1 Introduction

One of the basic questions in the design of routing schemes for communication networks is to construct a spanning network (a so-called spanner) which has two (often conflicting) properties: it should have simple structure and nicely approximate distances in the network. This problem fits in a larger framework of combinatorial and algorithmic problems that are concerned with distances in a finite metric space induced by a graph. An arbitrary metric space (in particular a finite metric defined by a graph) might not have enough structure to exploit algorithmically. A powerful technique that has been successfully used recently in this context is to embed the given metric space in a simpler metric space such that the distances are approximately preserved in the embedding. New and improved algorithms have resulted from this idea for several important problems (see, e.g., [1,2,4,14,18]). There are several ways to measure the quality of this approximation, two of them leading to the notion of a spanner. For $t \geq 1$, a spanning subgraph H of $G = (V, E)$ is called a *(multiplicative) t-spanner* of G if $d_H(u, v) \leq t \cdot d_G(u, v)$ for all $u, v \in V$ [5,23,24]. If $r \geq 0$ and $d_H(u, v) \leq d_G(u, v) + r$, for all $u, v \in V$, then H is called an *additive r-spanner* of G [19,26].

P. van Emde Boas et al. (Eds.): SOFSEM 2013, LNCS 7741, pp. 194–206, 2013.

The parameter t is called the *stretch* (or *stretch factor*) of H, while the parameter r is called the *surplus* of H. In what follows, we will often omit the word "multiplicative" when we refer to multiplicative spanners.

Tree metrics are a very natural class of simple metric spaces since many algorithmic problems become tractable on them. A *(multiplicative) tree t-spanner* of a graph G is a spanning tree with a stretch t [3], and an *additive tree r-spanner* of G is a spanning tree with a surplus r [26]. If we approximate the graph by a tree spanner, we can solve the problem on the tree and the solution interpret on the original graph. The TREE t-SPANNER problem asks, given a graph G and a positive number t, whether G admits a tree t-spanner. Note that the problem of finding a tree t-spanner of G minimizing t is known in literature also as the Minimum Max-Stretch spanning Tree problem (see, e.g., [16] and literature cited therein). Unfortunately, not many graph families admit good tree spanners. This motivates the study of sparse spanners, i.e., spanners with a small amount of edges. There are many applications of spanners in various areas; especially, in distributed systems and communication networks. In [24], close relationships were established between the quality of spanners (in terms of stretch factor and the number of spanner edges), and the time and communication complexities of any synchronizer for the network based on this spanner. Another example is the usage of tree t-spanners in the analysis of arrow distributed queuing protocols [22]. Sparse spanners are very useful in message routing in communication networks; in order to maintain succinct routing tables, efficient routing schemes can use only the edges of a sparse spanner [25]. The SPARSEST t-SPANNER problem asks, for a given graph G and a number t, to find a t-spanner of G with the smallest number of edges. We refer to the survey paper of Peleg [21] for an overview on spanners.

Inspired by ideas from works of Bartal [1], Fakcharoenphol et al. [17], and to extend those ideas to designing compact and efficient routing and distance labeling schemes in networks, in [13], a new notion of *collective tree spanners* was introduced. This notion is slightly *weaker* than the one of a tree spanner and slightly *stronger* than the notion of a sparse spanner. We say that a graph $G = (V, E)$ *admits a system of μ collective additive tree r-spanners* if there is a system $\mathcal{T}(G)$ of at most μ spanning trees of G such that for any two vertices x, y of G a spanning tree $T \in \mathcal{T}(G)$ exists such that $d_T(x, y) \leq d_G(x, y) + r$ (a multiplicative variant of this notion can be defined analogously). Clearly, if G admits a system of μ collective additive tree r-spanners, then G admits an additive r-spanner with at most $\mu \times (n - 1)$ edges (take the union of all those trees), and if $\mu = 1$ then G admits an additive tree r-spanner. Recently, in [10], *spanners of bounded tree-width* were introduced, motivated by the fact that many algorithmic problems are tractable on graphs of bounded tree-width, and a spanner H of G with small tree-width can be used to obtain an approximate solution to a problem on G. In particular, efficient and compact distance and routing labeling schemes are available for bounded tree-width graphs (see, e.g., [12,18] and papers cited therein), and they can be used to compute approximate distances and route along paths that are close to shortest in G.

The k-Tree-width t-spanner problem asks, for a given graph G, an integers k and a positive number $t \geq 1$, whether G admits a t-spanner of tree-width at most k. Every connected graph with n vertices and at most $n-1+m$ edges is of tree-width at most $m+1$ and hence this problem is a generalization of the Tree t-Spanner and the Sparsest t-Spanner problems. Furthermore, spanners of bounded tree-width have much more structure to exploit algorithmically than sparse spanners.

Our Results and Their Place in the Context of the Previous Results. This paper was inspired by few recent results from [7,11,15,16]. Elkin and Peleg in [15], among other results, described a polynomial time algorithm that, given an n-vertex graph G admitting a tree t-spanner, constructs a t-spanner of G with $O(n \log n)$ edges. Emek and Peleg in [16] presented the first $O(\log n)$-approximation algorithm for the minimum value of t for the tree t-spanner problem. They described a polynomial time algorithm that, given an n-vertex graph G admitting a tree t-spanner, constructs a tree $O(t \log n)$-spanner of G. Later, a simpler and faster $O(\log n)$-approximation algorithm for the problem was given by Dragan and Köhler [11]. Their result uses a new necessary condition for a graph to have a tree t-spanner: if a graph G has a tree t-spanner, then G admits a Robertson-Seymour's tree-decomposition with bags of radius at most $\lceil t/2 \rceil$ in G.

To describe the results of [7] and to elaborate more on the Dragan-Köhler's approach, we need to recall definitions of a few graph parameters. They all are based on the notion of tree-decomposition introduced by Robertson and Seymour in their work on graph minors [27].

A *tree-decomposition* of a graph $G = (V, E)$ is a pair $(\{X_i | i \in I\}, T = (I, F))$ where $\{X_i | i \in I\}$ is a collection of subsets of V, called *bags*, and T is a tree. The nodes of T are the bags $\{X_i | i \in I\}$ satisfying the following three conditions: 1) $\bigcup_{i \in I} X_i = V$; 2) for each edge $uv \in E$, there is a bag X_i such that $u, v \in X_i$; 3) for all $i, j, k \in I$, if j is on the path from i to k in T, then $X_i \bigcap X_k \subseteq X_j$. Equivalently, this condition could be stated as follows: for all vertices $v \in V$, the set of bags $\{i \in I | v \in X_i\}$ induces a connected subtree T_v of T.

For simplicity we denote a tree-decomposition $(\{X_i | i \in I\}, T = (I, F))$ of a graph G by $T(G)$.

Tree-decompositions were used to define several graph parameters to measure how close a given graph is to some known graph class (e.g., to trees or to chordal graphs) where many algorithmic problems could be solved efficiently. The *width* of a tree-decomposition $T(G) = (\{X_i | i \in I\}, T = (I, F))$ is $max_{i \in I} |X_i| - 1$. The *tree-width* of a graph G, denoted by $\mathsf{tw}(G)$, is the minimum width, over all tree-decompositions $T(G)$ of G [27]. The trees are exactly the graphs with tree-width 1. The *length* of a tree-decomposition $T(G)$ of a graph G is $\lambda := \max_{i \in I} \max_{u,v \in X_i} d_G(u, v)$ (i.e., each bag X_i has diameter at most λ in G). The *tree-length* of G, denoted by $\mathsf{tl}(G)$, is the minimum of the length, over all tree-decompositions of G [8]. The chordal graphs are exactly the graphs with tree-length 1. Note that these two graph parameters are not related to each other. For instance, a clique on n vertices has tree-length 1 and tree-width $n-1$, whereas a cycle on $3n$ vertices has tree-width 2 and tree-length n. In [11], yet another graph parameter was introduced, which is very similar to the notion of tree-length and, as it turns out, is related to the TREE t-SPANNER problem. The *breadth* of a tree-decomposition $T(G)$ of a graph G is the minimum integer r such that for every $i \in I$ there is a vertex $v_i \in V(G)$ with $X_i \subseteq D_r(v_i, G)$ (i.e., each bag X_i can be covered by a disk $D_r(v_i, G) := \{u \in V(G) | d_G(u, v_i) \leq r\}$ of radius at most r in G). Note that vertex v_i does not need to belong to X_i. The *tree-breadth* of G, denoted by $\mathsf{tb}(G)$, is the minimum of the breadth, over all tree-decompositions of G. Evidently, for any graph G, $1 \leq \mathsf{tb}(G) \leq \mathsf{tl}(G) \leq 2\mathsf{tb}(G)$ holds. Hence, if one parameter is bounded by a constant for a graph G then the other parameter is bounded for G as well.

We say that a family of graphs $\mathcal{G}$ is *of bounded tree-breadth* (*of bounded tree-width, of bounded tree-length*) if there is a constant c such that for each graph G from $\mathcal{G}$, $\mathsf{tb}(G) \leq c$ (resp., $\mathsf{tw}(G) \leq c$, $\mathsf{tl}(G) \leq c$).

It was shown in [11] that if a graph G admits a tree t-spanner then its tree-breadth is at most $\lceil t/2 \rceil$ and its tree-length is at most t. Furthermore, any graph G with tree-breadth $\mathsf{tb}(G) \leq \rho$ admits a tree $(2\rho\lfloor \log_2 n \rfloor)$-spanner that can be constructed in polynomial time. Thus, these two results gave a new $\log_2 n$-approximation algorithm for the TREE t-SPANNER problem on general (unweighted) graphs (see [11] for details). The algorithm of [11] is conceptually simpler than the previous $O(\log n)$-approximation algorithm proposed for the problem by Emek and Peleg [16].

Dourisboure et al. in [7] concerned with the construction of additive spanners with few edges for n-vertex graphs having a tree-decomposition into bags of diameter at most λ, i.e., the tree-length λ graphs. For such graphs they construct additive 2λ-spanners with $O(\lambda n + n \log n)$ edges, and additive 4λ-spanners with $O(\lambda n)$ edges. Combining these results with the results of [11], we obtain the following interesting fact (in a sense, turning a multiplicative stretch into an additive surplus without much increase in the number of edges).

Theorem 1. *(combining [7] and [11]) If a graph G admits a (multiplicative) tree t-spanner then it has an additive $2t$-spanner with $O(tn + n \log n)$ edges and an additive $4t$-spanner with $O(tn)$ edges, both constructible in polynomial time.*

This fact rises few intriguing questions. Does a polynomial time algorithm exist that, given an n-vertex graph G admitting a (multiplicative) tree t-spanner, constructs an additive $O(t)$-spanner of G with $O(n)$ or $O(n \log n)$ edges (where the number of edges in the spanner is independent of t)? Is a result similar to one presented by Elkin and Peleg in [15] possible? Namely, does a polynomial time algorithm exist that, given an n-vertex graph G admitting a (multiplicative) tree t-spanner, constructs an additive $(t-1)$-spanner of G with $O(n \log n)$ edges? If we allow to use more trees (like in collective tree spanners), does a polynomial time algorithm exist that, given an n-vertex graph G admitting a (multiplicative) tree t-spanner, constructs a system of $\tilde{O}(1)$ collective additive tree $\tilde{O}(t)$-spanners of G (where $\tilde{O}$ is similar to Big-O notation up to a poly-logarithmic factor)? Note that an interesting question whether a multiplicative tree spanner can be turned into an additive tree spanner with a slight increase in the stretch is (negatively) settled already in [16]: if there exist some $\delta = o(n)$ and $\epsilon > 0$ and a polynomial time algorithm that for any graph admitting a tree t-spanner constructs a tree $((6/5 - \epsilon)t + \delta)$-spanner, then P=NP.

We give some partial answers to these questions in Section 2. We investigate there a more general question whether a graph with bounded tree-breadth admits a small system of collective additive tree spanners. We show that any n-vertex graph G has a system of at most $\log_2 n$ collective additive tree $(2\rho \log_2 n)$ -spanners, where $\rho \leq \mathsf{tb}(G)$. This settles also an open question from [7] whether a graph with tree-length λ admits a small system of collective additive tree $\tilde{O}(\lambda)$-spanners.

As a consequence, we obtain that there is a polynomial time algorithm that, given an n-vertex graph G admitting a (multiplicative) tree t-spanner, constructs: i) a system of at most $\log_2 n$ collective additive tree $O(t \log n)$-spanners of G (compare with [11,16] where a multiplicative tree $O(t \log n)$-spanner was constructed for G in polynomial time; thus, we "have turned" a multiplicative tree $O(t \log n)$-spanner into at most $\log_2 n$ collective additive tree $O(t \log n)$-spanners); ii) an additive $O(t \log n)$-spanner of G with at most $n \log_2 n$ edges (compare with Theorem 1).

In Section 3 we generalize the method of Section 2. We define a new notion which combines both the tree-width and the tree-breadth of a graph.

The *k-breadth* of a tree-decomposition $T(G) = (\{X_i | i \in I\}, T = (I, F))$ of a graph G is the minimum integer r such that for each bag $X_i, i \in I$, there is a set of at most k vertices $C_i = \{v_j^i | v_j^i \in V(G), j = 1, \ldots, k\}$ such that for each $u \in X_i$, we have $d_G(u, C_i) \leq r$ (i.e., each bag X_i can be covered with at most k disks of G of radius at most r each; $X_i \subseteq D_r(v_1^i, G) \cup \ldots \cup D_r(v_k^i, G)$). The *k-tree-breadth* of a graph G, denoted by $\mathsf{tb}_k(G)$, is the minimum of the k-breadth, over all tree-decompositions of G. We say that a family of graphs $\mathcal{G}$ is *of bounded k-tree-breadth,* if there is a constant c such that for each graph G from $\mathcal{G}$, $\mathsf{tb}_k(G) \leq c$. Clearly, for every graph G, $\mathsf{tb}(G) = \mathsf{tb}_1(G)$, and $\mathsf{tw}(G) \leq k - 1$ if and only if $\mathsf{tb}_k(G) = 0$. Thus, the notions of the tree-width and the tree-breadth are particular cases of the k-tree-breadth.

In Section 3, we show that any n-vertex graph G with $\mathsf{tb}_k(G) \leq \rho$ has a system of at most $k(1+\log_2 n)$ collective additive tree $(2\rho(1+\log_2 n))$-spanners. In Section 4, we extend a result from [11] and show that if a graph G admits a (multiplicative) t-spanner H with $\mathsf{tw}(H) = k - 1$ then its k-tree-breadth is at most $\lceil t/2 \rceil$. As a consequence, we obtain that, for every fixed k, there is a polynomial time algorithm that, given an n-vertex graph G admitting a (multiplicative) t-spanner with tree-width at most $k-1$, constructs: i) a system of at most $k(1+\log_2 n)$ collective additive tree $O(t \log n)$-spanners of G; ii) an additive $O(t \log n)$-spanner of G with at most $O(kn \log n)$ edges.

All proofs omitted in this extended abstract and a few illustrative figures can be found in the full version of the paper [9].

Preliminaries. All graphs occurring in this paper are connected, finite, unweighted, undirected, loopless and without multiple edges. We call $G = (V, E)$ an *n-vertex m-edge graph* if $|V| = n$ and $|E| = m$. A *clique* is a set of pairwise adjacent vertices of G. By $G[S]$ we denote a subgraph of G induced by vertices of $S \subseteq V$. Let also $G \setminus S$ be the graph $G[V \setminus S]$ (which is not necessarily connected). A set $S \subseteq V$ is called a *separator* of G if the graph $G[V \setminus S]$ has more than one connected component, and S is called a *balanced separator* of G if each connected component of $G[V \setminus S]$ has at most $|V|/2$ vertices. A set $C \subseteq V$ is called a *balanced clique-separator* of G if C is both a clique and a balanced separator of G. For a vertex v of G, the sets $N_G(v) = \{w \in V | vw \in E\}$ and $N_G[v] = N_G(v) \cup \{v\}$ are called the *open neighborhood* and the *closed neighborhood* of v, respectively.

In a graph G the *length* of a path from a vertex v to a vertex u is the number of edges in the path. The *distance* $d_G(u, v)$ between vertices u and v is the length of a shortest path connecting u and v in G. The *diameter* in G of a set $S \subseteq V$ is $\max_{x,y \in S} d_G(x, y)$ and its *radius* in G is $\min_{x \in V} \max_{y \in S} d_G(x, y)$ (in some papers they are called the *weak diameter* and the *weak radius* to indicate that the distances are measured in G not in $G[S]$). The *disk* of G of radius k centered at vertex v is the set of all vertices at distance at most k to v: $D_k(v, G) = \{w \in V | d_G(v, w) \leq k\}$. A disk $D_k(v, G)$ is called a *balanced disk-separator* of G if the set $D_k(v, G)$ is a balanced separator of G.

2 Collective Additive Tree Spanners and Tree-Breadth

In this section, we show that every n-vertex graph G has a system of at most $\log_2 n$ collective additive tree $(2\rho \log_2 n)$-spanners, where $\rho \leq \mathtt{tb}(G)$. We also discuss consequences of this result. Our method is a generalization of techniques used in [13] and [11]. We will assume that $n \geq 4$ since any connected graph with at most 3 vertices has an additive tree 1-spanner.

Note that we do not assume here that a tree-decomposition $T(G)$ of breadth ρ is given for G as part of the input. Our method does not need to know $T(G)$, our algorithm works directly on G. For a given graph G and an integer ρ, even checking whether G has a tree-decomposition of breadth ρ could be a hard problem. For example, while graphs with tree-length 1 (as they are exactly the chordal graphs) can be recognized in linear time, the problem of determining whether a given graph has tree-length at most λ is NP-complete for every fixed $\lambda > 1$ (see [20]). We will need the following results proven in [11].

Lemma 1 ([11]). *Every graph G has a balanced disk-separator $D_r(v, G)$ centered at some vertex v, where $r \leq \mathtt{tb}(G)$. For an arbitrary graph G with n vertices and m edges a balanced disk-separator $D_r(v, G)$ with minimum r can be found in $O(nm)$ time.*

Hierarchical Decomposition of a Graph with Bounded Tree-Breadth. In this subsection, following [11], we show how to decompose a graph with bounded tree-breadth and build a hierarchical decomposition tree for it. This hierarchical decomposition tree is used later for construction of collective additive tree spanners for such a graph.

Let $G = (V, E)$ be an arbitrary connected n-vertex m-edge graph with a disk-separator $D_r(v, G)$. Also, let $G_1, \ldots, G_q$ be the connected components of

$G[V \setminus D_r(v,G)]$. Denote by $S_i := \{x \in V(G_i) | d_G(x, D_r(v,G)) = 1\}$ the neighborhood of $D_r(v,G)$ with respect to G_i. Let also G_i^+ be the graph obtained from component G_i by adding a vertex c_i (*representative* of $D_r(v,G)$) and making it adjacent to all vertices of S_i, i.e., for a vertex $x \in V(G_i)$, $c_i x \in E(G_i^+)$ if and only if there is a vertex $x_D \in D_r(v,G)$ with $xx_D \in E(G)$. In what follows, we will call vertex c_i a *meta vertex representing disk $D_r(v,G)$ in graph G_i^+*. Given a graph G and its disk-separator $D_r(v,G)$, the graphs $G_1^+, \dots, G_q^+$ can be constructed in total time $O(m)$. Furthermore, the total number of edges in the graphs $G_1^+, \dots, G_q^+$ does not exceed the number of edges in G, and the total number of vertices (including q meta vertices) in those graphs does not exceed the number of vertices in $G[V \setminus D_r(v,G)]$ plus q.

Denote by $G_{/e}$ the graph obtained from G by contracting its edge e. Recall that edge e contraction is an operation which removes e from G while simultaneously merging together the two vertices e previously connected. If a contraction results in multiple edges, we delete duplicates of an edge to stay within the class of simple graphs. The operation may be performed on a set of edges by contracting each edge (in any order).

Lemma 2 ([11]). *For any graph G and its edge e, $\mathsf{tb}(G) \leq \rho$ implies $\mathsf{tb}(G_{/e}) \leq \rho$. Consequently, if $\mathsf{tb}(G) \leq \rho$, then $\mathsf{tb}(G_i^+) \leq \rho$ for each $i = 1, \dots, q$.*

Clearly, one can get G_i^+ from G by repeatedly contracting (in any order) edges of G that are not incident to vertices of G_i. In other words, G_i^+ is a minor of G. Recall that a graph G' is a *minor* of G if G' can be obtained from G by contracting some edges, deleting some edges, and deleting some isolated vertices. The order in which a sequence of such contractions and deletions is performed on G does not affect the resulting graph G'.

Let $G = (V,E)$ be a connected n-vertex, m-edge graph and assume that $\mathsf{tb}(G) \leq \rho$. Lemma 1 guarantees that G has a balanced disk-separator $D_r(v,G)$ with $r \leq \rho$, which can be found in $O(nm)$ time by an algorithm that works directly on graph G and does not require construction of a tree-decomposition of G of breadth $\leq \rho$. Using these and Lemma 2, we can build a (rooted) *hierarchical tree* $\mathcal{H}(G)$ for G as follows. If G is a connected graph with at most 5 vertices, then $\mathcal{H}(G)$ is one node tree with root node $(V(G), G)$. Otherwise, find a balanced disk-separator $D_r(v,G)$ in G with minimum r (see Lemma 1) and construct the corresponding graphs $G_1^+, G_2^+, \dots, G_q^+$. For each graph G_i^+ ($i = 1, \dots, q$) (by Lemma 2, $\mathsf{tb}(G_i^+) \leq \rho$), construct a hierarchical tree $\mathcal{H}(G_i^+)$ recursively and build $\mathcal{H}(G)$ by taking the pair $(D_r(v,G), G)$ to be the root and connecting the root of each tree $\mathcal{H}(G_i^+)$ as a child of $(D_r(v,G), G)$.

The depth of this tree $\mathcal{H}(G)$ is the smallest integer k such that $\frac{n}{2^k} + \frac{1}{2^{k-1}} + \dots + \frac{1}{2} + 1 \leq 5$, that is, the depth is at most $\log_2 n - 1$. It is also easy to see that, given a graph G with n vertices and m edges, a hierarchical tree $\mathcal{H}(G)$ can be constructed in $O(nm \log^2 n)$ total time. There are at most $O(\log n)$ levels in $\mathcal{H}(G)$, and one needs to do at most $O(nm \log n)$ operations per level since the total number of edges in the graphs of each level is at most m and the total number of vertices in those graphs can not exceed $O(n \log n)$.

For an internal (i.e., non-leaf) node Y of $\mathcal{H}(G)$, since it is associated with a pair $(D_{r'}(v', G'), G')$, where $r' \leq \rho$, G' is a minor of G and v' is the center of disk $D_{r'}(v', G')$ of G', it will be convenient, in what follows, to denote G' by $G(\downarrow Y)$, v' by $c(Y)$, r' by $r(Y)$, and $D_{r'}(v', G')$ by Y itself. Thus, $(D_{r'}(v', G'), G') = (D_{r(Y)}(c(Y), G(\downarrow Y)), G(\downarrow Y)) = (Y, G(\downarrow Y))$ in these notations, and we identify node Y of $\mathcal{H}(G)$ with the set $Y = D_{r(Y)}(c(Y), G(\downarrow Y))$ and associate with this node also the graph $G(\downarrow Y)$. Each leaf Y of $\mathcal{H}(G)$, since it corresponds to a pair $(V(G'), G')$, we identify with the set $Y = V(G')$ and use, for a convenience, the notation $G(\downarrow Y)$ for G'. If now $(Y^0, Y^1, \ldots, Y^h)$ is the path of $\mathcal{H}(G)$ connecting the root Y^0 of $\mathcal{H}(G)$ with a node Y^h, then the vertex set of the graph $G(\downarrow Y^h)$ consists of some (original) vertices of G plus at most h meta vertices representing the disks $D_{r(Y)}(c(Y^i), G(\downarrow Y^i)) = Y^i$, $i = 0, 1, \ldots, h-1$. Note also that each (original) vertex of G belongs to exactly one node of $\mathcal{H}(G)$.

Construction of Collective Additive Tree Spanners. Unfortunately, the class of graphs of bounded tree-breadth is not hereditary, i.e., induced subgraphs of a graph with tree-breath ρ are not necessarily of tree-breadth at most ρ (for example, a cycle of length ℓ with one extra vertex adjacent to each vertex of the cycle has tree-breadth 1, but the cycle itself has tree-breadth $\ell/3$). Thus, the method presented in [13], for constructing collective additive tree spanners for hereditary classes of graphs admitting balanced disk-separators, cannot be applied directly to the graphs of bounded tree-breadth. Nevertheless, we will show that, with the help of Lemma 2, the notion of hierarchical tree from previous subsection and a careful analysis of distance changes (see Lemma 3), it is possible to generalize the method of [13] and construct in polynomial time for every n-vertex graph G a system of at most $\log_2 n$ collective additive tree $(2\rho \log_2 n)$-spanners, where $\rho \leq \mathsf{tb}(G)$. Unavoidable presence of meta vertices in the graphs resulting from a hierarchical decomposition of the original graph G complicates the construction and the analysis. Recall that, in [13], it was shown that if every induced subgraph of a graph G enjoys a balanced disk-separator with radius $\leq r$, then G admits a system of at most $\log_2 n$ collective additive tree $2r$-spanners.

Let $G = (V, E)$ be a connected n-vertex, m-edge graph and assume that $\mathsf{tb}(G) \leq \rho$. Let $\mathcal{H}(G)$ be a hierarchical tree of G. Consider an arbitrary internal node Y^h of $\mathcal{H}(G)$, and let $(Y^0, Y^1, \ldots, Y^h)$ be the path of $\mathcal{H}(G)$ connecting the root Y^0 of $\mathcal{H}(G)$ with Y^h. Let $\widehat{G}(\downarrow Y^j)$ be the graph obtained from $G(\downarrow Y^j)$ by removing all its meta vertices (note that $\widehat{G}(\downarrow Y^j)$ may be disconnected).

Lemma 3. *For any vertex z from $Y^h \cap V(G)$ there exists an index $i \in \{0, \ldots, h\}$ such that the vertices z and $c(Y^i)$ can be connected in the graph $\widehat{G}(\downarrow Y^i)$ by a path of length at most $\rho(h+1)$. In particular, $d_G(z, c(Y^i)) \leq \rho(h+1)$ holds.*

Consider arbitrary vertices x and y of G, and let $S(x)$ and $S(y)$ be the nodes of $\mathcal{H}(G)$ containing x and y, respectively. Let also $NCA_{\mathcal{H}(G)}(S(x), S(y))$ be the nearest common ancestor of nodes $S(x)$ and $S(y)$ in $\mathcal{H}(G)$ and $(Y^0, Y^1, \ldots, Y^h)$ be the path of $\mathcal{H}(G)$ connecting the root Y^0 of $\mathcal{H}(G)$ with $NCA_{\mathcal{H}(G)}(S(x), S(y)) = Y^h$ (i.e., $Y^0, Y^1, \ldots, Y^h$ are the common ancestors of $S(x)$ and $S(y)$).

Lemma 4. *Any path $P^G_{x,y}$ connecting vertices x and y in G contains a vertex from $Y^0 \cup Y^1 \cup \ldots \cup Y^h$.*

Let $SP^G_{x,y}$ be a shortest path of G connecting vertices x and y, and let Y^i be the node of the path $(Y^0, Y^1, \ldots, Y^h)$ with the smallest index such that $SP^G_{x,y} \bigcap Y^i \neq \emptyset$ in G. The following lemma holds.

Lemma 5. *For each $j = 0, \ldots, i$, $d_G(x, y) = d_{G'}(x, y)$ where $G' := \widehat{G}(\downarrow Y^j)$.*

Let now $B^i_1, \ldots, B^i_{p_i}$ be the nodes at depth i of the tree $\mathcal{H}(G)$. For each node B^i_j that is not a leaf of $\mathcal{H}(G)$, consider its (central) vertex $c^i_j := c(B^i_j)$. If c^i_j is an original vertex of G (not a meta vertex created during the construction of $\mathcal{H}(G)$), then define a connected graph G^i_j obtained from $G(\downarrow B^i_j)$ by removing all its meta vertices. If removal of those meta vertices produced few connected components, choose as G^i_j that component which contains the vertex c^i_j. Denote by T^i_j a BFS–tree of graph G^i_j rooted at vertex c^i_j of B^i_j. If B^i_j is a leaf of $\mathcal{H}(G)$, then B^i_j has at most 5 vertices. In this case, remove all meta vertices from $G(\downarrow B^i_j)$ and for each connected component of the resulting graph construct an additive tree spanner with optimal surplus ≤ 3. Denote the resulting subtree (forest) by T^i_j. The trees T^i_j ($i = 0, 1, \ldots, depth(\mathcal{H}(G))$, $j = 1, 2, \ldots, p_i$), obtained this way, are called *local subtrees* of G. Clearly, the construction of these local subtrees can be incorporated into the procedure of constructing hierarchical tree $\mathcal{H}(G)$ of G.

Lemma 6. *For any two vertices $x, y \in V(G)$, there exists a local subtree T such that $d_T(x, y) \leq d_G(x, y) + 2\rho \log_2 n - 1$.*

This lemma implies two important results. Let G be a graph with n vertices and m edges having $\mathsf{tb}(G) \leq \rho$. Also, let $\mathcal{H}(G)$ be its hierarchical tree and $\mathcal{LT}(G)$ be the family of all its local subtrees (defined above). Consider a graph H obtained by taking the union of all local subtrees of G (by putting all of them together), i.e., $H := \bigcup \{T^i_j | T^i_j \in \mathcal{LT}(G)\} = (V, \cup\{E(T^i_j) | T^i_j \in \mathcal{LT}(G)\})$. Clearly, H is a spanning subgraph of G, constructible in $O(nm \log^2 n)$ total time, and, for any two vertices x and y of G, $d_H(x, y) \leq d_G(x, y) + 2\rho \log_2 n - 1$ holds. Also, since for every level i ($i = 0, 1, \ldots, depth(\mathcal{H}(G))$) of hierarchical tree $\mathcal{H}(G)$, the corresponding local subtrees $T^i_1, \ldots, T^i_{p_i}$ are pairwise vertex-disjoint, their union has at most $n-1$ edges. Therefore, H cannot have more than $(n-1) \log_2 n$ edges in total. Thus, we have proven the following result.

Theorem 2. *Every graph G with n vertices and $\mathsf{tb}(G) \leq \rho$ admits an additive $(2\rho \log_2 n)$–spanner with at most $n \log_2 n$ edges. Furthermore, such a sparse additive spanner of G can be constructed in polynomial time.*

Instead of taking the union of all local subtrees of G, one can fix i ($i \in \{0, 1, \ldots, depth(\mathcal{H}(G))\}$) and consider separately the union of only local subtrees $T^i_1, \ldots, T^i_{p_i}$, corresponding to the level i of the hierarchical tree $\mathcal{H}(G)$, and then extend in linear $O(m)$ time that forest to a spanning tree T^i of G (using, for example, a variant of the Kruskal's Spanning Tree algorithm for the unweighted graphs). We call this tree T^i the *spanning tree of G corresponding to the level i of the hierarchical tree $\mathcal{H}(G)$*. In this way we can obtain at most $\log_2 n$ spanning trees for G, one for each level i of $\mathcal{H}(G)$. Denote the collection of those spanning trees by $\mathcal{T}(G)$. Thus, we obtain the following theorem.

Theorem 3. *Every graph G with n vertices and $\mathtt{tb}(G) \leq \rho$ admits a system $\mathcal{T}(G)$ of at most $\log_2 n$ collective additive tree $(2\rho \log_2 n)$–spanners. Furthermore, such a system of collective additive tree spanners of G can be constructed in polynomial time.*

Additive Spanners for Graphs Having (Multiplicative) Tree t–spanners. Now we give implications of the above results for the class of tree t–spanner admissible graphs. In [11], the following important ("bridging") lemma was proven.

Lemma 7 ([11]). *If a graph G admits a tree t-spanner then its tree-breadth is at most $\lceil t/2 \rceil$.*

Note that the tree-breadth bounded by $\lceil t/2 \rceil$ provides only a necessary condition for a graph to have a multiplicative tree t-spanner. There are (chordal) graphs which have tree-breadth 1 but any multiplicative tree t-spanner of them has $t = \Omega(\log n)$ [11]. Furthermore, a cycle on $3n$ vertices has tree-breadth n but admits a system of 2 collective additive tree 0-spanners.

Combining Lemma 7 with Theorem 2 and Theorem 3, we deduce the following results.

Theorem 4. *Let G be a graph with n vertices and m edges having a (multiplicative) tree t–spanner. Then, G admits an additive $(2\lceil t/2 \rceil \log_2 n)$–spanner with at most $n \log_2 n$ edges constructible in $O(nm \log^2 n)$ time.*

Theorem 5. *Let G be a graph with n vertices and m edges having a (multiplicative) tree t–spanner. Then, G admits a system $\mathcal{T}(G)$ of at most $\log_2 n$ collective additive tree $(2\lceil t/2 \rceil \log_2 n)$–spanners constructible in $O(nm \log^2 n)$ time.*

3 Graphs with Bounded k-Tree-Breadth, $k \geq 2$

In this section, we extend the approach of Section 2 and show that any n-vertex graph G with $\mathtt{tb}_k(G) \leq \rho$ has a system of at most $k(1+\log_2 n)$ collective additive tree $(2\rho(1+\log_2 n))$-spanners constructible in polynomial time for every fixed k.

Balanced Separators for Graphs with Bounded k-Tree-Breadth. We say that a graph $G = (V, E)$ with $|V| \geq k$ has a *balanced $\mathbf{D_r^k}$-separator* if there exists a collection of k disks $D_r(v_1, G), D_r(v_2, G), \ldots, D_r(v_k, G)$ in G, centered at (different) vertices $v_1, v_2, \ldots, v_k$ and each of radius r, such that the union of those disks $\mathbf{D_r^k} := \bigcup_{i=1}^{k} D_r(v_i, G)$ forms a balanced separator of G, i.e., each connected component of $G[V \setminus \mathbf{D_r^k}]$ has at most $|V|/2$ vertices. The following result generalizes Lemma 1.

Lemma 8. *Every graph G with at least k vertices and $\mathsf{tb}_k(G) \leq \rho$ has a* balanced $\mathbf{D}_{\rho}^{\mathbf{k}}$-separator. *For an arbitrary graph G with $n \geq k$ vertices and m edges, a balanced $\mathbf{D}_{\mathbf{r}}^{\mathbf{k}}$-separator with the smallest radius r can be found in $O(n^k m)$ time.*

Collective Additive Tree Spanners of a Graph with Bounded k-Tree-Breadth. Using Lemma 8, we generalize the technique of Section 2 and obtain the following results for the graphs with bounded k-tree-breadth ($k \geq 2$). Details can be found in the full version of this extended abstract (see [9]).

Theorem 6. *Every graph G with n vertices and $\mathsf{tb}_k(G) \leq \rho$ admits an additive $(2\rho(1 + \log_2 n))$–spanner with at most $O(kn \log n)$ edges constructible in polynomial time for every fixed k.*

Theorem 7. *Every n-vertex graph G with $\mathsf{tb}_k(G) \leq \rho$ admits a system $\mathcal{T}(G)$ of at most $k(1+\log_2 n)$ collective additive tree $(2\rho(1+\log_2 n))$-spanners constructible in polynomial time for every fixed k.*

4 Additive Spanners for Graphs Admitting t–Spanners of Bounded Tree-Width

In this section, we show that if a graph G admits a (multiplicative) t-spanner H with $\mathsf{tw}(H) = k - 1$ then its k-tree-breadth is at most $\lceil t/2 \rceil$. As a consequence, we obtain that, for every fixed k, there is a polynomial time algorithm that, given an n-vertex graph G admitting a (multiplicative) t-spanner with tree-width at most $k - 1$, constructs a system of at most $k(1 + \log_2 n)$ collective additive tree $O(t \log n)$-spanners of G.

k-Tree-Breadth of a Graph Admitting a t-Spanner of Bounded Tree-width. Let H be a graph with tree-width $k - 1$, and let $T(H) = (\{X_i | i \in I\}, T = (I, F))$ be its tree-decomposition of width $k - 1$. For an integer $r \geq 0$, denote by $X_i^{(r)}$, $i \in I$, the set $D_r(X_i, H) := \bigcup_{x \in X_i} D_r(x, H)$. Clearly, $X_i^{(0)} = X_i$ for every $i \in I$. The following important lemmas hold.

Lemma 9. *For every integer $r \geq 0$, $T^{(r)}(H) := (\{X_i^{(r)} | i \in I\}, T = (I, F))$ is a tree-decomposition of H with k-breadth $\leq r$.*

Lemma 10. *If a graph G admits a t-spanner with tree-width $k-1$, then $\mathsf{tb}_k(G) \leq \lceil t/2 \rceil$.*

Consequences. Now we give two implications of the above results for the class of graphs admitting (multiplicative) t–spanners with tree-width $k - 1$. They are direct consequences of Lemma 10, Theorem 6 and Theorem 7.

Theorem 8. *Let G be a graph with n vertices and m edges having a (multiplicative) t–spanner with tree-width $k - 1$. Then, G admits an additive $(2\lceil t/2 \rceil(1 + \log_2 n))$–spanner with at most $O(kn \log n)$ edges constructible in polynomial time for every fixed k.*

Theorem 9. *Let G be a graph with n vertices and m edges having a (multiplicative) t–spanner with tree-width $k-1$. Then, G admits a system $\mathcal{T}(G)$ of at most $k(1+\log_2 n)$ collective additive tree $(2\lceil t/2\rceil(1+\log_2 n))$–spanners constructible in polynomial time for every fixed k.*

References

1. Bartal, Y.: Probabilistic approximations of metric spaces and its algorithmic applications. In: FOCS 1996, pp. 184–193 (1996)
2. Bartal, Y., Blum, A., Burch, C., Tomkins, A.: A polylog-competitive algorithm for metrical task systems. In: STOC 1997, pp. 711–719 (1997)
3. Cai, L., Corneil, D.G.: Tree spanners. SIAM J. Disc. Math. 8, 359–387 (1995)
4. Charikar, M., Chekuri, C., Goel, A., Guha, S., Plotkin, S.A.: Approximating a finite metric by a small number of tree metrics. In: FOCS 1998, pp. 379–388 (1998)
5. Chew, L.P.: There are planar graphs almost as good as the complete graph. J. of Comp. and Sys. Sci. 39, 205–219 (1989)
6. Diestel, R.: Graph Theory, 2nd edn. Graduate Texts in Math., vol. 173. Springer (2000)
7. Dourisboure, Y., Dragan, F.F., Gavoille, C., Yan, C.: Spanners for bounded tree-length graphs. Theor. Comput. Sci. 383, 34–44 (2007)
8. Dourisboure, Y., Gavoille, C.: Tree-decompositions with bags of small diameter. Disc. Math. 307, 2008–2029 (2007)
9. Dragan, F.F., Abu-Ata, M.: Collective Additive Tree Spanners of Bounded Tree-Breadth Graphs with Generalizations and Consequences (Full version of this extended abstract), CoRR abs/1207.2506 (2012)
10. Dragan, F.F., Fomin, F.V., Golovach, P.A.: Spanners in sparse graphs. J. Comput. Syst. Sci. 77, 1108–1119 (2011)
11. Dragan, F.F., Köhler, E.: An Approximation Algorithm for the Tree t-Spanner Problem on Unweighted Graphs via Generalized Chordal Graphs. In: Goldberg, L.A., Jansen, K., Ravi, R., Rolim, J.D.P. (eds.) APPROX/RANDOM 2011. LNCS, vol. 6845, pp. 171–183. Springer, Heidelberg (2011)
12. Dragan, F.F., Yan, C.: Collective Tree Spanners in Graphs with Bounded Parameters. Algorithmica 57, 22–43 (2010)
13. Dragan, F.F., Yan, C., Lomonosov, I.: Collective tree spanners of graphs. SIAM J. Disc. Math. 20, 241–260 (2006)
14. Elkin, M., Emek, Y., Spielman, D.A., Teng, S.H.: Lower-stretch spanning trees. SIAM J. Comput. 38, 608–628 (2008)
15. Elkin, M., Peleg, D.: Approximating k-spanner problems for $k \geq 2$. Theor. Comput. Sci. 337, 249–277 (2005)
16. Emek, Y., Peleg, D.: Approximating minimum max-stretch spanning trees on unweighted graphs. SIAM J. Comput. 38, 1761–1781 (2008)
17. Fakcharoenphol, F., Rao, S., Talwar, K.: A tight bound on approximating arbitrary metrics by tree metrics. J. Comput. Syst. Sci. 69, 485–497 (2004)
18. Gupta, A., Kumar, A., Rastogi, R.: Traveling with a pez dispenser (or, routing issues in mpls). SIAM J. Comput. 34, 453–474 (2004)
19. Liestman, A.L., Shermer, T.: Additive graph spanners. Networks 23, 343–364 (1993)
20. Lokshtanov, D.: On the complexity of computing tree-length. Disc. Appl. Math. 158, 820–827 (2010)

21. Peleg, D.: Low Stretch Spanning Trees. In: Diks, K., Rytter, W. (eds.) MFCS 2002. LNCS, vol. 2420, pp. 68–80. Springer, Heidelberg (2002)
22. Peleg, D., Reshef, E.: Low complexity variants of the arrow distributed directory. J. Comput. System Sci. 63, 474–485 (2001)
23. Peleg, D., Schäffer, A.A.: Graph Spanners. J. Graph Theory 13, 99–116 (1989)
24. Peleg, D., Ullman, J.D.: An optimal synchronizer for the hypercube. SIAM J. Comput. 18, 740–747 (1989)
25. Peleg, D., Upfal, E.: A tradeoff between space and efficiency for routing tables (extended abstract). In: STOC 1988, pp. 43–52 (1988)
26. Prisner, E.: Distance Approximating Spanning Trees. In: Reischuk, R., Morvan, M. (eds.) STACS 1997. LNCS, vol. 1200, pp. 499–510. Springer, Heidelberg (1997)
27. Robertson, N., Seymour, P.D.: Graph minors. II. Algorithmic aspects of tree-width. J. of Algorithms 7, 309–322 (1998)