Industrial Mathematics Institute

2004:04

On subtrees of trees

L.A. Székely and H. Wang

IMI
Preprint Series

Department of Mathematics
University of South Carolina

On Subtrees of Trees*

L. A. Székely and Hua Wang
Department of Mathematics,
University of South Carolina, Columbia, SC 29208
{szekely,hwang0}@math.sc.edu

February 11, 2004

*This research was supported in part by the NSF contracts Nr. 007 2187 and 0302307.

Abstract

We study that over a certain type of trees (e.g. all trees or all binary trees) with a given number of vertices, which trees minimize or maximize the total number of subtrees (or subtrees with at least one leaf). Trees minimizing the total number of subtrees (or subtrees with at least one leaf) usually maximize the Wiener index, and vice versa. In [10], we described the structure of binary trees maximizing the total number of subtrees, here we provide a formula for this maximum value. We extend here the results from [10] to binary trees maximizing the total number of subtrees with at least one leaf—this was first investigated by Knudsen [8] to provide upper bound for the time complexity of his multiple parsimony alignment with affine gap cost using a phylogenetic tree.

Also, we show that the techniques of [10] can be adapted to the minimization of Wiener index among binary trees, first solved in [5] and [6].

Using the number of subtrees containing a particular vertex, we define the *subtree core* of the tree, a new concept analogous to, but different from the concepts of *center* and *centroid*.

Keywords: center, centroid, subtree core, number of subtrees, Wiener index, multiple parsimony alignment with affine gap cost, caterpillar, binary tree, tree

1 Terminology

All graphs in this paper will be finite, simple and undirected. A *tree* $T = (V, E)$ is a connected, acyclic graph. We refer to vertices of degree 1 of T as *leaves*. The unique path connecting two vertices v, u in T will be denoted by $P_T(v, u)$. For a tree T and two vertices v, u of T, the *distance* $d_T(v, u)$ between them is the number of edges on the connecting path $P_T(v, u)$. For a vertex v of T, define the *distance of the vertex* as

$$g_T(v) = \sum_{u \in V(T)} d_T(v, u),$$

the sum of distances from v to all other vertices. Let

$$\sigma(T) = \frac{1}{2} \sum_{v \in V(T)} g_T(v)$$

denote the *Wiener index* of T, which is the sum of distances for all unordered pairs of vertices.

We call a tree (T, r) *rooted at the vertex* r (or just by T if it is clear what the root is) by specifying a vertex $r \in V(T)$. For any two different vertices u, v in a rooted tree (T, r), we say that v is a *successor* of u, if $P_T(r, u) \subset P_T(r, v)$. Furthermore, if u and v are adjacent to each other and $d_T(r, u) = d_T(r, v) - 1$, we say that u is a *parent* of v and v is a *child* of u.

If v is any vertex of a rooted tree (T, r), let $T(v)$, *the subtree induced by* v, denote the rooted subtree of T that is induced by v and all its successors in T, and is rooted at v.

The *height* of a vertex v of a rooted tree T with root r is $h_T(v) = d_T(r, v)$, and the *height* of a rooted tree T is $h(T) = \max_{v \in T} h_T(v)$, the maximum height of vertices.

A *binary* tree is a tree T such that every vertex of T has degree 1 or 3. A *rooted binary tree* is a tree T with root r, which has exactly two children, while every other vertex of T has degree 1 or 3. A rooted binary tree T is *complete*, if it has height h and 2^h leaves for some $h \geq 0$. In addition, we also take a single vertex to be a rooted binary tree of height 0.

A *caterpillar tree* is a tree, which has a path, such that every vertex not on the path is adjacent to some vertex on the path. A *binary caterpillar tree* is a caterpillar tree, which is also a binary tree.

For a tree T and a vertex v of T, let $f_T(v)$ denote the number of subtrees of T that contain v, let $F(T)$ denote the number of non-empty subtrees of T.

2 The subtree core of a tree

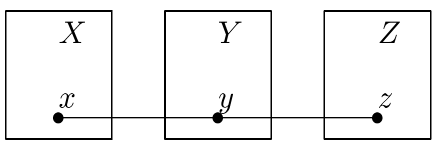
Much research has been devoted to define the "middle part" of a tree. The first such result is due to Jordan [7]. In a tree T, the *branch weight* of a vertex v, $bw(v)$, is the maximum number of edges over all subtrees of T which contain v as a leaf. By definition, the *centroid* $C(T)$ of T is the set of vertices minimizing the branch weight. Jordan [7] showed that either $C(T) = \{c\}$, and $bw(c) \leq \frac{n-1}{2}$, or $C(T) = \{c_1, c_2\}$, where c_1 and c_2 are adjacent vertices with $bw(c_1) = bw(c_2) = \frac{n-1}{2}$, and in both cases all other vertices have branch weight strictly exceeding $\frac{n}{2}$. Zelinka [11] gave an alternative characterization of the centroid: $C(T)$ contains exactly those vertices u of $V(T)$, which minimize the distance function of vertices, i.e. $g_T(u) = \sum_{v \in V} d_T(u, v)$.

Jordan [7] also defined the *center* of a tree T, as the set of vertices minimizing the function *eccentricity* $ecc(u) = \max_{v \in V(T)} d_T(u, v)$, and showed that the center contains one vertex or two adjacent vertices. (For a contemporary reference, see [9] 6.21 and 6.22.) Ádám [1] studied further concepts of centrality in trees.

Here we are going to define the "middle part" of a tree in a new way. Recall that $f_T(v)$ denotes the number of those subtrees of T, which contain v. Define the *subtree core* of T as the set of vertices maximizing $f_T(v)$.

Theorem 2.1. *The subtree core of any tree T contains one or two vertices, and if the subtree core contains two vertices, then they must be adjacent.*

Proof. First we are going to show that f_T is strictly concave along any path of T, and hence f_T is maximized at a single vertex or two adjacent vertices on any path of T.



T

Figure 1: x, y, z are the roots of X, Y, Z respectively.

For any tree T (Fig. 1), consider three vertices x, y, z such that $xy, yz \in E(T)$. Let X, Y, Z denote the components containing x, y, z after the removal of the edges xy and yz from T. Observe the identities

$$f_T(x) = f_X(x) + f_X(x)f_Y(y) + f_X(x)f_Y(y)f_Z(z),$$

$$f_T(z) = f_Z(z) + f_Z(z)f_Y(y) + f_Z(z)f_Y(y)f_X(x),$$

$$f_T(y) = f_Y(y) + f_X(x)f_Y(y) + f_Z(z)f_Y(y) + f_X(x)f_Y(y)f_Z(z).$$

Comparing $f_T(x) + f_T(z)$ and $2f_T(y)$, we obtain

$$2f_T(y) - f_T(x) - f_T(z) = 2f_Y(y) + (f_X(x) + f_Z(z))(f_Y(y) - 1) > 0,$$

and therefore $f_T(.)$ is strictly concave along any path of T. If $f_T(v)$ were maximized in 3 different vertices of T, then any two of them must be consecutive on some path, which yields a contradiction. $\square$

Next, we are going to show that the concept of the subtree core differs from both of the concepts of the center and centroid. Consider tree T_0 in Fig. 2. The center is $\{x\}$, the centroid is $\{y\}$, while the subtree core is $\{z\}$.

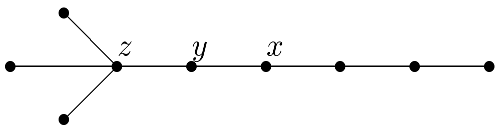


Figure 2: An example showing that the three "middle of the tree" concepts are distinct.

3 Extremal trees for the number of subtrees

It is well-known that the Wiener index among trees on n vertices is minimized by the star $K_{1,n-1}$ and is maximized by the n-vertex path P_{n-1}, see Entringer, Jackcon, and Snyder [4], or Lovász [9] 6.23. We are going to show the counterparts of these simple results for the number of subtrees.

Theorem 3.1. *The n-vertex path P_{n-1} has $\binom{n+1}{2}$ subtrees, fewer than any other tree on n vertices. The star $K_{1,n-1}$ has $2^{n-1}+n-1$ subtrees, more than any other tree on n vertices.*

Proof. For $T = P_{n-1}$ (the path with n vertices), $F(T)$ is the number of ways to choose a sub-path (the number of ways to choose 2 out of n vertices as the end-vertices for the sub-path, allowing that the 2 vertices being the same), so $F(P_{n-1}) = \binom{n+1}{2}$.

For any n-vertex tree T, let $V(T) = \{v_1, v_2, \ldots, v_n\}$. Then, for any $1 \le i \le j \le n$, $P_T(v_i, v_j)$ is a subtree of T, so $F(T) \ge \binom{n+1}{2} = F(P_{n-1})$. If T is not a path, then it has a vertex of degree ≥ 3. This vertex and its 3 neighbors define a subtree not counted by the $P_T(v_i, v_j)$'s, and therefore $F(T) > \binom{n+1}{2}$.

It is easy to see that $F(K_{1,n-1}) = 2^{n-1}+n-1$. We will show by induction on n, that for any non-star n-vertex tree T, $F(K_{1,n-1}) > F(T)$. The base case $n = 1$ holds vacuously.

For any $n \ge 2$, suppose the claim holds for trees with fewer than n vertices. Let T be a tree that maximizes $F(T)$ among n-vertex trees. Consider 2 adjacent vertices x, y of T, let X, Y be the two components of $T - xy$ after deleting the edge xy, such that $x \in X$ and $y \in Y$. Let us use the notation $a = |V(X)|$, $b = |V(Y)|$. Then we have $a + b = n$. According to the decomposition,

$$\begin{aligned} F(T) &= F(X) + F(Y) + f_X(x) f_Y(y) \le F(K_{1,a-1}) + F(K_{1,b-1}) + 2^{a-1+b-1} && (1)\\ &\le 2^{a-1} + 2^{b-1} + n - 2 + 2^{n-2} \le 2^{n-1} + 1 + n - 2 = F(K_{1,n-1}), && (2)\end{aligned}$$

since the function $2^{x-1} + 2^{n-x-1}$ is maximized on the interval $[1, n-1]$ precisely in the endpoints of that interval.

Equality holds in (1) and (2) if and only if $a = 1$ and Y is a star, or $b = 1$ and X is a star. In both cases, T is a star as well. Thus, the induction step is completed. $\square$

To present our main results, we have to give more definitions. Call a rooted binary tree *ordered*, if for every $k \ge 1$, the vertices at height k are put in a linear order, such that if u and v are vertices at height $k+1$, and they have distinct parents, then the order between u and v at height $k+1$ is the same as the order of their parents at height k.

A rooted binary tree is *good*, if (i) the heights of any two of its leaf vertices differ by at most 1; (ii) the tree can be ordered such that the parents of the leaves at the greatest height make a final segment in the ordering of vertices at the next-to-greatest height. For brevity, we often refer to such trees as *rgood binary trees.* A single-vertex rooted binary tree is also rgood.

A binary tree is *good*, if it is obtained from two rgood binary trees T_1 and T_2 by joining their roots with an edge, if (i) for any two leaves, their respective heights in T_1 and/or T_2 differ by at most 1; (ii) at least one of T_1 and T_2 is complete.

Note that good and rgood binary trees are *unique* in the following sense: if we have two good (rgood) binary trees with same number of vertices, then we can label their vertices such that they are isomorphic to each other. The concept of *height* can be naturally extended to vertices of good binary trees, as shown on Fig. 3.

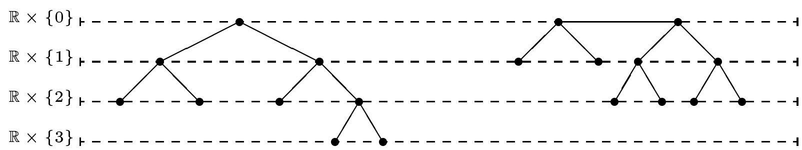


Figure 3: An rgood binary tree (on the left) and a good binary tree (on the right). Vertices at height k of the rgood binary tree and of the two rgood parts of the good binary tree are shown on the line $\mathbb{R} \times k$.

Fischermann, Hoffmann, Rautenbach, Székely, and Volkmann [5] proved:

Theorem 3.2. *Among binary trees with n leaves, precisely the binary caterpillar tree maximizes the Wiener index.*

Fischermann *et al.* [5], and independently Jelen and Triesch [6] proved:

Theorem 3.3. *Among binary trees with n leaves, precisely the good binary tree minimizes the Wiener index.*

We proved in [10]:

Theorem 3.4. *Among binary trees with n leaves, precisely the good binary tree maximizes the number of subtrees.*

We publish the proof of Theorem 3.4 in a separate paper because of its length. We also proved:

Theorem 3.5. *For any $n \geq 2$, precisely the n-leaf binary caterpillar tree, which has $2^{n+1} + 2^{n-2} - n - 4$ subtrees, minimizes the number of subtrees among n-leaf binary trees.*

We postpone the proof of Theorem 3.5 to a later Section.

We see here an amazing and not yet understood relationship between the Wiener index and the number of subtrees. Unfortunately this relationship does not extend as expected. After the results presented in this Section, it might be natural to conjecture that "within certain classes of trees of a fixed parameter, the smaller $F(T)$ is, the bigger $\sigma(T)$ is".

However, using the tree in Fig. 4 we construct binary trees T' and T'', such that $F(T') > F(T'')$ and $\sigma(T') > \sigma(T'')$.

In the binary tree T in Fig. 4, x and y are leaves; $T_1 - \{v_1\}$ is a complete binary tree of height 3 on 15 vertices; $T_2 - \{v_2\}$ is a complete binary tree of height 2 on 7 vertices; T_3 is a binary caterpillar tree on 10 vertices; T_4 is a binary caterpillar tree on 16 vertices.

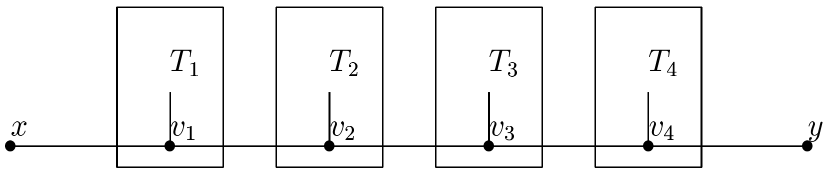


Figure 4: Constructing a counterexample.

Let $A_i = f_{T_i}(v_i)$, $B_i = g_{T_i}(v_i)$, $N_i = |V(T_i)|$ for $i = 1, 2, 3, 4$. Simple calculations show that $A_1 = 677, A_2 = 26, A_3 = 47, A_4 = 383$, $N_1 = N_4 = 16, N_2 = 8, N_3 = 10$. It is easy to verify that

$$f_T(x) = 1 + A_1 + A_1A_2 + A_1A_2A_3 + 2A_1A_2A_3A_4,$$

$$f_T(y) = 1 + A_4 + A_4A_3 + A_4A_3A_2 + 2A_4A_3A_2A_1,$$

$$g_T(x) = \sum_{i=1}^{4}(B_i + iN_i) = 126 + \sum_{i=1}^{4} B_i,$$

$$g_T(y) = \sum_{i=1}^{4}(B_{5-i} + iN_{5-i}) = 124 + \sum_{i=1}^{4} B_i,$$

and therefore $g_T(x) > g_T(y)$. Also,

$$f_T(x) - f_T(y) = (A_1 - A_4)(1 + A_2A_3) + (A_1A_2 - A_3A_4) = 359163 > 0.$$

Take any rooted binary tree X with root r, which has more than one vertex. Define T' as the union of T and X with x being identified with r, and define T'' be the union of T and X with y being identified with r. Then we have the counterexample by

$$\begin{aligned} F(T') - F(T'') &= f_X(r)(f_T(x) - f_T(y)) > 0, \\ \sigma(T') - \sigma(T'') &= \sum_{v \in V(X)} d_X(v, r)(g_T(x) - g_T(y)) > 0. \end{aligned}$$

4 Alternative binary representation of integers

To find a formula for the number of subtrees of rgood and good binary trees will require a novel unique representation of the number $n > 1$ as a sum of powers of 2 that we will write as

$$n = \sum_{i=1}^{l} 2^{k_i}. \tag{3}$$

We describe this representation recursively. We define k_1 by the inequality $2^{k_1} \le \frac{2}{3}n < 2^{k_1+1}$. If we have already defined $k_1, k_2, ..., k_{i-1}$ and $\sum_{t=1}^{i-1} 2^{k_t} < n$, then k_i is defined as follows: if $n-\sum_{t=1}^{i-1} 2^{k_t}=2^m$ for some m, then $k_i=m$ and we have the terminal term in the representation. Otherwise define k_i by the inequality $2^{k_i}\leq \frac{2}{3}\left(n-\sum_{t=1}^{i-1} 2^{k_t}\right)< 2^{k_i+1}$.

The definition of k_1 differs only in one aspect from the definition of the generic k_i: in the first step powers of two are split further, while in the generic step they are not. This means in particular that for any $n>1$, we have $l\geq 2$. If $l=2$, then $k_2+1\geq k_1\geq k_2\geq 0$, and $k_1=k_2$ if and only if $n=2^{k_1+1}$. We always have $k_1=\lfloor \log_2(\frac{2}{3}n)\rfloor$.

The representation is clearly unique and has the properties that the terms are decreasing: $k_1\geq k_2\geq ...$; and that the representation is hereditary in the following sense: if n is represented as $\sum_{i=1}^{l} 2^{k_i}$, then for all $j\leq l-1$

$$\sum_{i=j}^{l} 2^{k_i} \tag{4}$$

is the representation of the numerical value of the sum in (4).

We use a simple Lemma from [10]:

Lemma 4.1. *Removing the root of a rooted binary tree T, we obtain two rgood induced subtrees, T_1 and T_2. Assume that T_1 has no more leaves than T_2. Now T is rgood if and only if one of the following conditions hold:*
i) $h(T_1)=h(T_2)$, and T_2 is complete;
ii) $h(T_1)=h(T_2)-1$, and T_1 is complete. □

Lemma 4.1 immediately implies that the terms in the novel binary representation of n correspond to the following procedure decomposing rgood binary trees into a sequence of complete binary trees with the same total number of leaves. We give the decomposition recursively. In the first step, if the rgood binary tree T on $n>1$ leaves is complete, then the decomposition splits it into two isomorphic complete (and rgood) binary trees. In later steps, if an emerging rgood binary tree is complete, we do not split it further. If the emerging rgood binary tree is not complete, remove the root to obtain two induced rooted binary trees T_1 and T_2. If (i) from Lemma 4.1 applies, write down T_2 and consider T_1 for further splitting. If (ii) from Lemma 4.1 applies, write down T_1 and consider T_2 for further splitting. It is clear that in any case the first complete binary tree in the decomposition has at most $2/3$ of the leaves of T, but has more than $1/3$ of them.

There is another simple Lemma in [10] describing the structure of good binary trees:

Lemma 4.2. *Let us be given two rgood binary trees, T' and T'', such that $h(T')\leq h(T'')$. Join with an edge the roots of T' and T'' to obtain the binary tree T. Now T is good if and only if one of the following conditions hold:*
i) $h(T')=h(T'')$, and one or both of T' and T'' is complete;
ii) $h(T')=h(T'')-1$, and T' is complete. □

It is clear from Lemma 4.2 that the novel binary representation also descibes numerically splitting the good binary tree into two rgood binary trees by deleting the edge on $\mathbb{R}\times 0$, and then splitting further the arising rgood binary trees as described above for the decomposition of rgood binary trees.

5 Closed formula for the number of subtrees in good binary trees

An interesting question remaining after Theorem 3.4 is to calculate $F(T)$ when T is a good binary tree with n leaves. This will be done by solving a number of recurrences. Let R_n denote the rgood binary tree on n leaves, rooted at the vertex r of degree 2. Let f_n denote the number of subtrees of R_n containing the root, i.e. $f_n = f_{R_n}(r)$. Notice that we suppressed the root and the tree in the notation f_n. Observe the initial values $f_1 = 1,\ f_2 = 4$. Next, let F_n denote the total number of subtrees in R_n, i.e. $F_n = F(R_n)$. Observe the initial values $F_1 = 1,\ F_2 = 6$. Let G_n denote the good binary tree on n leaves. The plan to compute $F(G_n)$ is the following: we evaluate f_{2^k}, F_{2^k}, F_n, and $F(G_n)$ in this order.

Counting the *empty* subtree as well with $f_n + 1$, it is not hard to see the following recurrence relationship for all $k \geq 1$:

$$(f_{2^k} + 1) = (f_{2^{k-1}} + 1)^2 + 1, \tag{5}$$

and $f_1 = 1$. Fortunately, Aho and Sloane [2] solved the recurrence relation (5) explicitly:

$$f_{2^k} = \lfloor q^{2^{k+1}} \rfloor - 1 \tag{6}$$

for $k \geq 0$, where $\lfloor a \rfloor$ is the floor function of a, and

$$q = \exp\Big(\sum_{i=0}^{\infty} 2^{-i-1} \ln\big(1 + \frac{1}{f_{2^i}}\big)\Big) = \exp\Big(\frac{1}{2}\ln 2 + \frac{1}{4}\ln\frac{5}{4} + \frac{1}{8}\ln\frac{26}{25} + \frac{1}{16}\ln\frac{677}{676} + \ldots\Big).$$

Numerically $q = 1.502837\ldots$. For further details, see [2].

Observe that $F_1 = 1$ and that for all $k \geq 1$ the following recurrence relation holds:

$$F_{2^k} = 2F_{2^{k-1}} + f_{2^k} = f_{2^k} + 2f_{2^{k-1}} + 4f_{2^{k-2}} + \ldots + 2^{k-1} f_{2^1} + 2^k F_1. \tag{7}$$

Using (6), it is easy to solve (7) by

$$F_{2^k} = \sum_{i=0}^{k-1} 2^i \lfloor q^{2^{k-i+1}} \rfloor + 1. \tag{8}$$

Next, we try to compute f_n based on the representation of n in (3), using the following more general version of (5):

$$(f_n + 1) = (f_{2^{k_1}} + 1)(f_{n-2^{k_1}} + 1) + 1. \tag{9}$$

As we noted in Section 4, for every $n > 1$ we have $l \geq 2$. Therefore, iterating (9) yields: $(f_n + 1) =$

$$= \sum_{i=1}^{l-2} \prod_{j=1}^{i} (f_{2^{k_j}} + 1) + \prod_{j=1}^{l} (f_{2^{k_j}} + 1) + 1 = \sum_{i=1}^{l-2} \prod_{j=1}^{i} \lfloor q^{2^{k_j+1}} \rfloor + \prod_{j=1}^{l} \lfloor q^{2^{k_j+1}} \rfloor + 1. \tag{10}$$

Observe that using the decomposition of R_n described in Section 4 to generalize (7), we obtain a recursion for F_n as well:

$$F_n = F_{2^{k_1}} + F_{n-2^{k_1}} + f_n. \tag{11}$$

Solving (11) by iteration over the same decomposition, we obtain

$$F_n = \sum_{i=1}^{l} F_{2^{k_i}} + \sum_{i=1}^{l-1} f_{\sum_{j=i}^{l} 2^{k_j}} = \sum_{i=1}^{l} (\sum_{j=0}^{k_i-1} 2^j \lfloor q^{2^{k_i-j+1}} \rfloor + 1) + \sum_{i=1}^{l-1} f_{\sum_{j=i}^{l} 2^{k_j}}. \tag{12}$$

Notice that (12) still contains f-terms. Using (10) we substitute them by explicit terms for $i \le l-1$:

$$f_{\sum_{j=i}^{l} 2^{k_j}} = \sum_{j=i}^{l-2} \prod_{s=i}^{j} \lfloor q^{2^{k_s+1}} \rfloor + \prod_{j=i}^{l} \lfloor q^{2^{k_j+1}} \rfloor;$$

and then express explicitly F_n:

$$F_n = \sum_{i=1}^{l} (\sum_{j=0}^{k_i-1} 2^j \lfloor q^{2^{k_i-j+1}} \rfloor + 1) + \sum_{i=1}^{l-1} (\sum_{j=i}^{l-2} \prod_{s=i}^{j} \lfloor q^{2^{k_s+1}} \rfloor + \prod_{j=i}^{l} \lfloor q^{2^{k_j+1}} \rfloor). \tag{13}$$

Next, observe for all $n > 1$ the identity

$$F(G_n) = F_n - 1 - f_{2^{k_1}} - f_{n-2^{k_1}} \tag{14}$$

holds, and gives an explicit formula to $F(G_n)$ in view of (3), (6), (10), and (13). Indeed, (14) is true for the following reason. Let r denote the root of R_n, let its neighbors be x and y, such that x is the root of the subtree of 2^{k_1} leaves. Categorise the subtrees of R_n by the following cases: (1) does not contain any of r, x, y; (2) contains x but not r; (3) contains y but not r; (4) contains all of x, y, r. (5) the one-vertex tree r; (6) contains r and x but not y—there are $f_{2^{k_1}}$ of them; (7) contains r and y but not x—there are $f_{n-2^{k_1}}$ of them. Deleting r and joining x to y establishes a bijection between subtrees of G_n and subtrees of R_n in the cases (1)–(4).

From the formula (14) and (16) one can obtain $F(G_n)$ for small values of n as shown in the table below. The table also includes $F^*(G_n)$, the number of subtrees of G_n containing at least one leaf. Formula (16) will determine $F^*(G_n)$.

n	1	2	3	4	5	6	7
f_n	1	4	10	25	55	130	286
F_n	1	6	17	37	78	173	340
$F(G_n)$		3	11	28	63	143	304
$F^*(G_n)$		3	10	25	57	132	287

6 Some proofs

The main goal of this section is to prove Theorem 3.5. We do not give however, the simplest proof that we know. Instead, we prove two lemmas that we need in [10] to prove Theorem 3.4. In Sections 7 and 8 we twist these lemmas further to adapt them for the solution of two other extremal problems, also optimized by good trees among binary trees with given number of vertices and leaves.

Consider the tree T in Fig. 5, with leaves x and y, and $P_T(x,y) = xx_1 \dots x_n z y_n \dots y_1 y$ $(xx_1 \dots x_n y_n \dots y_1 y)$ if $d_T(x,y)$ is even (odd), for any $n \ge 0$

After the deletion of all the edges of $P_T(x,y)$ from T, some connected components will remain. Let X_i denote the component that contains x_i, let Y_i denote the component that contains y_i, for $i = 1, 2, \dots, n$, and let Z denote the component that contains z. Set

$a_i = f_{X_i}(x_i)$ and $b_i = f_{Y_i}(y_i)$ for $i = 1, \dots, n$, and $c = f_Z(z)$.

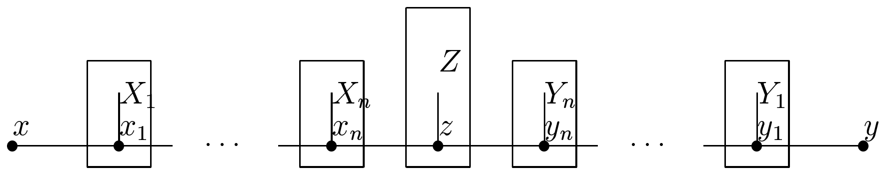


Figure 5: Path $P_T(x, y)$ connecting leaves x and y.

Lemma 6.1. *In the situation described above, if $a_i \geq b_i$ for $i = 1, 2, \ldots, n$, then $f_T(x) \geq f_T(y)$. Furthermore, $f_T(x) = f_T(y)$ if and only if $n = 0$ or $a_i = b_i$ for all i.*

Proof. We cover the case when z and Z occur, a similar argument works when z and Z do not occur. Denote by $N = c\prod_{i=1}^{n}(a_i b_i)$ the number of subtrees that contain both x and y. We have

$$f_T(x) = 1 + \sum_{k=1}^{n}(\prod_{i=1}^{k} a_i) + c(\prod_{i=1}^{n} a_i) + c(\prod_{i=1}^{n} a_i)(\sum_{k=1}^{n}(\prod_{j=n+1-k}^{n} b_j)) + N;$$

$$f_T(y) = 1 + \sum_{k=1}^{n}(\prod_{j=1}^{k} b_j) + c(\prod_{j=1}^{n} b_j) + c(\prod_{j=1}^{n} b_j)(\sum_{k=1}^{n}(\prod_{i=n+1-k}^{n} a_i)) + N;$$

Then we have $f_T(x) - f_T(y) =$

$$\sum_{k=1}^{n}(\prod_{i=1}^{k} a_i - \prod_{j=1}^{k} b_j) + c(\prod_{i=1}^{n} a_i - \prod_{j=1}^{n} b_j) + c\sum_{k=1}^{n}(\prod_{i=1}^{n-k} a_i - \prod_{j=1}^{n-k} b_j)(\prod_{l=n+1-k}^{n} a_l b_l) \geq 0,$$

with strict inequality if $a_i > b_i$ for any $i \in \{1, 2, \ldots, n\}$. $\square$

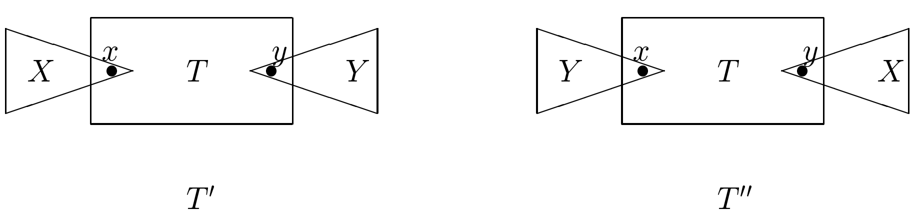


Figure 6: Switching subtrees rooted at x and y.

If we have a tree T with leaves x and y, and two rooted trees X and Y, then we can build two new trees, first T', by identifying the root of X with x and the root of Y with y, second T'', by identifying the root of X with y and the root of Y with x. Under the circumstances below we can tell which composite tree has more subtrees.

Lemma 6.2. *If $f_T(x) > f_T(y)$ and $f_X(x) < f_Y(y)$, then we have*

$$F(T'') > F(T').$$

Proof. When T' changes to T'', the number of subtrees which contain both or neither of x and y do not change, so we only need to consider the number of subtrees which contain precisely one of x and y. For T', the number of subtrees which contain x but not y is

$$f_X(x)(f_T(x) - N),$$

the number of the subtrees which contain y but not x is

$$f_Y(y)(f_T(y) - N),$$

where N is the number of subtrees of T that contain both x and y. Similarly, for T'', these two numbers are

$$f_Y(y)(f_T(x) - N) \quad \text{and} \quad f_X(x)(f_T(y) - N).$$

We have

$$F(T'') - F(T') = (f_Y(y) - f_X(x))(f_T(x) - f_T(y)) > 0.$$

□

We use Lemmas 6.1 and 6.2 to prove Theorem 3.5.

Proof. Let C_n denote the binary caterpillar with $n \geq 2$ leaves as in Fig. 7. First we are going to calculate $F(C_n)$. We start with observing $F(C_2) = 3$. For $n \geq 3$ we have the following recurrence relationship for $F(C_n)$:

$$F(C_n) = F(C_{n-1}) + 3f_{C_{n-1}}(v_{n-1}) + 2,$$

where $F(C_{n-1})$ is the number of subtrees of C_n which contain neither of v_n nor u_{n-1}; $3f_{C_{n-1}}(v_{n-1}) + 2$ is the number of subtrees of C_n which contain u_{n-1} but not v_n, v_n but not u_{n-1}, or both u_{n-1} and v_n. Also, we have the following recurrence relationship for $f_{C_n}(v_n)$:

$$f_{C_n}(v_n) = 2f_{C_{n-1}}(v_{n-1}) + 1,$$

since for each subtree S of C_{n-1} counted in $f_{C_{n-1}}(v_{n-1})$, $T_1 = S \cup \{v_n\}$ and $T_2 = S \cup \{v_n\} \cup \{u_{n-1}\}$ is each a subtree of C_n that contains v_n. And 1 in the formula counts v_n itself as a subtree of C_n that contains v_n.

It follows that $f_{C_n}(v_n) = 2^{n-1} + 2^{n-2} - 1$, as $f_{C_2}(v_2) = 2$. Thus, we can easily calculate that $F(C_n) = 2^{n+1} + 2^{n-2} - n - 4$.

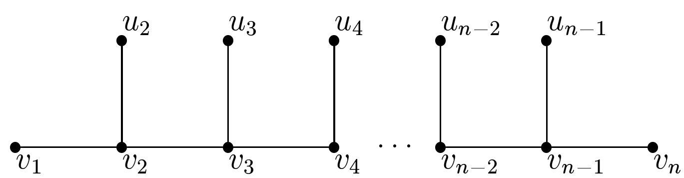


Figure 7: A caterpillar tree with n leaves

Let now T be a binary tree with n leaves that minimizes $F(T)$, and suppose (for contradiction) that T is not a caterpillar. Note that this implies $n \geq 3$.

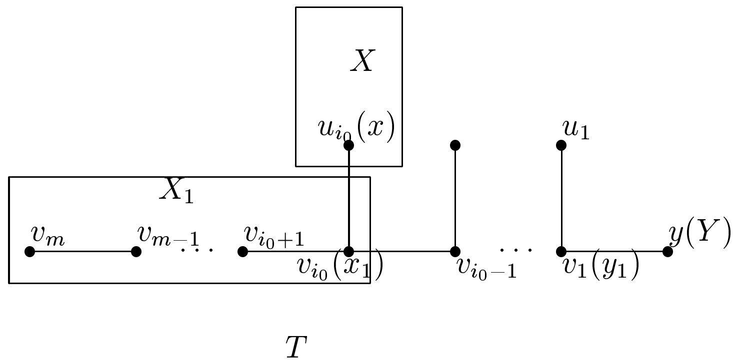


Figure 8: T not being a binary caterpillar tree.

Let $P = v_m v_{m-1} \dots v_1 y$ be a longest path in T. Clearly $m \geq 2$. Then v_m and y must be leaves. Let u_i be the neighbor of v_i that is not on P, for $i = 1, 2, \dots, m-1$. Note that the u_i's exist, since T is a binary tree. It is easy to see that u_{m-1} and u_1 must be leaves by the choice of P. Let

$$i_0 = \min\{\ i \in \{1, 2, \dots, m-1\} \text{ s.t. } u_i \text{ is not a leaf }\},$$

i_0 exists since T is not a caterpillar tree, and we have T as shown in Fig. 8.

To use Lemma 6.1, substitute

$$x \leftarrow u_{i_0}; x_1 \leftarrow v_{i_0}; y_1 \leftarrow v_1; \dots,$$

then we have $X, X_1, \dots,\ Y, Y_1, \dots,$ (and Z if necessary) respectively as in Lemma 6.1. Notice that we obtain $Y \leftarrow \{y\}$, a single vertex tree, and observe that $f_X(x) > f_Y(y)$, and $a_1 > b_1 = 2$, $a_i = b_i = 2$ for all the other i. By Lemma 6.1, we have $f_S(x) > f_S(y)$, where $S = (T \backslash X) \cup \{x\}$.

Hence, we can apply Lemma 6.2 and it follows that if we interchange X and Y (which is actually moving X to y), we will decrease $F(T)$, while not changing the number of leaves Thus, we have a contradiction, and hence T is a caterpillar. □

7 Further relation between $F(T)$ and $\sigma(T)$

The lemmas in this paper can be modified (which we will outline below) to prove Theorems 3.2 and 3.3. First, notice that for two rgood binary trees T and T' with roots r and r' respectively, then one is always a subtree of the other. Therefore we have

$$\begin{aligned} g_T(r) > g_{T'}(r') \quad &\Leftrightarrow \quad |V(T)| > |V(T')| \qquad \text{and} \\ g_T(r) = g_{T'}(r') \quad &\Leftrightarrow \quad |V(T)| = |V(T')|. \end{aligned}$$

Consider now the same tree as in the Lemma 6.1, shown at Fig. 5, and set

$a'_i = g_{X_i}(x_i)$, $N_i = |V(X_i)|$ and $b'_i = g_{Y_i}(y_i)$, $M_i = |V(Y_i)|$ for $i = 1, 2, \dots, n$, and $c' = g_Z(z)$, $N = |V(Z)|$. (Note that z and Z exist if and only if $d_T(x, y)$ is even).

Lemma′ 6.1 *If $N_i \geq M_i$ for $i = 1, 2, \dots, n$, then $g_T(x) \leq g_T(y)$. Furthermore, $g_T(x) = g_T(y)$ if and only if $n = 0$ or $N_i = M_i$ for any $1 \leq i \leq n$.*

Proof. If z and Z occur, we have

$$g_T(x) = \sum_{i=1}^{n} iN_i + (n+1)N + \sum_{j=1}^{n}(2n+2-j)M_j + \sum_{i=1}^{n} a'_i + \sum_{j=1}^{n} b'_j + c' + 2n,$$

$$g_T(y) = \sum_{i=1}^{n} iM_i + (n+1)N + \sum_{j=1}^{n}(2n+2-j)N_j + \sum_{i=1}^{n} a'_i + \sum_{j=1}^{n} b'_j + c' + 2n.$$

It is not hard to see that $g_T(x) - g_T(y) =$

$$= \sum_{i=1}^{n} i(N_i - M_i) + \sum_{j=1}^{n}(2n+2-j)(M_j - N_j) = \sum_{i=1}^{n}(2i-2n-2)(N_i - M_i) \leq 0,$$

with strict inequality if $N_i > M_i$ for any $i \in \{1, 2, \ldots, n\}$.

A similar argument works if z and Z do not occur. □

Consider the trees T' and T'' in Lemma 6.2. Then we have

Lemma′ 6.2 *If $g_T(x) < g_T(y)$ and $|V(X)| > |V(Y)|$, then $\sigma(T'') > \sigma(T')$.*

Proof. Similar to the proof of Lemma 6.2, we only need to consider the paths which contain one and only one of x and y. For T', the sum of the lengths of the paths which contain x but not y is

$$g_X(x)|V(T)| + (g_T(x) - d_T(x,y))|V(X)|,$$

the sum of the lengths of the paths which contain y but not x is

$$g_Y(y)|V(T)| + (g_T(y) - d_T(x,y))|V(Y)|.$$

Similarly, for T'', these two numbers are

$$g_Y(y)|V(T)| + (g_T(x) - d_T(x,y))|V(Y)|,$$

and

$$g_X(x)|V(T)| + (g_T(y) - d_T(x,y))|V(X)|.$$

Therefore $\sigma(T') - \sigma(T'') =$

$$\begin{aligned} &= g_T(x)|V(X)| + g_T(y)|V(Y)| - g_T(x)|V(Y)| - g_T(y)|V(X)| \\ &= (g_T(x) - g_T(y))(|V(X)| - |V(Y)|) < 0. \end{aligned}$$

□

Using arguments similar to those in the proofs of Theorems 3.5, Theorem 3.2 about the maximization of the Wiener index among binary trees can be proved using the lemmas above. To obtain an alternative proof to Theorem 3.3 of Fischermann *et al.* [5] about the minimization of the Wiener index among binary trees, one has to repeat, mutatis mutandis, the proof of Theorem 3.4, as it is written in [10]. For guidance, we state below the most crucial lemma. Before that, we have to make some additional definitions and conventions.

If T is a rooted binary tree with root r, and r_1, r_2 are the children of r, then we will simply write T_1 for $T(r_1)$ and T_2 for $T(r_2)$. We assign the labels r_1 and r_2 according the following rule: $|V(T_2)| \geq |V(T_1)|$. T_i will be rooted at r_i, $i = 1, 2$. We define recursively $T_{i_1 i_2 \ldots i_k 1}$ and $T_{i_1 i_2 \ldots i_k 2}$ to be the two rooted binary trees induced by the children of the root of $T_{i_1 i_2 \dots i_k}$, if $T_{i_1 i_2 \dots i_k}$ is not a single vertex, where $i_j \in \{1, 2\}$, $j = 1, 2, \dots, k$. We assign the labels $r_{i_1 i_2 \dots i_k 1}$ and $r_{i_1 i_2 \dots i_k 2}$ according the following rule:

$$|V(T_{i_1 i_2 \dots i_k 2})| \geq |V(T_{i_1 i_2 \dots i_k 1})|. \tag{15}$$

We complete the recursive definition by letting $r_{i_1 i_2 \dots i_k}$ be the root for $T_{i_1 i_2 \dots i_k}$.

Lemma 7.1. *Assume T is a binary tree that minimizes $\sigma(T)$ among n-leaf binary trees. Assume that T is divided into two rooted subtrees T', T'' by the removal of the edge $v'v''$ as shown in Fig. 4 in [10]. Then, if for all $k \geq 1$ the inequalities*

$$|V(T')| > |V((T'')_{\underbrace{2 \dots 21}_{k\ 2's}})|$$

hold, then T'' is rgood. □

8 Subtrees with at least one leaf

Knudsen [8] provided a multiple parsimony alignment with affine gap cost using a phylogenetic tree. In bounding the time complexity of his algorithm, a factor was the number of so-called "acceptable residue configuration". In our terms, it is the number of subtrees containing at least one leaf vertex. Knudsen estimated the maximum number of acceptable residue configurations over all binary trees. Here we show that good binary trees have the largest number of acceptable residue configurations and provide a formula to compute the number of them. Knudsen's estimate easily follows from the formula.

To complement our earlier results, we show that caterpillar trees minimize the number of acceptable residue configurations, and also study Knudsen's problem for arbitrary trees.

We give some formal definitions. For a tree T and a vertex $v \in V(T)$, let $f^*_T(v)$ denote the number of subtrees of T which contain v and at least one leaf different from v; and let $F^*(T)$ denote the number of subtrees of T which contain at least one leaf. If T is a single-vertex tree, then f^*_T and F^* vanishes on it.

Theorem 8.1. *Among trees on $n \geq 3$ vertices, the path P_{n-1} minimizes F^* with $F^*(P_{n-1}) = 2n - 1$; while the star $K_{1,n-1}$ maximizes F^* with $F^*(K_{1,n-1}) = 2^{n-1} + n - 2$.*

Proof. For $T = P_{n-1}$, $F^*(T) = F(P_{n-1}) - F(P_{n-3}) = 2n - 1$.

For any n-vertex tree T, let $V(T) = \{v_1, v_2, \dots, v_n\}$. Let v_1, v_n be two of the leaves of T (since a tree T has at least 2 leaves for $n \geq 3$). Then, for any $1 \leq i \leq n$, $P_T(v_1, v_i)$ and $P_T(v_n, v_i)$ are subtrees of T that contain at least a leaf, so $F^*(T) \geq 2n - 1$ ($P_T(v_1, v_n)$ was counted twice in the above analysis). If T is not a path, then it has at least another leaf, say v_2, then the single vertex v_2 is not counted, and therefore $F^*(T) > 2n - 1$.

It is easy to see that $F^*(K_{1,n-1}) = F(K_{1,n-1}) - 1 = 2^{n-1} + n - 2$.

For any n-vertex tree T, $F^*(T) = F(T) - F(H)$ where H is the subtree obtained by deleting all the leaves of T. We already know that $F(T) \leq F(K_{1,n-1}) = 2^{n-1} + n - 1$. For $n \geq 3$, T have at least one vertex that is not a leaf, and hence $F(H) \geq 1$. Thus, $F^*(T) \leq F(K_{1,n-1}) - 1 = 2^{n-1} + n - 2$.

Equality holds in the above if and only if $T = K_{1,n-1}$. □

Theorem″ 3.4 *If T is a binary tree that maximizes $F^*(T)$ among n-leaf binary trees, then T must be good.* □

Theorem″ 3.5 *Among n-leaf binary trees, the caterpillar tree minimizes $F^*(T)$.* □

We postpone the sketch of Theorem″ 3.4 to the next section, and we leave the proof of Theorem″ 3.5 to the reader.

Using the notation G_n from Section 5, we have that

Theorem 8.2. *The maximum value of $F^*(T)$ among n-leaf binary trees is*

$$F^*(G_n) = \begin{cases} F(G_n) - F(G_{\frac{n}{2}}) & \text{if } n \text{ even,} \\ F(G_n) - \frac{2}{3}F(G_{\frac{n-1}{2}}) - \frac{1}{3}F(G_{\frac{n+1}{2}}) - \frac{1}{3} & \text{if } n \text{ odd.} \end{cases} \tag{16}$$

Proof. The maximizing tree is good, so $T = G_n$, and $F(G_n)$ counts all of its subtrees. If n is even, then for correction, from $F(G_n)$ we have to subtract the number of subtrees of H, where H is the tree obtained from T deleting all leaves. If n is even, then H is also a good binary tree, but on $\frac{n}{2}$ vertices.

If n is odd, then there is a problem: after the deletion of leaves the remaining tree is not binary.

Make a drawing of G_n as described at the definition of goodness, and label the leaves from left to right as $v_1, v_2, \ldots, v_n$. Let v_k be the last leaf of height $h(G_n) - 1$. Then k must be odd. Let x be the parent of v_k and u be the other child of x. Observe that the children of u are v_{k+1} and v_{k+2} (see Fig. 9). Again, let H be obtained from G_n by deleting all leaves, and let $G_{\frac{n-1}{2}}$ be obtained from H by deleting u. Then we have

$$F^*(G_n) = F(G_n) - F(H) \tag{17}$$

and

$$F(H) = F(G_{\frac{n-1}{2}}) + 1 + f_{G_{\frac{n-1}{2}}}(x). \tag{18}$$

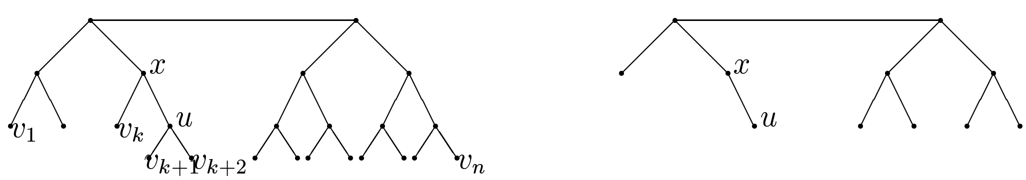


Figure 9: Left: G_n; right: H, $H \setminus \{u\} = G_{\frac{n-1}{2}}$ and $H \cup \{(x, v_k)\} = G_{\frac{n+1}{2}}$.

Observe the identity

$$3f_{G_{\frac{n-1}{2}}}(x) = F(G_{\frac{n+1}{2}}) - F(G_{\frac{n-1}{2}}) - 2. \tag{19}$$

We justify (19) with a case analysis referring to Fig. 9. A subtree of $G_{\frac{n+1}{2}} = H \cup \{(x, v_k)\}$ can be a $\{u\}$, $\{v_k\}$, a subtree of $G_{\frac{n-1}{2}} = H \setminus \{u\}$, or a subtree of $G_{\frac{n-1}{2}} = H \setminus \{u\}$ containing x with one or two elements of $\{u, v_k\}$ added.

Now the second case of (16) follows from equations (17), (18), and (19). □

Similar to the subtree core, we define the *f^*-subtree core* of a tree T as the set of vertices maximizing $f_T^*(v)$.

Theorem 8.3. *Assume that the tree T has no vertices of degree 2. Then the f^*-subtree core of the tree T contains one or two vertices, and if the f^*-subtree core contains two vertices, then they must be adjacent.*

Proof. As in the proof of Theorem 2.1, it is enough to show that f_T^* is strictly concave along any path of T.

For any tree T (Fig. 1), consider three vertices x, y, z such that $xy, yz \in E(T)$. Let X, Y, Z denote the components containing x, y, z after the removal of the edges xy and yz from T. "Cancelling" with subtrees containing both x, y and a leaf, (both y, z and a leaf) we obtain the identities

$$f_T^*(y) - f_T^*(x) = f_{Y\cup Z}^*(y) - f_X^*(x),$$

$$f_T^*(y) - f_T^*(z) = f_{X\cup Y}^*(y) - f_Z^*(z).$$

Since the degree of x and y is not 2, $Y \setminus \{y\}$ is not empty and hence $f_{Y\cup Z}^*(y) > f_Z^*(z)$ and $f_{X\cup Y}^*(y) > f_X^*(x)$. Comparing $f_T^*(x) + f_T^*(z)$ and $2f_T^*(y)$, we obtain

$$2f_T^*(y) - f_T^*(x) - f_T^*(z) = f_{Y\cup Z}^*(y) + f_{X\cup Y}^*(y) - f_X^*(x) - f_Z^*(z) > 0.$$

□

Note: The result above is not true for every tree. Take for example the path $T = P_n$, where every non-leaf vertex is in the f^*-subtree core.

9 Key Lemmas toward Theorem″ 3.4

Consider again the tree in Lemma 6.1 (Fig. 5), and use from there the notation a_i, b_i, and c. Let $a_i'' = f_{X_i}^*(x_i)$, $b_i'' = f_{Y_i}^*(y_i)$, for $i = 1, 2, \ldots, n$ and $c'' = f_Z^*(z)$. (Note that z and Z exist if and only if $d_T(x, y)$ is even.)

Lemma″6.1 *If $a_i \geq b_i$ and $a_i'' \geq b_i''$ for $i = 1, 2, \ldots, n$, then*

$$f_T(x) \geq f_T(y) \tag{20}$$

and

$$f_T^*(x) \geq f_T^*(y). \tag{21}$$

Furthermore, if a strict inequality $a_i > b_i$ holds for any i, $i \in \{1, 2, \ldots, n\}$, then we have strict inequalities in (20) and (21).

Proof. We consider only the case when z and Z occur, the other case is similar. Lemma 6.1 already proved part of the conclusion (20) in Lemma 6.1. Let us consider the following families of subtrees of T:

$$\begin{aligned}
A_i &= \text{the set of subtrees of } T \text{ which contain } x,\ x_i, \text{ and at least one other} \\
&\quad\ \text{leaf, and do not contain } x_{i+1} \text{ (or } z \text{ if } i = n\text{);} \\
A &= \text{the set of subtrees of } T \text{ which contain } x,\ z \text{ and at least one other} \\
&\quad\ \text{leaf, and do not contain any } y_n\text{;} \\
C_i &= \text{the set of subtrees of } T \text{ which contain } x,\ y_i, \text{ and at least one} \\
&\quad\ \text{other leaf, and do not contain } y_{i-1} \text{ (or } z \text{ if } i = 1\text{);}
\end{aligned}$$

$$
\begin{aligned}
B_i &= \text{the set of subtrees of } T \text{ which contain } y,\ y_i \text{ and at least one other} \\
&\ \text{leaf, and do not contain } y_{i+1} \text{ or } (z \text{ for } i = n); \\
B &= \text{the set of subtrees of } T \text{ which contain } y,\ z \text{ and at least one} \\
&\ \text{other leaf, and do not contain } x_n; \\
D_i &= \text{the set of subtrees of } T \text{ which contain } y,\ x_i, \text{ and at least one} \\
&\ \text{other leaf and do not contain } x_{i-1} \text{ (or } z \text{ if } i = 1).
\end{aligned}
$$

The following identities follow from a simple case analysis:

$$
f_T^*(x) = |A| + \sum_{i=1}^{n}(|A_i| + |C_i|); \qquad f_T^*(y) = |B| + \sum_{i=1}^{n}(|B_i| + |D_i|).
$$

We are going to show by establishing injections that for $i = 1, 2, \ldots, n$

$$
|A_i| \geq |B_i|; \qquad |A| \geq |B|; \qquad |C_i| \geq |D_i|,
$$

and hence

$$
f_T^*(x) \geq f_T^*(y),
$$

with strict inequality if and only if any of the conditions has strict inequality.

We give an injection from B_i to A_i as follows.

For $t = 1, 2, ..., n$, observe that Y_t has no more subtrees containing y_t than subtrees of X_t containing x_t by the assumption $a_t \leq b_t$; and furthermore, Y_t has no more subtrees containing y_t and at least one more leaf than subtrees of X_t containing x_t and at least one more leaf by the assumption $a_t'' \leq b_t''$. Therefore, one can construct an a map τ_t, which maps subtrees of X_t containing x_t to subtrees of Y_t containing y_t in an injective way, which has an additional property that subtrees containing at least one more leaf are mapped to subtrees containing at least one more leaf.

Consider the map ϵ which assigns to an $F \in A_i$ the ordered i-tuple of trees $(F \cap X_1, F \cap X_2, ..., F \cap X_i)$; and the map μ which assigns to an $H \in B_i$ the ordered i-tuple of trees $(H \cap Y_1, H \cap Y_2, ..., H \cap Y_i)$. These maps are injective and their ranges include all i-tuples in which every component restricted to X_i (Y_i) also contains x_i (y_i). Now the injection from B_i to A_i is the following:

$$
H \in B_i \mapsto \epsilon^{-1}\Big(\tau_1(H \cap Y_1), \tau_2(H \cap Y_2), ..., \tau_i(H \cap Y_i)\Big).
$$

Also, this map puts subtrees containing other leaves than y to subtrees containing other leaves than x.

We give an injection from D_i to C_i as below. The injection from B to A can be constructed similarly and we leave it to the reader.

Consider the map ϵ' which assigns to an $F \in C_i$ the ordered i-tuple of trees $(F \cap X_1, F \cap X_2, ..., F \cap X_i - 1, F \cap ((\cup_{j=i}^n X_j) \cup Z \cup (\cup_{j=i}^n Y_j)))$; and the map μ' which assigns to an $H \in D_i$ the ordered i-tuple of trees $(H \cap Y_1, H \cap Y_2, ..., H \cap Y_i - 1, H \cap ((\cup_{j=i}^n X_j) \cup Z \cup (\cup_{j=i}^n Y_j)))$. Now the injection from D_i to C_i is the following:

$$
H \in D_i \mapsto \epsilon'^{-1}\Big(\tau_1(H \cap Y_1), \tau_2(H \cap Y_2), ..., \tau_{i-1}(H \cap Y_{i-1}, H \cap ((\cup_{j=i}^n X_j) \cup Z \cup (\cup_{j=i}^n Y_j)))\Big).
$$

Also, this map puts subtrees containing other leaves than y to subtrees containing other leaves than x. $\square$

We introduce a new notation $f'_T(v) = f_T(v) - f^*_T(v)$ to denote the number of subtrees of a tree T which contain $v \in V(T)$ and no other leaf.
Lemma$''$6.2 *Consider now T' and T'' from Lemma 6.2. If*

$$f_T(x) > f_T(y), \qquad f^*_T(x) > f^*_T(y)$$

and

$$f'_X(x) < f'_Y(y), \qquad f^*_X(x) < f^*_Y(y),$$

then $F^*(T'') > F^*(T')$.

Proof. Let N^* denote the number of subtrees of T which contain at least one leaf other than x, y and contain both x and y; and let N denote the number of subtrees of T which contain both x and y Let $c(T', x, y)$ $(c(T'', x, y))$ denote the number of subtrees of T' (T''), which contain both x, y, and in addition at least one more leaf. A simple bijective argument shows $c(T', x, y) = c(T'', x, y)$. Similarly, let $d(T', x, y)$ $(d(T'', x, y))$ denote the number of subtrees of T' (T''), which contain none of x, y, but has at least one leaf. Again, a simple bijective argument shows $d(T', x, y) = d(T'', x, y)$. We make a case analysis as $x \in, y \in$; $x \notin y \notin$; $x \in y \notin$; $x \notin y \in$, and use inclusion-exclusion in the last two cases:

$$\begin{aligned}
F^*(T') &= c(T', x, y) + d(T', x, y) \\
&+ f_X(x)(f^*_T(x) - N^*) + f^*_X(x)(f_T(x) - N) - f^*_X(x)(f^*_T(x) - N^*) \\
&+ f_Y(y)(f^*_T(y) - N^*) + f^*_Y(y)(f_T(y) - N) - f^*_Y(y)(f^*_T(y) - N^*); \\
F^*(T'') &= c(T'', x, y) + d(T'', x, y) \\
&+ f_Y(y)(f^*_T(x) - N^*) + f^*_Y(y)(f_T(x) - N) - f^*_Y(y)(f^*_T(x) - N^*) \\
&+ f_X(x)(f^*_T(y) - N^*) + f^*_X(x)(f_T(y) - N) - f^*_X(x)(f^*_T(y) - N^*).
\end{aligned}$$

Hence we have

$$\begin{aligned}
& F^*(T') - F^*(T'') \\
= \ & (f'_X(x) - f'_Y(y))(f^*_T(x) - f^*_T(y)) + (f^*_X(x) - f^*_Y(y))(f_T(x) - f_T(y)) < 0.
\end{aligned}$$

□

For the study of F^* and f^* on binary trees, we are going to label vertices and subtrees as described as follows. Note that this is different from the labelling in [10] and from the labelling in Section 7, (15). However, for rgood trees, *all three labellings are the same*, as we will see in Lemma 9.1.

If T is a rooted binary tree with root r, and r_1, r_2 are the children of r, then we will simply write T_1 for $T(r_1)$ and T_2 for $T(r_2)$. We assign the labels r_1 and r_2 according to the following rule: $h(T_2) \geq h(T_1)$, and $f_{T_2}(r_2) \geq f_{T_1}(r_1)$ in case equality holds in the first inequality. T_i will be rooted at r_i, $i = 1, 2$. We define recursively $T_{i_1 i_2 \ldots i_k 1}$ and $T_{i_1 i_2 \ldots i_k 2}$ to be the two rooted binary trees induced by the children of the root of $T_{i_1 i_2 \ldots i_k}$, when $T_{i_1 i_2 \ldots i_k}$ is not a single vertex, where $i_j \in \{1, 2\}$, $j = 1, 2, \ldots, k$. We assign the labels $r_{i_1 i_2 \ldots i_k 1}$ and $r_{i_1 i_2 \ldots i_k 2}$ according to the following rule:

$$h(T_{i_1 i_2 \ldots i_k 2}) \geq h(T_{i_1 i_2 \ldots i_k 1}),$$

and in the case of equality,

$$f_{T_{i_1 i_2 \dots i_k 2}}(r_{i_1 i_2 \dots i_k 2}) \geq f_{T_{i_1 i_2 \dots i_k 1}}(r_{i_1 i_2 \dots i_k 1}). \tag{22}$$

We complete the recursive definition by letting $r_{i_1 i_2 \dots i_k}$ be the root for $T_{i_1 i_2 \dots i_k}$.

The following observation is trivial and we leave the proof to the reader.

Lemma 9.1. *For rgood trees (T_1, v_1) and (T_2, v_2), the following are equivalent:*

$$f^*_{T_2}(v_2) > f^*_{T_1}(v_1), \qquad f'_{T_2}(v_2) > f'_{T_1}(v_1), \qquad f_{T_2}(v_2) > f_{T_1}(v_1), \qquad |V(T_2)| > |V(T_1)|.$$

Therefore, for subtrees of rgood trees, we can go back and forth with our different ways of labelling in [10] and in this present proof.

Lemma 9.2. *For any rooted binary tree T with root r,*

$$f^*_T(r) > f'_T(r) \qquad \text{and} \qquad f^*_T(r) > \tfrac{1}{2} f_T(r). \tag{23}$$

Proof. Assume T has m leaves. Since T is a rooted binary tree, it has $m-2$ non-leaf vertices.

Then $f^*_T(r) \geq 2^m - 1$, since different non-empty subsets of leaves, with r added, span different subtrees. On the other hand, $f'_T(r) \leq 2^{m-2}$, since the number of leafless subtrees containing r is at most the number of subsets of all non-leaf, non-root vertices.

The second inequality easily follows. $\square$

Lemma 9.3. *For any rooted binary tree (T, r), and the induced subtree of the son r_2, (T_2, r_2), we have*

$$f^*_T(r) \geq 2 f^*_{T_2}(r_2).$$

Proof. For every subtree of T_2 which contains r_2 and a leaf, we construct two subtrees of T containing r and a leaf. For the first, add to the tree the rr_2 edge, for the second, add to the tree the rr_1 and rr_2 edges. $\square$

Lemma 9.4. *For any rooted tree T with root r, and any $r' \in V(T)$ $(r' \neq r)$, consider the induced subtree $T' = T(r')$ rooted at r'. Then we have*

$$f^*_T(r) > f^*_{T'}(r'). \tag{24}$$

If T'' is obtained from T by deleting some vertices, but not r, then

$$f^*_T(r) > f^*_{T''}(r). \tag{25}$$

Proof. To prove (24) with an injection, extend with the $r'r$ path the subtrees of T' that we count. To prove (25), if l'' is a leaf of T'' but not a leaf of T with an injection, assign to l'' a leaf l of T, such that l'' separates r and l in T, and for every l'' and corresponding (distinct!) l fix the path ll''. Extend every subtree of T'' that we have to count, for all its new leaves, with the ll'' paths. This is an injection into the set of subtrees of T that we have to count. $\square$

Lemma 9.5. *For any rooted binary tree (T, r), we have*

$$f_T(r) \geq 2^{k+1} f^*_{T_{\underbrace{2\dots21}_{k\ 2's}}}(r_{\underbrace{2\dots21}_{k\ 2's}}). \tag{26}$$

Proof. By Lemma 9.3, we have

$$f_T^*(r) \geq 2f_{T_2}^*(r_2) \geq \ldots \geq 2^k f^*_{T_{\underbrace{2\ldots2}_{k\ 2's}}}(r_{\underbrace{2\ldots2}_{k\ 2's}}). \tag{27}$$

Also, Lemma 9.3 implies for any rooted binary tree (T,r) that $f_T^*(r) \geq 2f_{T_1}^*(r_1)$, and thus

$$f^*_{T_{\underbrace{2\ldots2}_{k\ 2's}}}(r_{\underbrace{2\ldots2}_{k\ 2's}}) \geq 2f^*_{T_{\underbrace{2\ldots2}_{k\ 2's}1}}(r_{\underbrace{2\ldots2}_{k\ 2's}1}). \tag{28}$$

Thus, (27), (28) and the fact that $f_T(r) \geq f_T^*(r)$ yield (26). $\square$

10 Proof to Theorem$''$ 3.4

Based on Lemmas above, Theorem$''$ 3.5 and Theorem$''$ 3.4 can be proved. We outline only the proof of Theorem$''$ 3.4. The reader has to read [10] parallel, since we highlight only the steps that require modification. The figures in [10] will help at reading. As in [10], we use superscripts on some inequalities to indicate their proofs.

Lemma 10.1. *Assume T is an optimal binary tree that maximizes $F^*(T)$. Assume that T is divided into two rooted subtrees T', T'' by the removal of an edge $v'v''$. Then, if for all $k \geq 1$ the inequalities*

$$f_{T'}(v') > f_{(T'')_{\underbrace{2\ldots2}_{k\ 2's}1}}(v''_{\underbrace{2\ldots2}_{k\ 2's}1}) \text{ and } f^*_{T'}(v') > f^*_{(T'')_{\underbrace{2\ldots2}_{k\ 2's}1}}(v''_{\underbrace{2\ldots2}_{k\ 2's}1}) \tag{29}$$

hold as far as vertex $v''_{\underbrace{2\ldots2}_{k\ 2's}1}$ exists, then T'' is rgood.

Note: We understand that (29) holds if $(T'')_{21}$ does not exist. Then $(T'')_2$ is a single vertex, and by (22) $(T'')_1$ is also a single vertex. Therefore T'' is rgood as Lemma 10.1 requires.

Proof. The proof goes by induction on $|V(T'')|$. The base case: if $|V(T'')| = 1$, then by definition, T'' is rgood. Now, suppose that Lemma 10.1 holds for any induced subtree in place of T'' with fewer vertices. We are going to show the following:

Claim 10.1. *$(T'')_1$ and $(T'')_2$ are rgood.*

Proof. We prove the statement for $(T'')_2$ and $(T'')_1$ in a different order than in [10]. Proof for $(T'')_2$. Consider T as being divided into $T''' = ((T'')_2, v''_2)$ and $T^* = (T' \cup (T'')_1 \cup \{v''\}, v'')$. Notice that for any $k \geq 1$

$$f_{T^*}(v'') > f_{T'}(v') >^{(29)} f_{(T'')_{\underbrace{2\ldots2}_{k+1\ 2's}1}}(v''_{\underbrace{2\ldots2}_{k+1\ 2's}1}) = f_{(T''')_{\underbrace{2\ldots2}_{k\ 2's}1}}(v''_{\underbrace{2\ldots2}_{k+1\ 2's}1}),$$

and

$$f^*_{T^*}(v'') >^{(25)} f^*_{T'}(v') >^{(29)} f^*_{(T'')_{\underbrace{2\ldots2}_{k+1\ 2's}1}}(v''_{\underbrace{2\ldots2}_{k+1\ 2's}1}) = f^*_{(T''')_{\underbrace{2\ldots2}_{k\ 2's}1}}(v''_{\underbrace{2\ldots2}_{k+1\ 2's}1}),$$

thus (29) holds for T^* and T'''. By hypothesis, it follows that $(T'')_2$ must be rgood.

Proof for $(T'')_1$. Since $(T'')_2$ is rgood and of height at least $h((T'')_1)$, then $(T'')_2$ contains a complete subtree of height $h(T'')_1 - 1$ and hence a subtree that is isomorphic to $(T'')_{\underbrace{12\ldots21}_{k\ 2's}}$. This will explain the middle inequalities in the next two displayed formulas.
Now consider T as being divided into $T''' = ((T'')_1, v_1'')$ and $T^* = (T' \cup (T'')_2 \cup \{v''\}, v'')$. We have for any $k \geq 1$,

$$f_{T^*}(v'') > f_{(T'')_2}(v_2'') > f_{(T'')_{\underbrace{12\ldots21}_{k\ 2's}}}(v''_{\underbrace{12\ldots21}_{k\ 2's}}) = f_{(T''')_{\underbrace{2\ldots21}_{k\ 2's}}}(v''_{\underbrace{12\ldots21}_{k\ 2's}}),$$

and

$$f^*_{T^*}(v'') >^{(25)} f^*_{(T'')_2}(v_2'') > f^*_{(T'')_{\underbrace{12\ldots21}_{k\ 2's}}}(v''_{\underbrace{12\ldots21}_{k\ 2's}}) = f^*_{(T''')_{\underbrace{2\ldots21}_{k\ 2's}}}(v''_{\underbrace{12\ldots21}_{k\ 2's}}),$$

thus (29) holds for T^* and T'''. By hypothesis, it follows that $(T'')_1$ is rgood. $\square$

After this point, since $((T'')_1, v_1'')$ and $((T'')_2, v_2'')$ are both rgood, we have that

$$f^*_{H_{i_1 i_2 \ldots i_k 2}}(v_{i_1 i_2 \ldots i_k 2}) \geq f^*_{H_{i_1 i_2 \ldots i_k 1}}(v_{i_1 i_2 \ldots i_k 1}),$$

$$f'_{H_{i_1 i_2 \ldots i_k 2}}(v_{i_1 i_2 \ldots i_k 2}) \geq f'_{H_{i_1 i_2 \ldots i_k 1}}(v_{i_1 i_2 \ldots i_k 1}),$$

$$f_{H_{i_1 i_2 \ldots i_k 2}}(v_{i_1 i_2 \ldots i_k 2}) \geq f_{H_{i_1 i_2 \ldots i_k 1}}(v_{i_1 i_2 \ldots i_k 1})$$

holds for $(H, v) = ((T'')_1, v_1'')$ or $(H, v) = ((T'')_2, v_2'')$.

Knowing that $(T'')_1$ and $(T'')_2$ are rgood, we return to the inductive step in the proof of Lemma 10.1. We consider the following cases: (i) $h((T'')_1) < h((T'')_2)$ and (ii) $h((T'')_1) = h((T'')_2)$. (Note that the third inequality $h((T'')_1) > h((T'')_2)$ is impossible by the rgoodness of $(T'')_1$ and $(T'')_2$ and (22)).

Case (i): $h((T'')_1) < h((T'')_2)$.
By Lemma 9.1, we also have $|V((T'')_2)| > |V((T'')_1)|$, $f^*_{(T'')_2}(v_2'') > f^*_{(T'')_1}(v_1'')$, and $f_{(T'')_2}(v_2'') > f_{(T'')_1}(v_1'')$.

Claim 10.2. *For any $k \geq 0$ such that $(T'')_{\underbrace{1\ldots1}_{k}}$ is not empty, we have*

$$|V((T'')_{\underbrace{1\ldots1}_{k}})| \geq |V((T'')_{\underbrace{22\ldots2}_{k+1}})|. \tag{30}$$

Proof. Apply the same induction as in [10] till (13) in [10], then we are in the position to apply Lemma$''$ 6.1 in the same setting as in [10].

Using the notation in Lemma$''$ 6.1, we have

$$a_i = f_{(T'')_{\underbrace{1\ldots12}_{l+1-i\ 1's}}}(v''_{\underbrace{1\ldots12}_{l+1-i\ 1's}}) + 1 \geq f_{(T'')_{\underbrace{22\ldots21}_{l+2-i\ 2's}}}(v''_{\underbrace{22\ldots21}_{l+2-i\ 2's}}) + 1 = b_i \tag{31}$$

exactly as in [10], and

$$a_i'' = f^*_{(T'')_{\underbrace{1\ldots12}_{l+1-i\ 1's}}}(v''_{\underbrace{1\ldots12}_{l+1-i\ 1's}}) + 1 \geq f^*_{(T'')_{\underbrace{22\ldots21}_{l+2-i\ 2's}}}(v''_{\underbrace{22\ldots21}_{l+2-i\ 2's}}) + 1 = b_i'' \tag{32}$$

for $i = 1, 2, \ldots, l$ by Lemma 9.1 and the rgoodness of proper induced rooted subtrees of T''. We also have

$$a_{l+1} = f_{T'}(v') + 1 > f_{(T'')_{21}}(v''_{21}) + 1 = b_{l+1}$$

as in [10] by (29) , and then Lemma 9.1 implies

$$a''_{l+1} = f^*_{T'}(v') + 1 > f^*_{(T'')_{21}}(v''_{21}) + 1 = b''_{l+1}.$$

From here, we obtain the conclusion of Lemma$''$ 6.1, which is exactly the first condition of Lemma$''$ 6.2 as well:

$$f_S(x) > f_S(y) \text{ and } f^*_S(x) > f^*_S(y).$$

Note that we have

$$f_X(x) = f_{(T'')\underbrace{1\ldots11}_{l+1}}(v''\underbrace{1\ldots11}_{l+1}) < f_{(T'')\underbrace{22\ldots22}_{l+2}}(v''\underbrace{22\ldots22}_{l+2}) = f_Y(y) \tag{33}$$

exactly as in [10], then (33) implies $f^*_X(x) < f^*_Y(y)$ and $f'_X(x) < f'_Y(y)$ (the second condition of Lemma$''$ 6.2) by Lemma 9.1 and the rgoodness of proper rooted subtrees of T''.

Thus, by Lemma$''$ 6.2, interchanging X and Y increases $F^*(T)$, contradicting the optimality of T. Hence (30) holds for $k = l + 1$, and we completed the induction proof. $\square$

With Claim 10.2, the same proof as in [10] shows that T'' is rgood. End of Case (i).

Case (ii): $h((T'')_1) = h((T'')_2)$.

Claim 10.3. *For any $k \geq 0$ such that $(T'')\underbrace{21\ldots1}_{k\ 1's}$ is not empty, we have*

$$|V((T'')\underbrace{21\ldots1}_{k\ 1's})| \geq |V((T'')\underbrace{12\ldots2}_{k\ 2's})| \tag{34}$$

Proof. Apply the same induction as in [10] till (20) in [10], then we are in the position to apply Lemma$''$ 6.1 in the same setting as in [10]. Using the notation in Lemma$''$ 6.1, we have

$$a_i = f_{(T'')\underbrace{21\ldots12}_{l+1-i\ 1's}}(v''\underbrace{21\ldots12}_{l+1-i\ 1's}) + 1 \geq f_{(T'')\underbrace{12\ldots21}_{l+1-i\ 2's}}(v''\underbrace{12\ldots21}_{l+1-i\ 2's}) + 1 = b_i \tag{35}$$

and

$$a''_i = f^*_{(T'')\underbrace{21\ldots12}_{l+1-i\ 1's}}(v''\underbrace{21\ldots12}_{l+1-i\ 1's}) + 1 \geq f^*_{(T'')\underbrace{12\ldots21}_{l+1-i\ 2's}}(v''\underbrace{12\ldots21}_{l+1-i\ 2's}) + 1 = b''_i \tag{36}$$

for $i = 1, 2, \ldots, l + 1$ by Lemma 9.1 and the rgoodness of proper induced subtrees of T''. In fact, strict inequality holds in (35) for $i = 1$ and therefore $a_1 > b_1$. From here, we obtain the conclusion of Lemma$''$ 6.1, which is exactly the first condition of Lemma$''$ 6.2 as well:

$$f_S(x) > f_S(y) \text{ and } f^*_S(x) > f^*_S(y).$$

Note that we have:

$$f_X(x) = f_{(T'')\underbrace{21\ldots11}_{l+1\ 1's}}(v''\underbrace{21\ldots11}_{l+1\ 1's}) < f_{(T'')\underbrace{12\ldots22}_{l+1\ 2's}}(v''\underbrace{12\ldots22}_{l+1\ 2's}) = f_Y(y) \tag{37}$$

exactly as in [10], then (37) implies $f^*_X(x) < f^*_Y(y)$ and $f'_X(x) < f'_Y(y)$ (the second condition of Lemma$''$ 6.2) by Lemma 9.1 and the rgoodness of proper rooted subtrees of T''.

Thus, by Lemma$''$ 6.2, interchanging X and Y increases $F^*(T)$, contradicting the optimality of T. Hence (34) holds for $k = l + 1$. Using induction, we proved Claim 10.3.

□

With Lemma 10.3, same proof as in [10] shows that T'' is rgood. End of Proof of Lemma 10.1. □

Lemma 10.2. *Consider an optimal tree T and its two rooted subtrees T' and T'' after an edge deletion, as in Lemma 10.1. If $|h(T'') - h(T')| \leq 1$, then T' and T'' both must be rgood.*

Note that if we choose a longest path P and choose (v', v'') as the closest to middle edge on P, we obtain such a T' and T''.

Proof. Assume without loss of generality that $f_{T''}(v'') \geq f_{T'}(v')$. Then, we have for $k \geq 1$ that

$$f_{T''}(v'') \geq f_{T'}(v') > f_{(T')\underbrace{2\ldots21}_{k\ 2's}}(v'\underbrace{2\ldots21}_{k\ 2's})$$

and

$$\begin{aligned} f^*_{T''}(v'') >^{(23)} \frac{1}{2} f_{T''}(v'') &\geq \frac{1}{2} f_{T'}(v') \geq^{(26)} 2^k f^*_{(T')\underbrace{2\ldots21}_{k\ 2's}}(v'\underbrace{2\ldots21}_{k\ 2's}) \\ &> f^*_{(T')\underbrace{2\ldots21}_{k\ 2's}}(v'\underbrace{2\ldots21}_{k\ 2's}). \end{aligned}$$

Thus condition (29) holds, T' is rgood.

On the one hand, since T' is rgood, T' must contain a complete rooted binary tree T^*, with the same root, of height at least $h(T') - 1 \geq h(T'') - 2$. On the other hand, $(T'')\underbrace{2\ldots21}_{k\ 2's}$ is of height at most $h(T'') - 2$ and is isomorphic to a subtree of T' (sharing the same root). Therefore

$$f_{T'}(v') \geq f_{(T'')\underbrace{2\ldots21}_{k\ 2's}} \text{ and } f^*_{T'}(v') \geq f^*_{(T'')\underbrace{2\ldots21}_{k\ 2's}} \tag{38}$$

for $k \geq 1$. In fact, (38) are always strict inequalities, since T' has some other vertices than those in the complete rooted binary tree with height $h(T') - 1$. So condition (29) holds, T'' is also rgood.

□

Lemma 10.3. *Consider an optimal tree T and its two rooted subtrees T' and T'' after an edge deletion, such that $|h(T'') - h(T')| \leq 1$. Assume that $f_{T''}(v'') \geq f_{T'}(v')$ (and also $f^*_{T''}(v'') \geq f^*_{T'}(v')$ by Lemma 9.1). Then T' is complete or $T^* = (T' \cup (T'')_1 \cup \{v''\}, v'')$ is rgood.*

Proof. Consider T as being divided into T^* and $(T'')_2$. The proof in [10] yields for any $k \geq 0$

$$f_{(T')\underbrace{2\ldots21}_{k\ 2's}}(v'_{\underbrace{2\ldots21}_{k\ 2's}}) < f_{(T'')_2}(v''_2). \tag{39}$$

Then Lemma 9.1 yields for any $k \geq 0$

$$f^*_{(T')\underbrace{2\ldots21}_{k\ 2's}}(v'_{\underbrace{2\ldots21}_{k\ 2's}}) < f^*_{(T'')_2}(v''_2). \tag{40}$$

Similarly, notice that $(T'')_1$ is rgood, and then for $k \geq 0$,

$$f_{(T'')_2}(v''_2) \geq f_{(T'')_1}(v''_1) > f_{(T'')_{11}}(v''_{11}) \geq f_{(T'')\underbrace{12\ldots21}_{k\ 2's}}(v''_{\underbrace{12\ldots21}_{k\ 2's}}) \tag{41}$$

and by Lemma 9.1

$$f^*_{(T'')_2}(v''_2) > f^*_{(T'')\underbrace{12\ldots21}_{k\ 2's}}(v''_{\underbrace{12\ldots21}_{k\ 2's}}). \tag{42}$$

Combining (39) and (41), we obtain that for any $k \geq 0$,

$$f_{(T'')_2}(v''_2) > \max\left(f_{(T')\underbrace{2\ldots21}_{k\ 2's}}(v'_{\underbrace{2\ldots21}_{k\ 2's}}), f_{(T'')\underbrace{12\ldots21}_{k\ 2's}}(v''_{\underbrace{12\ldots21}_{k\ 2's}})\right). \tag{43}$$

Since $(T^*)_2 = T'$ or $(T'')_1$, we have from (43) that

$$f_{(T'')_2}(v''_2) > f_{(T^*)\underbrace{2\ldots21}_{k+1\ 2's}}(r^*) \text{ for } k \geq 0,$$

where r^* is the root of $(T^*)_{\underbrace{2\ldots21}_{k+1\ 2's}}$.

Similarly, combining (40) and (42), we obtain that for any $k \geq 0$,

$$f^*_{(T'')_2}(v''_2) > \max\left(f^*_{(T')\underbrace{2\ldots21}_{k\ 2's}}(v'_{\underbrace{2\ldots21}_{k\ 2's}}), f^*_{(T'')\underbrace{12\ldots21}_{k\ 2's}}(v''_{\underbrace{12\ldots21}_{k\ 2's}})\right). \tag{44}$$

Since $(T^*)_2 = T'$ or $(T'')_1$, we have from (44) that

$$f^*_{(T'')_2}(v''_2) > f^*_{(T^*)\underbrace{2\ldots21}_{k+1\ 2's}}(r^*) \text{ for } k \geq 0, \tag{45}$$

where r^* is the root of $(T^*)_{\underbrace{2\ldots21}_{k+1\ 2's}}$.

So by (43) and (45) condition (29) holds, T^* is rgood by Lemma 10.1. □

With the above Lemmas, the proof of Theorem$''$ 3.4 is almost exactly the same as in [10], one has to change only (34), (36), (37), and the formula after (37). (For each of these formulas, if the condition holds for f in [10], then it holds for both f and f^*.)

Acknowledgement We are indebted to (i) István Miklós for letting us know Knudsen's results, (ii) to A. Kostochka, J. Skokan, and D. B. West for conversations on the topic of this paper during his visit at UIUC, (iii) to Éva Czabarka for several important suggestions after her reading of a draft of this paper.

References

[1] Ádám, A., The centrality of vertices in trees, *Studia Sci. Math. Hung.* **9** (1974), 285–303.

[2] Aho, A. V.; Sloane, N. J. A., Some doubly exponential sequences, *Fibonacci Quart.* **11** (4) (1973), 429–437.

[3] Dobrynin, A.A., Entringer, R., Gutman, I., Wiener index of trees: Theory and applications, *Acta Appl. Math.* **66** (3) (2001), 211–249.

[4] Entringer, R.C., Jackson, D.E., Snyder, D.A., Distance in graphs, *Czechoslovak Math. J.* **26** (101) (1976), 283–296.

[5] Fischermann, M., Hoffmann, A., Rautenbach, D., Székely, L.A., Volkmann, L., Wiener index versus maximum degree in trees, *Discrete Appl. Math.* **122** (1–3) (2002), 127–137.

[6] Jelen, F., Triesch, E., Superdominance order and distance of trees with bounded maximum degree, *Discrete Appl. Math.* **125** (2–3) (2003), 225–233.

[7] Jordan, C., Sur les assemblages de lignes, *J. Reine Angew. Math.* **70** (1869), 185–190.

[8] Knudsen, B., Optimal multiple parsimony alignment with affine gap cost using a phylogenetic tree, Lecture Notes in Bioinformatics 2812, Springer Verlag, 2003, 433–446.

[9] Lovász, L., *Combinatorial Problems and Exercises*, 2nd ed., North–Holland Publishing Co., Amsterdam, 1993.

[10] Székely, L.A., Wang, H., Binary trees with the largest number of subtrees, submitted.

[11] Zelinka, H., Medians and peripherians of trees, *Arch. Math.* (Brno) **4** (1968), 87–95.