

LEAF STRIPPING ON UNIFORM ATTACHMENT TREES

LOUIGI ADDARIO-BERRY*, ANNA BRANDENBERGER^o, SIMON BRIEND[†], NICOLAS BROUTIN[‡],
AND GÁBOR LUGOSI[§]

ABSTRACT. In this note we analyze the performance of a simple root-finding algorithm in uniform attachment trees. The *leaf-stripping* algorithm recursively removes all leaves of the tree for a carefully chosen number of rounds. We show that, with probability $1 - \varepsilon$, the set of remaining vertices contains the root and has a size only depending on ε but not on the size of the tree.

1. INTRODUCTION

The problem of localizing the root vertex in various random tree models goes back to Haigh [11] and recently has been intensively studied; see [3, 5–7, 12, 14, 15, 18, 19].

In this paper we are interested in root-finding algorithms for random recursive trees (also known as uniform attachment trees). A random recursive tree T_n of size n is defined as follows. T_1 consists of a single vertex with label 1. Then, we inductively obtain T_{i+1} from T_i by connecting a new vertex labelled $i + 1$ to an independently and uniformly chosen vertex $v \in [n]$. An equivalent way of viewing this tree is as uniformly chosen *increasing tree* of size n : these are rooted trees with vertex set $[n]$ whose labels increase along any path from the root to a leaf (see, e.g., [2] or [10, Example II.18]).

Write $\mathcal{T}_n$ for the set of all unrooted trees with vertex set $[n]$. A *root-finding algorithm* for trees of size n is a function $A : \mathcal{T}_n \rightarrow 2^{[n]}$ which is invariant to relabeling: if $T, T' \in \mathcal{T}_n$ and T' can be obtained from T by a permutation $\sigma : [n] \rightarrow [n]$ of vertex labels, then $A(T') = \sigma(A(T)) := \{\sigma(v), v \in A(T)\}$. Its *error probability* is $\mathbf{P}\{1 \notin A(T_n)\}$, where we abuse notation and write $A(T_n)$ to mean $A(T)$ where T is the unrooted tree corresponding to T_n . The *size* of A is $\max(|A(T)| : T \in \mathcal{T}_n)$. The set $A(T)$ is often called a *confidence set* for the root vertex.

The paper [5] showed that there exist root finding algorithms with low error probability whose size is independent of n ; more precisely, there exists $c > 0$ such that for all $\varepsilon \in (0, 1/e)$, for all $n \geq 1$, there exists a root-finding algorithm A for trees of size n with error at most ε and size at most $\exp(c \log(1/\varepsilon) / \log \log(1/\varepsilon))$. This result is proved via a rather delicate analysis of the *maximum likelihood* root-finding algorithm, which first lists nodes in decreasing order of their likelihood of being the root (given the shape of the tree but not its label information), then returns the k most likely nodes, for a suitable value of k . In [5] a simpler method was also analyzed that ranks vertices according to their *Jordan centrality*, defined by the size of the largest component of the forest obtained by removing the vertex from the tree. This method is guaranteed to output a confidence set whose size is at most $(11/\varepsilon) \log(1/\varepsilon)$ and contains the root vertex with probability $1 - \varepsilon$.

The purpose of this paper is to show that a different algorithm which we term *leaf-stripping*, which is very simple to describe and easy to implement, also yields root finding algorithms with small error probability and with size independent of n . The algorithm proceeds as follows. For $n \in \mathbf{N}$ define

$$m_n = \left\lceil e \log n - \frac{3}{2} \log \log(n+1) \right\rceil; \quad (1)$$

then for $k \in \mathbf{N}$, define the leaf-stripping algorithm $R_k(T_n)$ as follows.

Algorithm 1. Let $T^{(0)} = T_n$. For $1 \leq i \leq m_n - k$, let $T^{(i)}$ be the tree obtained from $T^{(i-1)}$ by removing all its leaves. Then let $R_k(T_n)$ be the vertex set of $T^{(m_n - k)}$.

*DEPARTMENT OF MATHEMATICS AND STATISTICS, MCGILL UNIVERSITY LOUIGI.ADDARIO@MCGILL.CA

^oDEPARTMENT OF MATHEMATICS, MIT ABRANDE@MIT.EDU

[†]DEPARTMENT OF MATHEMATICS, UNIDISTANCE SUISSE SIMON.BRIEND@UNIDISTANCE.CH

[‡]LABORATOIRE DE PROBABILITÉS, STATISTIQUE ET MODÉLISATION SORBONNE UNIVERSITÉ NICOLAS.BROUTIN@LPSM.PARIS

[§]DEPARTMENT OF ECONOMICS AND BUSINESS, POMPEU FABRA UNIVERSITY, BARCELONA, SPAIN; ICREA, PG. LLUÍS COMPANYYS 23, 08010 BARCELONA, SPAIN; BARCELONA SCHOOL OF ECONOMICS GABOR.LUGOSI@GMAIL.COM

We have defined the algorithm only for T_n since that is the only tree we apply it to. Note that the algorithm succeeds if and only if there are at least two edge-disjoint paths of length at least $m_n - k$ starting from vertex 1 in T_n . Equivalently, $1 \in R_k(T_n)$ if and only if 1 has at least two children u, v in T_n such that the subtrees rooted at u and v have height at least $m_n - k - 1$. This observation allows us to explain the role of the constant m_n ; it is known from [1] that the expected height of T_n of size n is $m_n + O(1)$, and that the height is exponentially concentrated around m_n . Therefore, stripping leaves m_n times will erase most or all of the tree. We instead strip $m_n - k$ times for a suitably chosen $k = k_\varepsilon$, to ensure that on one hand, the resulting set is not too large, and on the other hand, the root is not likely to be removed. Our main results state that leaf-stripping yields a family of root-finding algorithms for which the size and the error probability are polynomially related.

Theorem 1. *There exist $c, \gamma > 0$ and $c' > 0$ such that for all $\varepsilon \in (0, 1)$, setting $k = \lceil c \log(2/\varepsilon) \rceil$, for all n sufficiently large, $R_k(T_n)$ satisfies*

$$\mathbf{P}\{1 \in R_k(T_n) \text{ and } |R_k(T_n)| \leq \varepsilon^{-\gamma}\} \geq 1 - \varepsilon.$$

Theorem 1 shows that the performance of the leaf-stripping algorithm does not deteriorate as the size n of the tree grows and the size of the confidence set $R_k(T_n)$ may be bounded by a function of the probability of error only. Note however, that the dependence on the probability of error is inferior to that of the maximum likelihood mentioned above. In particular, while ranking vertices by their likelihood of being the root produces a confidence set that is smaller than any power of $1/\varepsilon$, Theorem 1 only implies a bound that is polynomial in $1/\varepsilon$. Theorem 2 below implies that this polynomial dependence is necessary (though we do not claim optimality of the obtained exponents).

Theorem 2. *There exists $\gamma' > 0$ such that for all $\varepsilon > 0$ sufficiently small, for all n sufficiently large, for all $k \in \mathbf{N}$ for which $\mathbf{P}\{1 \in R_k(T_n)\} \geq 1 - \varepsilon$, we also have $\mathbf{P}\{|R_k(T_n)| \geq \varepsilon^{-\gamma'}\} \geq 1 - \varepsilon$.*

Remarks.

- In Theorem 1 the size $|R_k(T_n)|$ is not deterministically bounded, but we may easily modify the leaf-stripping algorithm to obtain an algorithm for which both the size and error probability are bounded. Specifically, consider the algorithm R'_k defined by setting $R'_k(T_n) = R_k(T_n)$ if $|R_k(T_n)| \leq \varepsilon^{-\gamma}$ and setting $R'_k(T_n) = \emptyset$ otherwise. Then R'_k has size at most $\varepsilon^{-\gamma}$ and error probability less than ε .
- Sreedharan, Magner, Grama, and Szpankowski [20] address the problem of (partially) recovering the labeling of a Barabási-Albert preferential attachment graph. The first step of the procedure analyzed in [20] is a “peeling” algorithm whose aim is to recover the root. When the graph is a tree, peeling is equivalent to the leaf-stripping method analyzed here. In [20] (Lemma 6.1 in the supplementary material) it is claimed that peeling deterministically finds the root vertex, which is not the case. In this note we offer an analysis of the peeling algorithm for uniform attachment trees. The analogous problem for preferential attachment trees – and more generally for preferential attachment graphs – remains a challenge for future research.
- Navlakha and Kingsford [16] proposed a general framework for recovering the past of growing networks from a present-day snapshot. As a general principle, they propose *greedy likelihood* for root reconstruction. Greedy likelihood, at each step, deletes the vertex which is most likely to have been added most recently, breaking ties uniformly at random. In a uniform attachment tree, greedy likelihood deletes a uniformly random leaf at each step. This method is closely related to but not equivalent to leaf stripping. We conjecture that greedy likelihood has similar performance to that of leaf stripping, though we do not see an easy way to extend the analysis of this paper to an analysis of the greedy likelihood algorithm.
- *Random recursive dags* [8] are growing random graphs $(D_i)_{i \geq 1}$ that generalize random recursive trees. In a random recursive k -dag, for each $i \geq 1$, vertex $i + 1$ samples a set of k parents uniformly at random, with replacement, from among the integers $\{1, \dots, k\}$. It was suggested by an anonymous referee that a modification of the leaf-stripping algorithm may be effective for root finding in random recursive k -dags. In fact, the peeling algorithm of [20] is a natural generalization of leaf-stripping to dags. We do not see how to straightforwardly generalize the arguments of this work to the setting of dags — the information available in the literature on path lengths in random recursive k -dags [4] does not seem strong enough for our techniques to directly apply — but we view this as an interesting direction for future research.

1.1. Notation. Given a rooted tree t and a node $v \in t$, we denote its distance from the root by $\text{ht}(v)$. We further denote the height of t by $\text{ht}(t) := \max_{v \in t} \text{ht}(v)$; the subtree of t rooted at v by $t^{v, \downarrow}$ (defined as the subtree containing all vertices $u \in t$ for which the path between u and the root contains v); and the parent of v by $\text{par}(v)$. Some more general notation includes: for any set A , let $A^0 = \emptyset$. The set of positive integers is denoted by $\mathbf{N}$, and we write $[n] = \{1, \dots, n\}$ for any $n \in \mathbf{N}$. We omit ceilings and floors for legibility whenever possible.

2. PRELIMINARIES

2.1. Embedding in the Ulam–Harris tree. We introduce another representation for increasing trees that relies on the *Ulam–Harris tree* $\mathcal{U}$ [17]. This is the infinite ordered rooted tree with node set $\bigcup_{j=0}^{\infty} \mathbf{N}^j$ (root $\emptyset$ and nodes on level j of the form $n_1 \cdots n_j$) and edges from $n_1 \cdots n_j$ to $n_1 \cdots n_j n$ for all $j \geq 0$ and $n \in \mathbf{N}$; see Figure 1.

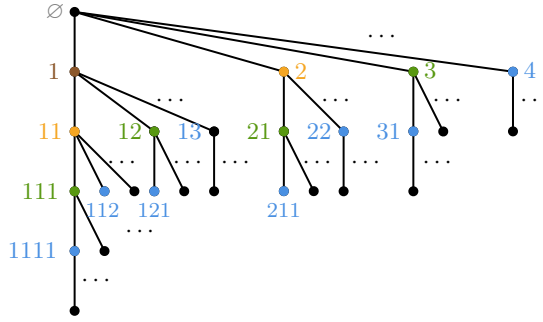


Figure 1. The Ulam–Harris tree $\mathcal{U}$, with zones 1 through 4 in different colours.

An ordered rooted tree can be canonically embedded in $\mathcal{U}$. Similarly, an increasing tree can be embedded in $\mathcal{U}$ by ordering vertices according to their labels: given an increasing tree t on $[n]$, we define a map $\varphi = \varphi_t : [n] \rightarrow \mathcal{U}$ by

$$\varphi(1) = \emptyset, \tag{2a}$$

$$\varphi(v) = n_1 \cdots n_j n_{j+1} \implies \varphi^{-1}(n_1 \cdots n_j) = \text{par}(v) < v, \tag{2b}$$

$$\varphi(v) = n_1 \cdots n_j n, \quad n > 1 \implies \varphi^{-1}(n_1 \cdots n_j(n-1)) < v. \tag{2c}$$

Condition (2a) ensures that the root of t is mapped to the root of $\mathcal{U}$; (2b) ensures that parent-child relations in t are preserved in $\mathcal{U}$, which in particular implies that $\text{ht}(\varphi(v)) = \text{ht}(v)$ for all $v \in t$; and (2c) ensures that for any node in $\mathcal{U}$, its left sibling is a vertex in t with a smaller label. Then $\{\varphi(i)\}_{i \in [n]}$ is a size- n subtree of $\mathcal{U}$, and clearly t can be recovered from the map φ ; we sometimes refer to t as the increasing tree encoded by φ . See Figure 2 for an example increasing tree and its embedding into $\mathcal{U}$.

We define the *zone* of a node $u = n_1 \cdots n_j \in \mathcal{U}$, $u \neq \emptyset$, to be $z(u) = n_1 + \cdots + n_j$. A node being in zone $z+1$ corresponds to it being either the first child of a node of zone z or the next sibling of a node in zone z . In particular, the number of nodes of $\mathcal{U}$ in zone z is $|\{u \in \mathcal{U} : z(u) = z\}| = 2^{z-1}$. See Figure 1 for an illustration of zones 1–4 of the Ulam–Harris tree.

We now embed the random recursive tree T_n on $[n]$ into $\mathcal{U}$ via φ_{T_n} . This embedding helps us analyze the sizes of subtrees. Indeed, note that due to the Pólya urn structure of the subtree sizes of T_n , we have that for vertices $v \in [n]$ in zone $z(\varphi_{T_n}(v)) = z$,

$$|T_n^{v, \downarrow}| \stackrel{d}{=} \lfloor \cdots \lfloor nU_1 \rfloor U_2 \rfloor \cdots U_z \rfloor, \tag{3}$$

where $U_1, \dots, U_z$ are i.i.d. Uniform $[0, 1]$ random variables. Indeed, for zone 1, the unique vertex with $z(\varphi_{T_n}(v)) = 1$ is $v = 2$. Since the two sets of vertices that respectively connect to vertex 1 and vertex 2 have sizes distributed according to a Pólya urn with two colours, we have $|T_n^{2, \downarrow}| \stackrel{d}{=} \lfloor nU \rfloor$ (see for instance [13, p. 177]). The general result follows straightforwardly by induction using the fact that $1 - U \stackrel{d}{=} U$.

In particular, (3) implies that the subtree sizes of two nodes in the same zone have the same distribution.

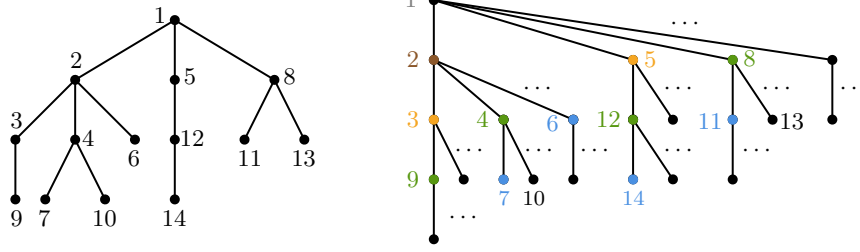


Figure 2. An instance of an increasing tree on $\{1, \dots, 14\}$ and its embedding via φ into the Ulam–Harris tree.

2.2. Tail bounds on the height of random recursive trees. In order to prove our main result, we use exponential tail bounds on $\text{ht}(T_n)$, see e.g., [1] or [9, Theorem 6.32], which we state as follows: there exist $\alpha > 0$ and $0 < \alpha' < 1/(2e)$ such that for all $n \in \mathbf{N}$ sufficiently large and for all $k \in \mathbf{N}$,

$$\mathbf{P}\{|\text{ht}(T_n) - m_n| \geq k\} \leq \alpha e^{-\alpha' k}. \quad (4)$$

3. PROOF OF THEOREM 1

Recall from (1) that $m_n = \lceil e \log n - \frac{3}{2} \log \log n \rceil$. Recall that $R_k(T_n)$ is the vertex set of the tree obtained from T_n after performing $m_n - k$ rounds of leaf stripping. For simplicity, we write $R_k \equiv R_k(T_n)$.

3.1. Detection of the root. First, we need to make sure that leaf stripping will “detect” vertex 1, that is, we show that $\mathbf{P}\{1 \in R_k\} \geq 1 - \varepsilon$.

Lemma 3. *There exist $c, c' > 0$ such that for all $\varepsilon > 0$, for all $n \in \mathbf{N}$, if $k \geq c \log(c'/\varepsilon)$, then $\mathbf{P}\{1 \notin R_k\} \leq \varepsilon$.*

Proof. Recall that $T_n^{2,\downarrow}$ is the subtree of vertex 2, and let $T_n^{2,\uparrow}$ be the tree T_n with $T_n^{2,\downarrow}$ removed, i.e., the subtree of vertex 1 excluding vertex 2 and its descendants. It is straightforward to see that, conditionally given their sizes, both $T_n^{2,\uparrow}$ and $T_n^{2,\downarrow}$ are distributed as random recursive trees and are (conditionally) independent. Then, we have the following lower and upper bounds:

$$\mathbf{P}\{1 \notin R_k\} \geq \mathbf{P}\{\text{ht}(T_n^{2,\uparrow}) < m_n - k\} \quad (5)$$

and

$$\begin{aligned} \mathbf{P}\{1 \notin R_k\} &\leq \mathbf{P}\{\text{ht}(T_n^{2,\uparrow}) < m_n - k \text{ or } \text{ht}(T_n^{2,\downarrow}) < m_n - k\} \\ &\leq 2\mathbf{P}\{\text{ht}(T_n^{2,\uparrow}) < m_n - k\}, \end{aligned} \quad (6)$$

where the second inequality comes from a union bound, and the fact that $T_n^{2,\uparrow}$ and $T_n^{2,\downarrow}$ are identically distributed when seen as rooted unlabelled trees. Therefore

$$\frac{\mathbf{P}\{1 \notin R_k\}}{\mathbf{P}\{\text{ht}(T_n^{2,\uparrow}) < m_n - k\}} \in [1, 2]. \quad (7)$$

From this we see that controlling (6), that is, controlling $\mathbf{P}\{\text{ht}(T_n^{2,\uparrow}) < m_n - k\}$, is sufficient to prove the lemma. From (3) we have that $|T_n^{2,\uparrow}| = n - |T_n^{2,\downarrow}| \stackrel{d}{=} \lfloor nU \rfloor \stackrel{d}{=} \text{Unif}(1, \dots, n-1)$ and since $T_n^{2,\uparrow}$ is distributed as a random recursive tree conditioned on its size,

$$\begin{aligned} \mathbf{P}\{\text{ht}(T_n^{2,\uparrow}) < m_n - k\} &= \sum_{j=1}^{n-1} \frac{1}{n-1} \mathbf{P}\{\text{ht}(T_j) < m_n - k\} \\ &\leq 1/e^{k/3} + \frac{1}{n-1} \sum_{j=n/e^{k/3}}^n \mathbf{P}\{\text{ht}(T_j) \leq m_n - k\}. \end{aligned}$$

For $j \geq n/e^{k/3}$ we have $|m_n - m_j - e \log(n/j)| \leq 1 + o(1)$; so, using the upper tail bound from (4), we obtain that

$$\mathbf{P}\{\text{ht}(T_j) \leq m_n - k\} \leq \mathbf{P}\{\text{ht}(T_j) \leq m_j + 2 - k(1 - e/3)\} \leq \alpha e^{-\alpha' k} \quad (8)$$

for some $\alpha, \alpha' > 0$. Thus,

$$\mathbf{P}\{\text{ht}(T_n^{2,\uparrow}) < m_n - k\} \leq e^{-k/3} + \alpha e^{-\alpha'k}$$

and therefore

$$\mathbf{P}\{1 \notin R_k\} \leq c'e^{-\alpha'k}$$

for some $0 < c' \leq 4\alpha$. The result follows by taking $k = c \log(c'/\varepsilon)$, where $c = 1/\alpha'$. $\square$

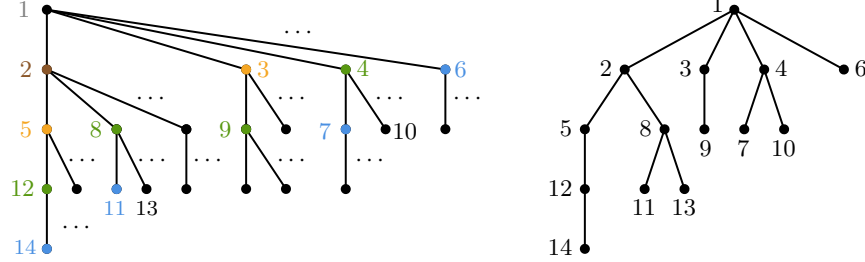


Figure 3. Image of the tree on $\{1, \dots, 14\}$ from Figure 2 under the tree flipping involution $\ell^{-1} \circ f_2 \circ \ell$, i.e., flipping up to zone $z = 2$.

3.2. Size of the confidence set. We must now prove that $|R_k|$ is small, with high probability.

Lemma 4. *For any c, c' as in Lemma 3, there exists $\gamma > 0$ such that for all $\varepsilon > 0$, for all $n \in \mathbf{N}$, if $k = c \log(c'/\varepsilon)$, we have $\mathbf{P}\{|R_k| > \varepsilon^{-\gamma}\} \leq \varepsilon$.*

Proof. We use the embedding of T_n into $\mathcal{U}$ by $\varphi = \varphi_{T_n}$; recall from (3) that under this embedding, if for $v \in T_n$, $z(\varphi_{T_n}(v)) = z$, then $|T_n^{v,\downarrow}| \stackrel{d}{=} [\dots [nU_1] \dots U_z]$.

Let $S = \{v \in [n] : z(\varphi_{T_n}(v)) = 4k, \text{ht}(T_n^{v,\downarrow}) \geq m_n - k\}$ be the set of vertices of T_n in zone $4k$ with subtrees of height $\geq m_n - k$. Note that if S is empty, then R_k only includes vertices in zones $1, \dots, 4k - 1$, so

$$|R_k| = \sum_{z=1}^{4k-1} |\{v \in [n] : z(\varphi_{T_n}(v)) = z\}| \leq 1 + \sum_{z=1}^{4k-1} 2^{z-1} = 2^{4k-1}, \quad (9)$$

and $k = c \log(c'/\varepsilon)$ suffices for $|R_k| = \varepsilon^{-\gamma}$, where $\gamma \geq 4c \log 2 > 8e \log 2$.

On the other hand, if S is non-empty, then since the height of T_n is at least $\text{ht}(v) + m_n - k$ for all $v \in S$, we have

$$\mathbf{P}\{\text{ht}(T_n) \geq m_n + k \mid S \neq \emptyset\} \geq \mathbf{P}\{\exists v \in S : \text{ht}(\varphi_{T_n}(v)) \geq 2k \mid S \neq \emptyset\}. \quad (10)$$

The heart of the proof then lies in the following claim:

$$\mathbf{P}\{\exists v \in S : \text{ht}(\varphi_{T_n}(v)) \geq 2k \mid S \neq \emptyset\} \geq \frac{1}{2}. \quad (11)$$

We complete the proof that $|R_k|$ is likely to be small before proving (11). Using (11), we have the following:

$$\begin{aligned} \mathbf{P}\{|R_k| \geq 2^{4k}\} &\leq \mathbf{P}\{S \neq \emptyset\} \\ &\leq 2\mathbf{P}\{\text{ht}(T_n) \geq m_n + k \mid S \neq \emptyset\} \cdot \mathbf{P}\{S \neq \emptyset\} \\ &\leq 2\mathbf{P}\{\text{ht}(T_n) \geq m_n + k\} \\ &\leq c'e^{-k/c}, \end{aligned}$$

where the first inequality is due to (9), the second is by multiplying by $1 \leq 2\mathbf{P}\{\text{ht}(T_n) \geq m_n + k \mid S \neq \emptyset\}$ using (10) and (11), and the fourth is due to the lower tail bound from (4), with c and c' as in Lemma 3. Therefore, assuming (11) we have that $\mathbf{P}\{|R_k| > \varepsilon^{-\gamma}\} \leq c'e^{-k/c} = \varepsilon$, which proves the lemma. $\square$

In order to prove (11), we define an involution on increasing trees which, intuitively, flips vertices in zone $4k$ at large heights $\text{ht}(\varphi_{T_n}(v))$ to lower heights, while preserving their subtree structure. See Figure 3 for an illustration of the involution applied to the tree from Figure 2 (where for clarity we illustrate it flipping vertices in zone 2 rather than in a zone $4k$ for $k \in \mathbf{N}$); note how the heights of vertices 3 and 5 are swapped, while their subtrees are preserved. We formalize the involution below.

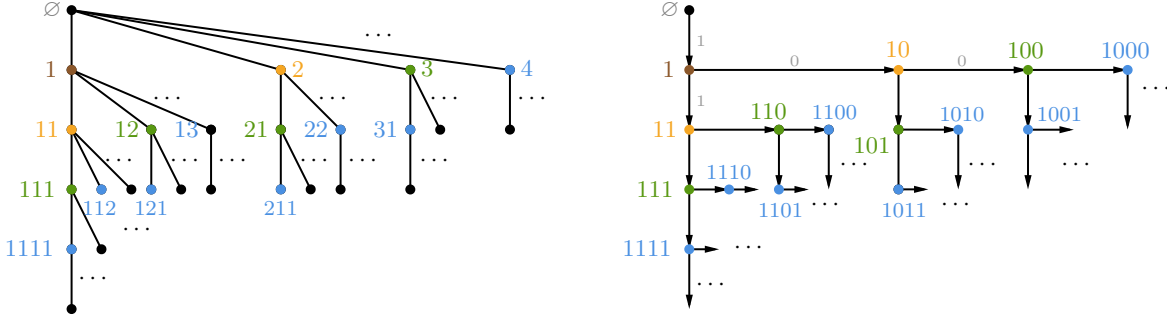


Figure 4. The Ulam–Harris tree (left) and its corresponding child-sibling binary tree (right) given by the bijection ℓ in (12), both with zones 1 through 4 illustrated in different colours.

We first define the child-sibling bijection $\ell : \mathcal{U} \mapsto \bigcup_{k=0}^{\infty} \{0, 1\}^k$ by $\ell(\emptyset) := \emptyset$ and, for $n_1, \dots, n_j \in \mathbb{N}^j$,

$$\ell(n_1 \cdots n_j) = 10^{n_1-1} 10^{n_2-1} \cdots 10^{n_j-1}. \quad (12)$$

This corresponds to the classic bijection between an ordered rooted tree and its child-sibling binary tree representation: ℓ relabels each node according to its identifying path in the binary tree. For instance, $\ell(1) = 1$ and $\ell(12) = 110$; see Figure 4. Note that under the image of ℓ , all non-root nodes in $\mathcal{U}$ have first coordinate in the child-sibling bijection equal to 1. Also, note that the zone $z(v)$ of any $v \in \mathcal{U} \setminus \{\emptyset\}$ is the length of $\ell(v)$.

For each $j \geq 2$, we now define the tree flipping involution which “exchanges the roles of child and sibling for all nodes in the first j zones in the child-sibling encoding, and preserves the subtrees of nodes in zone j ”. Formally, define $f_j : \bigcup_{k=0}^{\infty} \{0, 1\}^k \rightarrow \bigcup_{k=0}^{\infty} \{0, 1\}^k$ by $f_j(\emptyset) = \emptyset$, $f_j(1) = 1$, and

$$f_j(i_1 i_2 \cdots i_{j-1} i_j i_{j+1} \cdots i_k) = \overline{i_1 i_2 \cdots i_{j-1} i_j i_{j+1} \cdots i_k}, \quad (13)$$

where $\bar{i} = 1 - i$ for $i \in \{0, 1\}$. See Figure 5 for an illustration for $j = 3$.

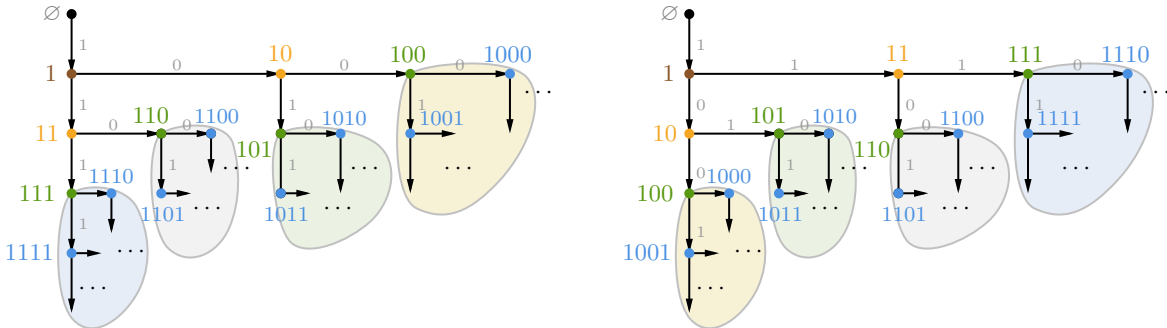


Figure 5. Illustration of f_3 : the second and third coordinates are flipped for every node. Notice that for each node in zone 3, its subtree remains identical under the map.

Then, the final involution $b : \mathcal{U} \rightarrow \mathcal{U}$ that we consider is

$$b := \ell^{-1} \circ f_{4k} \circ \ell, \quad (14)$$

and, given an increasing tree t on $[n]$, we define a function $\bar{\varphi} = \bar{\varphi}_t : [n] \rightarrow \mathcal{U}$ by

$$\overline{\varphi}_t(v) = b(\varphi_t(v)). \quad (15)$$

The following lemma ensures that we indeed obtain an involution between increasing trees. See Figure 3 for an illustration of the involution applied to the tree from Figure 2, with f_2 rather than f_{4k} for clarity.

Lemma 5. *For t an increasing tree on $[n]$, the map $\overline{\varphi}_t$ encodes an increasing tree on $[n]$.*

Proof. We check the conditions (2a), (2b) and (2c) for $\bar{\varphi} = \bar{\varphi}_t$. Since $b(\emptyset) = \emptyset$, (2a) is satisfied, and since $b(u) \neq \emptyset$ for $u \neq \emptyset$, $1 < b(\varphi(v))$ for all $1 \neq v \in [n]$. Then, for (2b) and (2c), it suffices to show that, for any $w \in [n-1]$ with $\bar{\varphi}(w) = n_1 \cdots n_j$ for some $j \geq 1$, for any $v, v' \in [n]$,

- i) if $\bar{\varphi}(v) = n_1 \cdots n_j 1$ then $w < v$;
- ii) if $\bar{\varphi}(v') = n_1 \cdots n_{j-1}(n_j + 1)$ then $w < v'$.

Note that by definition, $\ell(\bar{\varphi}(u)) = f_{4k}(\ell(\varphi(u)))$ for any $u \in [n]$. First, suppose $n_1 + \cdots + n_j + 1 \leq 4k$. If $v, v' \in [n]$ satisfy (2b) and (2c), then since

$$\ell(\bar{\varphi}(w)) = 10^{n_1-1} \cdots 10^{n_j-1}, \quad \ell(\bar{\varphi}(v)) = 10^{n_1-1} \cdots 10^{n_j-1} 1 \quad \text{and} \quad \ell(\bar{\varphi}(v')) = 10^{n_1-1} \cdots 10^{n_j-1} 0,$$

we have

$$\begin{aligned} \ell(\varphi(w)) &= 11^{n_1-1} 01^{n_2-1} \cdots 01^{n_j-1}, \\ \ell(\varphi(v)) &= 11^{n_1-1} 01^{n_2-1} \cdots 01^{n_j-1} 0 = \ell(\varphi(w)) 0, \text{ and} \\ \ell(\varphi(v')) &= 11^{n_1-1} 01^{n_2-1} \cdots 01^{n_j-1} 1 = \ell(\varphi(w)) 1. \end{aligned}$$

That is, $\varphi(v)$ and $\varphi(v')$ are respectively the right sibling and the first child of $\varphi(w)$ in $\mathcal{U}$. Therefore, by (2c) and (2b) for φ , we have $w < v$ and $w < v'$, as required.

Now, if $n_1 + \cdots + n_j + 1 > 4k$, by expanding analogously, we have $\ell(\varphi(v)) = \ell(\varphi(w)) 1$ and $\ell(\varphi(v')) = \ell(\varphi(w)) 0$. Then, $\varphi(v)$ and $\varphi(v')$ are respectively the first child and the right sibling of $\varphi(w)$ in $\mathcal{U}$, so $w < v, v'$. $\square$

We can now apply this function to the random recursive tree T_n . We denote by $\bar{T}_n$ the increasing tree encoded by $\bar{\varphi} := \bar{\varphi}_{T_n}$. Let $\bar{S} = \{v \in [n] : z(\bar{\varphi}(v)) = 4k, \text{ ht}(\bar{T}_n^{v, \downarrow}) \geq m_n - k\}$ be the set of vertices in zone $4k$ that have tall subtrees after applying $\bar{\varphi}$. The involution $\bar{\varphi}$ has the following properties:

Lemma 6. *Let T_n be a random recursive tree and write $\varphi = \varphi_{T_n}$. Then, the random tree $\bar{T}_n$ encoded by the map $\bar{\varphi} := \bar{\varphi}_{T_n}$, and the set $\bar{S}$ satisfy the following properties:*

- i) *For all $v \in [n]$, $z(\bar{\varphi}(v)) = z(\varphi(v))$, i.e., the zone of all vertices stays the same,*
- ii) *For all $v \in [n]$ with $z(\varphi(v)) = 4k$, $\bar{T}_n^{v, \downarrow} = T_n^{v, \downarrow}$, and $S = \bar{S}$.*
- iii) *For all $v \in [n]$ with $z(\varphi(v)) = z \in [4k]$, $\text{ht}(\varphi(v)) + \text{ht}(\bar{\varphi}(v)) = z + 1$. In particular, if $z(\varphi(v)) = 4k$ and $\text{ht}(\varphi(v)) \leq 2k + 1$ then $\text{ht}(\bar{\varphi}(v)) \geq 2k$.*

Proof. For (i), recall that for $v = n_1 \cdots n_j \in \mathcal{U}$, $z(v) = n_1 + \cdots + n_j$. Also, $\ell(v)$ is a string of length $z(v)$ for all $v \in \mathcal{U}$. The statement then holds since f_{4k} does not change the length of the string $\ell(\varphi(v))$.

For (ii), we argue similarly to the proof of Lemma 5. Let $v \in [n]$ such that $z(\varphi(v)) = 4k$. For any $w \in T_n^{v, \downarrow}$, we can write $\varphi(w) = \varphi(v) n_1 \cdots n_k$ for some $k \geq 1$. Then, $\ell(\varphi(w)) = \ell(\varphi(v)) 10^{n_1-1} \cdots 10^{n_j-1}$. Since $\ell(\varphi(v))$ has length $4k$, writing $\ell(\bar{\varphi}(v)) = f_{4k}(\ell(\varphi(v)))$, we have

$$f_{4k}(\ell(\varphi(w))) = f_{4k}(\ell(\varphi(v))) 10^{n_1-1} \cdots 10^{n_j-1} = \ell(\bar{\varphi}(v)) 10^{n_1-1} \cdots 10^{n_j-1}.$$

Therefore $\bar{\varphi}(w) = \bar{\varphi}(v) n_1 \cdots n_k$, i.e., w is mapped to a node in the same relative position in $\mathcal{U}$ with respect to v . As this holds for all $w \in T_n^{v, \downarrow}$, we have $\bar{T}_n^{v, \downarrow} = T_n^{v, \downarrow}$. Combined with (i), this also implies that $S = \bar{S}$.

For (iii), notice that $\text{ht}(\varphi(v))$ is the number of 1's in $\ell(\varphi(v))$, which has total length z . Then, applying f_{4k} and noting that the first coordinate stays fixed, we see that $\ell(\bar{\varphi}(v))$ has $1 + z - \text{ht}(\varphi(v))$ 1's. $\square$

Armed with $\bar{\varphi}$ and the properties established in Lemma 6, we complete the proof.

Proof of (11). Since b is an involution on the set of increasing trees on $[n]$, and due to property (ii) from Lemma 6, we have that

$$(\varphi, S) \stackrel{d}{=} (\bar{\varphi}, \bar{S}) = (\bar{\varphi}, S); \tag{16}$$

in particular, $\bar{\varphi}$ also encodes a random recursive tree. Furthermore, due to (iii), if $S \neq \emptyset$, then either S contains some $v \in [n]$ with $\text{ht}(\varphi(v)) \geq 2k$, or $\bar{S}$ contains some $v \in [n]$ with $\text{ht}(\bar{\varphi}(v)) \geq 2k$. That is, letting $E_S = \{\exists v \in S : \text{ht}(\varphi_{T_n}(v)) \geq 2k\}$ and $\bar{E}_S = \{\exists v \in \bar{S} : \text{ht}(\bar{\varphi}(v)) \geq 2k\}$, we have $(E_S \cup \bar{E}_S) \cap \{S \neq \emptyset\} = \{S \neq \emptyset\}$. Therefore, we have

$$\mathbf{P}\{E_S \cup \bar{E}_S, S \neq \emptyset\} = \mathbf{P}\{S \neq \emptyset\}. \tag{17}$$

Also, (16) implies that

$$\mathbf{P}\{E_S, S \neq \emptyset\} = \mathbf{P}\{\bar{E}_S, \bar{S} \neq \emptyset\} = \mathbf{P}\{\bar{E}_S, S \neq \emptyset\},$$

so (17) can be rewritten as

$$2\mathbf{P}\{E_S, S \neq \emptyset\} \geq \mathbf{P}\{S \neq \emptyset\},$$

proving (11). $\square$

4. PROOF OF THEOREM 2

Proof. First, let $f(k) : \mathbf{N} \rightarrow \mathbf{R}^+$ be any non decreasing function and consider $n' = \lfloor n/f(k) \rfloor$. For any $k \in \mathbf{N}$, for n sufficiently large, we have $n' \geq n/(2f(k))$, so $\mathbf{P}\{|T_n^{2,\uparrow}| \leq n'\} \geq 1/(2f(k))$. Also, for fixed k we have $|m_n - m_{n'} - e \log f(k)| \leq 1 + o(1)$ as $n \rightarrow \infty$, so from (4), for n large enough,

$$\mathbf{P}\{\text{ht}(T_{n'}) \geq m_n - k\} \leq \mathbf{P}\{\text{ht}(T_{n'}) \geq m_{n'} + e \log f(k) - 2 - k\} \leq \alpha e^{-\alpha'(e \log f(k) - k)} \quad (18)$$

and

$$\mathbf{P}\{\text{ht}(T_{n'}) \geq m_n - k\} \geq \mathbf{P}\{\text{ht}(T_{n'}) \geq m_{n'} + e \log f(k) + 2 - k\}. \quad (19)$$

Recalling from (5) that $\mathbf{P}\{1 \notin R_k\} \geq \mathbf{P}\{\text{ht}(T_n^{2,\uparrow}) < m_n - k\}$, it follows that for all $k \in \mathbf{N}$, for all n sufficiently large we have

$$\begin{aligned} \mathbf{P}\{1 \notin R_k\} &\geq \mathbf{P}\{\text{ht}(T_n^{2,\uparrow}) < m_n - k \mid |T_n^{2,\uparrow}| \leq n'\} \mathbf{P}\{|T_n^{2,\uparrow}| \leq n'\} \\ &\geq \mathbf{P}\{\text{ht}(T_{n'}) < m_n - k\} (2f(k))^{-1} \\ &\geq (1 - \alpha e^{-\alpha'(e \log f(k) - k)}) (2f(k))^{-1}. \end{aligned} \quad (20)$$

Setting $f(k) \equiv 2$, (20) yields that for all $k > 0$,

$$\liminf_{n \rightarrow \infty} \mathbf{P}\{1 \notin R_k(T_n)\} > 0.$$

Let

$$\varepsilon_0 = \min \left(\liminf_{n \rightarrow \infty} \mathbf{P}\{1 \notin R_k(T_n)\} : 1 \leq k \leq \log(2\alpha)/(\alpha'(e-1)) \right);$$

we hereafter assume that $\varepsilon \in (0, \varepsilon_0)$. Since the theorem only makes an assertion about values k for which $\mathbf{P}\{1 \notin R_k(T_n)\} < \varepsilon$, we can thus also assume that $k \geq \log(2\alpha)/(\alpha'(e-1))$ from now on. Taking $f(k) \equiv e^k$, this then implies that the lower bound in (20) is at least $e^{-k}/4$. Therefore, we must take $k \geq \log(1/(4\varepsilon))$ to ensure $\mathbf{P}\{1 \notin R_k\} < \varepsilon$.

We now prove the polynomial lower bound on $|R_k(T_n)|$ for any $k \geq \log(1/(4\varepsilon))$.

Fix $\delta > 0$. Set $K = \lceil (1 + \delta)^k/4 \rceil$ and $n' = \lfloor n/(1 + \delta)^k \rfloor \sim n/(4K)$, i.e., $f(k) = (1 + \delta)^k$. Let T_K be the subtree of T_n consisting of vertices $[K]$ and consider the K subtrees $T_n^{K,1}, \dots, T_n^{K,K}$, where $T_n^{K,i}$ is the tree containing vertex i in the forest obtained by removing edges between vertices in $[K]$. The vector of subtree sizes $(|T_n^{K,1}|, \dots, |T_n^{K,K}|)$ is distributed as a standard Pólya urn with n balls of K colours. Therefore,

$$\mathbf{P}\left\{ \# \{i \in [K] : |T_n^{K,i}| > n'\} < \frac{K}{4} \right\} \leq \binom{K}{K/4} (1/4)^{3K/4} \leq 2^{-K/4} \leq 2^{-(1+\delta)^k/16} \leq \varepsilon/2,$$

where the final inequality holds provided $\varepsilon > 0$ is sufficiently small, using that $k \geq \log(1/(4\varepsilon))$. Next, using that $\log f(k) = k \log(1 + \delta)$, we have from (19) that

$$\mathbf{P}\{\text{ht}(T_{n'}) \geq m_n - k\} \geq \mathbf{P}\{\text{ht}(T_{n'}) \geq m_{n'} + ek \log(1 + \delta) + 2 - k\} \geq 1/2$$

for $\delta > 0$ such that $\log(1 + \delta) < 1/e$, and for ε sufficiently small, since $\text{med}(\text{ht}(T_{n'})) = m_{n'} + O(1)$.

To combine these bounds, let $S = \{i \in [K] : |T_n^{K,i}| > n'\}$. On the event $\{|S| \geq K/4\}$, since the events $(\{\text{ht}(T_n^{K,i}) \geq m_n - k\})_{i \in [K]}$ are independent conditioned on the subtree sizes, we have that $\# \{i \in [K] : \text{ht}(T_n^{K,i}) \geq m_n - k\} \succeq_{st} \text{Bin}(K/4, 1/2)$. We have

$$\begin{aligned} &\mathbf{P}\left\{ \# \{i \in [K] : \text{ht}(T_n^{K,i}) \geq m_n - k\} < \frac{K}{8} \right\} \\ &\leq \mathbf{P}\left\{ \# \{i \in [K] : \text{ht}(T_n^{K,i}) \geq m_n - k\} < \frac{K}{8} \cap \left(|S| \geq \frac{K}{4} \right) \right\} + \mathbf{P}\left\{ |S| < \frac{K}{4} \right\} \\ &\leq \mathbf{P}\left\{ \text{Bin}\left(\frac{K}{4}, \frac{1}{2}\right) < \frac{K}{8} \right\} + \varepsilon/2 \\ &\leq \exp(-K/32) + \varepsilon/2 \leq \exp(-(1 + \delta)^k/128) + \varepsilon/2 \leq \varepsilon \end{aligned}$$

for sufficiently small $\varepsilon > 0$, where we again used that $k \geq \log(1/(4\varepsilon))$. On the other hand, if $\#\{i \in [K] : \text{ht}(T_n^{k,i}) \geq m_n - k\} \geq K/8$, there are at least $K/8$ vertices in $[K]$ with large subtree height, which are all then included in R_k , so $|R_k| \geq K/8 = c''\varepsilon^{-\log(1+\delta)}$ for some $c'' > 0$. This proves the theorem. $\square$

ACKNOWLEDGEMENTS

This research was initiated during the Seventeenth Annual Workshop on Probability and Combinatorics at McGill University's Bellairs Institute in Holetown, Barbados. We thank Bellairs Institute for its hospitality. LAB was supported by an NSERC Discovery Grant and by the CRC program. AB was supported by NSF GRFP DGE-2141064 and Simons Investigator Award 622132 to Elchanan Mossel. GL acknowledges support from the Spanish MINECO grant PID2022-138268NB-I00 and Ayudas Fundación BBVA a Proyectos de Investigación Científica 2021. We also thank two anonymous referees for helpful comments.

REFERENCES

- [1] L. Addario-Berry and K. Ford. Poisson–Dirichlet branching random walks. *The Annals of Applied Probability*, 23(1):283 – 307, 2013.
- [2] F. Bergeron, P. Flajolet, and B. Salvy. Varieties of increasing trees. In *CAAP*, volume 581 of *Lecture Notes in Computer Science*, pages 24–48. Springer, 1992.
- [3] A. M. Brandenberger, L. Devroye, and M. K. Goh. Root estimation in Galton–Watson trees. *Random Structures & Algorithms*, 61(3):520–542, 2022.
- [4] S. Briend, F. Calvillo, and G. Lugosi. Archaeology of random recursive dags and Cooper–Frieze random networks. *Combin. Probab. Comput.*, 32(6):859–873, 2023.
- [5] S. Bubeck, L. Devroye, and G. Lugosi. Finding Adam in random growing trees. *Random Structures & Algorithms*, 50(2):158–172, 2017.
- [6] H. Crane and M. Xu. Inference on the history of a randomly growing tree. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(4):639–668, 2021.
- [7] H. Crane and M. Xu. Root and community inference on latent network growth processes using noisy attachment models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, page qkad102, 2023.
- [8] L. Devroye and S. Janson. Long and short paths in uniform random recursive dags. *Ark. Mat.*, 49(1):61–77, 2011.
- [9] M. Drmota. *Random Trees: an Interplay between Combinatorics and Probability*. Springer Science & Business Media, 2009.
- [10] P. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge University Press, Cambridge, UK, 2009.
- [11] J. Haigh. The recovery of the root of a tree. *Journal of Applied Probability*, 7(1):79–88, 1970.
- [12] V. Jog and P.-L. Loh. Analysis of centrality in sublinear preferential attachment trees via the Crump–Mode–Jagers branching process. *IEEE Transactions on Network Science and Engineering*, 4(1):1–12, 2016.
- [13] N. Johnson and S. Kotz. *Urn models and their application*. Wiley, New York, 1977.
- [14] J. Khim and P.-L. Loh. Confidence sets for the source of a diffusion in regular trees. *IEEE Transactions on Network Science and Engineering*, 4(1):27–40, 2016.
- [15] G. Lugosi and A. S. Pereira. Finding the seed of uniform attachment trees. *Electronic Journal of Probability*, 24:1–15, 2019.
- [16] S. Navlakha and C. Kingsford. Network archaeology: Uncovering ancient networks from present-day interactions. *PLoS Comput Biol*, 7(4):e1001119, 2011.
- [17] J. Neveu. Arbres et processus de Galton–Watson. *Annales de l'I. H. P. Probabilités et statistiques*, 22:199–207, 1986.
- [18] T. Reddard and L. Devroye. On the discovery of the seed in uniform attachment trees. *Internet Mathematics*, page 7593, 2019.
- [19] D. Shah and T. R. Zaman. Rumors in a network: Who's the culprit? *IEEE Transactions on Information Theory*, 57(8):5163–5181, 2011.
- [20] J. K. Sreedharan, A. Magner, A. Grama, and W. Szpankowski. Inferring temporal information from a snapshot of a dynamic network. *Nature Scientific Reports*, 9(1):3057, 2019.