Uniform random generation of some families of random trees and simulations of Voronoi tessellations on the CRT

Vincenzo Politelli
Advisor: Prof. Jérémie Bettinelli*

Abstract

In the paper [AAC+18], the authors study the behaviour of Voronoi tessellations induced on large graphs by $k \geq 2$ agents, when such agents are placed uniformly at random on the "vertices" of the graph. The same is done for $k \geq 2$ agents placed uniformly at random on the vertices of the CRT (Continuum Random Tree), which constitutes the scaling limit of uniformly random labeled trees, rooted plane trees, rooted unembedded binary trees, rooted unembedded trees and unrooted unembedded trees and other tree-like families of graphs. The aim of this thesis is that of giving a pedagogical description of the construction of the CRT, on the uniform random generation of some families of trees, and to run computer simulations to corroborate the results presented in [AAC+18]. We will provide proof of correctness for the algorithms used to randomly generate graphs, either by re-elaborating existing proofs or by writing new ones, with the aim of providing straightforward and complete explanations of the results presented. Finally, we code a library (linked in section 6) in the Python programming language, which allows to run the simulations presented in this thesis.

1 Acknowledgements

I am sincerely grateful to my advisor for his invaluable guidance and support throughout the two months during which I wrote this thesis.

2 Introduction

In this section we introduce the formalism used for the remainder of the thesis along some elementary facts in graph theory and we will present the main results that we will discuss.

One usually defines graphs as an ordered pair $G = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$ a finite set of vertices (also called nodes) and $\mathcal{E} \subseteq V^{(2)}$ a set of edges.

Given a subset of vertices $\mathcal{V}' \subseteq \mathcal{V}$, the subgraph induced by $\mathcal{V}'$ is the graph $G' = (\mathcal{V}', \mathcal{E}')$ where $\mathcal{E}'$ is defined as the subset of edges of $\mathcal{E}$ that only contain elements of $\mathcal{V}'$ i.e., $\mathcal{E}' = \mathcal{E} \cap \mathcal{V}'^{(2)}$.

Given two vertices $v, w \in \mathcal{V}$, a path γ from v to w is a sequence $e_1, \ldots, e_n$ of distinct edges such that $e_i \cap e_{i+1}$ is non-empty for all $1 \leq i \leq n-1$ and $v \in e_1$ and $w \in e_n$. The integer n is denoted by $|\gamma|$ and we will refer to it as the *length* of the path.

Two graphs $(\mathcal{V}, \mathcal{E})$, $(\mathcal{V}', \mathcal{E}')$ are said *isomorphic* if there is a bijection $f : \mathcal{V} \to \mathcal{E}$, called an isomorphism, such that $\{v, w\} \in \mathcal{E}$ if and only if $\{f(v), f(w)\} \in \mathcal{E}'$.

A graph is said to have a cycle if there exists a vertex $v \in \mathcal{V}$ and a path of non-zero length from v to itself. If the graph does not have a cycle, we we say that it is *acyclic*.

If a graph contains a path connecting any two vertices, we say that it is *connected*, otherwise that is *disconnected*.

Finally, an acyclic graph is called a *forest*. If such forest is also connected we call it a *tree*.

One can also endow graphs with a distance function.

*École polytechnique, Laboratoire d'informatique (LIX), 91128, Palaiseau Cedex, FRANCE.

Definition 1. Let $G = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$ and $\mathcal{E}$ defined as above. We define the graph distance $d_G \colon \mathcal{V} \times \mathcal{V} \to \mathbb{N}$ by letting

$$d_G(v, w) := \inf \{|\gamma| : \gamma \text{ is a path connecting } v \text{ and } w\}$$

for all $(v, w) \in \mathcal{V} \times \mathcal{V}$.

One easily verifies that d_G is a distance function. Thus, we have that $(\mathcal{V}, d_G)$ is a metric space. Notice that, given the set of vertices $\mathcal{V}$, the function d_G is univocally determined. The converse is also true, given $\mathcal{V}$ and d_G, there exists only one possible set of edges $\mathcal{E}$. If we also endow the set $\mathcal{V}$ of vertices with the uniform probability measure μ_G we get that $(\mathcal{V}, d_G, \mu_G)$ is a measured metric space. This leads us to the definition of graph that we will use throughout this thesis

Definition 2. Let $\mathcal{V}$ be a set of vertices and $\mathcal{E} \subseteq \mathcal{V}^{(2)}$ and let $G = (\mathcal{V}, \mathcal{E})$. The measured metric space $(\mathcal{V}, d_G, \mu_G)$ is called a *graph*. We will often abuse notation and write (G, d, μ) instead.

This formalism will be more suitable to the way we will treat graphs. As pointed above, this definition completely characterises the set of edges $\mathcal{E}$ of the graph. So, we will be able to use the notion of edges if needed.

When dealing with graphs, one often treats them as objects embedded on a surface.

Definition 3 ([Ale22])**.** Given an orientable surface S of genus g. A celullar embedding of a graph G is a crossing-free drawing of G on S or, more precisely, a one-to-one map ι from G to S such that every vertex of G is mapped to a point on S and every edge (v, w) is represented as a simple path (curve) between the images of u and v. Moreover, the drawing is required to be crossing-free: the images of two distinct edges cannot cross (they meet only at their endpoints), and the images of edges cannot pass through the images of vertices.

Moreover, if the components of $S \setminus G$ are homeomorphic to a topological disk, one calls ι a topological embedding. In this case, the connected components of $S \setminus G$ are called the faces of a graph.

By abuse of notation, we will refer to embedded graphs as topologically embedded graphs.

Among the most studied class of graphs are those embedded on the sphere. Graphs that can be embedded on the sphere are called *planar*. The reason of this definition is that one can consider a graph G (topologically) embedded on a sphere S with a stereographic projection p onto $\mathbb{R}^2$ such that the preimage of the *point at infinity* of p is in $S \setminus G$. The connected component of $S \setminus G$ that contains the preimage point at infinity of p is called the *external face*. This constitutes an embedding of the graph G on $\mathbb{R}^2$. Therefore, we can talk about graph embedded on the sphere and graphs embedded on the plane interchangeably.

Another notion that we will need in this text is the notion of Voronoi vector.

Definition 4. Let (G, d, μ) be a graph on N vertices endowed with the graph distance d and the uniform probability measure μ. Let $2 \leq k \leq N$ be a natural number. Sample distinct vertices $v_1, v_2, \ldots, v_k$ according to the probability measure μ. Such vertices are called *agents*. We define and denote the Voronoi vector as follows:

$$\text{Vor} = \left(\frac{V_1}{N}, \frac{V_2}{N}, \ldots, \frac{V_k}{N} \right),$$

where V_i is the number of vertices of G which are closer to v_i than to any other v_j with $j \neq i$. In other words

$$V_i = \# \{v \in \mathcal{V} : d(v_i, v) \leq d(v_j, v), \forall j \neq i\}.$$

The main question addressed in [AAC+18] is that of determining which models of random graphs (which we shall define more thoroughly later) give rise to a uniform Voronoi vector when the agents are chosen uniformly at random.

One first step to answering this question is given by the following theorem from [AAC+18].

Theorem 1 ([AAC+18])**.** Fix $k \geq 2$. Let T_N be a uniform random labeled tree (resp. rooted plane tree, rooted unembedded binary tree, rooted unembedded tree, unrooted unembedded tree) on N vertices and let $v_1, v_2, \ldots, v_k$ be k random uniform vertices of T_N. Then the k-dimensional vector whose i'th coordinate is equal to the proportion of vertices of T that are closer to v_i than to any other v_j converges to a uniform vector on the $(k-1)$-dimensional simplex, as N goes to infinity.

Note that, given $x_1, x_2, \ldots, x_k$ points in $\mathbb{R}^k$ which are in convex position, the $(k-1)$-dimensional simplex is the convex hull of such points.

In order to corroborate the statement of this theorem, one could run the following simulation.

Algorithm 1: Simulation for Theorem 1

input : $k \geq 2$ and N large
do M **times**
  sample T_N a uniform random labeled tree on N vertices
  sample $v_1, v_2, \ldots, v_k$ distinct vertices uniformly at random from T_N
  compute the corresponding Voronoi vector
perform a *goodness-of-fit* test between the distribution of the computed Voronoi vectors and the predicted distribution

However, the main result of [AAC+18] is a more simple yet stronger version of Theorem 1, that is, its infinitary counterpart.

Theorem 2 ([AAC+18])**.** Let (T, d, μ) be the CRT (Continuum Random Tree), with distance d and mass measure μ. Let $X_1, X_2, \ldots, X_k \in T$ be independent samples for some fixed $k \geq 2$. Partition T into the Voronoi cells $C_1, C_2, \ldots C_k$ where C_i is the part of T closer to X_i than to any other X_j for $j \neq i$. Then the vector $(\mu(C_i))_{1 \leq i \leq k}$ is a uniform partition of unity (i.e. uniformly distributed on the $(k-1)$-dimensional simplex).

Intuitively, the CRT can be thought of as the (scaling) limit of some families of uniformly random trees. We will make this idea precise in section 4.4.

2.1 Trees

As previously defined, trees are connected acyclic graph. It is easy to see that for any tree T and any two nodes v, w of T, there exists a unique path connecting v and w. Given a tree T and a node v of T, a subtree of T at v is any of the connected components that are produced when removing all the edges of T incident to v and the vertex v itself. It is easy to see that a subtree is itself a tree.

There are several families of trees one can define by adding some combinatorial or topological structure on them. Here are some of them

- **Free trees**. Acyclic connected graphs with no additional structure on them.
- **Rooted unembedded (or unordered) trees**. Free trees with a distinguished vertex, called the *root*. In this case we can orient each edge (that is, assign to each edge a starting and an ending point) away from the root. For every vertex v different from the root, consider the unique vertex w such that there exists an edge $\{v, w\}$ oriented from w to v. Such edge w is called the *parent* of v. For every vertex v, consider the set of vertices $w_1, w_2, \ldots, w_m$ such that there exist edges $\{v, w_1\}, \{v, w_2\}, \ldots, \{v, w_m\}$ oriented from v to w_i for all $0 \leq i \leq m$, Such vertices $w_1, w_2, \ldots, w_3$ are called the *children* of v. If a node does not have any children we call it a *leaf*.
- **Labeled trees**. Free trees in which each vertex has a distinct label (for trees with n vertices we will use the labels in $\{0, 1, \ldots, n-1\}$).
- **Rooted plane trees**. Rooted trees where, for each vertex v, the children of v are totally ordered. We call these plane trees because, if one embeds a tree on the plane, children get a natural counter-clockwise ordering.
- **Binary trees**. Rooted trees where every node has either two or no children. The nodes which are not leaves are called *internal* nodes. It is well-known that the number of binary trees with n internal nodes (or, alternatively, $n+1$ leaves) is the n-th Catalan number C_n.

We remark that two rooted unordered trees are isomorphic if there exists an isomorphism that preserves the property of being a root node. Similarly, for two rooted plane trees to be isomorphic, the isomorphism must preserve the property of being a root node and it must also preserve the ordering of children. In case of labelled trees, the isomorphism must preserve labels.

3 Encoding of rooted plane trees through Dyck paths

We present an important fact about rooted plane trees, namely a bijection between rooted plane trees and Dyck paths.

Definition 5. Let $n \in \mathbb{N}$. A sequence $p_0, p_1, \ldots, p_{2n}$ that takes values in $\mathbb{N}$ such that:

(i) $p_0 = p_{2n} = 0$,

(ii) $p_{i+1} - p_i = a_i, \qquad \forall\, 0 \leq i \leq 2n-1$,

is called a tied down, positive walk on the integers, with unit steps, of length $2n$.

Definition 6. Let $\mathbb{N}$. Consider a sequence $a_0, a_1, \ldots, a_{2n-1}$ that takes values in $\{-1, +1\}$ that follows the two requirements below:

(i) $\sum_{j \leq i} a_j \geq 0, \qquad \forall\, 0 \leq i \leq 2n-1$;

(ii) $\sum_{j=0}^{2n-1} a_j = 0$.

Such sequence is called a Dyck word of length $2n$.

Now, consider the sequnce $p_0, p_1, \ldots, p_{2n}$ defined as

$$p_0 := 0; \;\; p_{i+1} := \sum_{j \leq i} a_i, \qquad \forall\, 0 \leq i \leq 2n-1.$$

Such sequence is called the walk generated by the Dyck path $a_0, a_1, \ldots, a_{2n-1}$.

The Dyck path of length $2n$ generated by $a_0, \ldots, a_{2n-1}$ or by $p_0, \ldots, p_{2n}$ is the piecewise affine function from $[0, 2n]$ to $\mathbb{R}$ connecting the points (i, p_i) and $(i+1, p_{i+1})$ with a straight-line path for all $0 \leq i \leq 2n-1$.

Remark 1. In Definition 6, condition (i) is equivalent to asking that the first i element of the word contain at most as many -1s as $+1$s and condition (ii) is equivalent to saying that $a_0, a_1, \ldots, a_{2n-1}$ contains the same number of $+1$s and -1s.

Remark 2. The sequence $p_0, p_1, \ldots, p_{2n}$ described in Definition 6 is a tied down, positive walk on the integers, with unit steps.

Remark 3. Notice that the set of Dyck words, the set of tied down, positive random walks of unit steps, and the set of Dyck paths where all of the objects are of length $2n$ are naturally bijective.

Indeed, given two distinct Dyck paths of length $2n$, they generate distinct walks. Similarly, given a tied down, positive walk with unit steps of length $2n$, one can obtain the Dyck word that generates it through the inversion formula

$$a_i = p_{i+1} - p_i, \qquad \forall\, 0 \leq i \leq 2n-1.$$

Clearly, two distinct walks yield two distinct Dyck words. Therefore, we have a bijection between Dyck words and tied down, positive walks with unit step, of length $2n$.

Also, it is clear that the set of Dyck paths of length $2n$ is bijective to the set of tied down, positive walks with unit step of length $2n$.

The three sets mentioned above are indeed bijective as claimed.

It turns out that the set of Dyck words of length $2n$ corresponds to the set of well-parenthesised words of length $2n$ from the alphabet $\{(,)\}$ (here '(' plays the role of '+1' and ')' that of '-1'). We leave some reference to consult for additional detail [HMU03].

Given a rooted plane tree, we can generate a Dyck word through the following Algorithm 2.

Algorithm 2: TreeToDyckWord

input : a rooted plane tree T on n vertices.
output: the Dyck word $\texttt{WORD}(T)$ of lenght $2(n-1)$.

```
WORD(Tree)
└ WORD_AUX(root)

WORD_AUX(current_node)
  for child of current_node in ascending order do
    yield +1
    WORD_AUX(child)
    yield -1
```

Given a rooted plane tree T on n vertices, Algorithm 2 generates a sequence

$$\texttt{WORD}(T) = a_0, a_1, \ldots, a_{2(n-1)-1}$$

which takes values in $\{-1, +1\}$. Consider the sequence $p_0, p_2, \ldots, p_{2(n-1)}$ defined as

$$p_0 := 0, \qquad p_{i+1} := p_i + a_i = \sum_{j \leq i} a_j, \qquad \forall i \in \{0, 2(n-1)-1\}.$$

We define

$$\texttt{WALK}(T) := p_0, p_1, \ldots, p_{2(n-1)}.$$

Let us prove that, for any tree T, the sequence $\texttt{WORD}(T)$ is actually a Dyck word and the sequence $\texttt{WALK}(T)$ is the walk associated to it.

Proposition 1. Let $n \geq 1$ and T be a rooted plane tree on n vertices. Then, the sequence $\texttt{WALK}(T) = p_0, p_1, \ldots, p_{2(n-1)}$ is a tied down, positive walk with unit steps, that is:

(i) $p_0 = p_{2(n-1)} = 0$.

(ii) $p_{i+1} = p_i \pm 1$;

(iii) $p_i \geq 0$.

We also have that

(iv) The sequence $\texttt{WORD}(T) = a_0, a_1, \ldots, a_{2(n-1)-1}$ is the Dyck word that generates the walk $\texttt{WALK}(T)$.

Proof. The facts that $p_0 = 0$ and $p_{i+1} = p_i \pm 1$ are trivial. We now show that $p_{2(n-1)} = 0$ and $p_i \geq 0$ by induction on $n \geq 1$.

Base case. There is only one rooted plane tree on 1 vertex, for which the statements are easily verifiable.

Induction step. Let $n \geq 1$ and suppose that the statements are true for all the rooted plane trees on $1 \leq m \leq n$ vertices. We want to show that the statements are true for any rooted plane tree on $n+1$ vertices.

Let T be a rooted plane tree on $n+1$ vertices. Let $c_1, c_2, \ldots, c_k$ be the children of the root of T listed in ascending order. Let $T_1, T_2, \ldots, T_k$ be the subtrees of T where T_i is the subtree of T rooted at c_i. Notice that the sequence $\texttt{WORD}(T)$ generated with Algorithm 2 follows the following formula

$$\begin{aligned} \texttt{WORD}(T) &= a_0, a_1, \ldots, a_{2n-1} \\ &= +1, \texttt{WORD}(T_1), -1, +1, \texttt{WORD}(T_2), -1, \ldots, +1, \texttt{WORD}(T_k), -1. \end{aligned} \tag{1}$$

By the induction hypothesis we have that the partial sums of the first j elements of $\texttt{WORD}(T_i)$ (i.e. the j-th element of the sequence $\texttt{WALK}(T_i)$) is non-negative for all $1 \leq i \leq k$. Combining this fact with (1) we can see that the sum of the first i elements of $\texttt{WALK}(T)$ is non-negative, which means that $p_i \geq 0$ for all $1 \leq i \leq 2n$.

By definition, we have that $p_{2n} = \sum_{i=0}^{2n-1} a_i = \sum_{l \in \texttt{WORD}(T)} l$. By summing the terms appearing in (1) and rearranging the terms so that the $+1$s that appear in it cancel with the -1s, we get

$$\sum_{l \in \texttt{WORD}(T)} l = \sum_{i=1}^{k} \sum_{l_i \in \texttt{WORD}(T_i)} l_i \ + \underbrace{(1-1) + \cdots + (1-1)}_{k \text{ times}} = \sum_{i=1}^{k} \sum_{l_i \in \texttt{WORD}(T_i)} l_i.$$

Since $\sum_{l_i \in \texttt{WORD}(T_i)} l_i = 0$ by the induction hypothesis, we get that $p_{2n} = \sum_{l \in \texttt{WORD}(T)} l = 0$. Thus $p_0, p_1, \ldots, p_{2n}$ is a tied down positive walk with unit steps.

The definition of $\texttt{WALK}(T)$ implies that $a_i = p_{i+1} - p_i$ for all $0 \leq i \leq 2n-1$. By Remark 3, we get that WORD(T) is the Dyck word that generates the walk $p_0, p_1, \ldots, p_{2n}$. This proves (iv). □

A corollary of Proposition 1, is that we can associate a tied down positive walk with unit steps, a Dyck word, or a Dyck path of length $2(n-1)$ to any rooted plane tree on n vertices. Indeed, given a rooted plane tree T on n vertices, we can generate the sequences $\texttt{WORD}(T)$, or $\texttt{WALK}(T)$. Such sequences uniquely define a Dyck path (which is the same for both sequences by Remark 3).

We remark that the Dyck word associated to the rooted tree with only one vertex is the empty sequence and the corresponding Dyck path is trivially the function $x \in \{0\} \mapsto 0$.

However, given a Dyck word D, one can also generate a rooted plane tree T such that $\texttt{WORD}(T) = D$. We will prove that we can do so by the following algorithm.

Algorithm 3: DyckWordToTree

input : a Dyck word $D = a_0, \ldots, a_{2n-1}$.
output: a rooted plane tree T on $n+1$ nodes such that $\texttt{WORD}(T) = D$.

```
T ← the tree containing only one root node
current_node ← the root node of T
for i from 0 to 2n − 1 do
    if a_i = +1 then
        let '<' be the total order associated to the children of current_node.
        let b_1 < ··· < b_k be the children of current_node.
        let b_{k+1} be a new children of current_node such that b_k < b_{k+1}.
        current_node ← b_{k+1}.
    if a_i = −1 then
        current_node ← the parent of current_node.
return T
```

Proposition 2. Let $n \geq 1$ and $\mathbb{T}_n$ be the set of rooted ordered trees on n vertices. Let $\mathbb{D}_{n-1}$ be the set of Dyck words of length $2(n-1)$. The function Φ_n that associates each element of $\mathbb{T}_n$ to its corresponding Dyck word (that is to $\texttt{WORD}(T)$) is a bijection.

Proof. We want to prove that Φ_n is bijective by providing an inverse of it. Such inverse is given by Algorithm 3.

Let $n \geq 0$ and define $P(n) :=$ "Given a Dyck word $D = a_0, \ldots, a_{2(n-1)-1}$, the algorithm DyckWordToTree that takes D as input, returns a tree T such that $\Phi_n(T) = \texttt{WORD}(T) = D$. Moreover, at the end of this procedure, the variable current_node stores the root node r".

We will prove $P(n)$ by induction on $n \geq 0$.

Base case. There only exists one element in $\mathbb{T}_1$ and $\mathbb{D}_0$. Therefore the statement is easily verified.

Induction step. Let $n \geq 0$ and suppose that $P(m)$ is verified for all $m \leq n$. We want to show that $P(n+1)$ is true.

Let $D = a_0, a_1, \ldots, a_{2n-1}$ be a Dyck word of length $2(n+1)$. We first prove that D has a particular structure.

Let $W = p_0, p_1, \ldots, p_{2n}$ be the Dyck path generated by D. We have

$$p_0 = 0, \ \ p_{i+1} = \sum_{j \leq i} a_j \qquad \forall\, 0 \leq i \leq 2n-1$$

Let $0 = i_0 < i_1 < \cdots, < i_k = 2n$ be the indices such that $p_{i_0} = p_{i_1} = \cdots = p_{i_k} = 0$. Since $p_i \geq 0$ for all $0 \leq i \leq 2n$, and $p_{i+1} - p_i = \pm 1$, we have that

- $p_{i_j+1} = 1$ for all $0 \leq j \leq k-1$;
- $a_{i_j} = +1$ for all $0 \leq j \leq k$;
- $a_{i_j-1} = -1$ for all $1 \leq j \leq k$.

Therefore

$$D = +1, (a_{i_0+1}, \ldots, a_{i_1-2}), -1, +1, (a_{i_1+1}, \ldots, a_{i_2-2}), -1, \ldots, +1, \left(a_{i_{k-1}+1}, \ldots, a_{i_k-2}\right), -1.$$

We claim that $a_{i_j+1}, \ldots, a_{i_{j+1}-2}$ is a Dyck word for all $0 \leq j \leq k-1$. Indeed, for all $i_j + 1 \leq l \leq i_{j+1} - 2$, we have that

$$\sum_{m \leq l} a_m = p_l - p_{i_j+1} + 1 = p_m \geq 0.$$

Also, for all $0 \leq j \leq k-1$, we have that

$$\sum_{m \leq i_{j+1}-2} a_m = p_{i_{j+1}-2} - 1 + 1 = p_{i_{j+1}-2} + a_{i_j-1} + a_{i_j} = p_{i_j} = 0.$$

Therefore $D_j := a_{i_j+1}, \ldots, a_{i_{j+1}-2}$ is a Dyck word. Since D_j has length at most $2n$, the induction hypothesis tells us that there exists a tree T_j such that $\texttt{WORD}(T_j) = D_j$.

We construct the tree T on $n+1$ vertices as follows. Connect a root node r to each of the subtrees $T_1, \ldots, T_k$ with the edges $\{r, c_1\}, \ldots, \{r, c_k\}$. The list of children of r in ascending order is $c_1, \ldots, c_k$. By construction we have

$$\begin{aligned}
\texttt{WORD}(T) &= +1, \texttt{WORD}(T_1), -1, +1, \texttt{WORD}(T_2), -1, \ldots, +1, \texttt{WORD}(T_k), -1 \\
&= +1, D_1, -1, +1, D_2, -1, \ldots, +1, D_k, -1 \\
&= a_{i_0}, (a_{i_0+1}, \ldots, a_{i_1-2}), a_{i_1-1}, a_{i_1}, (a_{i_1+1}, \ldots, a_{i_2-2}), a_{i_2-1}, \ldots, a_{i_{k-1}}, \left(a_{i_{k-1}+1}, \ldots, a_{i_k-2}\right), a_{i_k-1} \\
&= a_0, \ldots, a_{2n-1}
\end{aligned}$$

We now prove that T is the tree returned by the algorithm DyckWordToTree when it takes D as input and that, when the execution of such algorithm ends, the variable `current_node` stores the root node r.

When the algorithm DyckWordToTree takes D as input, we have the following. When $i = i_0 = 0$, we have that $a_{i_0} = a_0 = +1$. Thus, the first children c_1 of the root node r is created and the variable `current_node` is updated to store c_1. The algorithm then runs through the Dyck word $D_1 = a_1, \ldots, a_{i_1-2}$. By induction hypothesis, we get that the algorithm constructs the tree T_1 in which the root node corresponds to c_1. At the end of this process (that is, at the end of the iteration step when $i = a_{i_0-2}$), the induction hypothesis tells us that the variable `current_node` stores the node c_1. Since $a_{i_1-1} = -1$, at the end of the iteration step with $i = i_{1-1}$, we will have the variable `current_node` storing the root node r, which is the parent of c_1.

We can repeat the present argument to argue that the tree which will be constructed at the end of this procedure is the rooted plane tree T' with root node r; with the children of r being $c_1, \ldots c_k$ (listed in ascending order) and the subtrees of T at r are the trees $T_1, \ldots, T_k$ with the root node of T_j corresponding to the node c_j for all $1 \leq j \leq k$. We have that $T' = T$.

Finally, we observe that, by induction hypothesis, at the end of the iteration step with $i = i_{i_k-2}$, the variable `current_node` is storing the node c_k. Since $a_{i_k-1} = -1$, the variable `current_node` will be storing the root node r at the end of the algorithm.

Thus, $P(n+1)$ is verified. □

A corollary of Proposition 2 together with Remark 3 is that the notion of rooted plane trees and that of Dyck words of Dyck paths are strongly interconnected. In particular, the encoding of rooted plane trees through Dyck paths will give us a solid intuition of what is the CRT.

4 The Brownian motion and the Brownian excusion

4.1 Definition and elementary properties

The Brownian motion is a fundamental object in probability theory both because of its theoretical interest and its ubiquitousness in natural science. It is named after Robert Brown, who, in 1827, was studying the trajectory of some pollen immersed in water. Albert Einstein would give a quantitative description of such motion in 1905, which helped him indirectly confirm the existence of atoms and molecules. The experimental works of Jean Baptiste Perrin held in 1908 confirmed the exquations used by Einstein to describe the Brownian motion. We now proceed to give an rigorous description of it.

Consider some probability triple $(\Omega, \mathscr{F}, P)$. A stochastic process on it is a sequences of random variable on such probability space depending on a parameter.

Definition 7 ([RW00])**.** A real-valued stochastic process $\{B_t \colon t \in \mathbb{R}_+\}$ is a standard Brownian motion if it has the following properties:

(i) $B_0(\omega) = 0$, for all $\omega \in \Omega$;

(ii) the map $t \in \mathbb{R}_+ \mapsto B_t(\omega)$ is a continuous function for all $\omega \in \Omega$.

(iii) for every $t, h \geq 0$, $B_{t+h} - B_t$ is independent of $\{B_s \colon 0 \leq s \leq t\}$ and has a Gaussian distribution with zero mean and variance h.

There are some simple transformations that preserve standard Brownian motion.

Fact 1. Let $(B_t)_{t\geq 0}$ be a standard Brownian motion.

(i) Let $Y_t := -B_t$ for all $t \geq 0$. Then $\{Y_t \colon t \geq 0\}$ is also a standard Brownian motion.

(ii) Let $s \in [0, \infty)$ and $Y_t := B_{s+t} - B_s$ for all $t \geq 0$. Then $\{Y_t \colon t \geq 0\}$ is also a standard Brownian motion.

(iii) Let $T \in [0, \infty)$ and $Y_t := B_{T-t} - B_T$. Then $\{Y_t \colon t \in [0, T]\}$ is also a standard Brownian motion.

(iv) Let $a \in (0, \infty)$ and define $Y_t := \frac{1}{a} B_{a^2 t}$ for all $t \geq 0$. Then $\{Y_t \colon t \geq 0\}$ is also a standard Brownian motion.

The proof of Fact 1 is obtained by unravelling the definitions. Notice that by point (iv) of Fact 1, for any $a \in (0, \infty)$, rescaling the x-axis and y-axis of a standard Brownian motion by $1/a^2$ and $1/a$ respectively, still yields a standard Brownian motion. This property is called self-similarity.

Another important property of the Brownian motion is that, given a Brownian motion $(B_t)_{t\geq 0}$, the function $t \mapsto B_t$ is almost surely continuous. We actually know some more. To explain this, we need the following definition.

Definition 8. A function $f : X \subseteq \mathbb{R} \to \mathbb{R}$ is said to be locally γ-Hölder continuous at x if there exists $\delta > 0$ and a constant $C > 0$ such that for all y with $|y - x| < \delta$, we have $|f(y) - f(x)| < C\,|y - x|^{\gamma}$.

If f is locally γ-Hölder at x for all $x \in X$, we say that f is everywhere locally γ-Hölder. In this case we refer to γ as the Hölder exponent of f.

We have the following well-known proposition.

Proposition 3. Let $(B_t)_{t\geq 0}$ be a standard Brownian motion. The function $t \mapsto B_t$ is everywhere locally γ-Hölder for all $\gamma < 1/2$.

Finally, we give the following essential property

Proposition 4 ([RW00])**.** There exists a probability space $(\Omega, \mathscr{F}, \mathbb{P})$ on which one can define a standard Brownian motion.

Thoughout this section we will be working on such probability space.

4.2 Skorokhod embedding and Donsker Invariance Principle

One way to think about a standard Brownian motion is that of the "limit" of a normalized random walk. We can make this idea precise by looking at a construction named Skorokhod embedding, which allows to embed any zero-mean, finite-variance random walk into a standard Browninan motion. A description of the construction that we are about to describe can be found in [RW00].

Let $(B_t)_{t\geq 0}$ be a standard Brownian motion. We admit the following proposition without proof.

Proposition 5 ([RW00])**.** Fix $a < 0 < b$. Let $\tau := \{t\colon B_t \notin (a,b)\}$. Then $\tau < \infty$ almost surely and

(i) $\mathbb{P}(B_t = b) = -\frac{a}{b-a}$;

(ii) $\mathbb{E}(\tau) = |ab|$.

Consider a random walk with step distribution F such that

$$\int_{-\infty}^{\infty} xF(dx) \ , \qquad \int_{-\infty}^{\infty} x^2 F(dx) = \sigma^2 < \infty.$$

We aim to find a stopping time T for the Brownian motion such that $B_T \sim F$ and $\mathbb{E}(T) = \sigma^2$.

Let F_+ (resp. F_-) be a restriction of F to $\mathbb{R}_+$ (resp. $\mathbb{R}_-^*$). We randomly and independently pick $\alpha < 0 < \beta$ following the probability distribution μ defined as

$$\mu(da, db) := \gamma \cdot (b-a) \cdot F_+(db) \cdot F_-(da)$$

where the normalization constant γ is defined as

$$\gamma^{-1} = \int_0^{\infty} bF_+(db) = \int_{-\infty}^{0} aF_-(da).$$

The second equality in the above equation holds because F has mean zero. The construction named Skorokhod embedding picks α and β according to μ and lets the Brownian motion $(B_t)_{t\geq 0}$ run until it hits α or β. We have the following theorem

Theorem 3 (Skorokhod embedding, [RW00])**.** Let $T := \inf\{s\colon B_s \notin (\alpha,\beta)\}$. We have that $B_T \sim F$ and $\mathbb{E}(T) = \sigma^2$.

Proof. Using the definition of γ provided above and point (i) of Proposition 5, we get that

$$\mathbb{P}(B_T \in db) = \int_{\infty}^{0} -\frac{a}{b-a} \cdot \gamma \cdot (b-a) \cdot F_+(db) \cdot F_-(da) = F_+(db).$$

Similarly, we have $\mathbb{P}(B_T \in da) = F_-(da)$. Therefore, $B_T \sim F$.

Now, using point (ii) and Proposition 5, we get that

$$\begin{aligned}
\mathbb{E}(B_T) &= \int_0^{\infty}\int_{-\infty}^{0} \mu(da,db)\,|ab| \\
&= \gamma \cdot \left[\left(\int_0^{\infty} F_+(db) \cdot b^2 \underbrace{\int_{-\infty}^{0} F_-(da)\cdot|a|}_{\gamma^{-1}} \right) + \left(\int_{-\infty}^{0} F_-(da)\cdot a^2 \underbrace{\int_0^{\infty} F_+(db)\cdot|b|}_{\gamma^{-1}} \right) \right] \\
&= \int_{-\infty}^{\infty} x^2 F(dx).
\end{aligned}$$

□

Now, we can see how to embed a random walk with step distribution F in a Brownian motion. One can take $T_0 := 0$, then define $T_1 := \inf\{t\colon B_t \notin (\alpha,\beta)\}$. We can then set $T_2' := \inf\{(B_{T_1+t} - B_{T_1}) \notin (\alpha,\beta)\}$ and then set $T_2 := T_1 + T_2'$. In general, supposing that $T_0, \ldots, T_i$ are already chosen, we set $T_{i+1}' := \inf\{t > 0\colon (B_{T_i+t} - B_{T_i}) \notin (\alpha,\beta)\}$ and $T_{i+1} \cdot T_{i+1}' + T_i$. Notice that, the fact that $B_{T_i+t} - B_{T_i}$ is a Brownian motion does not simply follow from point (ii) of Fact 1, since the T_is are random. It follows, instead, from the "Strong Markov Property" of the Brownian motion, whose description and proof can be found in section 12 of [RW00].

We thus have a sequence $0 = T_0 < T_1 < T_2 < \cdots$ of stopping times for which the following holds.

Theorem 4 ([RW00])**.** Define $S_n := B_{T_n}$ for all $n \geq 0$. Then $(S_n)_{n\geq 0}$, is a random walk with step distribution F. We also have $\mathbb{E}(T_n) = n\sigma^2$.

The Skorokhod embedding is a powerful tool which allows us to prove a theorem known as Donsker's Invariance Principle, also called the Functional Central Limit Theorem. This theorem affirms that the Brownian motion is a weak limit of random walks. So, consider a zero-mean random walk with unit variance and step distribution F. Let $(S_n)_{n\geq 0}$ be the random walk whose existence is guaranteed by Theorem 4. For all $n \geq 1$, we define the random piecewise affine function

$$S_t^{(n)} = \sqrt{n}\left[\left(t - \frac{k}{n}\right) S_{k+1} + \left(\frac{k+1}{n} - t\right) S_k\right], \qquad \forall\ \frac{k}{n} \leq t \leq \frac{k+1}{n} \tag{2}$$

for all $0 \leq k \leq n-1$. This function is piecewise continuous on $[0,1]$. Also $S_t^{(n)} = \frac{1}{\sqrt{n}} S_k$ for all $t = k/n$. We have the following theorem.

Theorem 5 (Donsker's Invariance Principle, [RW00])**.** The process $\left(S_t^{(n)}\right)_{0\leq t\leq 1}$ weakly converges to $(B_t)_{0\leq 1}$ as n tends to $+\infty$.

Proof. The proof of this theorem amounts to prove that, given $\varepsilon > 0$, there exists N large enough such that

$$\mathbb{P}\left(\sup_{0\leq t\leq 1}\left|S_t^{(n)} - B_t^{(n)}\right| > \varepsilon\right) \leq \varepsilon,$$

where $B_t^{(n)} := \frac{1}{\sqrt{n}} B_{nt}$, for all $0 \leq t \leq 1$. Point (iv) of Fact 1 tells us that $\left(B_t^{(n)}\right)_{0\leq t\leq 1}$ is a Brownian motion.

First, the continuity of the Brownian motion tell us that for all $\varepsilon > 0$, there exists $\delta > 0$ such that

$$\mathbb{P}\left(\exists\ s,t \in [0,1] \text{ with } |s-t| < \delta \text{ and } \left|B_s^{(n)} - B_t^{(n)}\right| > \varepsilon\right) \leq \frac{\varepsilon}{2}$$

We also notice that the sequence $(T_n)_{n\geq 0}$ of stopping times as defined in the Skorokhod embedding is a random walk. Thus, the Strong Law of Large Numbers yields

$$\frac{T_n}{n} \longrightarrow \mathbb{E}(T_1) = 1$$

almost surely as n tends to $+\infty$. Therefore,

$$\frac{1}{n}\sup_{k\leq n}|T_k - k| \longrightarrow 0$$

almost surely as n tends to $+\infty$. Therefore, there exists N_1 such that, for all $n \geq N_1$,

$$\mathbb{P}\left(\frac{1}{n}\sup_{k\leq n}|T_k - k| > \frac{\delta}{3}\right) \leq \frac{\varepsilon}{2}.$$

Finally, by the definition of $S^{(n)}$ and the continuity of $\left(B_s^{(n)}\right)_{0\leq s\leq 1}$, for all $t \in \left[\frac{k}{n}, \frac{(k+1)}{n}\right]$, there exists $s \in \left[\frac{T_k}{n}, \frac{T_{k+1}}{n}\right]$ such that

$$S^{(n)}(t) = \frac{1}{\sqrt{n}} B_{ns} = B_s^{(n)}. \tag{3}$$

We can thus define $N := \max\left(N_1, \frac{3}{\delta}\right)$. For all $n \geq N$, we have

$$\begin{aligned}
&\mathbb{P}\left(\sup_{0\leq t\leq 1}\left|S_t^{(n)} - B_t^{(n)}\right| > \varepsilon\right) \\
&\leq \mathbb{P}\left(\sup_{0\leq t\leq 1}\left|S_t^{(n)} - B_t^{(n)}\right| > \varepsilon \text{ with } \frac{1}{n}|T_k - k| \leq \frac{\delta}{3},\ \forall\ k \leq n\right) + \mathbb{P}\left(\frac{1}{n}\sup_{k\leq n}|T_k - k| > \frac{\delta}{3}\right) \\
&\overset{(*)}{\leq} \mathbb{P}\left(\exists\ s,t \in [0,1] \text{ with } |s-t| \leq \delta \text{ and } \left|B_s^{(n)} - B_t^{(n)}\right| > \varepsilon\right) + \frac{\varepsilon}{2} \\
&\leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.
\end{aligned}$$

Here, inequality $(*)$ follows from the fact that, if there exists $t \in [k/n, (k+1)/n]$ such that $\left|S_t^{(n)} - B_t^{(n)}\right| > \varepsilon$, then equation (3) tells us that there exists $s \in [k/n, (k+1)/n]$ with $\left|B_s^{(n)} - B_t^{(n)}\right| > \varepsilon$. Moreover, if $|T_k - k| \leq \frac{nd}{3}$ for all k, we have that $|s-t| \leq \delta$. This yields that

$$\begin{aligned}
&\mathbb{P}\left(\sup_{0\leq t\leq 1}\left|S_t^{(n)} - B_t^{(n)}\right| > \varepsilon \text{ with } \frac{1}{n}|T_k - k| \leq \frac{\delta}{3},\ \forall\ k \leq n\right) \\
&\leq \mathbb{P}\left(\exists\ s,t \in [0,1] \text{ with } |s-t| < \delta \text{ and } \left|B_s^{(n)} - B_t^{(n)}\right| > \varepsilon\right) \leq \frac{\varepsilon}{2}
\end{aligned}$$

Which concludes the proof. □

4.3 Brownian excursion

We now define an object which is closely related to the Brownian motion, the Brownian excursion.

Definition 9. Let $(B(t))_{t\geq 0}$ be a Brownian motion. Let $\tau_- := \sup\{\tau < 1 \colon B_\tau = 0\}$ and $\tau_+ := \inf\{\tau > 1 \colon B_\tau = 0\}$. A standard Brownian excursion is a random function $\mathfrak{e} \colon [0,1] \leftrightarrow \mathbb{R}_+$ defined as

$$\mathfrak{e}(t) := \frac{|B\left[(1-t)\tau_- + t\tau_+\right]|}{\sqrt{\tau_+ - \tau_-}}\ ,\qquad \forall\ 0 \leq t \leq 1.$$

Remark 4. We remark that the definition of Brownian excursion implies that $\mathfrak{e}(0) = \mathfrak{e}(1) = 0$ and $\mathfrak{e}(t) > 0$ for all $0 < t < 1$. Another way of defining a standard Brownian excursion is to say that it is the absolute value of a standard Brownian motion on $[0,1]$ conditioned on taking value of zero at 0 and 1 and on not changing sign between those two values.

Now, take a simple symmetric random walk on the integers $\left(\tilde{D}_k\right)_{n=0}^{2n}$, that is, with step distribution $\frac{1}{2}\delta_{-1} + \frac{1}{2}\delta_{+1}$; here δ_x is the Dirac distribution centered at x. We have that $\left(\tilde{D}_k\right)_{n=0}^{2n}$ has zero mean and unitary variance. Condition such random walk on being positive and to start and end at zero. This is equivalent to generating a Dyck path from the set $\mathbb{D}_{2n}$ uniformly at random. In fact, a Dyck path can be obtained from $\left(\tilde{D}_k\right)_{n=k}^{2n}$ by a similar construction as the one used for Theorem 5. The Dyck path associated to $\left(\tilde{D}_k\right)_{n=0}^{2n}$ is the function $D \colon [0,2n] \to \mathbb{R}_+$ to

$$D_t^{(2n)} := (t-k)\cdot \tilde{D}_{k+1} + (k+1-t)\cdot \tilde{D}_k\ ,\qquad \forall\ k \leq t \leq k+1$$

for all $0 \leq k \leq n-1$. The intuition provided by Donsker's Invariance Principle suggests that the process given by a Dyck path in $\mathbb{D}_n$ chosen uniformly at random, whose x and y-axes are rescaled by $1/2n$ and $1/\sqrt{2n}$ respectively, that is, $\left(\frac{1}{\sqrt{2n}} D_{2nt}^{(2n)}\right)_{0\leq t\leq 1}$, weakly converges to the standard Brownian excursion. This is indeed the case as shown in [Kai76].

Now, since Dyck paths are associated to rooted plane trees with the bijection presented in Proposition 2, there is a certain sense in which uniformly random rooted plane trees on n vertices converge (upon some rescaling condition) to a certain random object, as n tends to $+\infty$. We will make this idea precise in the following section.

4.4 The Gromov-Hausdorff-Prokhorov distance

As we previously defined, we treat graphs as measured metric spaces. We recall that, if $G = (\mathcal{V}, \mathcal{E})$ is a pair, with $\mathcal{V}$ being a set of vertices and $\mathcal{E} \subseteq \mathcal{V}^{(2)}$ being a set of edges, then the measured metric space $(\mathcal{V}, d_G, \mu_G)$ is a graph. Here, d_G being the graph distance on G and μ_G being the uniform measure on $\mathcal{V}$.

The bijection between rooted plane trees on n vertices and Dyck paths on length $2(n-1)$ that we proposed in Proposition 2, can be thought of as slicing each edge of the tree length-wise and pulling the tree apart from the root (this very explanatory parallel was taken from [DZ90]). We illustrate this idea in Figure 1.

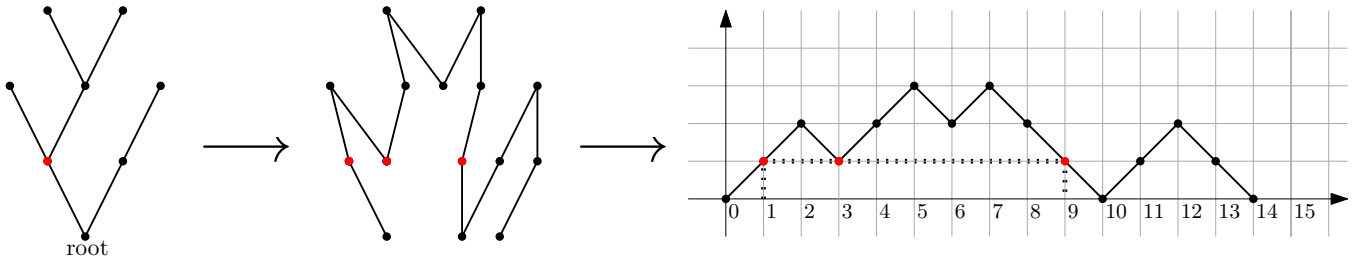

Figure 1: Transformation of a rooted plane tree into a Dyck path

Let $T = (\mathcal{V}_T, \mathcal{E}_T)$ be a rooted plane tree and let D be the Dyck path associated to T (here we see D as a function from $[0, 2n]$ to $\mathbb{R}_+$). With the help of Figure 1, notice that every point of the form $(i, D(i))$, with $i \in \mathbb{N}$, in the graph of the Dyck path, can be mapped back to the node associated to it. Notice that such mapping is not injective. Indeed, distinct points can be mapped to the same node of the tree. For example, every point of the form $(i, 0)$ maps to the root node. Notice that, given two points $(i, D(i))$, $(j, D(j))$ with $i, j \in \mathbb{N}$ and $i < j$, we have that the graph distance between the corresponding nodes in the tree is given by the function $d_D \colon \{0, \dots, 2n\} \to \mathbb{R}_+$ defined as:

$$d_D(i, j) := D(i) + D(j) - 2 \min_{i \le r \le j} D(r).$$

In particular, the graph distance of the nodes corresponding to $(i, D(i))$, $(j, D(j))$ is zero (that is, $(i, D(i))$ and $(j, D(j))$ correspond to the same node in the tree) if and only if $D(i) = D(j)$ and the minimum value of $D(r)$ is not lower than $D(i) = D(j)$ for all integers $i \le r \le j$. In other words, the graph distance of the nodes corresponding to $(i, D(i))$, $(j, D(j))$ is zero if and only if $D(i) = D(j) = \min_{i \le r \le j} D(r)$. Which corresponds to our construction in Figure 1.

Notice that d_D is not a proper distance function, it is only a pseudo-metric on the set $\Gamma := \{0, 1, \dots, 2n\}$ since, as seen above, it does not separate distinct points. One can consider the equivalence relation $\sim$ on Γ where $i \sim j$ if and only if $d_D(i, j) = 0$. Then, the pair $(\Gamma/\!\sim, d_D)$ is a metric space isomorphic to $(\mathcal{V}_T, d_T)$, where d_T is the graph distance on T.

Now, take the standard Brownian excursion $\mathfrak{e} \colon [0, 1] \to \mathbb{R}_+$. As previously remarked, in [Kai76] it is proven that $\mathfrak{e}$ can be approximated by appropriately rescaled uniformly random Dyck paths of length $2n$. Moreover, such approximation will get finer and finer as n tends to $+\infty$. Inspired by our previous discussion about how to retrieve a tree from the corresponding Dyck path, we define the following pseudo-metric $d_{\mathfrak{e}}$ on $[0, 1]$. Let $s < t$, we have

$$d_{\mathfrak{e}}(s, t) := \mathfrak{e}(s) + \mathfrak{e}(t) - 2 \min_{s \le r \le t} \mathfrak{e}(r).$$

Notice that, again, $d_{\mathfrak{e}}$ is only a pseudo-metric as, for example, $d_{\mathfrak{e}}(0, 1) = 0$. Similarly as in the discrete case, we can define the equivalence relation $\sim$ on $[0, 1]$ such that $s \sim t$ if and only if $d_{\mathfrak{e}}(s, t) = 0$. We set T to be the set $[0, 1]/\!\sim$. We also let $p \colon [0, 1] \to T$ be the projection map onto T. Let μ be the push-forward of the Lebesgue measure on T. We call μ the uniform measure on T. This leads us to the definition of the CRT.

Definition 10. Let $\mathfrak{e}$ be the standard Brownian excursion. Then, the triple $(T, d_{\mathfrak{e}}, \mu)$ as defined above is called the Continuum Random Tree.

There exist ways of making sense of the notion of a sequence of compact measured metric space converging to another compact measured metric space. One way of doing so is the Gromov-Hausdorff-Prokhorov distance. In order to let this text be self contained, we give a complete description of it. We first define the notion of Hausdorff distance. Let (X, d_X) be a compact metric space. Given a closed subset $F \subseteq X$, let F_ε be the ε-neighborhood of F, that is $F_\varepsilon := \{x \in X \colon \inf_{y \in F} d(x, y) < \varepsilon\}$. The Hausdorff distance between closed subsets C, D of X is defined as follows (see [Mie09]):

$$d_H(C, D) := \inf\{\varepsilon > 0 \colon C \subseteq D_\varepsilon,\ D \subseteq C_\varepsilon\}.$$

Now, we define the so-called Prokhorov distance. Let μ, ν be two Borel probability measures on (X, d_X). We define the Prokhorov distance as follows (see [Mie09]):

$$d_P(\mu, \nu) := \inf\left\{\varepsilon > 0 \colon \mu(F) \le \nu(F_\varepsilon) + \varepsilon \text{ for all closed } F \subseteq X\right\}.$$

As pointed out in [Mie09], the topology associated to d_P is that of weak convergence of probability spaces. Let (Y, d_Y) be another compact metric space. A function $\varphi\colon (X, d_X) \to (Y, d_Y)$ is called an isometry if it is bijective and distance-preserving. That is, if $d_X(x, y) = d_Y(\varphi(x), \varphi(y))$ for all $x, y \in X$. Two metric spaces are said to be in the same isometry class if there exists an isometry from one to the other. We define the Gromov-Hausdorff distace between metric spaces (X, d_X), (Y, d_Y) as follows (see [Mie09]):

$$d_{GH}(X, Y) := \inf_{\varphi,\psi} d_H(\varphi(X), \psi(Y)),$$

where the infimum is taken over all isometries from (X, d_X), (Y, d_Y) onto a common compact metric space (Z, d_Z).

Remark 5. We remark that d_{GH} is not a proper distance function over the set of all metric spaces. Indeed, one can see that two isometric space would have null distance. Since being isometric is an equivalence relation (see [BBI01]), d_{GH} would be a distance function over the set of all isometry classes of compact metric spaces. However, as pointed out in [BBI01], the "set" of all isometry classes of compact metric spaces is not a proper set in the sense of Set Theory, since it leads to set-theoretic paradoxes. However, we can refer to such "set" as a collection of objects and not as an object in Set-Theory. This is merely a way of shortening definitions and to properly talk about d_{GH} being a distance function.

Now, let μ, ν be Borelian probability measures on (X, d_X) and (Y, d_Y), respectively. Then (X, d_X, μ) and (Y, d_Y, ν) are measured metric spaces. Two measured metric spaces are said to be isometry-equivalent if there exists an isometry $\varphi\colon (X, d_X) \to (Y, d_Y)$ and, in addition, we have $\varphi_*(\mu) = \nu$, where $\varphi_*(\mu)$ is the pushforward measure on μ. We define the Gromov-Hausdorff-Prokhorov distance between two measured metric spaces (X, d_X, μ), (Y, d_Y, ν) as follows (see [Mie09]):

$$d_{GHP}(X, Y) := \inf_{\varphi,\psi} \{\max[d_H(\varphi(X), \psi(Y)), d_P(\varphi_*\mu, \psi_*\nu])\},$$

where the infimum is taken over all the isometries from (X, d_X), (Y, d_Y) to a common compact metric space (Z, d_Z). The same set-theoretic remarks contained in Remark 5 apply here.

We have the following theorem. The proof of its statements can be found in [Ald91], [Car16], [AM08], [PSW16], [Stu19], [MM11].

Theorem 6. Let (T_n, d_n, μ_n) be a uniform random labeled tree (resp. rooted plane tree, rooted unembedded tree, rooted unembedded binary tree, unrooted unembedded tree) on n vertices. Let (T, d, μ) be the CRT. We have that

$$\left(T_n, \frac{1}{\sqrt{n}} d_n, \mu_n\right) \xrightarrow{d_{GHD}} (T, d, \mu)$$

up to a constant factor depending on the specific family considered.

This theorem motivates the fact that Theorem 2 is stronger than Theorem 1. It also tells us that we can approximate the CRT by picking uniformly at random large trees (say, on N vertices, for N large) from the families metioned in Theorem 6. We will use this observation in section 6.

5 Random generation of trees

In this section we will take care of providing the methods that we used in our implementation in order to sample from some families of trees uniformly at random. We will also prove the correctness of such methods.

5.1 Rooted plane trees

Our strategy for sampling uniformly at random from the set $\mathbb{T}_n$ of rooted plane trees on n vertices, is that of sampling a Dyck word of length $2(n-1)$ form $\mathbb{D}_{2n-1}$ uniformly at random and subsequently construct the corresponding tree. In order to do so, we need some technical notions.

Definition 11. A sequence $x_1, x_2, \ldots, x_n$ that takes values in $\{-1, +1\}$ is k-dominating if for all $1 \leq j \leq n$, the subsequence $x_1, \ldots, x_j$ contains a number of $+1$s which is strictly greater than k times the number of -1s. If the sequence is 1-dominating we say that it is dominating.

We have this technical lemma called the 'cycle lemma'.

Lemma 1 ([DZ90])**.** Given a sequence $x_1, \dots, x_{n+m}$ valued in $\{-1, +1\}$, containing m many $+1$s and n many -1s, there are exactly $m + kn$ cyclic permutations of it which are k-dominant.

Proof ([DZ90]). Arrange the $n + m$ elements of the sequence on a cycle. Removing a subsequence of k many $+1$s followed by one -1 (note: all the removed element must be consecutive on the cycle) does not change the number of k-dominating cyclic permutations since no k-dominating permutation of the sequence could have begun with any of the $k+1$ deleted terms. The pigeon-hole principle tells us that, as long as the number of elements in the sequence is greater than or equal to $kn > 0$, there must be one such subsequence in the cycle. Let us remove such subsequences until only $+1$s remain. Any of the cyclic permutations of the original sequence that start at anyone of the $m - kn$ remaining elements is k-dominating. □

We have another technical lemma.

Lemma 2. Let $n \in \mathbb{N}$ and consider the word

$$x_1 \dots, x_{2n+1} = \underbrace{+1, \dots, +1}_{n+1 \text{ times}} \underbrace{-1, \dots, -1}_{n \text{ times}}.$$

Let also

$$S_n := \left\{ \left(x_{\sigma(1)}, \dots, x_{\sigma(2n+1)} \right) : \sigma \text{ is a permutation of the set } \{1, \dots, 2n+1\} \right\}.$$

We have that $|S_n| \geq (2n+1) \cdot C_n$, where C_n is the n-th Catalan number.

Proof. Let $y_1, \dots, y_{2n+1}$ be a permutation of the original sequence $x_1, \dots, x_{2n+1}$. There are $n!$ ways of permuting only the -1s and $(n+1)!$ ways of permuting only the $+1$s of the sequence $y_1, \dots, y_{2n+1}$. Therefore there are at least $n! \cdots (n+1)!$ ways of permuting the elements of $y_1, \dots, y_{2n+1}$ leaving it unchanged. Since there are $(2n+1)!$ permutations of the set $\{1, \dots, 2n+1\}$, we have that

$$|S_n| \geq \frac{(2n+1)!}{n! \cdot (n+1)!} = \frac{2n+1}{n+1} \cdot \begin{pmatrix} 2n \\ n \end{pmatrix} = (2n+1) \cdot C_n.$$

□

We now define a new object that will be useful in this section.

Definition 12. A quasi-Dyck word of length $2n+1$ is a sequence $a_0, a_1, \dots, a_{2n}$ that takes values in $\{-1, +1\}$ such that $a_0 = +1$ and $a_1, \dots, a_{2n}$ is a Dyck word.

It is easy to see that quasi-Dyck word are dominant sequences. Moreover, each quasi-Dyck word contains $n+1$ many $+1$s and n many -1s.

We also have the following lemma.

Lemma 3. Let P_n be the set of cyclic permutations of quasi-Dyck word of length $2n+1$. We have that $|P_n| = (2n+1) \cdot C_n$, where C_n is the n-th Catalan number.

Proof. Since C_n counts the number of Dyck words of length $2n$, it also counts the number of quasi-Dyck words of length $2n+1$.

For each quasi-Dyck word of length $2n+1$, there exist at most $2n+1$ distinct cyclic permutations of it. Therefore $|P_n| \leq (2n+1) \cdot C_n$. □

We finally obtain the following proposition.

Proposition 6. Given $n \geq 1$, we have that $S_n = P_n$ and $|S_n| = |P_n| = (2n+1) \cdot C_n$.

Proof. By the cycle lemma, applied with $m = n+1$, $n = n$ and $k = 1$, we have that, for all $y_1, \dots, y_{2n+1} \in S_n$, there exists a cyclic permutation of it that is dominant. Therefore we have the existence of a (unique) cyclic permutation σ of the set $\{1, \dots, 2n+1\}$ such that the sequence $y_{\sigma(1),\dots,\sigma(2n+1)}$ is dominant. In particular, this means that $y_{\sigma(1)} = +1$ and

$$\sum_{j=2}^{i} y_{\sigma(j)} \geq 0 \qquad \forall\, 2 \leq i \leq 2n+1 \qquad \text{and} \qquad \sum_{j=2}^{2n+1} y_{\sigma(j)} = 0.$$

This means that $y_{\sigma(2)}, \dots, y_{\sigma(2n+1)}$ is a Dyck word and, since $y_{\sigma(1)} = +1$, we get that $y_{\sigma(1),\dots,\sigma(2n+1)}$ is a quasi-Dyck word. Since the inverse of a cyclic permutation is still a cyclic permutation, we have that $y_1, \dots, y_{2n-1}$ is the cyclic permutation of a quasi-Dyck word. Therefore, $S_n \subseteq P_n$. Thus, combining lemmas 2 and 3, we have that

$$(2n+1) \cdot C_n \leq |S_n| \leq |P_n| \leq (2n+1) \cdot C_n$$

and so $S_n = P_n$ and $|S_n| = |P_n| = (2n+1) \cdot C_n$. □

Now, the fact that for any quasi-Dyck word there are at most $2n+1$ distinct cyclic permutations, together with the fact $|P_n| = (2n+1) \cdot C_n$, implies that all the cyclic permutations of a given quasi-Dyck word are distinct. Moreover, given two distinct quasi-Dyck words, the relation $\sim$ defined as

$$w_1 \sim w_2 \iff w_1 \text{ and } w_2 \text{ are cyclic permutations of the same quasi-Dyck word}$$

is an equivalence relation on P_n. Moreover, all the equivalence classes in $P_n/\sim$ have cardinality $2n+1$.

In light of this discussion, we propose the following algorithm to sample uniformly at random from the set $\mathbb{D}_n$ of Dyck words of length $2n$.

1. Generate a permutation σ of $\{1, \dots, 2n+1\}$ uniformly at random.
2. Consider the sequence $x_{\sigma(1)}, \dots, x_{\sigma(2n+1)}$ where $x_1 \dots, x_{2n+1} = \underbrace{+1, \dots, +1}_{n+1 \text{ times}} \underbrace{-1, \dots, -1}_{n \text{ times}}$.
3. Compute the unique quasi-Dyck word that is in the same equivalence class (induced by $\sim$) as $x_{\sigma(1)}, \dots, x_{\sigma(2n+1)}$ and take the corresponding Dyck word.

In order to choose a permuation of the set $\{1, \dots, 2n+1\}$ uniformly at random, we can use the Knuth shuffle that is described in [Ebe16]. We will denote the function that returns a uniformly random permutation of the set $\{1, \dots, 2n+1\}$ by KnuthShuffle$(2n+1)$.

In order to compute the quasi-Dyck word in the same equivalence class as $x_{\sigma(1)}, \dots, x_{\sigma(2n+1)}$, it suffices to take the index i such that $\sum_{j=1}^{i} x_{\sigma(j)}$ is minimal and the cumulative sum $\sum_{j=i+1}^{k} x_{\sigma(j)} > 0$ for all $i+1 \leq k \leq 2n$. Then, take the sequence $x_{\sigma(i)}, \dots, x_{\sigma(2n+1)}, x_{\sigma(1)}, \dots, x_{\sigma(i-1)}$, which is a cyclic permutation of $x_{\sigma(1)}, \dots, x_{\sigma(2n+1)}$. We can see that $x_{\sigma(i)}, \dots, x_{\sigma(2n+1)}, x_{\sigma(1)}, \dots, x_{\sigma(i-1)}$ is a Dyck word as follows. Since $x_{\sigma(1)}, \dots, x_{\sigma(2n+1)}$ is the cyclic permutation of a quasi-Dyck word, which is a dominating sequce, we get that i is unique. From this we get that $\sum_{j=i}^{k} x_{\sigma(j \mod 2n+1)} \geq 1$ for all k. Finally, as $x_1, \dots, x_{2n+1}$ contains $n+1$ many $+1$s and n many -1s, we get that $\sum_{j} x_{\sigma(j \mod 2n+1)} = 1$. Therefore, $x_{\sigma(i)}, \dots, x_{\sigma(2n+1)}, x_{\sigma(1)}, \dots, x_{\sigma(i-1)}$ is the quasi-Dyck word we were looking for.

Finally, in order to generate the corresponding rooted plane tree, we can use the algorithm DyckWordToTree of section 3.

Algorithm 4: RandomRootedPlaneTree

input : An integer n which represents the number of nodes of the trees.
output: A rooted plane tree sampled uniformly at random from the set $\mathbb{T}_n$ of rooted plane trees on n vertices.

$\sigma \leftarrow$ KnuthShuffle$(2n+1)$
apply σ to the indices of the sequence $x_1, \ldots, x_{2n+1} = \underbrace{+1, \ldots, +1}_{n+1 \text{ times}}, \underbrace{-1, \ldots, -1}_{n \text{ times}}$.
compute the unique index $i \in \{1, \ldots, 2n+1\}$ such that $\sum_{j=1}^{i} x_{\sigma(j)}$ reaches its minimum.
obtain the sequence $x_{\sigma(i+1)}, \ldots, x_{\sigma(2n+1)}, x_{\sigma(1)}, \ldots, x_{\sigma(i)}$, which is a Dyck word.
return DyckWordToTree$(x_{\sigma(i+1)}, \ldots, x_{\sigma(2n+1)}, x_{\sigma(1)}, \ldots, x_{\sigma(i)})$

5.2 Rooted unembedded trees

In order to generate random rooted unembedded trees, we use the algorithm RanRut contained in [NW75].

Throughout this section we will use the following notation. Let $T_1, T_2, \ldots, T_k$ be rooted unembedded trees with roots $c_1, c_2, \ldots, c_k$ respectively. For all $1 \leq i \leq k$, make m_i copies of T_i and call its roots $c_1^{(i)}, \ldots, c_{m_i}^{(i)}$. Let T the rooted unembedded tree that has a root node r and it is connected to the m_i copies of T_i for all $1 \leq i \leq k$ by the edges $\{r, c_1^{(i)}\}, \ldots, \{r, c_{m_i}^{(i)}\}$. We denote $T = m_1 \otimes T_1 + \ldots + m_k \otimes T_k$. Notice that, in this case, the operation '+' is commutative since T is unembedded. Also, the expression '$m_i \otimes T_i$' can be replaced by $\underbrace{T_i + \cdots + T_i}_{m_i \text{ times}}$.

We also note that the set of rooted unembedded trees on n vertices will be denoted by $\mathbb{U}_n$.

We have the following important lemma.

Lemma 4 (e.g. [NW75])**.** Let $n \geq 1$ and let $t_n = |\mathbb{U}_n|$. If we set $t_j = 0$ for all $j \leq 0$, we get that

$$(n-1)t_n = \sum_{d \geq 1} \sum_{k \geq 1} d \cdot t_d \cdot t_{n-kd}.$$

Proof. We propose the following counting argument. Consider the set $\overline{\mathbb{U}}_n$ of unembedded rooted trees on n vertices with a distinguished vertex v different from the root. That is

$$\overline{\mathbb{U}}_n := \{(T, v) \colon T \in \mathbb{U}_n \text{ and } v \text{ is a non-root node of } T\}.$$

Since $|\mathbb{U}_n| = t_n$, then $\left|\overline{\mathbb{U}}_n\right| = (n-1)t_n$.

We now count the number of elements in $\overline{\mathbb{U}}_n$ in a different way. Let $(T, v) \in \overline{\mathbb{U}}_n$ and let $c_1, \ldots, c_k$ be the children of the root r of T. Let $T_1, \ldots, T_k$ be rooted unembedded subtrees of T at r, where the root of T_i corresponds to the node c_i for all $1 \leq i \leq k$. Since T is unembedded, we can assume without loss of generality that T_1 is the subtree of T that contains v. Let $\delta(T, v) = d \geq 1$ be the number of vertices of T_1. Then, we have t_d choices for T_1 and d choices for v, which makes $d \cdot t_d$ choices for the pair (T_1, v). Again, since T is unembedded, we can suppose without loss of generality that $T_1, \ldots, T_k$ are isomorphic to T_1 and that T_l is not isomorphic to T_1 for all $l > k$, for some integer that we define $\kappa(T, v) = k \geq 1$. For a given value of d and a given value of k, we have $d \cdot t_{d^\cdot}, t_{n-kd}$ different choices for the pair (T_v).

Now, for any $d, k \geq 1$, let

$$\overline{\mathbb{U}}_n^{(d,k)} := \left\{(T, v) \in \overline{\mathbb{U}}_n \colon \delta(T, v) = d \text{ and } \kappa(T, v) = k\right\}.$$

We have

$$\overline{\mathbb{U}}_n = \bigsqcup_{d \geq 1} \bigsqcup_{k \geq 1} \overline{\mathbb{U}}_n^{(d,k)}.$$

Thus

$$\left|\overline{\mathbb{U}}_n\right| = \sum_{d \geq 1} \sum_{k \geq 1} \left|\overline{\mathbb{U}}_n^{(d,k)}\right| = \sum_{d \geq 1} \sum_{k \geq 1} d \cdot t_d \cdot t_{n-kd}.$$

Therefore we have that $(n-1)t_n = \sum_{d \geq 1} \sum_{k \geq 1} d \cdot t_d \cdot t_{n-kd}$, which is the desired result. □

In light of Lemma 4, we get that

$$\sum_{d\geq 1}\sum_{k\geq 1}\frac{d\cdot t_d\cdot t_{n-kd}}{(n-1)t_n}=1. \qquad (*)$$

We can interpret the addends of the LHS of $(*)$ as probabilities. This inspires the following algorithm for generating an element of $\mathbb{U}_n$ uniformly at random. Such algorithm is called RanRut and was first presented in [NW75].

1. Choose a pair of integers $d,k\geq 1$ at random with the probability distribution $\mathbb{P}(d,k)=\frac{d\cdot t_d\cdot t_{n-kd}}{(n-1)t_n}$.
2. Inductively choose a rooted unembedded tree S_1 on $n-kd$ vertices uniformly at random (that is, with probability $1/t_{n-kd}$).
3. Inductively choose a rooted unlabelled tree S_2 on d vertices uniformly at random (that is, with probability $1/t_d$).
4. Construct and return the tree $T=S_1+k\otimes S_2$.

To see why this yields an element of $\mathbb{U}_n$ uniformly at random, let T be a rooted unembedded tree on n vertices. There exist distincts trees $T_1,\ldots,T_j$ respectively with $l_1,\ldots,l_j$ vertices and integers $m_1,\ldots,m_j$ such that

$$T=m_1\otimes T_1+\cdots+m_j\otimes T_k \quad \text{ and } \quad l_1\cdot m_1+\cdots+l_j\cdot m_j=n-1.$$

The tree T is constructed by the procedure described above if and only if all of the following are verified. For some $1\leq i\leq j$ we get $S_1=T_i$ and $1\leq k\leq m_i$ and $d=l_i$ and $S_2=m_1\otimes T_1+\cdots+m_{i-1}\otimes T_{i-1}+(m_i-k)\otimes T_i+m_{i+1}\otimes T_{i+1}+\cdots m_j\otimes T_j$.

The probability of sampling T is thus,

$$\sum_{i=1}^{j}\sum_{k=1}^{m_i}\frac{l_i\cdot t_{n-k\cdot l_i}\cdot t_{l_i}}{(n-1)t_n}\cdot\frac{1}{t_{n-k\cdot l_i}}\cdot\frac{1}{t_{l_i}}=\frac{1}{(n-1)t_n}\sum_{i=1}^{j}\sum_{k=1}^{m_i}l_i=\frac{1}{(n-1)t_n}\sum_{i=1}^{j}m_i\cdot l_i=\frac{1}{(n-1)t_n}\cdot t_n=\frac{1}{t_n}.$$

Which is the desired result. We summarize this algorithm as follows. We remark that some memoization techniques should be implemented in order to make the algorithm run efficiently (i.e. in polynomial time).

Algorithm 5: RanRut

input : an integer $n \geq 1$ of vertices.
output: RANRUT(n), a rooted unembedded tree generated uniformly at random from $\mathbb{U}_n$.

```
RANRUT(n)
    if n=1 then
        return The tree containing only one node, which is also the root
    (d, k) ← integers chosen with probability (d·T_CARD(d)·T_CARD(n−kd)) / ((n−1)·T_CARD(n))
    S_1 ← RANRUT(n − kd)
    S_1 ← RANRUT(d)
    T ← S_1 + k ⊗ S_2
    return T

T_CARD(d)
    if d = 0 then
        return 0
    if d = 1, 2 then
        return 1
    cumulative_sum ← 0
    for d from 1 to n − 1 do
        for k from 1 to ⌊n/d⌋ do
            cumulative_sum += d · T_CARD(d) · T_CARD(n − kd)
    return cumulative_sum/(n − 1)
```

5.3 Free trees

In the following section, prove correctness and we describe the implementation of the algorithm Free contained in [Wil81]. Such algorithm allows us to sample uniformly at random elements from the set $\mathbf{T}_n$, which is the set of free trees on $n \geq 1$ vertices.

Given a vertex v, we define the *weight* of v as the number of vertices of the largest subtree containing v. A vertex of minimum weight is call a *centroid*. We proceed giving some facts about centroids form [Knu97].

Fact 2. Every tree has either one or two centroids. The trees with two centroids are called bi-centroidal trees.

Fact 3. If a tree T has two centroids, then they are connected by an edge e. If one removes e from T, one would have two trees with the same number of vertices.

Fact 4. A vertex v is the unique centroid of a tree with n vertices if an only if its weight is at most $(n-1)/2$.

Notice that Fact 3 implies that, if a tree has two centroids, then it must leave an even number of vertices.

We give two procedures to generate free trees with one centroid and two centroids separately. Let $n = 2k \geq 1$ and let $\mathbf{T}_n^{(2)}$ be the set of free trees on n vertices with two centroids. To randomly generate free trees with two centroids, Fact 3 tells us that we can generate two rooted unordered trees T_1, T_2 on k vertices with roots c_1 and c_2 respectively (for example with the algorithm RanRut), join their roots with the edge $\{c_1, c_2\}$, and subsequently "forget" the fact that roots are distinguished vertices. However, notice that if the generation of T_1 and T_2 is uniform and independent from the set $\mathbb{U}_k$, the free trees for which the removal of the edge $\{c_1, c_2\}$ (following the above construction), leaves two isomorphic trees (where the isomorphism preserves the property of being a root), will occur only half as often in our generation.

We can give more "weight" to such trees in the following way. Let $t_k^{(2)} = \left|\mathbf{T}_k^{(2)}\right|$.

1. With probability $1/(t_k+1)$, we generate uniformly at random a rooted tree T' from $\mathbb{U}_k$, we then set $T_1 := T_2 := T'$.
2. With probability $1 - 1/(t_k+1)$, we generate T_1 and T_2 uniformly at random and independently from $\mathbb{U}_k$.
3. We then return the tree constructed by joining the roots of T_1 and T_2 with an edge and forgetting the distinction of the roots.

To see why this method allows us to generate trees uniformly at random from $\mathbf{T}_n^{(2)}$, we compute directly the probability of generating a tree $T \in \mathbf{T}_n^{(2)}$.

Let $T \in \mathbf{T}_n^{(2)}$. Suppose that T is such that, by deleting the edge that connects the two centroids c_1, c_2, the two connected components T_1, T_2 seen as two rooted unembedded trees with roots c_1 and c_2 are isomorphic. Let T' be a tree isomorphic to T_1 and T_2. The probability of generating T is

$$\mathbb{P}(T) = \frac{1}{t_k+1}\mathbb{P}(T') + \frac{t_k}{t_k+1}\left(\mathbb{P}(T')\cdot\mathbb{P}(T')\right) = \frac{1}{t_k+1}\cdot\frac{1}{t_k} + \frac{1}{t_k+1}\cdot\frac{1}{t_k} = \frac{2}{(t_k+1)\cdot t_k}.$$

Similarly, if the (rooted unordered) trees T_1 and T_2 are not isomorphic, the probability of generating T is

$$\mathbb{P}(T) = \frac{t_k}{t_k+1}\cdot\left(\mathbb{P}(T_1)\cdot\mathbb{P}(T_2) + \mathbb{P}(T_2)\cdot\mathbb{P}(T_1)\right) = \frac{t_k}{t_k+1}\cdot\left(\frac{1}{t_k}\cdot\frac{1}{t_k} + \frac{1}{t_k}\cdot\frac{1}{t_k}\right) = \frac{2}{(t_k+1)\cdot t_k}.$$

Therefore, each tree is generated equally likely. The formulae derived above also tells us that $t_{2k}^{(2)} = \frac{(t_k+1)\cdot t_k}{2}$.

We summarize our algorithm as follows.

Algorithm 6: RandomBicentroidalTree

input : an even integer $n = 2k$ of nodes.
output: a free tree `BICENTER`(n) with two centroids, generated uniformly at random from the set $\mathbf{T}_n^{(2)}$.

```
BICENTER(n)
    m ← an integer drawn uniformly at radom from the set {1, ..., t_n + 1}
    if m = 1 then
        T' ← RANRUT(k)
        T_1, T_2 ← two copies of T'
    else
        T_1 ← RANRUT(k)
        T_2 ← RANRUT(k)
    T ← the tree obtained by joining T_1 and T_2 by an edge connecting their roots
    T ← the tree T, 'forgetting' the distinction of the root nodes
    return T
```

We now describe how to generate free trees uniformly at random from the set $\mathbf{T}_n^{(1)}$ of free trees with one centroid on n vertices. Let $t_n^{(1)} = \left|\mathbf{T}_n^{(1)}\right|$ One can see that the set $\mathbf{T}_n^{(1)}$ is in natural bijection with the set of rooted unlabelled tree on n vertices with the centroid as a root. By Fact 4, the problem can be reformulated as the generation of rooted unordered trees on n vertices in which all the subtrees of the root have at most $(n-1)/2$ vertices. In practice, this amounts to generating uniformly at random a forest on $n-1$ vertices of rooted plane trees $T_1, \ldots, T_k$, each of which having at most $(n-1)/2$ vertices. Then, one returns the tree $T = T_1 + \ldots + T_k$, 'forgetting' the distinction of the root.

Now, using the notation of [Wil81], let $\alpha(m,q)$ be the number of forests of rooted unordered trees in which each connected component has at most q vertices and the total number of vertices is m.

With a similar (but long) argument as the one used in section 5.2, one can prove for all $m \geq 1$, and setting $\alpha(0,1) = 1$, that

$$m\cdot\alpha(m,q) = \sum_{k\geq 1}\sum_{d=1}^{q} d\cdot\alpha(m-k\cdot d)\cdot\alpha(d-1,q).$$

A proof of a vast generalisation of the above formula can be found in [NW75]. We give an algorithm that generates a random forest of rooted unordered tree in which each connected component has at most q vertices and the total number of vertices is m A full proof of the correctness of the algorithm can be found in [Wil81].

Algorithm 7: RandomForest

input : integers m and q.
output: a forest, $\texttt{FOREST}(m, q)$ drawn uniformly at random from the set of all forest of rooted unembedded trees in which each connected components has at most q vertices and the total number of vertices is m.

$\texttt{FOREST}(m, q)$
  if $m = 0$ **then**
    return the empty forest
  $(d, k) \leftarrow$ a pair of integers with $1 \leq d \leq q$ and $k \geq 1$ with probability $\frac{d \cdot \texttt{ALPHA\_CARD}(m-k\cdot d,q)\cdot \texttt{T\_CARD}(d)}{m \cdot \texttt{ALPHA\_CARD}(m,q)}$
  $F' \leftarrow \texttt{FOREST}(n - k \cdot d, q)$
  $T' \leftarrow \texttt{RANRUT}(d)$
  $F \leftarrow$ the union of F' and k copies of T'
  return F

$\texttt{ALPHA\_CARD}(m, q)$
  input : integers m and q
  output: the value of $\alpha(m, q)$
  $\texttt{cumulative\_sum} \leftarrow 0$
  for k from 1 to m **do**
    for d from 1 to $\lfloor \frac{m}{q} \rfloor$ **do**
      $\texttt{cumulative\_sum}$ += $d \cdot \texttt{ALPHA\_CARD}(m - k \cdot d) \cdot \texttt{ALPHA\_CARD}(d - 1, q)$
  return $\texttt{cumulative\_sum}$

Finally, in order to generate trees uniformly at random from the set $\mathbf{T}_n$ of free trees on n vertices, we can combine the two above algorithms in the following way.

Case 1. If n is odd, we can just draw elements form the set $\mathbf{T}_n^{(1)}$ uniformly at random (see Fact 3).

Case 2. If $n = 2k$ is even, then, we have that $\mathbf{T}_n = \mathbf{T}_n^{(1)} \sqcup \mathbf{T}_n^{(2)}$. As we previously remarked, we have $\left|\mathbf{T}_n^{(1)}\right| = \alpha\left(n-1, \frac{n-1}{2}\right)$ and $\left|\mathbf{T}_n^{(2)}\right| = \frac{(t_k+1)\cdot t_k}{2}$. Thus, $f_n := |\mathbf{T}_n| = \alpha\left(n-1, \frac{n-1}{2}\right) + \frac{(t_k+1)\cdot t_k}{2}$. With these observations, in order to uniformly sample from $\mathbf{T}_n$, we can sample sample uniformly from $\mathbf{T}_n^{(2)}$ with probability $\frac{(t_k+1)\cdot t_k}{2f_n}$ and from $\mathbf{T}_n^{(1)}$ with probability $1 - \frac{(t_k+1)\cdot t_k}{2f_n}$. We can sample uniformly at random from $\mathbf{T}_n^{(2)}$ with Algorithm 6 and from $\mathbf{T}_n^{(1)}$ with the help of Algorithm 7.

We summarize this approach in the following algorithm.

Algorithm 8: Free

input : an integer $n \geq 1$ of nodes
output: a free tree sampled uniformly at random from the set $\mathbf{T}_n$.

if $n = 2k$ is even **then**
  $m \leftarrow$ an integer drawn uniformly at random from the set $\{1, \ldots, f_n\}$
  if $m < \frac{(\texttt{T\_CARD}(k)+1)\cdot \texttt{T\_CARD}(k)}{2}$ **then**
    $T \leftarrow \texttt{BICENTER}(n)$
  else
    $T_1, \ldots, T_k \leftarrow \texttt{FOREST}(n-1, \frac{n-1}{2})$
    $T \leftarrow T_1 + \cdots + T_k$
else
  $T_1, \ldots, T_k \leftarrow \texttt{FOREST}\left(n-1, \frac{n-1}{2}\right)$
  $T \leftarrow T_1 + \cdots + T_k$
  return T

5.4 Labeled trees

In order to generate labeled trees we exploit a known correspondence between labelled trees and the so-called Prüfer code/sequences [Prü18]. A Prüfer sequence of length $n-2$ is a word of length $n-2$ from the alphabet $\{0,\dots,n-1\}$ (unless differently specified).

Let $n \geq 2$ and $\mathbb{L}_n$ be the set of labelled trees on n vertices (the labels are all distinct and, unless differently specified, they are chosen from the set $\{0,\dots,n-1\}$). Let $\mathbb{P}_{n-2}$ be the set of Prüfer sequences of length $n-2$ (we adopt the convention that $\mathbb{P}_0$ is the set containing only the empty word). The main result of this section is the construction of a bijection Ψ_n between $\mathbb{P}_n$ and $\mathbb{L}_n$. The bijection Ψ_n is constructed as follows.

Algorithm 9: PRÜFER

input : a labeled tree T
output: the Prüfer sequence $\texttt{PRUFER}(T)$ associated to T

```
PRUFER(n)
    do until only one edge is left
        v ← the node of degree 1 with the lowest label
        yield the label of the unique neighbor of v
        delete the v and the only edge adjacent to it
```

We remark that in a tree with two or more nodes, there is always a node of degree 1, at least there would be a cycle. For all trees $T \in \mathbb{L}_n$, we set $\Psi_n(T) := \texttt{PRUFER}(T)$. By the analysis of this code we can deduce an important fact.

Fact 5. Given a labeled tree T, any vertex v of degree d and label l, appears exactly $d-1$ times in the Prüfer code.

We are ready to prove the following proposition.

Proposition 7. For all $n \geq 2$, we have that Ψ_n is a bijection.

Proof. We first prove that Ψ_n is injective by induction on n.

Base case. Since the sets $\mathbb{L}_2$ and $\mathbb{P}_0$ only contain one element, the statement is easy to prove.

Induction step. Supposet that the statement is true for some $n \geq 2$. Let $T_1, T_2 \in \mathbb{L}_{n+1}$ be distinct trees. Let v_1, v_2 be the lowest-labelled vertices of degree 1 of T_1 and T_2 respectively. If the labels of the unique neighbors of v_1 and v_2 are different, then $\texttt{PRUFER}(T_1)$ and $\texttt{PRUFER}(T_2)$ are clearly different. If the labels of v_1 and v_2 are different, then, Fact 5, tells us that $\texttt{PRUFER}(T_1)$ and $\texttt{PRUFER}(T_2)$ are different. Otherwise, suppose that the labels of v_1 and v_2 are the same and so are the labels of their unique neighbors. Denote the labels of the neighbors of v_1 and v_2 by $l_1 = l_2 = l$. Consider the trees T_1' and T_2' obtained by removing, respectively from T_1 and T_2, the vertices v_1 and v_2 and the unique edges adjacent to them. Then T_1' and T_2' are different. Since they are labeled trees on n vertices, the induction hypothesis tells us that $\texttt{PRUFER}(T_1')$ and $\texttt{PRUFER}(T_2')$ are different. Since $\texttt{PRUFER}(T_1) = l, \texttt{PRUFER}(T_1')$ and $\texttt{PRUFER}(T_2) = l, \texttt{PRUFER}(T_2')$, we have that $\texttt{PRUFER}(T_1)$ and $\texttt{PRUFER}(T_2)$ are different. Which is the desired result.

Thus Ψ_n is injective. We now prove that it is surjective. We do so by induction on $n \geq 2$.

Base case. Since the sets $\mathbb{L}_2$ and $\mathbb{P}_0$ only contain one element, the statement is easy to prove.

Induction step. Suppose that the statement is true for some $n \geq 2$. We are going to prove it for $n+1$. Let $S = s_1, \dots, s_{n-1}$ be a word on the alphabet $\{0,\dots,n\}$. Let i be the smallest element in the alphabet that does not appear in S and let $S' = s_2, \dots, s_{n-1}$. By the induction hypothesis, there exists a labelled tree on n vertices such that its Prüfer code (on the alphabet $\{0,\dots,n\} \setminus \{s_i\}$) is S'. Let w be the vertex of T' labelled by s_1 (which is different than s_i). We construct the tree T by adding to T' an new vertex v labeled i and the edge $\{v, w\}$.

By construction, we get that $\texttt{PRUFER}(T) = s_1, \texttt{PRUFER}(T') = s_1, s_2, \dots, s_{n-1}$, which is the desired result.

Therefore the map Ψ_n si bijective. □

Proposition 7 proves Cayleys formula, that is, $|\mathbb{L}_n| = |\mathbb{P}_{n-2}| = n^{n-2}$.

Proposition 7, naturally gives us a way of sampling from $\mathbb{L}_n$ uniformly at random. We can just generate a Prüfer sequence from $\mathbb{P}_{n-2}$ uniformly at random and construct the labeled tree associated to it. Such construction can be achieved as in the proof of surjectiveness of Ψ_n in Proposition 7. We summarize the above approach in the following algorithm.

Algorithm 10: RandomLabeledTree

input : an integer $n \geq 2$ of nodes
output: a labeled tree `LABELED`(n), chosen uniformly at random from the set $\mathbb{L}_n$

```
LABELED(n)
    T ← the empty tree
    s_1, ..., s_{n-2} ← integers chosen independently and uniformly at random form the alphabet {0, ..., n-1}
    S ← (s_1, ..., s_{n-2})
    alphabet ← {0, ..., n-1}
    while S is non-empty and alphabet has size strictly greater than 2 do
        i ← the lowest element in alphabet that does not appear in S
        s_j ← the first element of S
        if T does not contain a vertex labeled i then
            add a vertex v_i labeled i to T
        if T does not contain a vertex labeled s_j then
            add a vertex v_{s_j} labeled s_j to T
        add the vertex {v_i, v_{s_j}} to T
        alphabet ← alphabet \ {v_i}
        S ← s_{j+1}, ..., s_{n-2}
    return T
```

5.5 Rooted unembedded binary trees

Let $\mathbb{B}_n$ be the set of rooted unembedded binary trees on n vertices and let $b_n := |\mathbb{B}_n|$. We propose an algorithm to sample from $\mathbb{B}_n$ uniformly at random. We first introduce the notion of unbalanced right-heavy trees.

Definition 13. Let T be a rooted plane tree with root r. Let $c_1 < \cdots < c_k$ be the children of r. Let $T_1, \ldots, T_k$ be the subtrees of T at r, where T_i is rooted at c_i for all $1 \leq i \leq k$. Let $l_1, \ldots, l_k$ be the number of nodes of $T_1, \ldots, T_k$ respectively. T is said to be unbalanced right-heavy if we have $l_1 \leq \ldots \leq l_k$ and $T_1, \ldots, T_k$ are unbalanced right-heavy.

Let $\mathbb{H}_n$ be the set of rooted binary unbalanced right-heavy trees on n vertices. There is a natural bijection between $\mathbb{B}_n$ and $\mathbb{H}_n$. Indeed, Let T be a tree in $\mathbb{B}_n$. Let r be the root of T and c_1, c_2 be the children of r. Let T_1, T_k be the subtrees of T at r, where T_1 (resp. T_2) is rooted at c_1 (resp. c_2). Also, let l_1 (resp. l_2) be the number of nodes of T_1 (resp. T_2). One can obtain a corresponding tree in $\mathbb{H}_n$ by ordering the children c_1, c_2 of r in such a way that $c_i < c_j$ if $l_i < l_j$ for $i, j \in \{1, 2\}$. If $c_1 < c_2$ we say that T_1 (resp. T_2) is the left (resp. right) subtree of T at r and vice versa.

Thus, in order to sample elements uniformly at random from $\mathbb{B}_n$, we can just sample trees from $\mathbb{H}_n$ uniformly at random and subsequently "forget" the planar embedding.

Now, let $n \geq 1$ and let $j \geq 0$. We define $\mathbb{H}_n^{(j)}$ be the set of $T \in \mathbb{H}_n$ such that the left subtree of T at its root has j vertices. We have that

$$\mathbb{H}_n = \bigsqcup_{j=0}^{\lfloor (n-1)/2 \rfloor} \mathbb{H}_n^{(j)}.$$

Notice that, if $j < \lfloor (n-1)/2 \rfloor$, we have $\left|\mathbb{H}_n^{(j)}\right| = b_j \cdot b_{n-j-1}$. Otherwise, if $n = 2k+1$ is odd, we have

$\left|\mathbb{H}_n^{(k)}\right| = \begin{pmatrix} \mathbb{H}_n^{(k)}+1 \\ 2 \end{pmatrix} = \frac{1}{2}(c_k+1)\cdot c_k$. By which we have the formulae

$$c_{2k} = |\mathbb{H}_{2k}| = c_0 \cdot c_{2k-1} + c_1 \cdot c_{2k-2} + \cdots + c_{k-1} \cdot c_k = \sum_{j=0}^{k-1} c_j \cdot c_{n-j-1}, \tag{4}$$

$$c_{2k+1} = |\mathbb{H}_{2k+1}| = c_0 \cdot c_{2k} + c_1 \cdot c_{2k} + \cdots + c_{k-1} \cdot c_{k-1} \cdot c_{k+1} + \frac{1}{2}(c_k+1)\cdot c_k = \sum_{j=0}^{k-1} c_j \cdot c_{n-j-1} \ + \frac{1}{2}(c_k+1)\cdot c_k. \tag{5}$$

Formulae (4) and (5) give us the following algorithm to draw trees uniformly at random from the set $\mathbb{B}_n$.

1. Draw at number j from the set $\{0,\ldots,\lfloor(n-1)/2\rfloor\}$ with probability $\frac{c_j\cdot c_{n-j-1}}{c_k}$ if $j<(n-1)/2$ and with probability $\frac{(c_k+1)\cdot c_k}{2}$ if $j=(n-1)/2$.

2. Draw uniformly at random a tree from $\mathbb{H}_n^{(j)}$ and return the associated tree in $\mathbb{B}_n$.

Now, if $j<(n-1)/2$, drawing a tree from $\mathbb{H}_n^{(j)}$ uniformly at random, amounts to draw trees T_1 and T_2 form $\mathbb{H}_j$ and $\mathbb{H}_{n-j-1}$ independently and uniformly at random and then return the tree $T=T_1+T_j$, 'forgetting' the distinction of the root node.

This procedure does not work when $j=(n-1)/2$. This can only happen when $n=2k+1$ is odd. If we apply the above procedure to $\mathbb{H}_k$, then the trees which have isomorphic left and right subtrees occur only half as often. We use a similar trick as the one used in RanRut that is, we give more 'weight' to the symmetric trees.

We summarize this approach in the following algorithm. We notice that in the algorithm below, the function `RANBIN_AUX`(n), uniformly samples from $\mathbb{H}_n$ and the function `RANBIN`(n) returns the corresponding tree in $\mathbb{B}_n$.

Algorithm 11: RandomBinaryTree

input : a integer n of nodes
output: a tree, `RANBIN`(n), drawn uniformly at random from the set $\mathbb{B}_n$

`RANBIN`(n)
  $T \leftarrow$ `RANBIN_AUX`(n)
  $T \leftarrow$ the tree T, 'forgetting' the distinction of the root node and the ordering of the children of a node.

`RANBIN_AUX`(n)
  if $n=1$ **then**
    return the only tree in $\mathbb{H}_1$
  $j \leftarrow$ an integer drawn randomly from the set $\{0,\ldots,\lfloor(n-1)/2\rfloor\}$ where for all $i<(n-1)/2$, $\mathbb{P}(\text{drawing } i) = \frac{c_i\cdot c_{n-i-1}}{c_n}$ and for $i=n(n-1)/2$, $\mathbb{P}(\text{drawing } i) = \frac{(c_k+1)\cdot c_k}{2}$ if $j=(n-1)/2$
  $l \leftarrow$ an integer drawn uniformly at random from the set $\{1,\ldots,c_j+1\}$
  if $j=(n-1)/2$ and $l=c_j+1$ **then**
    $T' \leftarrow$ `RANBIN_AUX`(j)
    $T_1, T_2 \leftarrow$ two copies of T'
  else
    $T_1 \leftarrow$ `RANBIN_AUX`(j)
    $T_2 \leftarrow$ `RANBIN_AUX`$(n-j-1)$
  $T \leftarrow T_1 + T_2$

6 Simulations

In view of Theorem 6, we can test the predictions of Theorem 1 and 2 by studying the distribution of the Voronoi vector induced by k agents chosen uniformly at random among the vertices of a tree T, which is, in turn, chosen uniformly at random from any of the following families: labeled trees, rooted plane trees, rooted unembedded trees, rooted unembedded binary trees, unrooted unembedded trees on n vertices. To do so, we can use a variant of Algorithm 1.

Algorithm 12: Simulation for Theorem 1 and 2

input : a number $k \geq 2$ of agents, a number N (large) of vertices, and a family of graphs among the following: labeled trees, rooted plane trees, rooted unebedded binary trees, rooted unembedded trees, unrooted unembedded trees

do M **times**

  sample T_N a uniform random tree on N vertices from the given family

  sample $v_1, v_2, \ldots, v_k$ distinct vertices uniformly at random from T_N

  compute the corresponding Voronoi vector

compare the distribution of the computed Voronoi vectors and the predicted distribution

By Theorem 1 and 2, the distribution of the Voronoi vector $V = \left(V^{(1)}, \ldots, V^{(k)}\right)$ associated to random large trees with k agents placed uniformly at random on its vertices, tend to be uniformly distributed over the $(k-1)$-simplex as the trees get larger. In other words, V should follow a Dirichlet distribution with parameters $1, \ldots, 1$ (k times). We recall that the Dirichlet distribution with parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$, with $\alpha_i > 0$ for all $1 \leq i \leq k$, call it $\mathcal{D}(\boldsymbol{\alpha})$, has the following probability density function

$$p(x_1, \ldots, x_k; \alpha_1, \ldots, \alpha_k) := \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{k} x_i^{\alpha_i - 1}, \tag{6}$$

where $(x_1, \ldots, x_k)$ belongs to the standard $(k-1)$-dimensional simplex i.e., $\sum_{i=1}^{k} x_i = 1$ and $x_i \in [0, 1]$. Also, the normalization constant $\mathrm{B}(\boldsymbol{\alpha})$ is the multivariate beta function:

$$\mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{k} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)},$$

where $\Gamma \colon x \mapsto \int_0^\infty t^{z-1} e^{-t} dt$ is the gamma function.

Notice that, in our case, we have $(\alpha_1, \ldots, \alpha_k) = (1, \ldots, 1)$. Therefore, equation (6) reduces to

$$p(x_1, \ldots, x_k; \alpha_1, \ldots, \alpha_k) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})}$$

Finally, notice that, if the vector $(x_1, \ldots, x_k)$ follows a Dirichlet distribution with parameters $1, \ldots, 1$ (k times), then for all $1 \leq i \leq k$, x_i has a beta distribution with parameters 1, $k-1$; call such distribution $\mathcal{B}(1, k-1)$. The probability distribution function of $\mathcal{B}(1, k-1)$ is the function $x \in [0, 1] \mapsto \frac{1}{\mathrm{B}(1,k-1)} \cdot (1-x)^{k-2}$. Notice that, in our case, according to Theorem 1 and 2, $V^{(i)}$ follows a distribution $\mathcal{B}(1, k-1)$, for all $1 \leq i \leq k$.

Therefore, following the steps of Algorithm 12, we can generate M Voronoi vectors $V_1, \ldots, V_M$ associated to uniform random trees on N vertices with k agents placed uniformly at random on them, and check that the samples $V_i^{(j)}$ for $1 \leq i \leq M$ and $1 \leq j \leq k$ follow a distribution $\mathcal{B}(1, k-1)$.

We can do so with the Kolmogorov-Smirnov test or *goodness-of-fit* test. To explain how this process works, we first need some preliminary definitions.

We define the standard Brownian bridge $(B_t)_{0 \leq t \leq 1}$ as a standard Brownian motion conditioned on taking value zero at time $t = 1$. Consider the random variable $K := \sup_{t \in [0,1]} |B_t|$, where $(B_t)_{0 \leq t \leq 1}$ is a standard Brownian bridge. Also, let $X_1, \ldots, X_n$ be i.i.d. random variables defined on a probability space $(\Omega, \mathscr{F}, \mathbb{P})$, with cumulative distribution function (CDF) F. Define the empirical cumulative distribution function (ECDF) F_n associated to the samples $X_1, \ldots, X_n$ as

$$F_n(x) := \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{X_i \leq x}.$$

Define the Kolmogorov statistic to be

$$D_n := \sup_x |F_n(x) - F(x)|.$$

As proven in [Kol33], we get that

$$\sqrt{n}D_n \longrightarrow \sup_t |B(F(t))|$$

in distribution as n tends to $+\infty$.

Now, taken $Y_1, \dots, Y_n$ samples that take values in $\mathbb{R}$, we want to check the validity of the following null hypothesis: "the samples $Y_1, \dots, Y_n$ are sampled independently from the probability space $(\Omega, \mathscr{F}, \mathbb{P})$, with CDF equal to F". The Kolmogorov-Smirnov test (or *goodness-of-fit* test) can be constructed as follows. We compute the probability $\mathbb{P}(K \geq \sqrt{n}D_n)$. Such probability is called the p-value of the test. We then reject the null hypothesis if the p-value is below a significance level α. One usually chooses $\alpha = 0.05$.

Notice that this test gets more and more accurate as the sample size n gets larger. Moreover, the greater is the p-value, the more likely is the null-hypothesis. In order to have a more thorough explanation of the details of this test, see [Gas77].

We ran some simulations, following the scheme outlined in Algorithm 12, choosing a uniformly random tree on N vertices from a give family of trees and a number k of agents to placed uniformly at random on such vertices. We generate $M = 1000$ trees and the corresponding Voronoi vectors $V_1, \dots, V_M$. We then evaluate the p-value of the Kolmogorov-Smirnov test comparing the $k \cdot M$ samples $\left\{V_i^{(j)}\right\}_{\substack{1 \leq i \leq M \\ 1 \leq j \leq k}}$ against a distribution $\mathcal{B}(1, k-1)$. We give the following table summarizing the p-values resulting from such simulations.

p-values	labeled trees			rooted plane trees		
	$N = 50$	$N = 500$	$N = 1000$	$N = 50$	$N = 500$	$N = 1000$
$k = 2$	$8.75 \cdot 10^{-4}$	0.2334	0.2835	0.0064	0.6269	0.1513
$k = 3$	$7.02 \cdot 10^{-6}$	0.1642	0.0803	$2.42 \cdot 10^{-5}$	0.1692	0.3344
$k = 4$	$2.41 \cdot 10^{-7}$	0.0557	0.5269	$4.66 \cdot 10^{-10}$	0.2325	0.0635
$k = 5$	$4.86 \cdot 10^{-8}$	0.1819	0.3131	$1.02 \cdot 10^{-14}$	0.1315	0.2014

p-values	rooted unordered binary trees			rooted unordered trees		
	$N = 50$	$N = 500$	$N = 1000$	$N = 50$	$N = 500$	$N = 1000$
$k = 2$	$3.29 \cdot 10^{-5}$	0.2212	0.0591	$1.32 \cdot 10^{-5}$	0.1402	0.0852
$k = 3$	$4.84 \cdot 10^{-7}$	0.0874	0.0575	$2.08 \cdot 10^{-16}$	0.0707	0.1594
$k = 4$	$1.59 \cdot 10^{-7}$	0.0905	0.4935	$1.77 \cdot 10^{-27}$	0.0523	0.2514
$k = 5$	$2.72 \cdot 10^{-10}$	0.1014	0.4935	$1.97 \cdot 10^{-48}$	0.1465	0.5291

p-values	free trees		
	$N = 50$	$N = 500$	$N = 1000$
$k = 2$	$1.70 \cdot 10^{-6}$	0.2212	0.1315
$k = 3$	$5.80 \cdot 10^{-5}$	0.0874	0.5587
$k = 4$	$1.29 \cdot 10^{-27}$	0.0905	0.1122
$k = 5$	$4.19 \cdot 10^{-49}$	0.2929	0.3777

Table 1: p-values resulting from the simulation described in Algorithm 12,

We can see that, for $N = 50$ nodes, the corresponding p-values are below 0.05, the chosen significance level. However, for $N = 500, 1000$ nodes, the corresponding p-values are above the the significance value of 0.05. Thus, for N large, there is not enough evidence to reject the null hypothesis that the components of the Voronoi vectors, $\{V_i^{(j)}\}_{\substack{1 \leq i \leq M \\ 1 \leq j \leq k}}$ are sampled independently from a distribution $\mathcal{B}(1, k-1)$. This confirms the theoretical predictions of Theorem 1 and 2. We remark that the p-values in Table 1 seem to be higher for higher values of k. This confirms the intuitively true fact that convergence is faster if the ratio N/k is lower.

We present another simulation. We ran Algorithm 12 with $N = 200$, $M = 1000$ and $k = 2, 3, 4, 5$. We then computed the empirical probability distribution function (EPDF) of the samples $\{V_i^{(j)}\}_{\substack{1 \leq i \leq M \\ 1 \leq j \leq k}}$ and compared it with

the probability distribution function (PDF) of the predicted distribution $\mathcal{B}(1, k-1)$. We show our results only for the family of labeled trees and rooted plane trees in order to make the plot legible, but we provide the code to generate such plot for every family considered in this thesis in the library linked at the bottom of this section.

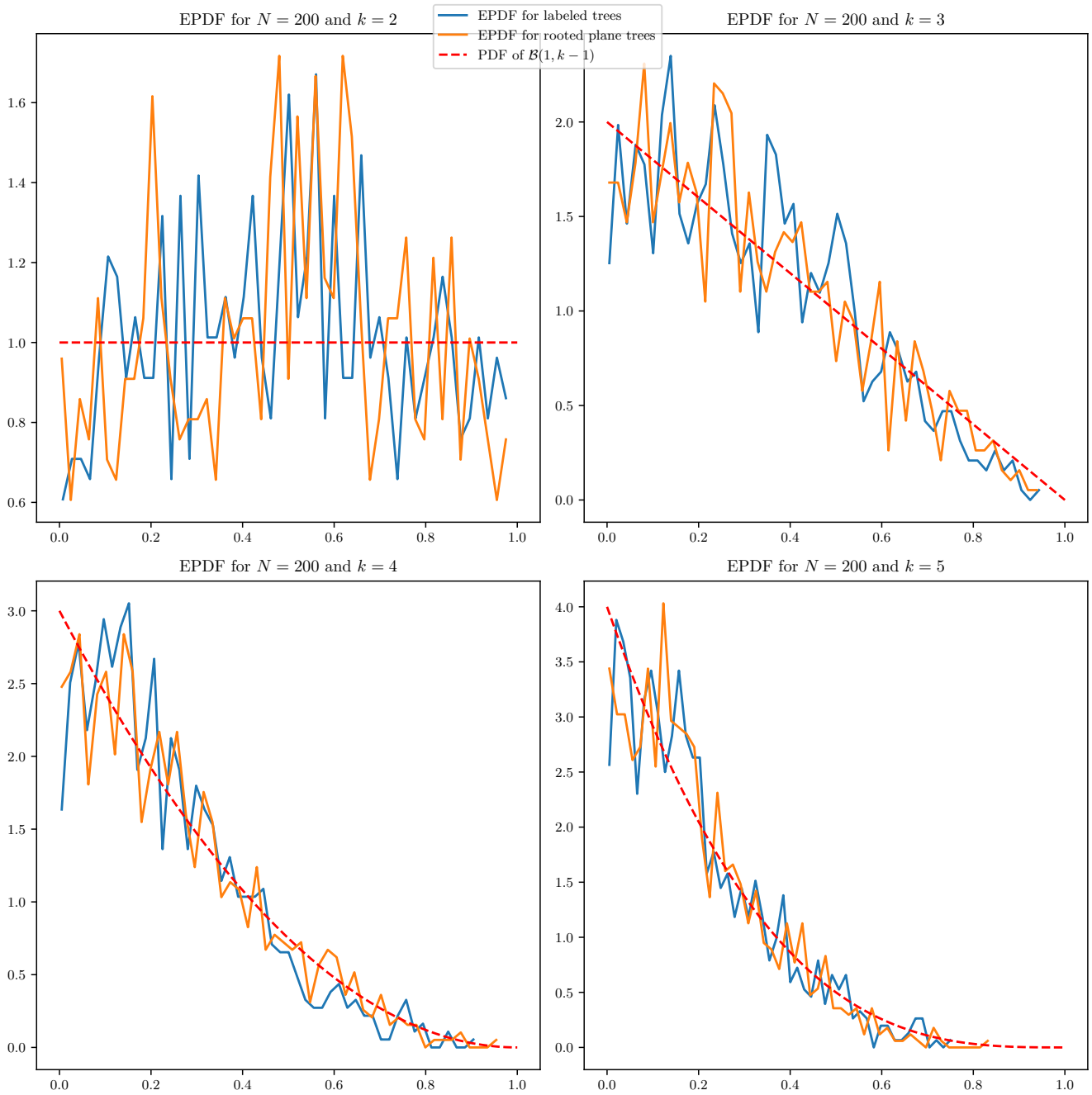

Figure 2: Comparison of the empirical probability distribution functions and the theoretical probability distribution function of $\mathcal{B}(1, k-1)$ for different values of k.

We see that the empirical probability distribution function and the theoretical probability distribution function are quite similar. Again, such similarity increases as k gets larger.

We have one last simulation to present. We ran Algorithm 12 on the family of rooted plane trees, with values of $k = 2$, $M = 2000$ and N ranging from 100 to 1000 with steps of 50 units. We plot the corresponding p-values that

we obtained.

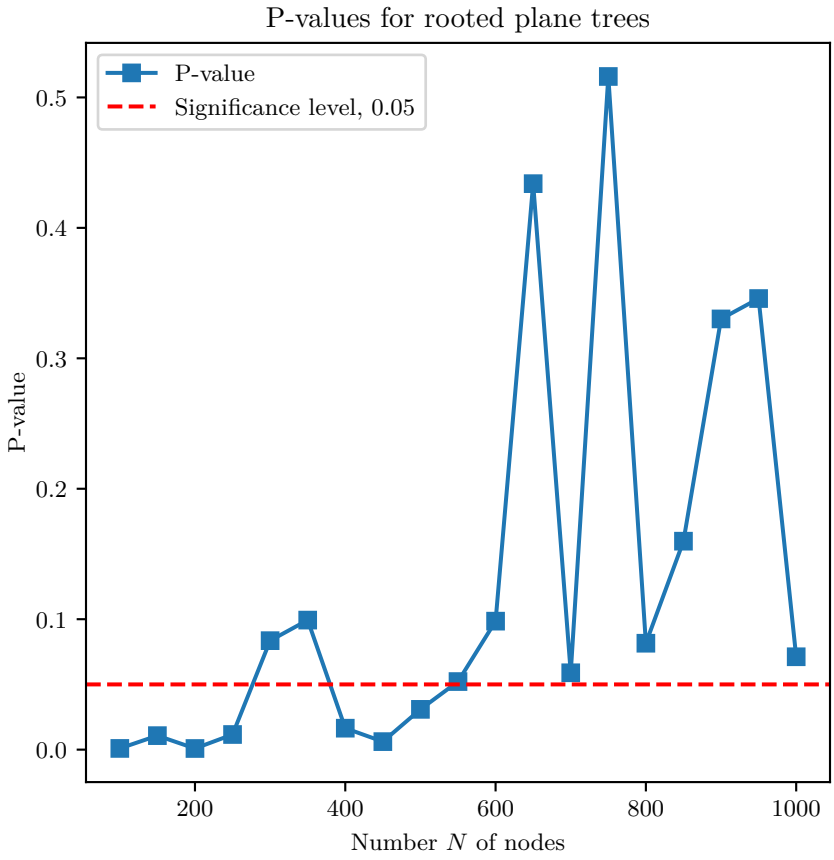

Figure 3: Graph of the p-values obtained by running Algorithm 12 with values of $k = 2$, $M = 1000$ and different values of N.

We remark that, for large values of N, the p-value obtained in the simulation is stably above the significance level of 0.05. This, again, corroborates Theorem 1 and 2.

The code used to run these simulations is contained in the git repository `https://github.com/vincenzo-politelli/Thesis_repo.git`.

7 Conclusions

We have strong evidence corroborating Theorem 1 and 2, which are proven in [AAC+18]. In order to run the simulations presented in section 6, we have designed, analysed (and proved the correctness of) algorithms that allow us to generate random trees from certain families. We also coded a library (`https://github.com/vincenzo-politelli/Thesis_repo.git`) implementing such methods and simulations. We remark that, in [AAC+18], the authors prove that some tree-like families, such as random embedded graphs with a fixed number of faces, enjoy similar properties related to the Voronoi competition on its vertices as the families of random trees analysed in this thesis. One possible development of this work would be to run simulations on those objects and test the predictions of [AAC+18]. In order to do so, one should also find algorithms to uniformly generate these more complicated families of random graphs, possibly providing a proof of the correctness of such algorithms.

References

[AAC+18] Louigi Addario-Berry, Omer Angel, Guillaume Chapuy, Éric Fusy, and Christina Goldschmidt. Voronoi tessellations in the CRT and continuum random maps of finite excess. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 933–946. SIAM, 2018.

[Ald91] David Aldous. The Continuum Random Tree. I. *The Annals of Probability*, 19(1):1 – 28, 1991.

[Ale22] Luca Castelli Aleardi. *Algorithms for graphs on surfaces: from graph drawing to graph encoding.* 2022.

[AM08] Marie Albenque and Jean-François Marckert. Some families of increasing planar maps. *Electron J Probab*, 13, 01 2008.

[BBI01] Dmitri Burago, Yuri Burago, and Sergei Ivanov. A course in metric geometry. *Graduate Studies in Math.*, 33, 01 2001.

[Car16] Alessandra Caraceni. The scaling limit of random outerplanar maps. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 52(4):1667 – 1686, 2016.

[DZ90] Nachum Dershowitz and Shmuel Zaks. The cycle lemma and some applications. *Eur. J. Comb.*, 11(1):35–40, 1990.

[Ebe16] Manuel Eberl. Fisher–yates shuffle. *Archive of Formal Proofs*, September 2016. `https://isa-afp.org/entries/Fisher_Yates.html`, Formal proof development.

[Gas77] Joseph Gastwirth. Review: E. l. lehmann, nonparametrics: Statistical methods based on ranks. *Bulletin of The American Mathematical Society - BULL AMER MATH SOC*, 83, 01 1977.

[HMU03] John Hopcroft, Rajeev Motwani, and Jeffrey Ullman. *Introduction to automata theory, languages, and computation - international edition (2. ed)*. 01 2003.

[Kai76] W. D. Kaigh. An Invariance Principle for Random Walk Conditioned by a Late Return to Zero. *The Annals of Probability*, 4(1):115 – 121, 1976.

[Knu97] Donald Ervin Knuth. *The art of computer programming, Volume I: Fundamental Algorithms, 3rd Edition.* Addison-Wesley, 1997.

[Kol33] A. Kolmogoroff. Sulla determinazione empirica di una legge di distribuzione. *G. Ist. Ital. Attuari*, 4:83–91, 1933.

[Mie09] Grégory Miermont. Tessellations of random maps of arbitrary genus. *Annales scientifiques de l'École Normale Supérieure*, Ser. 4, 42(5):725–781, 2009.

[MM11] Jean-François Marckert and Grégory Miermont. The crt is the scaling limit of unordered binary trees. *Random Structures & Algorithms*, 38(4):467–501, 2011.

[NW75] A. Nijenhuis and H.S. Wilf. *Combinatorial Algorithms.* Computer science and applied mathematics : a series of monographs and textbooks. Academic Press, 1975.

[Prü18] H. Prüfer. Neuer Beweis eines Satzes über Permutationen. *Arch. der Math. u. Phys. (3)*, 27:142–144, 1918.

[PSW16] Konstantinos Panagiotou, Benedikt Stufler, and Kerstin Weller. Scaling limits of random graphs from subcritical classes. *The Annals of Probability*, 44(5):3291–3334, 2016.

[RW00] L. C. G. Rogers and David Williams. *Diffusions, Markov Processes and Martingales.* Cambridge Mathematical Library. Cambridge University Press, 2 edition, 2000.

[Stu19] Benedikt Stufler. The continuum random tree is the scaling limit of unlabeled unrooted trees. *Random Structures & Algorithms*, 55(2):496–528, 2019.

[Wil81] Herbert S. Wilf. The uniform selection of free trees. *J. Algorithms*, 2(2):204–207, 1981.