Trabajo Final

Prof. Ph.D. José J. Cerda-Hernández

Instituto de Matemática y Ciencias Afines

Objetivo general del trabajo: Simular una triangulación y después analizar percolación sobre las triangulaciones.

1. Simular una triangulación
2. Colocar percolación de Bernoulli sobre la triangulación
3. Estimar el p_c
4. Estimar el Cluster Máximo
5. Estimar el Cluster Medio

1 Sampling critical random causal triangulations

Probably the easiest way to simulate infinite causal triangulation is to use the tree representation. The bijection between causal triangulations and Galton-Watson trees is well known, see Figure 1 and [6], [2] for details. Moreover, there exists an equivalence between the ensamble of critical random infinite causal triangulations $(\mathbb{LT}_\infty, P_{\mu_c})$ and the (critical) Galton-Watson process conditioned on non-extinction, see originally [10], [8], [1], and also [2], [3], these results are resumed in Theorem 3.1 from [4]. Thus, instead of simulation of triangulation we can simulate Galton-Watson trees followed by triangulation construction. The Galton-Watson process conditioned on non-extinction on infinity can be generated by spine decomposition in the following way, see [10], [8].

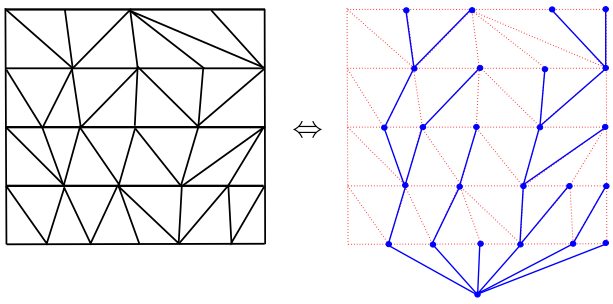

Figure 1: Left: A patch of a causal triangulation. Right: Illustration of the bijection between causal triangulations and planar rooted trees. In both figures the left and right of the strip are identified.

Let $\mathbf{p} = \{p_k : k \geq 0\}$ be a off-springs probability distribution, with finite positive first moment, and $p_0 \neq 0$. Denote by

$$\rho_k = \frac{kp_k}{\sum_{i\geq 1} ip_i},$$

its size-biased version. Let $\mathcal{T}(\mathbf{p}, \boldsymbol{\rho})$ denote the ensamble of infinite random rooted tree generated as follows. First draw an integer k with distribution ρ_k, and connect the root r to k offspring. Choose a new root r_1 with uniform distribution on $\{1, 2, \ldots, k\}$. For each vertex $v \in \{1, \ldots, k\} \setminus$

$\{r_1\}$ draw an integer n with distribution p_n, and connect the vertex v to n offspring. For the new root r_1 draw an integer k_1 with distribution ρ_{k_1}, and connect the new root r_1 to k_1 offspring. Then recursively, the last generation have a new root. The last root generates an integer k independently with the distribution ρ_k and the other vertices generate an integer l independently with distribution p_l. This process is repeated infinitely.

Proposition 1 *(see Theorem 3.1. [4]) There exist a equivalence between critical infinite causal triangulations* $(\mathbb{LT}_\infty, P_{\mu_c})$ *and the random trees* $\mathcal{T}(\mathbf{p}, \boldsymbol{\rho})$*, where*

$$\mathbf{p} = \left\{ p_k = \frac{1}{2^{k+1}} : k \geq 0 \right\} \quad \textit{and} \quad \boldsymbol{\rho} = \left\{ \rho_k = \frac{k}{2^{k+1}} : k \geq 1 \right\}. \tag{1}$$

2 Implementation of the numerical algorithm

Although our algorithm is based on the standard algorithms for percolation, in our case the lattice is a random causal triangulation reconstructed from a critical Galton-Watson tree. We sequentially generate the time-like edges of $\mathbf{t}_i, i = 1, \ldots, n$ by the following. With a given generated sequence $(\mathbf{t}_1, \ldots, \mathbf{t}_k)$ we start with vertices of on the last circle $S \times \{k\}$ and (i) we generate time-like edges of triangulation $\mathbf{t}_{k+1}$ of the strip $S \times [k, k+1]$ according the offspring distributions, then (ii) we add the auxiliary time-edges in order to reconstruct triangulation of the strip, and, finally, we add space-like edges connecting vertices on $S \times \{k+1\}$. Together with strip triangulation $\mathbf{t}_i$ we generate the configuration of percolation on edges of $\mathbf{t}_{k+1}$. Next, we explain in detail our algorithm.

As we mentioned before we will generate the conditioned Galton-Watson trees with the followed triangulations reconstruction. The algorithm constructs step-by-step a sequence of "coloring" vectors

$$v_n = [v_{n,1}, \ldots, v_{n,k_n}], \quad n = 0, 1, \ldots$$

that will indicate to which clusters ("colors") the corresponding k_n vertices on the n-th slice belong.

We start with one vertex, the root o, of the tree. For this vertex we assign the type cluster (or color) 1, i.e. $v_{0,1} = 1, k_0 = 1$, and we define the initial vector $v_0 = [1]$.

g-step: Here we generate the number of vertices on the next circle. Since the root belongs to the spine of the tree, we generate the next vector size k_1 (k_1 is the number of children of the root) according the offspring distribution (see Proposition 1)

$$\rho_k = \frac{k}{2^{k+1}}, \quad k \geq 1. \tag{2}$$

Let $\mathbf{r} = (r_{i,j}, i, j \in \mathbb{N})$ and $\mathbf{s} = (s_{i,j}, i, j \in \mathbb{N})$ be the series of i.i.d. random uniformly distributed on $[0, 1]$ variables, $r_{i,j}, s_{i,j} \sim U[0, 1]$. The vector v_1 will be generated into two steps: **r**-step and **s**-step.

r-step: (see the first plot on the Figure 2) Here we close or open the time-like edges. If $r_{1,i} < p, i = 1, \ldots, k_1 - 1$, then $v_{1,i} = v_{01} \equiv 1$, and since the last k_1-th vertex has two time-like edges the probability to be open by these two edges is $1 - (1-p)^2$. Thus, if $r_{1,k_1} < 1 - (1-p)^2$, then $v_{1,k_1} = v_{01} \equiv 1$. The others will be enumerated starting from 2 obtaining the preliminary vector $v_1 = [v_{1,1}, \ldots, v_{1,k_1}]$.

s-step: (see the second plot on the Figure 2) Here we close or open the space-like edges and update the values of v_1. If $s_{1,i} < p$ then two neighbor component $(v_{1,i}, v_{1,i+1})$ takes the same value (v, v), where $v = \min(v_{1,i}, v_{1,i+1})$. More precisely, the configuration of open-closed edges divides the set of vertices on the circle $\kappa_1 = \{1, \ldots, k_1\}$ into the set of b distinct clusters (maximal connected components):

$$\{1, \ldots, k_1\} \cup_{i=1}^{b} V_i, \quad |V_i| \geq 1 \text{ and } V_i \cap V_j = \emptyset, \ i \neq j.$$

For any $i \in \kappa_1$ let $c(i)$ be the number of cluster $V_{c(i)}$ which contains vertex i. Then

$$v_{1,i} = \min(v_{1,k}, k \in V_{c(i)}).$$

The **r**-step and **s**-step are illustrated in Figure 2.

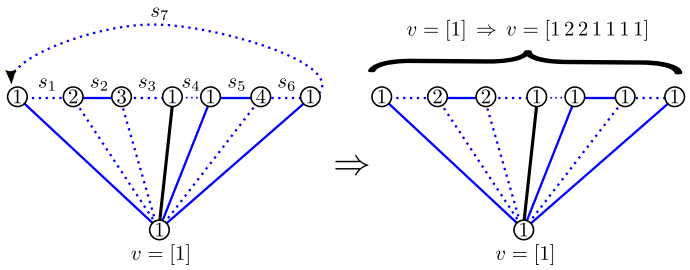

Figure 2: Example of as update the vector v starting of the root of the triangulation. After the first step, we have $v = [1] \to v = [1\ 2\ 3\ 1\ 1\ 4\ 1]$. Next, we generate a random vector $s = [s_1, \ldots, s_n]$, with $s_1, s_3, s_4, s_6, s_7 > p$ and $s_2, s_5 < p$, in order to update the vector v, obtaining $v = [1\ 2\ 3\ 1\ 1\ 4\ 1] \to v = [1\ 2\ 2\ 1\ 1\ 1\ 1]$.

The previous steps are summarized in the Algorithm 1. Figures 2 and 3 illustrate the steps of the Algorithm 1.

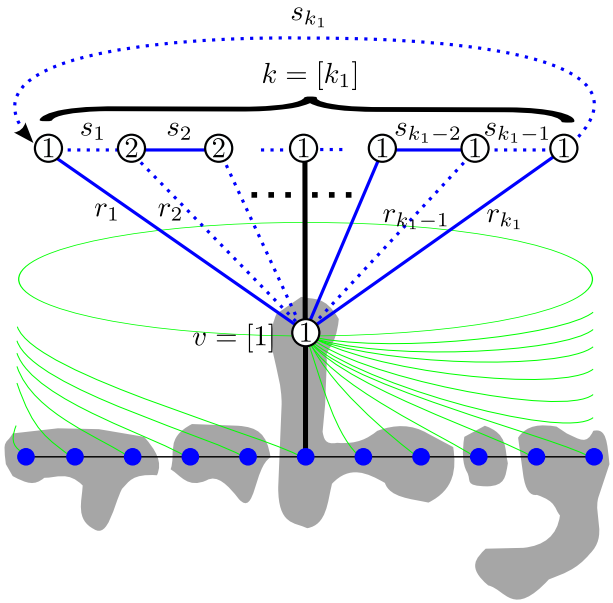

Figure 3: The figure shown the steps of the Algorithm 1. Vectors $r = [r_1, \ldots, r_{k_1}]$, $s = [s_1, \ldots, s_{k_1}]$ serve to create new clusters in the following level or generation.

Algorithm 1 is used when the last vector v has length $|v| = 1$, i.e we have only one vertex on the circle. Now we extend the algorithm for the case $|v| > 1$. We explain this new algorithm as follow:

Let $v_n = [v_{n,1}, \ldots, v_{n,l}]$, with $l > 1$, the vector of cluster configuration (type of cluster) in the n-th generation. Remember that we also know the exact position of the current *root* $\in \{1, \ldots, l\}$ (position of the spine in the GW tree) of the vector v_n. The update of the vector v_n will be represented in three parts.

Algorithm 1 Algorithm when we begin either of the root or when the number of children in the last generation is 1.

Input values: p : parameter of percolation.
Let $v = [v_1, \ldots, v_l]$ be the last generation of the random tree. We called n-th generation, where $n \in \mathbb{N} \cup \{0\}$.
if $l = 1$, that is, length$(v) = 1$ **then**
    We start with a coloring vector $v = [clus]$., where $clus$ is the last type of cluster.
    Create a vector of children $k = [k_1]$, where k_1 (number of children of v_1) is choose with offspring probabilities

$$\hat{p}_n = \frac{n}{2^{n+1}}, \quad n \geq 1. \tag{3}$$

    In this case, as length$(v) = 1$, then v_1 is the last root of the GW tree. **FIRST UPDATE CLUSTER:**
    Create a vector $r = [r_1, \ldots, r_{k_1}]$, where $r_i \sim \mathcal{U}(0,1)$ for all $1 \leq i \leq k_1$
    Create a auxiliar vector $w = [0, \ldots, 0]$ with lenght$(w) = k_1$.
    for $i = 1$ *to* k_1 **do**
        if $r_i < p$ **then**
            $w[i] = v_1$
        else
            $clus \leftarrow clus + 1$, we create a new type of cluster
            $w[i] = clus$
        end if
    end for
    SECOND UPDATE CLUSTER:
    Create a vector $s = [s_1, \ldots, s_{k_1}]$, where $s_i \sim \mathcal{U}(0,1)$ for all $1 \leq i \leq k_1$.
    if $s_i < p$ **then**
        $w[i]$ e $w[i+1]$ belong the same cluster, i.e.
        $w[i] \leftarrow \min(w[i], w[i+1])$
        $w[i+1] \leftarrow \min(w[i], w[i+1])$
    end if
end if
$v \leftarrow w$
Choose the new root: choose the number $root$ uniformly at random in the range $1 \leq root \leq$ lenght(v).

g-step: First, we create a vector of children $k = [k_1, \ldots, k_l]$ with its respective distributions. k_i is the number of children of the i-th vertex in the n-th generation: according the proposition 1, if $i \neq root$ we sample k_i according the distribution $\mathbf{p}$, and if $i = root$, then the distribution of k_i is $\boldsymbol{\rho}$, see (1).

The vector $k = [k_1, \ldots, k_l]$ will be called number of children in the $(n+1)$-th generation.

r-step: Second step is create the vector v in the $(n+1)$-th generation. Let $v = [v_1, \ldots, v_l]$ the cluster configurations in the nth generation and $k = [k_1, \ldots, k_l]$ the vector of children created in the previous step. For each vertex $i \in \{1, \ldots, l\}$ that belong to the nth generation draw k_i vertices in the $(n+1)$th generation, and conect i to k_i offspring. Note that $k_i \in \mathbb{N} \cup \{0\}$. Now, each vertex in the $(n+1)$th generation will be either linked with probability p, or not with probability $1-p$ with vertices in the nth generation, as follow. First, we create an auxiliar vector w of lenght $k_1 + \cdots + k_l$ to construct the vector of type of cluter to the $(n+1)$th generation. If $k_i = 0$ we do nothing. If $k_i > 0$ we denote by $m+1$ the distance between the vertex i and the next vertex in the same generation with at least one child, i.e., $k_i > 0$, $k_{i+j} = 0$ for $1 \leq j \leq m$ and $k_{i+m+1} > 0$ (see Figure 5). Each children $j \in \{1, \ldots, k_{i-1}\}$ of the vertex i is either linked with probability p, or not with probability $1-p$, as follow. For each child

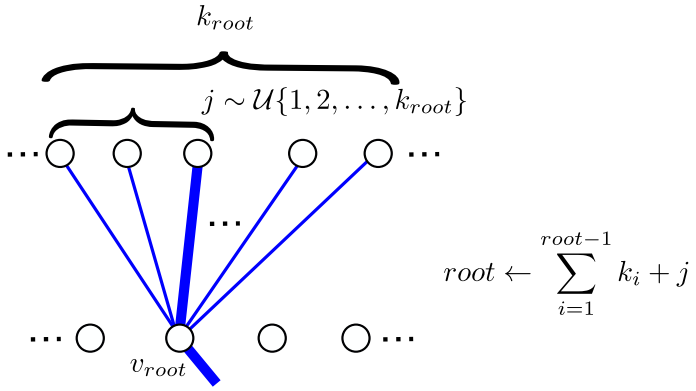

Figure 4: new root

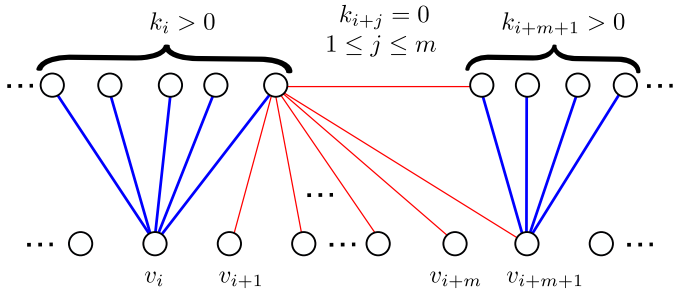

Figure 5: Steps to update the cluster type of the vertex $k_1 + \cdots + k_i$ in the $(n+1)$-th generation, for any i.

$j \in \{1, \ldots, k_{i-1}\}$ take $s \sim \mathcal{U}_{[0,1]}$. If $s < p$, then $w_{k_1+\cdots+k_{i-1}+j} = v_i$ (same cluster) else we create a new cluster for this vertex, i.e., $w_{k_1+\cdots+k_{i-1}+j} =$ new cluster type. In the case $j = k_i$, we take $s_i, s_{i+1}, \ldots, s_{i+m}, s_{i+m+1} \sim \mathcal{U}_{[0,1]}$. Using each uniform trial we either update the cluster type of the vertex $k_1 + \cdots + k_{i-1} + k_i$ with probability p or create a new type cluster with probability $1 - p$ in the $(n + 1)$th generation. Note that s_{i+j}, for all $j \in \{1, \ldots, m + 1\}$, is used to update the vertex $k_1 + \cdots + k_{i-1} + k_i$ in the $(n + 1)$th generation (see Figure 5). Note that the vertex $k_1 + \cdots + k_{i-1} + 1$ has three neighbors, the vertex i in the nth generation, and the vertices $k_1 + \cdots + k_{i-1} + 2$ and $k_1 + \cdots + k_r$ in the $(n + 1)$th generation, here r stand the previous left neighbor such that $k_r > 0$ and $k_j = 0$ for $r < j < i$, and that we do not update the cluster type of the vertex $k_1 + \cdots + k_{i-1} + 1$ with the vertex $k_1 + \cdots + k_r$ because that upgrade was done in the previous step. Note also that in this algorithm the last vertex in the $(n + 1)$th generation is linked with the last vertices of the nth generation, but also is linked with the first vertex of the nth generation.

The previous step will be called of **first update of cluster type**. This step create a new vector w in the $(n + 1)$-th generation only using the cluster type of the n-th generation.

s-step: Simultaneously, this step also update the vector v of the nth generation from left to right. Now update again the vector w but using only the vertices in the $(n + 1)$th generation as follows: we generate a random vector $s = [s_1, \ldots, s_{k_1+\cdots+k_l}]$, where $s_1, \ldots, s_{k_1+\cdots+k_l} \sim \mathcal{U}_{[0,1]}$. Next, we employ the percolation parameter p in order to update the cluster type vector w in the $(n + 1)$th generation. If $s_i < p$ then the edge $[i, i + 1]$ is open and update the type of cluster of both vertices. This update will be called of **second update of cluster type**. Next, we define the cluster type vector v of the $(n + 1)$th generation, as the vector of children after of the first and second update of cluster type v of the nth generation. Thus, the $(n + 1)$th generation will have $\sum_{i=1}^{l} k_i$ vertices. Finally, we chose the new root of the $(n + 1)$th generation and repeated this process infinitely times. The chose of the new root is present in the Algorithm 2 (see also

Figure 4).

Algorithm 2 Chosing the new root

1: If $k = [k_1, \ldots, k_l]$ is the vector of children and $root \in \{1, \ldots, l\}$ the root in the nth generation, then

2: Chose the number j uniformly at random in the range $1 \leq j \leq k_{root}$.

$root \leftarrow \sum_{i=1}^{root-1} k_i + j.$

One way of achieving the algorithm when $l = |v| > 1$ is described in the Algorithm 3.

Algorithm 3 Algorithm when $|v| > 1$.

Let $v = [v_1, \ldots, v_l]$, $k = [k_1, \ldots, k_l]$ and $clus = \max_{1 \le i \le l}\{v_i\}$.
Let $root \in \{1, \ldots, l\}$, the root in the nth generation.
FIRST UPDATE OF CLUSTER TYPE:
Create an auxiliar vector $w = [0, \ldots, 0]$ with $|w| = k_1 + \cdots + k_l$.
for $j = 1$ *to* l **do**
  if $k_j > 0$ **then**
    for $i = 1$ *to* k_j **do**
      $r \sim \mathcal{U}(0,1)$,
      if $r < p$ **then**
        $w_{k_1+\cdots+k_{j-1}+i} \leftarrow v_j$
      else
        $clus \leftarrow clus + 1$
        $w_{k_1+\cdots+k_{j-1}+i} \leftarrow clus$
      end if
    end for
    $s \leftarrow 1$
    while $k_{j+s} = 0$ **do**
      $r \sim \mathcal{U}(0,1)$,
      if $r < p$ **then**
        $w_{k_1+\cdots+k_j} \leftarrow \min\{w_{k_1+\cdots+k_j}, v_{j+s}\}$
        $v_{j+s} \leftarrow \min\{w_{k_1+\cdots+k_j}, v_{j+s}\}$
        $s \leftarrow s + 1$
      end if
    end while
    $r \sim \mathcal{U}(0,1)$,
    if $r < p$ **then**
      $w_{k_1+\cdots+k_j} \leftarrow \min\{w_{k_1+\cdots+k_j}, v_{j+s}\}$
      $v_{j+s} \leftarrow \min\{w_{k_1+\cdots+k_j}, v_{j+s}\}$
    end if
  end if
end for
SECOND UPDATE OF CLUSTER TYPE:
Let $s = [s_1, \ldots, s_{k_1+\cdots+k_l}]$, where $s_i \sim \mathcal{U}_{[0,1]}$ for any i.
if $s_i < p$ **then**
  $w[i]$ e $w[i+1]$ belong the same cluster, i.e.
  $w[i] \leftarrow \min(w[i], w[i+1])$
  $w[i+1] \leftarrow \min(w[i], w[i+1])$
end if
$v \leftarrow w$
Chose the new root using Algorithm 2.

3 Parte2: Definitions and preliminary results

We start with the definition of an ensamble of graphs called *rooted causal (or Lorentzian) triangulations* of the cylinder $\mathsf{C} = S \times [0, \infty]$, where S is a unite circle. Where possible, we follow definitions and notations of [11].

Causal triangulations. Consider a connected graph G with a countable set of vertices embedded into the cylinder C. Any connected component of $\mathsf{C} \setminus G$ is called a *face*. Let the size of a face be the number of edges incident to it, with the convention that an edge incident to the same face on both sides counts for two. We then call a face of size 3 (or 3-sided face) a *triangle*.

The graph G defines an infinite causal (or Lorentzian) triangulation $\mathbf{t}$ of if *(i)* all vertices lie

in circles $S\times\{j\}, j\in\mathbb{N}\cup\{0\}=\{0,1,\dots\}$; *(ii)* each face is triangle; *(iii)* each face of $\mathbf{t}$ belongs to some strip $S\times[j,j+1], j=0,1,\dots,$ and has all vertices and exactly one edge on the boundary $(S\times\{j\})\cup(S\times\{j+1\})$ of the strip $S\times[j,j+1]$; and *(iv)* the number of edges on $S\times\{j\}$ is positive and finite for any $j=0,1,\dots$. See Figure 1 for an example of causal triangulation.

We note that two vertices of a triangle on a same circle, say $S\times\{j\}$, may coincide (in this case the corresponding edge stretches over the whole circle $S\times\{j\}$, i.e. is a loop). The root in a triangulation $\mathbf{t}$ consists of a triangle Δ of $\mathbf{t}$, called the *root face*, with the anti-clockwise ordering on its vertices (o,x,y), where o and x lie in $S\times\{0\}$ (they can coincide) and y belongs to $S\times\{1\}$. The vertex o is called the *root vertex* or simply *root*. The edge (o,x) belongs to $S\times\{0\}$.

Two rooted triangulations, say $\mathbf{t}$ and $\mathbf{t}'$, are equivalent if $\mathbf{t}$ and $\mathbf{t}'$ are embeddings $i_{\mathbf{t}},\ i_{\mathbf{t}'}$ of the same graph G and there exists a self-homeomorphism $h:\mathsf{C}\to\mathsf{C}$ such that $hi_{\mathbf{t}}=i_{\mathbf{t}'}$. We suppose that the homeomorphism h transforms each slice $S\times\{j\}, j\in\mathbb{N}$ to itself and preserves the root: h sends the root of $\mathbf{t}$ to the root of $\mathbf{t}'$. The equivalence class of embedded rooted causal (Lorentzian) triangulations is called causal triangulation.

In the same way we can also define a causal triangulation of a cylinder $\mathsf{C}_N=S\times[0,N]$. Let $\mathbb{LT}_N$ and $\mathbb{LT}_\infty$ be the sets of all causal triangulations with the supports $\mathsf{C}_N=S\times[0,N]$ and $\mathsf{C}=S\times[0,\infty)$, respectively. The number of edges on the upper boundary $S\times\{N\}$ is not fixed. We introduce a Gibbs measure on the set $\mathbb{LT}_N$ as

$$P_{N,\mu}(\mathbf{t})=\frac{1}{Z_N(\mu)}e^{-\mu F_N(\mathbf{t})},\tag{4}$$

where $F_N(\mathbf{t})$ is the number of triangles in the first N strips of the triangulation $\mathbf{t}$, and $Z_N(\mu)$ is the partition function. Here μ is related to the fugacity g of a triangle via the relation $g=e^{-\mu}$.

The measure on the set of infinite triangulations $\mathbb{LT}_\infty$ is defined by the weak limit

$$P_\mu:=\lim_{n\to\infty}P_{N,\mu}.$$

It was shown in [6] that this limit exists for all $\mu\geq\mu_c:=\ln 2$. The latter also provided some properties of causal triangulations under the limit measure P_μ. The probability space $(\mathbb{LT}_\infty,\mathcal{F},P_\mu)$ we refer to as a *random causal triangulations* or *causal triangulations ensemble*, for any $\mu\geq\ln 2$.

Percolation model. Let $\mathbf{t}\in\mathbb{LT}_\infty$, and $E(\mathbf{t}),V(\mathbf{t})$ stand for the set of edges and vertices of the graph $\mathbf{t}$ respectively. The bond percolation on graph $\mathbf{t}$ is usually defined as follows. Each edge from the graph $\mathbf{t}$ is maintained (opened) with probability p or removed (closed) with probability $1-p$ independently. Thus the percolation model is essentially the product measure $\mathbb{P}_p^{\mathbf{t}}$ on the set of configurations $\Omega_{\mathbf{t}}=\{0,1\}^{E(\mathbf{t})}$, where the value 1 on the edge e means that the edge is open, and 0 – closed,

$$\mathbb{P}_p^{\mathbf{t}}(\omega)=\prod_{e\in E(\mathbf{t})}p^{\omega(e)}(1-p)^{1-\omega(e)},\quad\omega\in\Omega_{\mathbf{t}}.$$

$\mathbb{E}_p^{\mathbf{t}}$ denotes the expectation w.r.t. $\mathbb{P}_p^{\mathbf{t}}$. Recall that a cluster is a maximal connected component of open edges.

We define the *percolation function* $p\to\theta^{\mathbf{t}}(p)$ by

$$\theta^{\mathbf{t}}(p)=\mathbb{P}_p^{\mathbf{t}}(|C|=\infty)\tag{5}$$

C is percolation cluster containing the root of the triangulation $\mathbf{t}$, and $|C|$ is the number of vertices in the cluster C. In the literature the percolation function is also known as *strength of percolation cluster* or *percolation strength*. We define the *critical value* by

$$p_c=p_c(\mathbf{t})=\inf\{p:\theta^{\mathbf{t}}(p)>0\}.\tag{6}$$

An important question is whether the critical value is non-trivial, i.e., whether $p_c(\mathbf{t}) \in (0,1)$. In [11] we showed that the critical value obeys a zero-one law and is constant P_μ-a.s. for any $\mu \geq \ln 2$. Further, we show that the critical value is non-trivial only in the case $\mu = \mu_c = \ln 2$, P_μ-a.s. These results are summarized in Theorem 1 below.

Theorem 1 *For the considered percolation model on random causal triangulations the following statements hold.*

1. *The critical value $p_c(\mathbf{t})$ is constant P_μ-a.s.,*
2. *The critical value satisfy the following relation*

$$p_c(\mathbf{t}) = \begin{cases} 1 & \text{if} \quad \mu > \ln 2, \\ 0 < p_c(\mathbf{t}) < 1 & \text{if} \quad \mu = \ln 2. \end{cases} \qquad P_\mu - a.s. \tag{7}$$

3. *If $\mu = \mu_c = \ln 2$, then $p_c(\mathbf{t}) \leq \frac{1}{2}$, P_{μ_c}-a.s.*

As we mentioned in Introduction, the two facts are known: (i) the critical value is constant (it is the same for P_{μ_c} almost all triangulations $\mathbf{t}$), and (ii) Hausdorff dimension is constant and equal to 2 for P_μ almost all $\mathbf{t}$. These two facts suggest that any triangulation sampled with distribution P_{μ_c} belong to the same universality class, in the sense that they have identical critical exponent at the critical point (assuming the exponent exist). The behavior of percolation models is most interesting and richest for p values which are close to the critical value p_c. By drawing an analogy to physical systems, physicists predict that the behavior of percolate systems close to criticality is rather insensitive to the precise details of the model, and it is only characterized by the macroscopic behavior. Thus, percolation is expected to behave in a universal manner.

The main purpose of this work is to study percolation critical exponents. For this we will study numerically *two key functions* that describe the connections in bond percolation: percolation strength $\theta^{\mathbf{t}}(p)$ and *mean cluster size* $\chi^{\mathbf{t}}(p)$.

The *mean cluster size* $\chi^{\mathbf{t}}(p)$ is the expected cluster size of the root

$$\chi^{\mathbf{t}}(p) = \mathbb{E}_p^{\mathbf{t}}(|C|). \tag{8}$$

Clearly, $\chi^{\mathbf{t}}(p) = \infty$ if $p > p_c$, for P_μ almost all $\mathbf{t}$, and $p \to \chi^{\mathbf{t}}(p)$ is increasing, as $p \uparrow p_c$. Close to the critical point we expect that the mean cluster size behaves like

$$\chi^{\mathbf{t}}(p) \sim (p - p_c)^{-\gamma}, \qquad \text{as} \qquad p \uparrow p_c, \tag{9}$$

The symbol $\sim$ means that the critical exponent γ can be find as

$$\gamma = -\lim_{p \uparrow p_c} \frac{\ln \chi^{\mathbf{t}}(p)}{\ln(p - p_c)} \quad \text{for } P_{\mu_c} \text{ almost all } \mathbf{t}. \tag{10}$$

By universality class of criticality the critical exponent γ must be independent of $\mathbf{t}$, that is, γ is constant P_{μ_c} almost sure.

As $p_c(\mathbf{t})$ is constant P_{μ_c}-a.s. $\mathbf{t}$, we expect the following behavior of $\theta^{\mathbf{t}}(p)$ as p approaches p_c from above

$$\theta^{\mathbf{t}}(p) \sim (p - p_c)^{\beta}, \qquad \text{as} \qquad p \downarrow p_c, \tag{11}$$

or

$$\beta = \lim_{p \downarrow p_c} \frac{\ln \theta^{\mathbf{t}}(p)}{\ln(p - p_c)} \quad \text{for } P_{\mu_c} \text{ almost all } \mathbf{t}. \tag{12}$$

where the critical exponent β is independent of $\mathbf{t}$, that is, β takes the same value for P_{μ_c} almost all triangulations $\mathbf{t}$.

The existence of the above critical exponents is a priori unclear, and needs a mathematical proof. Unfortunately, in general such a proof is missing, and we can only give proofs of the existence in special cases (see [7]).

In this work we numerically estimate the critical exponents for the percolation strength and mean cluster size defined in (5) and (8), respectively.

References

[1] Aldous, D. and Pitman, J. *Tree-valued Markov chains derived from Galton-Watson processes.* Ann. Inst. H. Poincaré Probab. Statist. 34, no. 5, 637-686, (1998).

[2] B. Durhuus, T. Jonsson, and J. F. Wheater. On the spectral dimension of causal triangulations. *J. Stat. Phys.* **139** (2010) 859–881, 0908.3643.

[3] B. Durhuus. Probabilistic aspects of infinite trees and surfaces. Acta. Phys. Pol. B, 34(10):4795- 4811, (2006).

[4] V. Sisko, A. Yambartsev and S. Zohren, *A note on weak convergence results for uniform infinite causal triangulations*, Brazilian Journal of Probability and Statistics, 32(2), 597-615 (2018).

[5] M. Krikun and A. Yambartsev. Phase transition for the Ising model on the critical Lorentzian triangulation. *Journal of Statistical Physics*, v. 148, p. 422-439, 2012. 0810.2182.

[6] V. Malyshev, A. Yambartsev, and A. Zamyatin, *Two-dimensional Lorentzian models*, Moscow Mathematical Journal **1** (2001), no. 2, 1-18.

[7] G. Grimmett. Percolation (2nd edn). Springer, Berlin (1999).

[8] J. Geiger. *Elementary new proofs of classical limit theorems for Galton-Watson processes.* J. Appl. Probab. 36(2), 301-309 (1999).

[9] Giasemidis, G., Wheater, J. F. and Zohren, S. *Multigraph models for causal quantum gravity and scale dependent spectral dimension* J.Phys. A45, 355001, (2012).

[10] R. Lyons, R. Pemantle and Y. Peres. *Conceptual proofs of L log L criteria for mean behavior of branching processes.* Ann. Probab. 23(3), 1125-1138 (1995).

[11] Cerda-Hernández J , Yambartsev A, and Zohren S, *On the critical probability of percolation on random causal triangulations.* Brazilian Journal of Probability and Statistics, 31(2), 215-228 (2017).