HAL
open science

Easy computation of eccentricity approximating trees

Guillaume Ducoffe

To cite this version:

Guillaume Ducoffe. Easy computation of eccentricity approximating trees. Discrete Applied Mathematics, 2019, 260, pp.267-271. hal-02011252

HAL Id: hal-02011252

https://hal.science/hal-02011252

Submitted on 7 Feb 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Easy computation of eccentricity approximating trees

Guillaume Ducoffe[a,b,c]

[a] *National Institute for Research and Development in Informatics, Romania*
[b] *University of Bucharest, Faculty of Mathematics and Computer Science, Romania*
[c] *The Research Institute of the University of Bucharest ICUB, Romania*

Abstract

A spanning tree T of a graph $G = (V, E)$ is called *eccentricity k-approximating* if we have $ecc_T(v) \leq ecc_G(v) + k$ for every $v \in V$. Let $ets(G)$ be the minimum k such that G admits an eccentricity k-approximating spanning tree. As our main contribution in this paper, we prove that $ets(G)$ can be computed in $\mathcal{O}(nm)$-time along with a corresponding spanning tree. This answers an open question of [Dragan et al., DAM'17]. Moreover we also prove that for some classes of graphs such as chordal graphs and hyperbolic graphs, one can compute an eccentricity $\mathcal{O}(ets(G))$-approximating spanning tree in quasi linear time. Our proofs are based on simple relationships between eccentricity approximating trees and shortest-path trees.

Keywords: eccentricity-approximating tree; shortest-path tree; complexity; graph algorithms.

1. Introduction

We refer to [1] for any unspecified graph terminology. All graphs considered in this note are finite, simple, connected and unweighted. For every $u, v \in V$, let $dist_G(u, v)$ be the minimum length (number of edges) of a uv-path in G. A spanning tree T of G is a *k-additive tree spanner* if we have $dist_T(u, v) \leq dist_G(u, v) + k$ for every $u, v \in V$. There has been a great deal of research on additive tree spanners (*e.g.*, see [2, 7, 9, 12, 13, 11, 17]). This is in part motivated by their various applications, *e.g.* in distributed systems [8]. Unfortunately, even in restricted classes such as chordal graphs, there are graphs with no k-additive tree spanner for every fixed value of k [17]. Furthermore, computing the minimum k such that a given graph G admits a k-additive tree spanner is NP-hard [14].

The *eccentricity* of a vertex v is defined as $ecc_G(v) = \max\{dist_G(u, v) \mid u \in V\}$. A spanning tree T of G is called *eccentricity k-approximating* if we have $ecc_T(v) \leq ecc_G(v) + k$ for every $v \in V$. In what follows, we will denote by $ets(G)$ the minimum k such that G admits an eccentricity k-approximating spanning tree. Clearly, a k-additive tree spanner is also an eccentricity k-approximating spanning tree, but the converse is not true. In particular, Prisner proved in [19] that every chordal graph admits an eccentricity 2-approximating spanning tree. His result has been recently generalized to larger classes of graphs [6, 10]. – For instance, it was proved in [6] that we have $ets(G) = \mathcal{O}(\delta(G))$ for any graph G, with $\delta(G)$ being the graph hyperbolicity (see [15]). – However, the complexity of the following problem has been left open in [10]:

Problem 1 (Eccentricity Approximating Tree).

Input: *A graph $G = (V, E)$; an integer $k \geq 0$.*

Question: *Does G admit an eccentricity k-approximating spanning tree ?*

Our Results. We will prove in Section 3 that Eccentricity Approximating Tree can be solved in polynomial time (Theorem 1). Specifically, we will prove the following intermediate results, that are interesting in their own right. Given $G = (V, E)$ and $o \in V \cup E$ (either a vertex or an edge) we call *shortest-path tree* rooted at o any spanning tree T such that we have $dist_G(v, o) = dist_T(v, o)$ for any $v \in V$, and if $o \in E$ then $o \in E(T)$.

1. Our first result is that for any $k \geq ets(G)$, there always exists an eccentricity k-approximating spanning tree T which is a *shortest-path tree* rooted at some $o \in V \cup E$. Furthermore, o must coincide with the set of *central nodes* in T.
2. Conversely, for any $o \in V \cup E$, *any* shortest-path tree T that is rooted at o is eccentricity k_o-approximating, for some polynomial-time computable k_o. This upper-bound k_o on the additive distortion is always reached if o coincides with the set of *central nodes* in T.

We deduce from the above an $\mathcal{O}(nm)$-time algorithm for computing $ets(G)$. Unfortunately, this time bound prohibits to use our method in order to approximate all eccentricities in real-life graphs.

In Section 4, we focus our efforts on the design of faster *approximation algorithms* for the Eccentricity Approximating Tree problem. Doing so we answer another open question from [10], where the authors asked whether their proposed heuristics provide any provable good approximation. Our main result in this part is that *any* shortest-path tree that is rooted at some arbitrary almost central vertex in G is eccentricity $\mathcal{O}(ets(G))$-approximating (Theorem 2). In particular, this is the case for *all* the spanning trees which are outputted by the quasi linear-time algorithms given in [6, 10]. More generally, our approximation framework applies to any graph class where an almost central vertex can be computed efficiently (*e.g.*, see [4, 5]).

We left open whether the *value* of $ets(G)$ can be approximated in $o(nm)$-time. Indeed, given a spanning tree T of G, it is not even clear whether the minimum k such that T is eccentricity k-approximating can be approximated in $o(nm)$-time.

Our proofs are based on very simple properties of the eccentricity function in trees. For convenience of the reader, we gather all the properties that we will need in Section 2.

2. Preliminaries

We start introducing additional notations. Given a graph $G = (V, E)$ let $S \subseteq V$ be any vertex-subset (possibly, S is an edge of G). For every $v \in V$, we define $dist_G(v, S) = \min\{dist_G(v, s) \mid s \in S\}$. Let $ecc_G(S) = \max\{dist_G(v, S) \mid v \in V\}$. The *diameter* and the *radius* of G are defined, respectively, as $diam(G) = \max\{ecc_G(v) \mid v \in V\}$ and $rad(G) = \min\{ecc_G(v) \mid v \in V\}$. Finally, let $\mathcal{C}(G)$ contain all the vertices of G with minimum eccentricity. We call $\mathcal{C}(G)$ the center of G, while the vertices in $\mathcal{C}(G)$ are sometimes called central vertices.

The following result is the main ingredient in our proofs of Theorems 1 and 2.

Proposition 1. *Let T be any tree, let $S \subseteq V(T)$ induce a connected subtree $T[S]$, and let $x \in V(T)$. Then, $ecc_T(x) \leq dist_T(x, S) + diam(T[S]) + ecc_T(S)$, and this becomes an equality if $S = \mathcal{C}(T)$.*

This above property is sometimes called *unimodality* in the literature. Before we can give a proof of Proposition 1, we need to introduce some other well-known properties of trees:

Lemma 1 ([16]). *For any tree T, $\mathcal{C}(T)$ is either a single node or an edge.*

Lemma 2 ([1]). *Every edge in a tree T is a bridge.*

The next lemma is folklore and it can be proved the same way as [3, Theorem 2]. However since we did not find any reference, we also give its proof for self-containment.

Lemma 3. *If T is a tree of order at least three then, for any diametral pair (u, v), we have $dist_T(u, \mathcal{C}(T)) = dist_T(v, \mathcal{C}(T)) = ecc_T(\mathcal{C}(T))$. Furthermore nodes u and v are in different connected components of $T \setminus \mathcal{C}(T)$, and if $\mathcal{C}(T)$ is an edge then the components of $T \setminus \mathcal{C}(T)$ that contain u and v, respectively, are not adjacent to the same central node in $\mathcal{C}(T)$.*

Proof. By Lemma 1, we have that $\mathcal{C}(T)$ is either a single node or an edge. In particular we have $|\mathcal{C}(T)| \leq 2$ and so, since we assume $|V(T)| \geq 3$, we get $u, v \notin \mathcal{C}(T)$. There are two cases:

- Case $\mathcal{C}(T) = \{c\}$ is a single node. By the triangle inequality, $diam(T) = dist_T(u, v) \leq dist_T(u, c) + dist_T(c, v) \leq 2 \cdot rad(T)$. Let T_u be the connected component of $T \setminus \{c\}$ containing u. Since T is a tree, there is a unique neighbor x of node c in T_u. In particular, x is at a distance $\leq rad(T) - 1$ from every node in T_u. Since we have $x \notin \mathcal{C}(T)$ and so, $ecc_T(x) \geq rad(T) + 1$, this implies the existence of a node $w \notin V(T_u)$ such that $dist_T(c, w) \geq rad(T)$. Therefore, $diam(T) \geq 2 \cdot rad(T)$. Overall, $dist_T(u, v) = diam(T) = 2 \cdot rad(T)$, and so the unique uv-path in T goes by c. It implies that u and v are in different connected components of $T \setminus \{c\}$. Note that as a byproduct of our analysis, we also get in this case $dist_T(u, \mathcal{C}(T)) = dist_T(v, \mathcal{C}(T)) = rad(T) = ecc_T(\mathcal{C}(T))$.

- Case $\mathcal{C}(T) = \{c, c'\}$ is an edge. By Lemma 2, cc' is a bridge. Let $T_c, T_{c'}$ be the two subtrees in $T \setminus cc'$, with $c \in V(T_c)$ and $c' \in V(T_{c'})$. We must have that $ecc_{T_c}(c) \leq rad(T) - 1$ (resp., $ecc_{T_{c'}}(c') \leq rad(T) - 1$) since otherwise we would get $ecc_T(c') > rad(T)$ (resp. $ecc_T(c) > rad(T)$). Then, $ecc_{T_c}(c) = rad(T) - 1$ (resp., $ecc_{T_{c'}}(c') = rad(T) - 1$) since otherwise we would get $ecc_T(c') < rad(T)$ (resp. $ecc_T(c) < rad(T)$). In particular, $diam(T) = 2 \cdot rad(T) - 1$. If we had $u, v \in V(T_c)$ (resp., $u, v \in V(T_{c'})$) then we would get $dist_T(u, v) \leq 2(rad(T) - 1) < diam(T)$, a contradiction. Therefore, we must have either $u \in V(T_c)$ and $v \in V(T_{c'})$ or $u \in V(T_{c'})$ and $v \in V(T_c)$. As a byproduct of our analysis, we obtain in this case $dist_T(u, \mathcal{C}(T)) = dist_T(v, \mathcal{C}(T)) = rad(T) - 1 = ecc_T(\mathcal{C}(T))$.

Finally, in both cases we get as desired $dist_T(u, \mathcal{C}(T)) = dist_T(v, \mathcal{C}(T)) = ecc_T(\mathcal{C}(T))$. □

We are now ready to prove Proposition 1:

Proof of Proposition 1. We will assume for the proof that T is of order at least three (otherwise, T is either a single node or an edge, and so the result trivially holds in this case). First let S be arbitrary. We claim that the desired inequality directly follows from the triangle inequality. Indeed, for every $x, y \in V(T)$, let x' and y' be the closest nodes to x and y, respectively, in S. Then, $dist_T(x, y) \leq dist_T(x, x') + dist_T(x', y') + dist_T(y', y) \leq dist_T(x, S) + diam(T[S]) + dist_T(y, S)$. In particular, $ecc_T(x) \leq dist_T(x, S) + diam(T[S]) + ecc_T(S)$, and in the same way $ecc_T(y) \leq ecc_T(S) + diam(T[S]) + dist_T(y, S)$. Suppose now $S = \mathcal{C}(T)$ and let (u, v) be any diametral pair.

Recall that by Lemma 3, we have $dist_T(u, \mathcal{C}(T)) = dist_T(v, \mathcal{C}(T)) = ecc_T(\mathcal{C}(T))$. Furthermore let $x \in V(T) \setminus \mathcal{C}(T)$ be arbitrary (the proof if similar if $x \in \mathcal{C}(T)$). We also have by Lemma 3 that there exists $y \in \{u, v\}$ such that: x, y are in different components of $T \setminus \mathcal{C}(T)$; and if $\mathcal{C}(T)$ is an edge, the components of $T \setminus \mathcal{C}(T)$ that contain x and y, respectively, are not adjacent to the same central node in $\mathcal{C}(T)$. In this situation, the unique xy-path in T goes by all the nodes in $\mathcal{C}(T)$. Summarizing, we have $ecc_T(x) \geq dist_T(x, y) = dist_T(x, \mathcal{C}(T)) + diam(T[\mathcal{C}(T)]) + dist_T(y, \mathcal{C}(T)) = dist_T(x, \mathcal{C}(T)) + diam(T[\mathcal{C}(T)]) + ecc_T(\mathcal{C}(T))$. □

3. The algorithm

The purpose of this section is to prove that for any graph G, we can compute $ets(G)$ in polynomial time (Theorem 1). For that, we remind that for any vertex v a shortest-path tree rooted at v is any spanning tree T of G such that we have $dist_T(u, v) = dist_G(u, v)$ for any vertex u. We can compute such a tree T in linear time, using breadth-first search. In the same way, a shortest-path tree rooted at an edge e is any spanning tree T of G such that we have: $e \in E(T)$, and $dist_T(u, e) = dist_G(u, e)$ for any vertex u. We can also compute such a tree T in linear time, using a modified breadth-first search.

From Proposition 1, we will prove next that there always exists an eccentricity approximating tree that is a *shortest-path tree* (rooted either at a vertex or an edge of G). Our main result will follow from this nice property. We note that all the constructions of eccentricity approximating spanning trees that are given in [6, 10, 18] are also based on shortest-path trees.

Lemma 4. *Let $G = (V, E)$ have an eccentricity k-approximating spanning tree. There is one such a tree T such that, for every $v \in V$ we have $dist_T(v, \mathcal{C}(T)) = dist_G(v, \mathcal{C}(T))$.*

Proof. We choose an eccentricity k-approximating spanning tree T of G such that $\sum_{x \in V} ecc_T(x)$ is minimized. Suppose by contradiction there exists $v \in V$ such that $dist_T(v, \mathcal{C}(T)) > dist_G(v, \mathcal{C}(T))$. Without loss of generality, we choose such a vertex v such that $dist_G(v, \mathcal{C}(T))$ is minimized. In particular, for every fixed $u \in N_G(v)$ that is on a shortest $v\mathcal{C}(T)$-path in G we have that $dist_T(u, \mathcal{C}(T)) = dist_G(u, \mathcal{C}(T)) = dist_G(v, \mathcal{C}(T)) - 1$. Then, let $w \in N_T(v)$ be on the unique $v\mathcal{C}(T)$-path in T. We remove the edge vw from T and we replace it by vu. We first prove as a claim that doing so, we obtain a new spanning tree T'. Indeed, let T_v, T_w be the two subtrees in $T \setminus vw$ such that $v \in V(T_v)$, $w \in V(T_w)$. By the choice of w, we have $\mathcal{C}(T) \subseteq V(T_w)$. Furthermore, since we assume $dist_T(u, \mathcal{C}(T)) < dist_T(v, \mathcal{C}(T))$ we also have $u \in V(T_w)$. Then, by adding the edge uv between T_v, T_w we get a connected graph. Recall that a connected n-node graph is a tree if and only if it has exactly $n - 1$ edges [1]. As a result since T and T' have equal number of nodes and edges, it follows as claimed that T' is a tree. We continue proving as our second claim that we have $dist_{T'}(x, \mathcal{C}(T)) \leq dist_T(x, \mathcal{C}(T))$ for every $x \in V$. Indeed, this is true for any node $x \in V(T_w)$. By construction, this is also true for node v, and so this is also true for every node $x \in V(T_v)$. So, the claim is proved, and it directly implies $ecc_{T'}(\mathcal{C}(T)) \leq ecc_T(\mathcal{C}(T))$. Then by Proposition 1, we have for any node x: $ecc_{T'}(x) \leq dist_{T'}(x, \mathcal{C}(T)) + diam(T'[\mathcal{C}(T)]) + ecc_{T'}(\mathcal{C}(T)) \leq dist_T(x, \mathcal{C}(T)) + diam(T[\mathcal{C}(T)]) + ecc_T(\mathcal{C}(T)) = ecc_T(x) \leq ecc_G(x) + k$. However, since $dist_{T'}(v, \mathcal{C}(T)) = dist_G(v, \mathcal{C}(T)) < dist_T(v, \mathcal{C}(T))$, we obtain $ecc_{T'}(v) < ecc_T(v)$, and so, $\sum_{x \in V} ecc_{T'}(x) < \sum_{x \in V} ecc_T(x)$, that is a contradiction. □

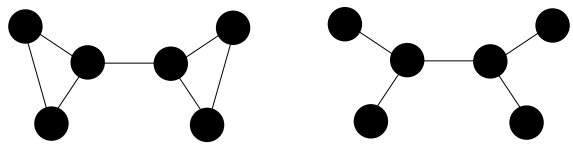

Figure 1: A graph G (left) and its unique eccentricity 0-approximating spanning tree (right).

We are now ready to prove the main result in this paper.

Theorem 1. *For every graph $G = (V, E)$, we can compute $ets(G)$ and a corresponding spanning tree in $\mathcal{O}(nm)$-time and $\mathcal{O}(n^2)$-space.*

Proof. For every $v \in V$, we define $k(v) = \max\{(dist_G(v, x) + ecc_G(v)) - ecc_G(x) \mid x \in V\}$. In the same way for every $e \in E$, we define $k(e) = \max\{(dist_G(v, e) + 1 + ecc_G(e)) - ecc_G(x) \mid x \in V\}$. We first prove the following equality (of which we will deduce a straightforward algorithm in order to prove the theorem):

$$ets(G) = \min\{k(o) \mid o \in V \cup E\}.$$

Indeed, let T be an eccentricity $ets(G)$-approximating tree of G that satisfies the additional condition of Lemma 4. By Lemma 1, the center of any tree is either a single node or an edge. So, there are two cases.

- **Case** $\mathcal{C}(T) = \{v\}$ for some $v \in V$. Then, by Proposition 1 we have:

  $$ets(G) = \max\{ecc_T(x) - ecc_G(x) \mid x \in V\} = \max\{(dist_T(x, v) + ecc_T(v)) - ecc_G(x) \mid x \in V\}.$$

  By Lemma 4, T is a shortest-path tree rooted at v, and so $ets(G) = k(v)$.

- **Case** $\mathcal{C}(T) = \{u, v\}$ for some $e = uv \in E$. Then by Proposition 1, we have:

  $$ets(G) = \max\{ecc_T(x) - ecc_G(x) \mid x \in V\} = \max\{(dist_T(x, e) + 1 + ecc_T(e)) - ecc_G(x) \mid x \in V\}.$$

  By Lemma 4, T is a shortest-path tree rooted at e, and so $ets(G) = k(e)$. – We observe that we cannot avoid this second case given that there are graphs G for which every eccentricity $ets(G)$-approximating spanning tree has an edge as its center; see Fig. 1. –

Overall, we proved with this above case analysis that $ets(G) \geq \min\{k(o) \mid o \in V \cup E\}$. Conversely, for any $o \in V \cup E$, let T^o be a shortest-path tree rooted at o. We have by Proposition 1 applied to $S = o$ that for any vertex x, $ecc_{T^o}(x) \leq dist_{T^o}(x, o) + diam(T^o[o]) + ecc_{T^o}(o) = dist_G(x, o) + diam(G[o]) + ecc_G(o)$. Recall that $k_o = \max\{(dist_G(v, o) + diam(G[o]) + ecc_G(o)) - ecc_G(x) \mid x \in V\}$. In particular, every shortest-path tree rooted at o is eccentricity $k(o)$-approximating. As a result, $ets(G) \leq \min\{k(o) \mid o \in V \cup E\}$, and so we proved the desired equality.

The algorithm now proceeds as follows. We compute $k(o)$ for every $o \in V \cup E$. Then, we choose $o_{\min} \in V \cup E$ such that $k(o_{\min})$ is minimized. Note that we proved above $k(o_{\min}) = ets(G)$. Finally, we output $k(o_{\min})$ and any shortest-path tree $T_{\min}$ which is rooted at $o_{\min}$. As already observed above, the spanning tree $T_{\min}$ is always eccentricity $k(o_{\min})$-approximating (Proposition 1). Therefore, the algorithm is correct.

Complexity. We start by computing all the distances in G. It takes $\mathcal{O}(nm)$-time and $\mathcal{O}(n^2)$-space. Then, for any $o \in V \cup E$, we can compute $ecc_G(o)$ in $\mathcal{O}(n)$-time by scanning all the vertices.

After this pre-processing, we can compute $k(o)$, for any $o \in V \cup E$, in $\mathcal{O}(n)$-time, by scanning a second time all the vertices. Overall, all the values $k(v),\ v \in V$ and $k(e),\ e \in E$ can be computed in total $\mathcal{O}(nm)$-time. If we have done that, we can compute $ets(G)$ and $o_{\min}$ in $\mathcal{O}(n+m)$-time by scanning the values $k_o,\ o \in V \cup E$. Finally, we can compute a shortest-path tree rooted at $o_{\min}$ in $\mathcal{O}(n+m)$-time by using breadth-first search. □

4. Approximation algorithms

We present in this section a general approximation framework for the Eccentricity Approximating Tree problem (Theorem 2). Before we state our main result in this part, let us further explain our approach. For every $i \geq 0$ we denote $\mathcal{C}^i(G)$ the set of all the vertices of G with eccentricity $rad(G)+i$. In particular, we have $\mathcal{C}^0(G) = \mathcal{C}(G)$. If $i = \mathcal{O}(1)$ then, we abusively call the vertices in $\mathcal{C}^i(G)$ almost central. Some heuristics were proposed in [10] for the Eccentricity Approximating Tree problem. Their output is a shortest-path tree rooted at some central vertex. We prove in what follows that such heuristics provide a quasi 2-approximation algorithm. Our analysis also extends to the recent construction given in [6] where the output is a shortest-path tree rooted at some almost central vertex.

Lemma 5. *Let $G = (V, E)$ be a graph and let T be an eccentricity k-approximating spanning tree of G. For every $v \in \mathcal{C}^i(G),\ i \geq 0$, we have $dist_G(v, \mathcal{C}(T)) \leq i + k$.*

Proof. On one hand we have $ecc_T(v) \leq ecc_G(v) + k = rad(G) + i + k \leq rad(T) + i + k$. On the other hand by Proposition 1, $ecc_T(v) = dist_T(v, \mathcal{C}(T)) + diam(T[\mathcal{C}(T)]) + ecc_T(\mathcal{C}(T)) \geq dist_G(v, \mathcal{C}(T)) + diam(T[\mathcal{C}(T)]) + ecc_T(\mathcal{C}(T))$. By another application of Proposition 1 to $x \in \mathcal{C}(T)$, we also obtain that $rad(T) = diam(T[\mathcal{C}(T)]) + ecc_T(\mathcal{C}(T))$. In particular, $ecc_T(v) \geq dist_G(v, \mathcal{C}(T)) + rad(T)$. Altogether combined, we have $dist_G(v, \mathcal{C}(T)) \leq i + k$. □

Our main result in this section somewhat confirms the intuition that, in order to obtain an eccentricity k-approximating spanning tree for some small value of k, one should take a shortest-path tree that is rooted at a vertex with small eccentricity.

Theorem 2. *Let $G = (V, E)$ admit an eccentricity k-approximating spanning tree. For every $v \in \mathcal{C}^i(G),\ i \geq 0$, every shortest-path tree rooted at v is an eccentricity $2(k+i)+1$-approximating spanning tree.*

Proof. We fix an eccentricity k-approximating spanning tree T of G, that exists by the hypothesis. Furthermore, let T' be any shortest-path tree rooted at v. For every $j \geq 0$ and $u \in \mathcal{C}^j(G)$, we will prove that $ecc_{T'}(u) \leq rad(G) + j + 2(k+i) + 1$, thereby proving the theorem. By Lemma 5, we obtain the following chain of inequalities:

$$dist_G(u, v) \leq dist_G(v, \mathcal{C}(T)) + diam(G[\mathcal{C}(T)]) + dist_G(u, \mathcal{C}(T)) \leq (i+k) + 1 + (j+k) = i + j + 1 + 2k.$$

Since we assume T' to be a shortest-path tree rooted at v, we so deduce from Proposition 1 (applied to $S = \{v\}$) that we have $ecc_{T'}(u) \leq dist_G(u, v) + ecc_G(v) \leq (i + j + 1 + 2k) + (rad(G) + i) = (rad(G) + j) + 2(k+i) + 1 = ecc_G(u) + 2(k+i) + 1$. □

We stress the following algorithmic implication of Theorem 2:

Corollary 1. *Let $i \geq 0$ be an integer. If there is an algorithm that computes for any graph G a vertex of eccentricity $\leq rad(G) + i$ in $T(n, m)$-time, then we can also compute in $\mathcal{O}(T(n, m) + m)$-time an eccentricity $(2 \cdot ets(G) + 2i + 1)$-approximating spanning tree of G.*

It is known that if G is chordal (resp., δ-hyperbolic) then a central vertex (resp., a vertex of eccentricity $\leq rad(G) + 5\delta$) can be computed in linear time [5, 4]. – We refer to the corresponding papers for the formal definitions of these classes of graphs. – Therefore, we can conclude as follows:

Corollary 2. *For every chordal graph $G = (V, E)$ we can compute an eccentricity $(2 \cdot ets(G) + 1)$-approximating spanning tree in linear time. For every δ-hyperbolic graph $G = (V, E)$ we can compute an eccentricity $(2 \cdot ets(G) + 10\delta + 1)$-approximating spanning tree in linear time.*

Acknowledgements

We wish to thank the referees for their careful reading of the first version of this manuscript, and their useful comments. This work was supported by the Institutional research programme PN 1819 "Advanced IT resources to support digital transformation processes in the economy and society - RESINFO-TD" (2018), project PN 1819-01-01"Modeling, simulation, optimization of complex systems and decision support in new areas of IT&C research", funded by the Ministry of Research and Innovation, Romania. This work was also supported by a grant of Romanian Ministry of Research and Innovation CCCDI-UEFISCDI. project no. 17PCCDI/2018.

References

[1] J. A. Bondy and U. S. R. Murty. *Graph theory.* Grad. Texts in Math., 2008.

[2] A. Brandstädt, V. Chepoi, and F. Dragan. Distance approximating trees for chordal and dually chordal graphs. *Journal of Algorithms*, 30(1):166–184, 1999.

[3] F. Buckley and M. Lewinter. Graphs with all diametral paths through distant central nodes. *Mathematical and computer modelling*, 17(11):35–41, 1993.

[4] V. Chepoi and F. Dragan. Finding a central vertex in an HHD-free graph. *Discrete applied mathematics*, 131(1):93–111, 2003.

[5] V. Chepoi, F. Dragan, B. Estellon, M. Habib, and Y. Vaxès. Diameters, centers, and approximating trees of δ-hyperbolic geodesic spaces and graphs. In *Proceedings of the twenty-fourth annual symposium on Computational geometry (SoCG)*, pages 59–68. ACM, 2008.

[6] V. Chepoi, F. Dragan, M. Habib, Y. Vaxès, and H. Al-Rasheed. Fast approximation of centrality and distances in hyperbolic graphs. In *COCOA*, 2018.

[7] D. Corneil, F. Dragan, E. Köhler, and C. Yan. Collective tree 1-spanners for interval graphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 151–162. Springer, 2005.

[8] M. Demmer and M. Herlihy. The arrow distributed directory protocol. In *International Symposium on Distributed Computing*, pages 119–133. Springer, 1998.

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

[10] F. Dragan, E. Köhler, and H. Al-Rasheed. Eccentricity approximating trees. *Discrete Applied Mathematics*, 232:142–156, 2017.

[11] F. Dragan and C. Yan. Collective tree spanners in graphs with bounded genus, chordality, tree-width, or clique-width. In *International Symposium on Algorithms and Computation*, pages 583–592. Springer, 2005.

[12] F. Dragan, C. Yan, and D. Corneil. Collective tree spanners and routing in AT-free related graphs. *Journal of Graph Algorithms and Applications*, 10(2):97–122, 2006.

[13] F. Dragan, C. Yan, and I. Lomonosov. Collective tree spanners of graphs. *SIAM Journal on Discrete Mathematics*, 20(1):240–260, 2006.

[14] B. Eckhardt. *Complexity Analysis of Tries and Spanning Tree Problems*. PhD thesis, Technische Universität München, 2010.

[15] M. Gromov. Hyperbolic groups. In *Essays in group theory*, pages 75–263. Springer, 1987.

[16] C. Jordan. Sur les assemblages de lignes. *J. Reine Angew. Math*, 70(185):81, 1869.

[17] D. Kratsch, H. Le, H. Müller, E. Prisner, and D. Wagner. Additive tree spanners. *SIAM Journal on Discrete Mathematics*, 17(2):332–340, 2003.

[18] R. Nandakumar and K. Parthasarathy. Eccentricity-preserving spanning trees. *J. Math. Phys. Sci*, 24(1):33–36, 1990.

[19] E. Prisner. Eccentricity-approximating trees in chordal graphs. *Discrete Mathematics*, 220(1-3):263–269, 2000.