On the average Steiner 3-eccentricity of trees*

Xingfu Li[a], Guihai Yu[a], Sandi Klavžar[b,c,d]

[a] College of Big Data Statistics, Guizhou University of Finance and Economics
Guiyang, Guizhou, 550025, China
xingfulisdu@qq.com; yuguihai@mail.gufe.edu.cn
[b] Faculty of Mathematics and Physics, University of Ljubljana, Slovenia
[c] Institute of Mathematics, Physics and Mechanics, Ljubljana, Slovenia
[d] Faculty of Natural Sciences and Mathematics, University of Maribor, Slovenia
sandi.klavzar@fmf.uni-lj.si

April 21, 2020

Abstract

The *Steiner k-eccentricity* of a vertex v of a graph G is the maximum Steiner distance over all k-subsets of $V(G)$ which contain v. In this paper Steiner 3-eccentricity is studied on trees. Some general properties of the Steiner 3-eccentricity of trees are given. Based on them, an $O(|V(T)|^2)$ time algorithm to calculate the average Steiner 3-eccentricity on a tree T is presented. A tree transformation which does not increase the average Steiner 3-eccentricity is given. As its application, several lower and upper bounds for the average Steiner 3-eccentricity of trees are derived.

Keywords: Steiner distance, Steiner tree, Steiner eccentricity, average Steiner eccentricity, graph algorithms

AMS Math. Subj. Class. (2010): 05C12, 05C05, 05C85

1 Introduction

Throughout this paper, all graphs considered are simple and connected. If $G = (V(G), E(G))$ is a graph, then its order and size will be denoted by $n(G)$ and $m(G)$, respectively. If $S \subseteq V(G)$, then the *Steiner distance* $d_G(S)$ of S is the minimum size among all connected subgraphs of G containing S, that is,

$$d_G(S) = \min\{m(T) :\ T \text{ subtree of } G \text{ with } S \subseteq V(T)\}\,.$$

If $k \geq 2$ is an integer and $v \in V(G)$, then the *Steiner k-eccentricity* $\mathrm{ecc}_k(v, G)$ of v in G is the maximum Steiner distance over all k-subsets of $V(G)$ which contain v, that is,

$$\mathrm{ecc}_k(v, G) = \max\{d_G(S) :\ v \in S \subseteq V(G), |S| = k\}\,.$$

*Supported by the National Natural Science Foundation of China (No. 11861019), Guizhou Talent Development Project in Science and Technology (No. KY[2018]046), Natural Science Foundation of Guizhou (Nos.[2019]1047, [2018]5774-006, [2018]5774-021, [2020]1Z001), Foundation of Guizhou University of Finance and Economics(No. 2019XJC04). Sandi Klavžar acknowledges the financial support from the Slovenian Research Agency (research core funding No. P1-0297 and projects J1-9109, J1-1693, N1-0095, N1-0108. Corresponding author: Sandi Klavžar.

Note that $\mathrm{ecc}_2(v, G)$ is the standard eccentricity of the vertex v, that is, the largest distance between v and the other vertices of G.

Li, Mao, and Gutman [17] proposed the *k-th Steiner Wiener index* $SW_k(G)$ of G as

$$SW_k(G) = \sum_{S \in \binom{V(G)}{k}} d_G(S)\,.$$

Note that $SW_2(G) = W(G)$, the celebrated Wiener index of G. Motivated by the k-th Steiner Wiener index, we introduce the *average Steiner k-eccentricity* $\mathrm{aecc}_k(G)$ of G as the mean value of all vertices' Steiner k-eccentricities in G, that is,

$$\mathrm{aecc}_k(G) = \frac{1}{n(G)} \sum_{v \in V(G)} \mathrm{ecc}_k(v, G)\,.$$

In this notation, $\mathrm{aecc}_2(G)$ is just the standard average eccentricity of G, cf. [3, 7, 8, 9, 24]).

The Steiner tree problem on general graphs is NP-hard to solve [10, 16], but it can be solved in polynomial time on trees [2]. The Steiner distance has been extensively studied on special graph classes such as trees, joins, standard graph products, corona products, and others, see [1, 4, 12, 22, 26]. The average Steiner k-distance and its close companion the k-th Steiner Wiener index have been studied on trees, complete graphs, paths, cycles, complete bipartite graphs, and others, see [6, 13]. The average Steiner distance and the Steiner Wiener index were also extensively studied, see [5, 17, 18, 19, 27, 28]. Some work on the Steiner diameter is present in [22, 26]. Other topological indices related to the Steiner distance have also been investigated: Steiner Gutman index in [23], Steiner degree distance in [14], Steiner hyper-Wiener index in [25], multi-center Wiener index in [15], Steiner Harary index in [21], and Steiner (revised) Szeged index in [11]. Y. Mao wrote an extensive survey paper on the Steiner distance in graphs [20].

In this paper we focus on the average Steiner 3-eccentricity of trees. In the rest of this section we list additional definitions needed in this paper. Then, in Section 2, we present several structural properties of the Steiner k-eccentricity of trees. Based on these results, in Section 3 we devise a quadratic-time algorithm to calculate the average Steiner 3-eccentricity of a tree and compare it with a much slower brute force method. In Section 4, the average Steiner 3-eccentricity of trees is investigated under a special transformation. Relying on this behavior, in the subsequent section we establish several lower and upper bonds on the average Steiner 3-eccentricity of trees. We conclude by presenting several topics for future research.

A vertex of a graph of degree 1 is a *leaf* or a *pendent vertex*, and it is of degree at least 2, then it is an *internal vertex*. With $\ell(G)$ we denote the number of leaves of a graph G. A vertex of a tree of degree at least 3 is a *branching vertex*. An edge is *pendent* if it is incident to a pendent vertex in a graph. A path P of a graph G is a *pendent path* if one endpoint of P has degree 1 and each internal vertex of P has degree 2.

If H_1 and H_2 are subgraphs of G, then the distance $d_G(H_1, H_2)$ between H_1 and H_2 is defined as $\min\{d_G(h_1, h_2) :\ h_1 \in V(H_1), h_2 \in V(H_2)\}$. In particular, if H_1 is the one vertex graph with u being its unique vertex, then we will write $d_G(u, H_2)$ for $d_G(H_1, H_2)$. The *eccentricity of a subgraph* H in G is $\mathrm{ecc}_G(H) = \max\{d_G(v, H) :\ v \in V(G)\}$.

If $S \subseteq V(G)$ and T is subtree of G with $S \subseteq V(T)$ and $m(T) = d_G(S)$, then we say that T is an *S-Steiner tree* and that a vertex of S is a *terminal* of T. If $k \geq 2$ and $v \in V(G)$, then a k-set $S \subseteq V(G)$ is a *Steiner k-ecc v-set* (or *k-ecc v-set* for short) if $v \in S$ and $d_G(S) = \mathrm{ecc}_k(v, G)$; a corresponding tree that realizes $\mathrm{ecc}_k(v, G)$ will be called a *Steiner k-ecc v-tree* (or *k-ecc v-tree* for short). A vertex v may have more than one k-ecc v-set, and each such set may have more than one Steiner k-ecc v-tree.

2 Preliminary results

The main topic of this paper is the average Steiner 3-eccentricity (of trees). We first give exact values of it for some classes of graphs, easy computations being omitted.

Proposition 2.1 *If $n \geq 3$, then* $\mathrm{aecc}_3(K_n) = 2$, $\mathrm{aecc}_3(P_n) = n-1$, $\mathrm{aecc}_3(K_{1,n-1}) = 3 - \frac{1}{n}$, *and* $\mathrm{aecc}_3(C_n) = n-1$. *Moreover, if $m, n \geq 3$, then* $\mathrm{aecc}_3(K_{m,n}) = 3$.

We now proceed with a series of lemmas.

Lemma 2.2 *If T is a tree and $S \subseteq V(T)$, then the S-Steiner tree is unique.*

Lemma 2.2 is implicitly used in the literature and also briefly mentioned in [20, p. 11]. It follows from the argument that two different S-Steiner trees would lead to a cycle in T. By Lemma 2.2, the formulation of the next lemma is justified.

Lemma 2.3 *Let T be a tree, $v \in V(T)$, and $v \in S \subseteq V(T)$, $|S| = k$. Let T_v be the unique S-Steiner tree and P a path in T with $V(P) \cap V(T_v) = \{x\}$. If*

(1) $x \in S$ and $x \neq v$, or

(2) $x \notin S$ and T_v has an internal vertex which is in S and is different from v,

then there exists a k-set $S' \neq S$ with $v \in S'$, such that the size of the S'-Steiner tree is strictly larger than the size of T_v.

Proof. Suppose first that $x \in S$ and $x \neq v$. Let u be the pendent vertex of P and set $S' = (S \cup \{u\}) - x$. Then the size of the S'-Steiner tree is $|E(T_v) \cup E(P)|$. Since $|V(P)| \geq 2$, we have $|E(T_v) \cup E(P)| \geq |E(T_v)| + 1 > |E(T_v)|$.

In the second case, let t be the internal vertex of T_v which is in S and different from v. Let again u be the pendent vertex of P. In this case we set $S' = (S \cup \{u\}) - t$ and obtain another k-set which induces a larger size Steiner tree than the original k-set S. ■

Recall that $\ell(T)$ denotes the number of leaves of a tree T.

Lemma 2.4 *Let T be a tree and $v \in V(T)$. If $k > \ell(T)$, then every k-ecc v-set contains all the leaves of T. The same conclusion holds if v is a leaf and $k = \ell(T)$.*

Proof. Trivially, $\mathrm{ecc}_k(v, T) \leq n(T) - 1$. Suppose that $k > \ell(T)$. Set $S = \{v\} \cup L \cup X$, where L is the set of leaves of T and X a set of arbitrary $k - \ell(T) - 1$ vertices from $V(T) \setminus (L \cup \{v\})$. Then $|S| = k$ and the S-Steiner tree it the whole tree T. Hence every k-ecc v-set is the whole tree T and thus contains all the leaves. If v is a leaf, then set $S = L \cup X$, where X a set of arbitrary $k - \ell(T)$ vertices from $V(T) \setminus L$ to reach the same conclusion. ■

Lemma 2.5 *Let T be a tree, $v \in V(T)$, and $\ell(T) \geq k \geq 2$. if S is a k-ecc v-set, then every vertex from $S \setminus \{v\}$ is a leaf of T.*

Proof. If $k = \ell(T)$ and v is a leaf of T, then the conclusion follows by Lemma 2.4. In the rest we may hence assume that $k < \ell(T)$ or v is not a leaf of T.

Let T_v be a k-ecc v-tree and suppose that there exists a vertex $u \in S \setminus v$ which is an internal vertex of T. There there exists a leaf x in T which does not lie in T_v. Let P be the unique x, T_v-path in T. Then P is a pendent path with at least one edge not in T_v and hence we can use Lemma 2.3 to obtain a larger S-Steiner tree, a contradiction. ■

Lemma 2.6 *Let T be a tree and $v \in V(T)$. Then every Steiner k-ecc v-tree contains a longest path starting at v.*

Proof. If $k = 2$, then $\mathrm{ecc}_2(v, T)$ is the length of a longest path from v to all the other vertices in T, so there is nothing to be proved. In the sequel we may thus assume $k \geq 3$. Suppose on the contrary that T_v is a k-ecc v-tree which contains no longest path starting at v in T. Let S be the k-ecc v-set corresponding to T_v. Let P be a longest path starting at v in the tree T, and let $v^{''}$ be the endpoint of P different from v. Let P_1 be the sub-path of P which is shared by T_v, and P_2 be the remaining sub-path of P. Then P_1 and P_2 share a unique vertex $v^{'} \in V(P)$. The described situation is illustrated in Fig. 1.

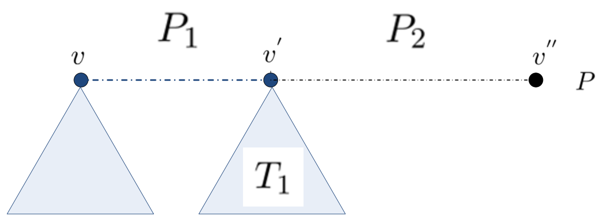

Figure 1: The situation from the proof of Lemma 2.6; the grey part is T_v

Note that v is an endpoint of P_1 and $v^{''}$ is an endpoint of P_2. By the assumption, P_2 is not empty. Let F be a forest obtained by deleting all the edges in $E(P_1) \subseteq E(T_v)$ from the tree T_v. Let T_1 be the tree in F which contains the vertex $v^{'}$, cf. Fig. 1 again. We now distinguish two cases.

Suppose first that $n(T_1) = 1$. Then $v^{'}$ is a leaf of T_v. So $v^{'}$ must be in the set S. We claim that $v^{'} \neq v$. Otherwise, the tree T_v would be a trivial tree and S contains the unique vertex v, which contradicts to the fact that $k \geq 3$. Let $S^{'} = (S \setminus \{v^{'}\}) \cup \{v^{''}\}$. Then $S^{'}$ is another k-set containing v and its $S^{'}$-Steiner tree is $T_v \cup P_2$. Since $S^{'}$ is a larger tree than T_v, we have a contradiction to the fact that T_v is a k-ecc v-tree.

Suppose second that $n(T_1) \geq 2$. Then there must be a vertex $u \in V(T_1)$ such that u is a leaf of T_v. Then u lies in the k-set S. We construct a path P_3 as follows.

- If there is no branching vertex in T_v, then set P_3 to be the path from $v^{'}$ to u in T_v.
- Suppose that there is at least one branching vertex in T_v. Let $w \in V(T_v)$ be the branching vertex nearest to u in T_v. If w is on the path from $v^{'}$ to u, then let P_3 be the path from w to u in the tree T_v. Otherwise, let P_3 be the path from $v^{'}$ to u in the tree T_v.

Let $S^{'} = (S \setminus \{u\}) \cup \{v^{''}\}$. Then the tree $T_v^{'} = (T_v \setminus P_3) \cup P_2$ is the $S^{'}$-Steiner tree. Since P is a longest starting from v and T_v contains no such longest path from v, the length of P_2 is strictly larger than the length of P_3. So $m(T_v^{'}) > m(T_v)$, a final contradiction. ■

In the rest of the section we focus on the structure of 3-ecc v-trees. By Lemma 2.6, the endpoint x of some longest path starting at v must be in some 3-ecc v-set. Here is now a property of the third terminal in a 3-ecc v-set.

Lemma 2.7 *Let v be a vertex of a tree T and let $S = \{v, x, y\}$ be a 3-ecc v-set, where the v, x-path P is a longest path in T starting from v. Then $d_T(y, P) = \mathrm{ecc}_T(P)$.*

Proof. Let T_v be the 3-ecc v-tree; so T_v contains P and the set $S = \{x, y, z\}$. The path P is thus fixed and hence the vertex y must be such that the y, P-path in T is as long as possible. But this in turn implies that for the third terminal y we must have $d_T(y, P) = \max\{d_T(s, P) : s \in V(T)\} = \mathrm{ecc}_T(P)$. ■

Combining with Lemma 2.7, the next lemma asserts that in the case of 3-ecc v-sets, in Lemma 2.6 an arbitrary longest path starting from v can be used.

Lemma 2.8 *Let v be a vertex of a tree T, and let P_1 and P_2 be distinct longest paths having v as an endpoint. Then* $\mathrm{ecc}_T(P_1) = \mathrm{ecc}_T(P_2)$.

Proof. Let w be the last common vertex of P_1 and P_2. Clearly w exists, it is possible that $w = v$. Let t_1 and t_2 be the other endpoints of P_1 and P_2, respectively. As T is a tree and $P_1 \neq P_2$ we have $t_1 \neq t_2$. Let u and s be vertices of T such that $d_T(u, P_1) = \mathrm{ecc}_T(P_1)$ and $d_T(s, P_2) = \mathrm{ecc}_T(P_2)$. Let further P_u be the shortest u, P_1-path, P_s the shortest s, P_2-path, u_0 the endpoint of P_u different from u, and s_0 the endpoint of P_s different from s.

We claim that u_0 lies in the v, w-subpath of P_1 (or P_2 for that matter). Suppose on the contrary that u_0 is an internal vertex of the w, t_1-subpath of P_1. Since $d_T(u, P_1) = \mathrm{ecc}_T(P_1)$ it follows that the length of P_u is at least the length of the w, t_2-subpath of P_2. Since the latter path is of the same length as the w, t_1-subpath of P_1, we get that the concatenation of P_u with the v, u_0-subpath of P_1 is a path strictly longer than P_1, a contradiction.

We have thus proved that u_0 lies in the v, w-subpath of P_1. By a parallel argument we also get that s_0 lies in the v, w-subpath of P_2 (or in the v, w-subpath of P_1 for that matter). But his means that $d_G(u, P_1) = d_G(s, P_2)$ and hence $\mathrm{ecc}_T(P_1) = \mathrm{ecc}_T(P_2)$. ■

3 Computing the average Steiner 3-eccentricity on trees

In this section, we design two polynomial algorithms to calculate the average Steiner 3-eccentricity on a tree. The first is a simple enumeration method, the other one is based on the results developed in Section 2.

If T is a tree and v its vertex, then the first algorithm computes $\mathrm{ecc}_3(v, T)$ by determining the S-Steiner tree for each 3-set S containing v, detecting in this way one of the largest size. This brute force strategy is written down in Algorithm 1.

Algorithm 1: Brute-Force-aecc(T)

Input: Tree T
Output: $\mathrm{aecc}_3(T)$

```
ecc=0;
for each vertex v in V(T) do
    max=0;
    for every two vertices u,w ∈ V(T) \ {v} do
        find the {v,u,w}-Steiner tree ST;
        if m(ST) > max then
            max = m(ST);
        end
    end
    ecc=ecc+max
end
return ecc/n(T);
```

Theorem 3.1 *If T is a tree of order $n = n(T)$, then Algorithm 1 can be implemented to run in $O(n^4)$ time.*

Proof. Step 5, which determines the $\{v, u, w\}$-Steiner tree, can be implemented in $O(n)$ time. For each vertex v, this computation has to be done for $\binom{n-1}{2} = O(n^2)$ pairs of vertices u and w, so for each $v \in V(T)$ we need $O(n^3)$ time. Hence the complete algorithm can be implemented to run in $O(n^4)$ time. ■

We next show how the results of the previous section can be used to design a faster algorithm for the average Steiner 3-eccentricity on trees. Let v be a vertex of a tree T. Then the main idea is to first apply Lemmas 2.6 and 2.8 to find a second vertex in a Steiner 3-ecc v-tree, and then to apply Lemma 2.7 to find a third vertex of a 3-ecc v-tree. The idea is implemented in Algorithm 2, where the Steiner 3-eccentricity for each vertex is computed in Steps 3–7. A longest path P starting in v is computed in Step 4 by calling the procedure Longest_Path, while a a longest shortest path from all vertices of T to P in computed in Steps 6-7 with the help of the procedure Path_Shrinking.

Algorithm 2: aecc(T)

```
   Input: Tree T
   Output: aecc_3(T)
1  ecc=0;
2  for each v ∈ V(T) do
3  |   path[1 : n] = ∅; //Each entry of path is initialized as ∅
4  |   ecc=ecc+Longest_Path(v, T, path);
5  |   T' = (V(T) ∪ {u}, E(T));
6  |   T' =Path_Shrinking(v, T', path);
7  |   ecc=ecc+Longest_Path(v, T', path);
8  end
9  return ecc/n(T);
```

The procedure Longest_Path can be implemented by modifying the classical depth-first search method (DFS) as formally written down in Algorithm 3. The parameter *path* in the algorithm is a linear array of $n(T)$ entries and used to store a longest path starting at v.

Algorithm 3: Longest_Path(v, T, path)

```
Input: A vertex v, a tree T rooted at v and an array named path
Output: the length of a longest path starting at v
max=0; temp=max
for each vertex u ∈ N_T(v) which has not been visited till now do
|   temp=Longest_Path(u, T, path);
|   if temp>max then
|   |   path[v]=u;
|   |   max=temp;
|   end
end
return max+1;
```

To implement the procedure Path_Shrinking in which a longest shortest path from all vertices of T to P must be computed, we reduce it to the problem of finding a longest path starting at a given vertex of a tree. The idea of the reduction is to shrink the longest path found in the initial step into a single vertex. The implementation is presented in Algorithm 4.

Algorithm 4: Path_Shrinking(v, u, T, path)

Input: A vertex v, a new vertex $u \notin V(T)$, a tree T rooted at v and an array named path
Output: A new tree after shrinking the longest path

```
w=v;
while path[w] ≠ ∅ do
    for each vertex x ∈ N_T(w) do
        remove the edge (w, x) from T;
        add a new edge between x and u in T;
    end
    w=path[w];
end
```

Theorem 3.2 *Let T be a tree of order $n = n(T)$. Then Algorithm 2 correctly computes* $\mathrm{aecc}_3(T)$ *and can be implemented to run in $O(n^2)$ time.*

Proof. The correctness of the algorithm follows by Lemmas 2.6, 2.7, and 2.8.

Using the adjacency list presentation of T, Algorithm 3 (which finds a longest path starting at a given vertex) can be implemented in $O(n)$ time. The same time can also be achieved in an implementation of Algorithm 4. Since in Algorithm 2 there is only one loop over all vertices of T, the whole algorithm can be implemented to run in $O(n^2)$ time. ■

4 A transformation on trees

Let T be a tree with the structure as schematically depicted in Fig. 2. Here the w, v_0-path P is a pendant path for which we require that $0 \le m(P) < \mathrm{ecc}_2(u, T_0)$ holds. (In case $m(P) = 0$, we have $v_0 = w$.) Then set $T' = T \setminus \{wx : x \in N_{T_1}(w)\} \cup \{uy : y \in N_{T_1}(w)\}$, see Fig. 2 again. We say that T' is obtained from T by a *π-transformation* and write $T' = \pi(T)$. The reverse transformation will be called a *π^{-1}-transformation*, that is, given T' as in Fig. 2, we set $T = T' \setminus \{ux : x \in N_{T_1}(u)\} \cup \{wy : y \in N_{T_1}(u)\}$ and write $T = \pi^{-1}(T')$.

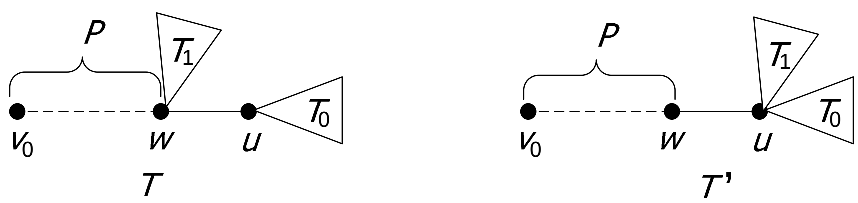

Figure 2: T and T'

Theorem 4.1 *Let T be a tree as in Fig. 2, and let $T' = \pi(T)$. Let P_0 be a longest path starting at u in T_0. If* $\mathrm{ecc}_{T_0}(P_0) \le \mathrm{ecc}_2(w, P) < \mathrm{ecc}_2(u, T_0)$ *and* $\mathrm{ecc}_2(w, T_1) \le \mathrm{ecc}_2(w, P)$*, then* $\mathrm{aecc}_3(T') = \mathrm{aecc}_3(T)$*. Otherwise,* $\mathrm{aecc}_3(T') < \mathrm{aecc}_3(T)$*.*

Proof. We are going to consider the behavior of the Steiner 3-eccentricity on the sets of vertices $V(P) \setminus \{w\}$, $V(T_1)$, and $V(T_0)$ on the following cases that cover all the possibilities. (Recall that in the definition of the π-transformation we have required that $m(P) = \mathrm{ecc}_2(w, P) < \mathrm{ecc}_2(u, T_0)$ holds.)

Case 1: $0 \leq \mathrm{ecc}_{T_0}(P_0) \leq \mathrm{ecc}_2(w, T_1) \leq \mathrm{ecc}_2(w, P) < \mathrm{ecc}_2(u, T_0)$. In this case it is evident that the following three statements hold.

(i) $\mathrm{ecc}_3(v, T) = \mathrm{ecc}_3(v, T^{'})$ for every vertex $v \in V(P) \setminus \{w\}$.

(ii) $\mathrm{ecc}_3(v, T) = \mathrm{ecc}_3(v, T^{'})$ for every vertex $v \in V(T_1)$.

(iii) $\mathrm{ecc}_3(v, T) = \mathrm{ecc}_3(v, T^{'})$ for every vertex $v \in V(T_0)$.

By the definition of the average Steiner 3-eccentricity it thus follows that $\mathrm{aecc}_3(T^{'}) = \mathrm{aecc}_3(T)$ holds in this case.

Case 2: $0 \leq \mathrm{ecc}_2(w, T_1) < \mathrm{ecc}_{T_0}(P_0) \leq \mathrm{ecc}_2(w, P) < \mathrm{ecc}_2(u, T_0)$. In this case we obtain the same conclusions as the Case 1. Hence we conclude that $\mathrm{aecc}_3(T^{'}) = \mathrm{aecc}_3(T)$ holds also in this case.

Case 3: $0 \leq \mathrm{ecc}_2(w, T_1) \leq \mathrm{ecc}_2(w, P) < \mathrm{ecc}_{T_0}(P_0) \leq \mathrm{ecc}_2(u, T_0)$. Now:

(i) $\mathrm{ecc}_3(v, T) = \mathrm{ecc}_3(v, T^{'})$ for every vertex $v \in V(P) \setminus \{w\}$.

(ii) $\mathrm{ecc}_3(v, T) - \mathrm{ecc}_3(v, T^{'}) = 1$ for every vertex $v \in V(T_1)$.

(iii) $\mathrm{ecc}_3(v, T) = \mathrm{ecc}_3(v, T^{'})$ for every vertex $v \in V(T_0)$.

Therefore, $\mathrm{aecc}_3(T^{'}) < \mathrm{aecc}_3(T)$.

Case 4: $0 \leq \mathrm{ecc}_{T_0}(P_0) \leq \mathrm{ecc}_2(w, P) < \mathrm{ecc}_2(w, T_1) \leq \mathrm{ecc}_2(u, T_0)$. Now:

(i) $\mathrm{ecc}_3(v, T) = \mathrm{ecc}_3(v, T^{'})$ for every vertex $v \in V(P) \setminus \{w\}$.

(ii) $\mathrm{ecc}_3(v, T) - \mathrm{ecc}_3(v, T^{'}) \geq 0$ for every vertex $v \in V(T_1)$.

(iii) $\mathrm{ecc}_3(v, T) - \mathrm{ecc}_3(v, T^{'}) \geq 0$ for every vertex $v \in V(T_0) \setminus \{u\}$.

(iv) $\mathrm{ecc}_3(u, T) - \mathrm{ecc}_3(v, T^{'}) = 1$.

We conclude that $\mathrm{aecc}_3(T^{'}) < \mathrm{aecc}_3(T)$ in this case.

Case 5: $0 \leq \mathrm{ecc}_2(w, P) < \mathrm{ecc}_{T_0}(P_0) \leq \mathrm{ecc}_2(w, T_1) \leq \mathrm{ecc}_2(u, T_0)$. Now:

(i) $\mathrm{ecc}_3(v, T) = \mathrm{ecc}_3(v, T^{'})$ for every vertex $v \in V(P) \setminus \{w\}$.

(ii) $\mathrm{ecc}_3(v, T) - \mathrm{ecc}_3(v, T^{'}) = 1$ for every vertex $v \in V(T_1)$.

(iii) $\mathrm{ecc}_3(v, T) - \mathrm{ecc}_3(v, T^{'}) \geq 0$ for every vertex $v \in V(T_0) \setminus \{u\}$.

(iv) $\mathrm{ecc}_3(u, T) - \mathrm{ecc}_3(v, T^{'}) = 1$ for the vertex u.

So also in this case $\mathrm{aecc}_3(T^{'}) < \mathrm{aecc}_3(T)$.

Case 6: $0 \leq \mathrm{ecc}_2(w, P) < \mathrm{ecc}_2(w, T_1) < \mathrm{ecc}_{T_0}(P_0) \leq \mathrm{ecc}_2(u, T_0)$. Now:

(i) $\mathrm{ecc}_3(v, T) = \mathrm{ecc}_3(v, T^{'})$ for every vertex $v \in V(P) \setminus \{w\}$.

(ii) $\mathrm{ecc}_3(v, T) - \mathrm{ecc}_3(v, T^{'}) = 1$ for every vertex $v \in V(T_1)$.

(iii) $\mathrm{ecc}_3(v, T) - \mathrm{ecc}_3(v, T^{'}) \geq 0$ for every vertex $v \in V(T_0)$.

Again $\mathrm{aecc}_3(T^{'}) < \mathrm{aecc}_3(T)$.

To be able to deal with the remaining possibilities, let P_1 be a longest path starting at w in T_1. The remaining cases to be considered are then as follows.

Case 7: $0 \leq \mathrm{ecc}_{T_1}(P_1) \leq \mathrm{ecc}_2(w, P) < \mathrm{ecc}_2(u, T_0) < \mathrm{ecc}_2(w, T_1)$. Now:

(i) $\mathrm{ecc}_3(v,T) = \mathrm{ecc}_3(v,T^{'})$ for every vertex $v \in V(P) \setminus \{w\}$.

(ii) $\mathrm{ecc}_3(v,T) - \mathrm{ecc}_3(v,T^{'}) \geq 0$ for every vertex $v \in V(T_1)$.

(iii) $\mathrm{ecc}_3(v,T) - \mathrm{ecc}_3(v,T^{'}) \geq 0$ for every vertex $v \in V(T_0) \setminus \{u\}$.

(iv) $\mathrm{ecc}_3(u,T) - \mathrm{ecc}_3(v,T^{'}) = 1$.

Once more $\mathrm{aecc}_3(T^{'}) < \mathrm{aecc}_3(T)$.

Case 8: $0 \leq \mathrm{ecc}_2(w,P) < \mathrm{ecc}_{T_1}(P_1) \leq \mathrm{ecc}_2(u,T_0) < \mathrm{ecc}_2(w,T_1)$. Now:

(i) $\mathrm{ecc}_3(v,T) = \mathrm{ecc}_3(v,T^{'})$ for every vertex $v \in V(P) \setminus \{w\}$.

(ii) $\mathrm{ecc}_3(v,T) - \mathrm{ecc}_3(v,T^{'}) \geq 0$ for every vertex $v \in V(T_1)$.

(iii) $\mathrm{ecc}_3(v,T) - \mathrm{ecc}_3(v,T^{'}) \geq 0$ for every vertex $v \in V(T_0) \setminus \{u\}$.

(iv) $\mathrm{ecc}_3(u,T) - \mathrm{ecc}_3(v,T^{'}) = 1$ for the vertex u.

Yet again $\mathrm{aecc}_3(T^{'}) < \mathrm{aecc}_3(T)$.

Case 9: $0 \leq \mathrm{ecc}_2(w,P) < \mathrm{ecc}_2(u,T_0) < \mathrm{ecc}_{T_1}(P_1) \leq \mathrm{ecc}_2(w,T_1)$. In this case, the following three statements hold.

(i) $\mathrm{ecc}_3(v,T) - \mathrm{ecc}_3(v,T^{'}) = -1$ for every vertex $v \in V(P) \setminus \{w\}$.

By Lemmas 2.6 and 2.7, the other two terminals must be in T_1. This remains true after the π-transformation is performed. So for every $v \in V(P) \setminus \{w\}$, $\mathrm{ecc}_3(v,T)$ increases by 1 after the transformation.

(ii) $\mathrm{ecc}_3(v,T) - \mathrm{ecc}_3(v,T^{'}) \geq 0$ for every vertex $v \in V(T_1)$.

(iii) $\mathrm{ecc}_3(v,T) - \mathrm{ecc}_3(v,T^{'}) = 1$ for every vertex $v \in V(T_0)$.

For every vertex $v \in V(T_0)$, by Lemma 2.6, the other endpoint of a longest path starting at v must be in T_1. This holds true after the π-transformation is performed. By Lemma 2.7, the third terminal could not be v_0. So after the π-transformation, $\mathrm{ecc}_3(v,T)$ decreases by 1.

Since we have assumed that $\mathrm{ecc}_2(w,P) < \mathrm{ecc}_2(u,T_0)$, we have $|V(P) \setminus \{w\}| < n(T_0)$. In summary, in Case 9, we have

$$\begin{aligned}
\mathrm{aecc}_3(T) - \mathrm{aecc}_3(T^{'}) &= \frac{1}{n}\Big\{ \sum_{v\in V(P)\setminus\{w\}} [\mathrm{ecc}_3(v,T) - \mathrm{ecc}_3(v,T^{'})] + \\
&\quad \sum_{v\in V(T_1)} [\mathrm{ecc}_3(v,T) - \mathrm{ecc}_3(v,T^{'})] + \sum_{v\in V(T_0)} [\mathrm{ecc}_3(v,T) - \mathrm{ecc}_3(v,T^{'})] \Big\} \\
&\geq \frac{1}{n}\Big[|V(T_0)| - |V(P) \setminus \{w\}| \Big] \\
&> 0\,.
\end{aligned}$$

We conclude that $\mathrm{aecc}_3(T^{'}) < \mathrm{aecc}_3(T)$ holds also in Case 9. ■

Corollary 4.2 *Let T' be a tree as in Fig. 2, and let $T = \pi^{-1}(T')$. Let P_0 be a longest path starting at u in T_0. If $\mathrm{ecc}_{T_0}(P_0) \leq \mathrm{ecc}_2(w,P) < \mathrm{ecc}_2(u,T_0)$ and $\mathrm{ecc}_2(w,T_1) \leq \mathrm{ecc}_2(w,P)$, then $\mathrm{aecc}_3(T^{'}) = \mathrm{aecc}_3(T)$. Otherwise, $\mathrm{aecc}_3(T) > \mathrm{aecc}_3(T')$.*

5 Some applications of the π-transformation

As an application of the π-transformation, we establish in this section several lower and upper bounds on the average Steiner 3-eccentricity of trees in terms of the order, the maximum degree, the number of pendent vertices, the matching number, the independent number, the diameter, and the radius.

5.1 Lower and upper bounds on general trees

Theorem 5.1 *If T is a tree on n vertices, then*

$$3 - \frac{1}{n} \leq \mathrm{aecc}_3(T) \leq n - 1\,.$$

The left equality holds if and only if $T \cong K_{1,n-1}$ and the right equality holds if and only if $T \cong P_n$.

Proof. Let T be an arbitrary tree of order n. Then repeatedly apply the π-transformation on T until no further π-transformation is possible. In the last step we must necessarily arrive at $K_{1,n-1}$. See Fig. 3 for an example of such a procedure.

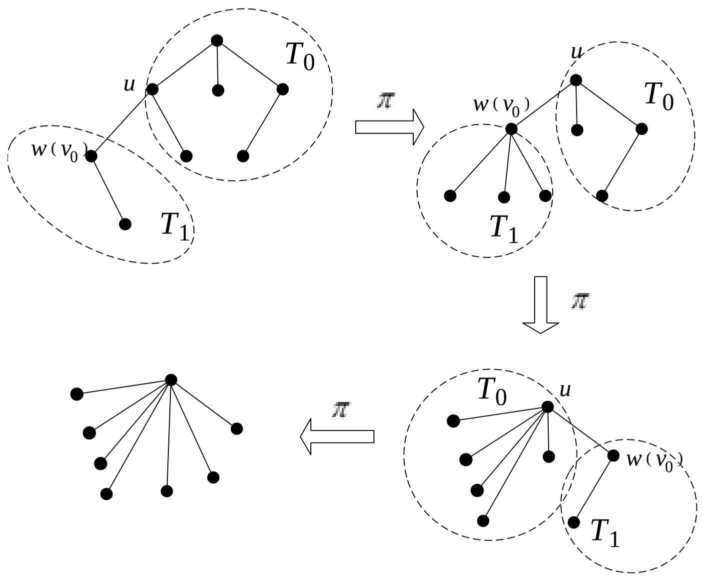

Figure 3: Transforming a tree with a sequences of π-transformations to a star

By Theorem 4.1, during the process, each π-transformation does not increase the average Steiner 3-eccentricity. Hence $\mathrm{aecc}(S_{1,n-1}) \leq \mathrm{aecc}_3(T)$. On the other hand, repeatedly applying the π^{-1}-transformation on T as long as it is possible, we must necessarily arrive at the path P_n, see Fig. 4 for an example.

By Corollary 4.2, at each step of this process the average Steiner 3-eccentricity does not decrease, hence $\mathrm{aecc}_3(P) \geq \mathrm{aecc}_3(T)$. Using the values from Proposition 2.1 we thus have $3 - \frac{1}{n} = \mathrm{aecc}(S_{1,n-1}) \leq \mathrm{aecc}_3(T) \leq \mathrm{aecc}_3(P_n) = n - 1$ and we are done. ■

5.2 An upper bound on trees with maximum degree

A *broom* $B(n, \Delta)$ is a tree obtained from $K_{1,\Delta}$ by attaching a path of length $n-\Delta-1$ to an arbitrary pendent vertex of the star. See Fig. 5 for an example.

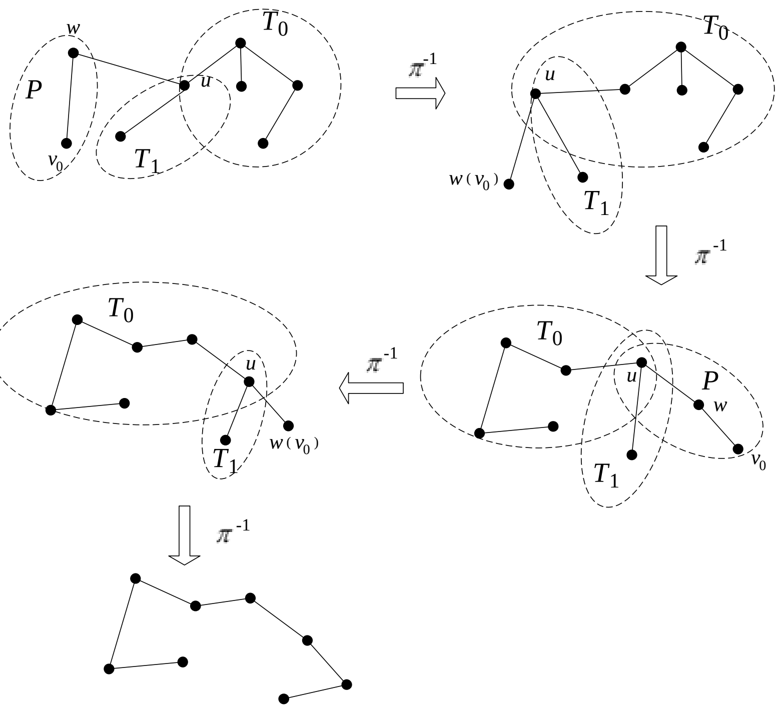

Figure 4: Transforming a tree with a sequences of π^{-1}-transformations to a path

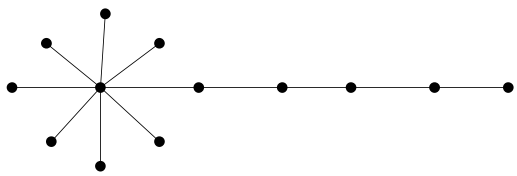

Figure 5: The broom $B(13, 8)$

Theorem 5.2 *If T is a tree of order $n = n(T)$ and maximum degree $\Delta = \Delta(T)$, then*

$$\mathrm{aecc}_3(T) \le \mathrm{aecc}_3(B(n, \Delta)) = n - \Delta + 1 + \frac{\Delta}{n}.$$

Proof. Let T be a tree with $n = n(T)$ and $\Delta = \Delta(T)$, and let r be the vertex of T with degree Δ. Consider T as a tree rooted in r. Let $T_1, \ldots, T_\Delta$ be the maximal subtrees of T that contain r and exactly one of the neighbor of r, respectively. We may also consider these Δ trees to be rooted at r. Repeatedly apply the π^{-1}-transformation on each subtree T_i, until T_i becomes a path. When all subtrees turn into paths, we can further proceed the π^{-1}-transformation until we arrive at the broom $B(n, \Delta)$, see Fig. 6 for an example.

By Corollary 4.2, during the whole process the Steiner 3-eccentricity does not decrease. This implies that $\mathrm{aecc}_3(T) \le \mathrm{aecc}_3(B(n, \Delta))$. Finally, the broom $B = B(n, \Delta)$ has Δ leaves, and $\mathrm{ecc}_3(v, B) = n - \Delta + 2$ holds for each of its leaves v. For each of the other $n - \Delta$ vertices w of B we have $\mathrm{ecc}_3(w, B) = n - \Delta + 1$. Hence $\mathrm{aecc}_3(B) = (\Delta(n - \Delta + 2) + (n - \Delta)(n - \Delta + 1))/n = n + 1 - \Delta + \frac{\Delta}{n}$, and we are done. ■

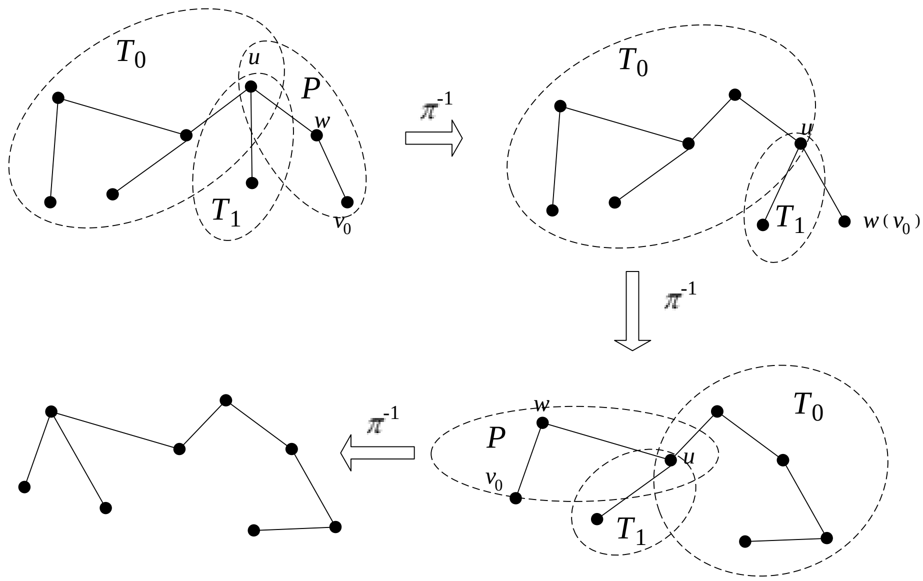

Figure 6: Transforming a tree with a sequences of π^{-1}-transformations to a broom

5.3 A lower bound on trees with constant number of leaves

A *starlike tree* is a tree with exactly one vertex of degree at least three. In other words, a starlike tree is a tree obtained by attaching to an isolated vertex $t \geq 3$ pendant paths. If the lengths of these pendant paths pairwise differ by at most one, then the starlike tree is called *balanced.* Note that if T is a balanced starlike tree of order n and with p leaves, then it is uniquely determined (up to isomorphism); we will denote it by $BS_{n,p}$.

Theorem 5.3 *Let T be a tree of order $n \geq 2$ and with p pendent vertices. Then*

$$\mathrm{aecc}_3(T) \geq \mathrm{aecc}_3(BS_{n,p}) \, .$$

Proof. Let T be a tree as stated. If T has exactly one vertex of degree at least three, then successively applying the π-transformation we obtain the balanced starlike tree $BS_{n,p}$. Suppose next that T has at least two vertices with degree greater than two. Repetitively balancing the pendent paths by the π-transformation method, the average Steiner 3-eccentricity does not increase. The balancing procedure may stuck in a state, where there are exactly two branching vertices and p pendent paths, which are all hanging under one of these two branching vertices and having the same length in T. We can reattach $p-2$ pendent paths to the same branching vertex without changing the average Steiner 3-eccentricity. In this way we arrive at the starlike tree $BS_{n,p}$, see Fig. 7 for an example.

Since in all the transformations made to reach $BS_{n,p}$ the average Steiner 3-eccentricity has not increased, we conclude that $\mathrm{aecc}_3(BS_{n,p}) \leq \mathrm{aecc}_3(T)$. ■

5.4 Lower bounds on trees with matching and independence number

If $m \geq 3$ and $m+2 \leq n \leq 2m+1$, then let $T_{n,m}$ be a tree obtained from $K_{1,m}$ by respectively adding a pendent edge to its $n-m-1$ pendent vertices. Note that $n(T_{n,m}) = n$. Observe further that $\alpha(T_{n,m}) = m$ and $\beta(T_{n,m}) = n-m$, where $\alpha(G)$ and $\beta(G)$ are the matching number and the independence number of G, respectively.

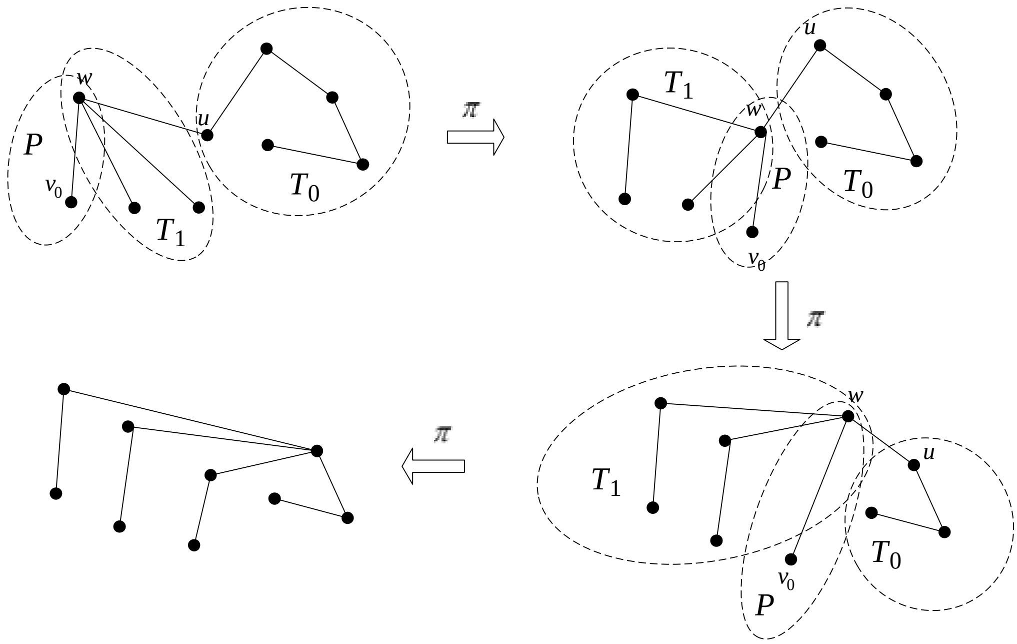

Figure 7: Transforming a tree with a sequences of π-transformations to a balanced starlike tree

Theorem 5.4 *If T is a tree with $n = n(T)$ and $\beta = \beta(T) \geq 2$, then*

$$\mathrm{aecc}_3(T) \geq \mathrm{aecc}_3(T_{n,n-\beta}) .$$

Proof. Let M be a maximum matching of T, so that $|M| = \beta$. Set further $\ell = \ell(T)$. If $e \in M$, then at most one of the endpoints of e is a leaf, hence $\ell \leq \beta + (n - 2\beta) = n - \beta$. By Theorem 5.3, we have $\mathrm{aecc}_3(T) \geq \mathrm{aecc}_3(BS_{n,\ell})$. Applying Theorem 4.1 again we can then estimate that

$$\mathrm{aecc}_3(T) \geq \mathrm{aecc}_3(BS_{n,\ell}) \geq \mathrm{aecc}_3(BS_{n,n-\beta}) = \mathrm{aecc}_3(T_{n,n-\beta}) ,$$

and we are done. ■

For trees with perfect matchings, Theorem 5.4 together with a straightforward computation of $\mathrm{aecc}_3(T_{n,n/2})$ yields the following consequence.

Corollary 5.5 *If T be a tree of order n with a perfect matching, then*

$$\mathrm{aecc}_3(T) \geq \mathrm{aecc}_3(T_{n,n/2}) = \begin{cases} 3; & n = 4, \\ \frac{9}{2}; & n = 6, \\ \frac{11}{2} - \frac{2}{n}; & n \geq 8. \end{cases}$$

We next give a bound with the independence number of a tree.

Theorem 5.6 *If T be a tree of order n and $\alpha = \alpha(T)$, then*

$$\mathrm{aecc}_3(T) \geq \mathrm{aecc}_3(T_{n,\alpha}) .$$

Proof. Set again $\ell = \ell(T)$. Clearly, $\alpha \geq \ell(T)$. By Theorem 5.3, we have $\mathrm{aecc}_3(T) \geq \mathrm{aecc}_3(BS_{n,\ell})$. By the aid of Theorem 4.1 we conclude that $\mathrm{aecc}_3(T) \geq \mathrm{aecc}_3(BS_{n,\ell}) \geq \mathrm{aecc}_3(BS_{n,\alpha})$. ■

5.5 Lower bounds on trees with constant diameter or radius

Recall that the *diameter* diam(G) and the *radius* rad(G) of a graph G are the maximum and the minimum, respectively, eccentricity of the vertices of G. The *center* of G is the set of its vertices with minimum eccentricity. Recall also that the center of a tree consists either of a single vertex or of two adjacent vertices.

Let $T_{n,d}(p_1,\ldots,p_{d-1})$ be a tree of order n obtained from a path $P_{d+1} = v_0v_1\ldots v_d$ by attaching $p_i \geq 0$ pendent vertices to v_i for every $i \in [d-1]$. Clearly, as the order of $T'_{n,d}(p_1,\ldots,p_{d-1})$ is n, we must have $\Sigma_{i=1}^{d-1} p_i = n-d-1$. In the special case when d is even and all the $n-d-1$ vertices are attached to the vertex $v_{d/2}$, we briefly denote the tree with $T'_{n,d}$. Similarly, if d is odd, then let $T'_{n,d}$ denote the graph in which $\lfloor (n-d-1)/2 \rfloor$ vertices are attached to $v_{\lceil d/2 \rceil}$ and $\lceil (n-d-1)/2 \rceil$ vertices are attached to $v_{\lceil d/2 \rceil}$.

Theorem 5.7 *If T is a tree of order n and* diam$(T) = d$*, then*

$$\text{aecc}_3(T) \geq \text{aecc}_3(T'_{n,d})\,.$$

Proof. Let T be a tree as stated and let $P = v_0v_1\ldots v_d$ be a longest path in T. Since P is a longest path in T, both v_0 and v_d are leaves of T. For $i \in [d-1]$ let T_i be a maximal subtree of T that contains v_i but no other vertex of P. Consider T_i as a rooted tree with the root v_i. Then the depth of the rooted tree T_i is at most the minimum of the lengths of the v_0, v_i-subpath of P and the v_i, v_d-subpath of P, that is, at most $\min\{i, d-i\}$. Therefore, for each $i \in [d-1]$, we can repeatedly apply the π-transformation on the subtree T_i respected to T so than T_i turns into a star rooted at v_i. The average Steiner 3-eccentricity has not increased along the way. After this procedure is over, $T_{n,d}(p_1,\ldots,p_{d-1})$ is constructed. Afterwards we repeatedly apply the π-transformation on each pendent vertex attached to v_i for each $i \in [d-1]$, to arrive at $T'_{n,d}$. ■

Note that if d is odd, then we could define $T'_{n,d}$ also by arbitrary distributing the $n-d-1$ vertices that are attached to $v_{\lceil d/2 \rceil}$ and to $v_{\lceil d/2 \rceil}$. That is, any such tree can serve for the lower bound of Theorem 5.7.

If the center of a tree T contains only one vertex, then diam$(T) = 2\,$rad(T), and if the center of T consists of two vertices, diam$(T) = 2\,$rad$(T) - 1$. Hence Theorem 5.7 yields the following consequence.

Corollary 5.8 *If T is a tree of order n and $r =$ rad(G), then* aecc$_3(T) \geq$ aecc$_3(T'_{n,2r-1})$.

6 Concluding remarks

Let T be a tree of order n. If $k \geq 4$ is a given, fixed integer, then the number of k-subsets of $V(T)$ is a polynomial in n (of degree k). Consequently, using a brute force approach, the average Steiner k-eccentricity of T can be computed in polynomial time. It is unpractical thought. Hence it would be of interest to design faster algorithms for $k \geq 4$, just as we did for the average Steiner 3-eccentricity. Moreover, it would be interesting to know whether there is a polynomial algorithm with time complexity not related to k.

We have derived several lower and upper bounds for the average Steiner 3-eccentricity on a tree with different constrained parameters. It would be interesting to see if and how these bounds extend to $k \geq 4$.

Just a little research has been done by now on the (average) Steiner k-eccentricity for $k \geq 3$. Hence a lot of work still has to be done.

References

[1] B. S. Anand, M. Changat, S. Klavžar, I. Peterin, Convex sets in lexicographic product of graphs, Graphs Combin. 28 (2012) 77–84.

[2] L. W. Beineke, O. R. Oellermann, R. E. Pippert, On the Steiner median of a tree, Discrete Appl. Math. 68 (1996) 249–258.

[3] R. M. Casablanca, P. Dankelmann, Distance and eccentric sequences to bound the Wiener index, Hosoya polynomial and the average eccentricity in the strong products of graphs, Discrete Appl. Math. 263 (2019) 105–117.

[4] G. Chartrand, O. R. Oellermann, S. Tian, H. B. Zou, Steiner distance in graphs, Časopis Pěst. Mat. 114 (1989) 399–410.

[5] P. Dankelmann, H. C. Swart, O. R. Oellermann, The average Steiner distance of graphs with prescribed properties, Discrete Appl. Math. 79 (1997) 91–103.

[6] P. Dankelmann, O. R. Oellermann, H. C. Swart, The average Steiner distance of a graph, J. Graph Theory 22 (1996) 15–22.

[7] P. Dankelmann, W. Goddard, H. C. Swart, The average eccentricity of a graph and its subgraphs, Util. Math. 65 (2004) 41–51.

[8] P. Dankelmann, S. Mukwembi, Upper bounds on the average eccentricity, Discrete Appl. Math. 167 (2014) 72–79.

[9] P. Dankelmann, F. J. Osaye, S. Mukwembi, B. G. Rodrigues, Upper bounds on the average eccentricity of K_3-free and C_4-free graphs, Discrete Appl. Math. 270 (2019) 106–114.

[10] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman & Company, New York, 1979.

[11] M. Ghorbani, X. Li, H. R. Maimani, Y. Mao, S. Rahmani, M R. Parsa, Steiner (revised) Szeged index of graphs, MATCH Commun. Math. Comput. Chem. 82 (2019) 733–742.

[12] T. Gologranc, Steiner convex sets and Cartesian product, Bull. Malays. Math. Sci. Soc. 41 (2018) 627–636.

[13] I. Gutman, X. Li, Y. Mao, Inverse problem on the Steiner Wiener index, Discuss. Math. Graph Theory 38 (2017) 83–95.

[14] I. Gutman, On Steiner degree distance of trees, Appl. Math. Comput. 283 (2016) 163–167.

[15] I. Gutman, B. Furtula, X. Li, Multicenter Wiener indices and their applications, J. Serb. Chem. Soc. 80 (2015) 1009–1017.

[16] F. K. Hwang, D. S. Richards, P. Winter, The Steiner Tree Problem, North-Holland, Amsterdam, 1992.

[17] X. Li, Y. Mao, I. Gutman, The Steiner Wiener index of a graph, Discuss. Math. Graph Theory 36 (2016) 455–465.

[18] X. Li, Y. Mao, I. Gutman, Inverse problem on the Steiner Wiener index, Discuss. Math. Graph Theory 38 (2018) 83–95.

[19] L. Lu, Q. Huang, J. Hou, X. Chen, A sharp lower bound on the Steiner Wiener index for trees with given diameter, Discrete Math. 341 (2018) 723–731.

[20] Y. Mao, Steiner distance in graphs—a survey, arXiv:1708.05779 [math.CO] (18 Aug 2017).

[21] Y. Mao, Steiner Harary index, Kragujevac J. Math. 42 (2018) 29–39.

[22] Y. Mao, E. Cheng, Z. Wang, Steiner distance in product networks, Discrete Math. Theor. Comput. Sci. 20 (2018) paper 8, 25 pp.

[23] Y. Mao, K. C. Das, Steiner Gutman index, MATCH Commun. Math. Comput. Chem. 79 (2018) 779–794.

[24] L.-D. Pei, X.-F. Pan, J. Tian G.-Q. Peng, On AutoGraphiX conjecture regarding domination number and average eccentricity, Filomat 33 (2019) 699–710.

[25] N. Tratnik, On the Steiner hyper-Wiener index of a graph, Appl. Math. Comput. 337 (2019) 360–371.

[26] Z. Wang, Y. Mao, C. Melekian, E. Cheng, Steiner Distance in join, corona, cluster, and threshold graphs, J. Inform. Sci. Eng. 35 (2019) 721–735.

[27] C. Wang, C. Ye, S. Zhang, Z. Wang, H. Li, The Steiner Wiener index of trees with given diameter, Util. Math. 108 (2018) 45–52.

[28] J. Zhang, G.-J. Zhang, H. Wang, X. D. Zhang, Extremal trees with respect to the Steiner Wiener index, Discrete Math. Algorithms Appl. 11 (2019) paper 1950067, 16 pp.