Cutting down trees with a Markov chainsaw

L. Addario–Berry [1] **N. Broutin** [2] C. Holmgren [2]

[1]Department of Mathematics and Statistics
McGill University

[2]Projet Algorithms
INRIA Paris-Rocquencourt

October 18, 2010

Plan of the talk

1. Cutting down trees
2. The record point of view
3. Generating random Cayley trees
4. Large random Cayley trees and the CRT
5. Cutting down trees to plant a forest
6. Cutting down lattice paths
7. Algorithms and the additive coalescent

Introduction

Natural setting: cutting at edges

- cut a random edge
- keep the bit with the root
- stop when no more edges.

Practical setting: cutting at nodes

- pick a random node
- remove the subtree
- stop when no more nodes

Two kinds of result:

- for deterministic and $\log n$ trees: (weakly) 1-stable laws
- for Galton–Watson trees ($\sqrt{n}$ trees), the case we are interested in here.

History: Meir-Moon, Janson, Chassaing, Panholzer, Fill, etc... method of moments :(

Union-Find algorithms and the fragmentation/coalescent

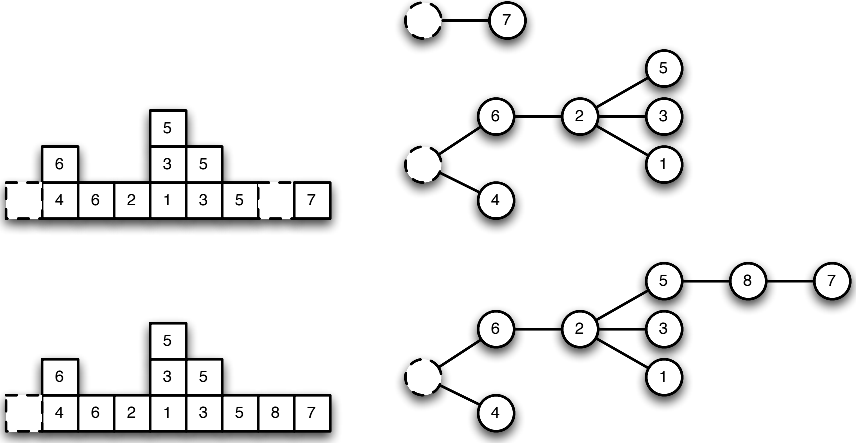
A natural model for merge queries:

- initially have n sets $\{1\}, \{2\}, \ldots, \{n\}$
- there is an unknown spanning tree T on $1, 2, \ldots, n$
- the merge queries are the edges of T in a random order
- after $n - k$ merges the collection of sets is forest of $k + 1$ rooted trees $\Rightarrow$ kernel is **additive**

An other important one:

- the queries are edges of K_n
- the merge queries arrive in random order
- growth of random graphs/minimum spanning tree: kernel is **multiplicative**

Hashing with linear probing/parking



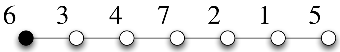
Records I

Usual records

- take a random permutation $\pi = (\pi_1, \ldots, \pi_n)$ of $\{1, 2, \ldots, n\}$
- i is a record if $\pi_i = \min\{j \leq i : \pi_j\}$.

$$\mathbf{E}\left[\sum_i \mathbf{1}[i \text{ is a record}]\right] = \sum_{i=1}^{n} 1/i \sim \log n$$

Equivalent to records in a rooted path P



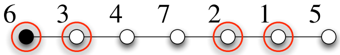
Records I

Usual records

- take a random permutation $\pi = (\pi_1, \ldots, \pi_n)$ of $\{1, 2, \ldots, n\}$
- i is a record if $\pi_i = \min\{j \leq i : \pi_j\}$.

$$\mathbf{E}\left[\sum_i \mathbf{1}[i \text{ is a record}]\right] = \sum_{i=1}^{n} 1/i \sim \log n$$

Equivalent to records in a rooted path P



Records II: a tree version

Records in rooted trees: $R(T)$ the number of records in a tree T

- for a tree T rooted at r and a random permutation π
- a vertex $u \in T$ is a record if $\pi_u = \min\{\pi_v : v \in [\![u, r]\!]\}$.

$$\mathbf{E}\left[R(T)\right] = \mathbf{E}\left[\sum_{u\in T} \mathbf{1}[u \text{ is a record}]\right] = n\mathbb{P}\left(U \text{ is a record}\right) = n\mathbf{E}\left[\frac{1}{D_U + 1}\right]$$

If T_n is a rinary search tree

$$\mathbf{E}\left[R(T_n)\right] \sim \frac{n}{2\log n}$$

If T_n is a simply generated tree

$$\mathbf{E}\left[R(T_n)\right] \approx \sqrt{n}$$

Records II: a tree version

Records in rooted trees: $R(T)$ the number of records in a tree T

- for a tree T rooted at r and a random permutation π
- a vertex $u \in T$ is a record if $\pi_u = \min\{\pi_v : v \in [\![u, r]\!]\}$.

$$\mathbf{E}\left[R(T)\right] = \mathbf{E}\left[\sum_{u \in T} \mathbf{1}[u \text{ is a record}]\right] = n\mathbb{P}\left(U \text{ is a record}\right) = n\mathbf{E}\left[\frac{1}{D_U + 1}\right]$$

If T_n is a rinary search tree

$$\mathbf{E}\left[R(T_n)\right] \sim \frac{n}{2\log n}$$

If T_n is a simply generated tree

$$\mathbf{E}\left[R(T_n)\right] \approx \sqrt{n}$$

A coupling

Cut the vertices in the order given by π. Then:

$$u \text{ is cut} \quad \Leftrightarrow \quad u \text{ is a record}$$

Generating rooted labelled trees uniformly

Cayley tree:

- uniformly chosen rooted labelled tree on $\{1, 2, \ldots, n\}$
- uniformly random rooted spanning tree of K_n
- can also generate the shape as a Poisson(1)-Galton–Watson tree conditioned on the size being n.

Algorithm 1

- Start with vertex 1
- For $2 \leq i \leq n$ connect i to

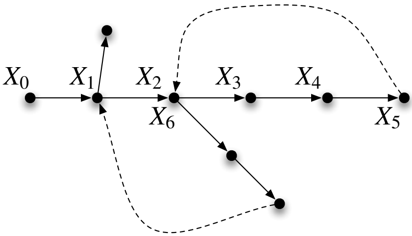
$$V_i = \begin{cases} j & w.p.\ 1/n \qquad\qquad 1 \leq j \leq i-2 \\ i-1 & w.p.\ 1-(i-2)/n \end{cases}$$

- Randomly permute the labels

Generating rooted labelled trees uniformly

Algorithm 2 (Aldous–Broder)

- start from a random vertex X_0;
- move in the graph according to a simple random walk X;
- if X visits X_i for the first time add the edge $X_{i-1} \to X_i$.
- At the cover time, return the tree of directed edges.



Consequences for large random trees

A "building branches" point of view

Walk until the first time we go to a vertex already discovered:

- when the tree has k vertices, we stop the current branch with probability k/n.
- given that we stop, the current vertex is uniform on the already constructed tree

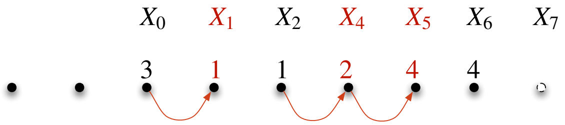
Lengths in Cayley trees

- $\mathbb{P}\left(\text{first branch } \geq k \text{ edges}\right) = \prod_{i=1}^{k}(1 - i/n)$
- critical scaling $k \sim x\sqrt{n}$ (birthday paradox)
- $\mathbb{P}\left(\text{first branch} \geq x\sqrt{n}\right) \sim e^{-x^2/2}$.

A Markov chain of spanning trees

Following the algorithm

- $\{X_j, j \geq 0\}$ a sequence of i.i.d. r.v. uniform in $\{1, 2, \ldots, n\}$
- $\tau(k) = \inf\{j \geq 0 : X_j = k\}$
- T the tree consisting of $\{(X_{\tau(k)-1} \to X_{\tau(k)}) : \tau(k) > 0\}$



A stationary sequence of rooted trees: change the starting point

- $\{X_j, j \in \mathbb{Z}\}$ a sequence of i.i.d. r.v. uniform in $\{1, 2, \ldots, n\}$
- $\tau_i(k) = \inf\{j \geq i : X_j = k\}$
- T^i the tree consisting of $\{(X_{\tau_i(k)-1} \to X_{\tau_i(k)}), \tau_i(k) > i\}$

A Markov chain of spanning trees

Following the algorithm

- $\{X_j, j \geq 0\}$ a sequence of i.i.d. r.v. uniform in $\{1, 2, \ldots, n\}$
- $\tau(k) = \inf\{j \geq 0 : X_j = k\}$
- T the tree consisting of $\{(X_{\tau(k)-1} \to X_{\tau(k)}) : \tau(k) > 0\}$

X_{-2}	X_{-1}	X_0	X_1	X_3	X_4	X_5	X_6	X_7
3	2	3	1	1	2	4	4	1

A stationary sequence of rooted trees: change the starting point

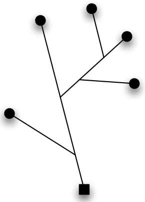
- $\{X_j, j \in \mathbb{Z}\}$ a sequence of i.i.d. r.v. uniform in $\{1, 2, \ldots, n\}$
- $\tau_i(k) = \inf\{j \geq i : X_j = k\}$
- T^i the tree consisting of $\{(X_{\tau_i(k)-1} \to X_{\tau_i(k)}), \tau_i(k) > i\}$

Stick breaking construction for the limit of large trees

Theorem (Scaling limit of Cayley trees)

Viewed as a metric space, with the graph distance,

$$n^{-1/2} T_n \xrightarrow[n\to\infty]{d-GH} \mathcal{T}.$$



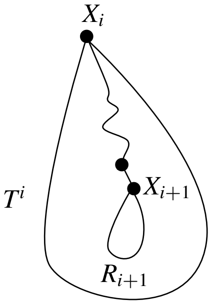
Scaling limit of the Aldous–Broder algorithm

- a Poisson point process with rate xdx splits $\mathbb{R}_{>0}$ into segments;
- Take the segments in order, and make them into a tree shape hooking them at a uniform position
- Taking the closure yields the continuum random tree.

A closer look at the Markov chain construction

Reformulation

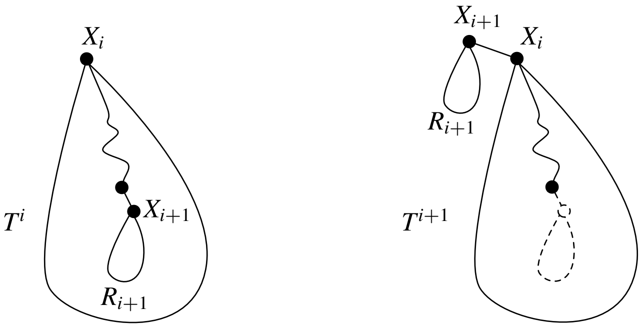
- pick a random node X_{i+1}
- cut it off, together with its subtree R_{i+1}
- Build T^i by appending $T^i \setminus R_{i+1}$ as a child of X_{i+1}



A closer look at the Markov chain construction

Reformulation

- pick a random node X_{i+1}
- cut it off, together with its subtree R_{i+1}
- Build T^i by appending $T^i \setminus R_{i+1}$ as a child of X_{i+1}



Observations about the random walk construction

Facts about the construction

Conditional on their respective sizes (subset of labels):

- left-over tree $T^1 \setminus R_1$ is a Cayley tree
- the cut-off tree R_1 is a Cayley tree

Idea:

- Run the procedure without putting edges from X_{-1} or X_0
- Conditional on $X_{-1} \neq X_0$, this is a random forest of 2 rooted trees.

A probabilistic correspondence

Theorem

There exists a correspondence between:

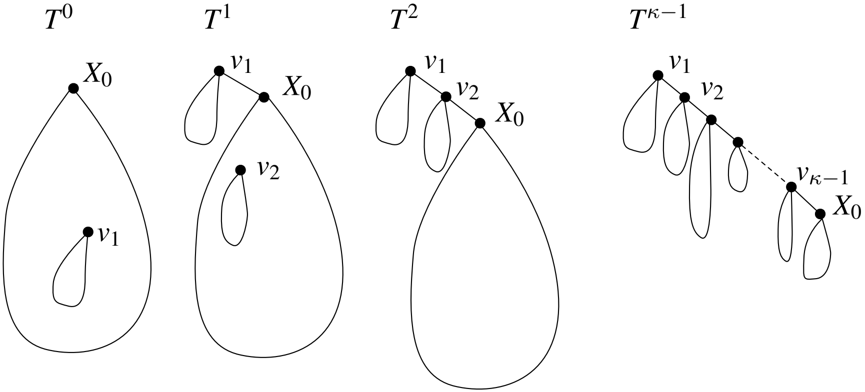
- *a uniform Cayley tree T and a cutting down sequence, and*
- *a uniform Cayley tree $T^\star$ and a uniformly random node $u \in T^\star$.*

In the correspondence, the length of the cutting sequence is turned into the depth of u in $T^\star$(plus one).

Idea of the proof:

- one step of the Markov chain of spanning tree cuts at a uniformly random node;
- for the second step, rather than doing it on the whole tree, just do it on $T^0 \setminus R_1$: replace $T^0 \setminus R_1$ by the its tranformation in one step of the Markov chain
- stop when you hit the root of initial tree T^0.

A picture is worth a thousand words



Sampling a random vertex in a tree

Algorithm

- with probability $|T|^{-1}$, return the root u_1;
- otherwise, pick a child u_2 of u_1 with probability proportional to $|T(u_2)|$ and recurse in $T(u_2)$

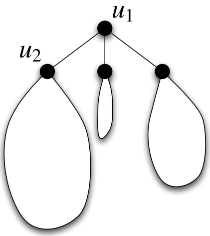
Lemma

The vertex U returned by the procedure is uniform in T.

Proof.

For all $u \in T$, there is a unique path $u_1, u_2, \ldots, u_\kappa = u$. Then $\mathbb{P}(U = u)$:

$$\prod_{i=1}^{\kappa-1} \left(1 - \frac{1}{|T(u_i)|}\right) \frac{|T(u_{i+1})|}{|T(u_i)| - 1} \times \frac{1}{|T(u_\kappa)|} = \frac{1}{|T|} \qquad \square$$



Why does it work?

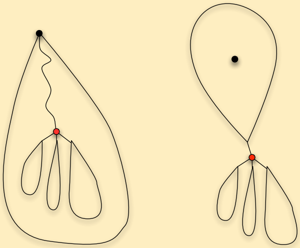
Lemma

At time κ, when $v_\kappa = X_0$, the vertex X_0 is uniformly random in $T^{\kappa-1}$

Proof.

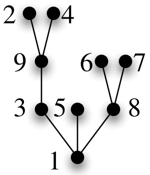
The growth of the path $v_1, v_2, \ldots, v_\kappa$ follows the distribution of the path to a uniformly random node:

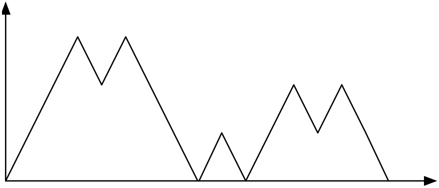
- with probability $|T^i(X_0)|^{-1}$ return X_0
- otherwise $T^{i+1}(X_0)$ is size-biased



$\square$

About walks associated with trees





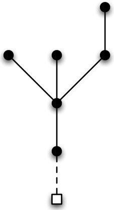
Let T_n be a Cayley tree of size n, with contour function $C(\cdot)$

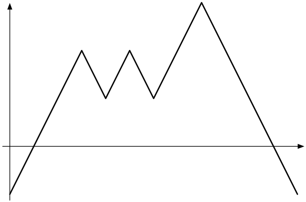
Theorem (Marckert–Mokkadem)

Let $e = (e(t), 0 \leq t \leq 1)$ be a standard Brownian excursion. Then,

$$\frac{C(\lfloor 2n \cdot \rfloor)}{\sqrt{n}} \xrightarrow[n\to\infty]{d} 2e(\cdot)$$

Towards cutting down lattice paths



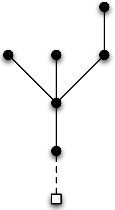


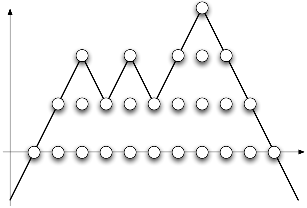
Let $A = \{(x,y) : 0 \leq x \leq 2n \text{ and } 0 \leq y \leq C_T(x)\}$
Say $\pi(p) = i$ if $p \in A$ correspond to the point $u \in T$
For $p \in A$ let $\ell_p = \#\{p' : \pi(p') = \pi(p)\}$

Sampling nodes uniformly at random

- pick $p = (x,y)$ with probability proportional to ℓ_p^{-1}
- then, $\pi(p)$ is uniform in T.

Towards cutting down lattice paths





Let $A = \{(x, y) : 0 \le x \le 2n \text{ and } 0 \le y \le C_T(x)\}$

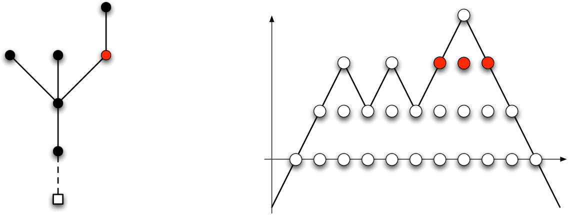
Say $\pi(p) = i$ if $p \in A$ correspond to the point $u \in T$

For $p \in A$ let $\ell_p = \#\{p' : \pi(p') = \pi(p)\}$

Sampling nodes uniformly at random

- pick $p = (x, y)$ with probability proportional to ℓ_p^{-1}
- then, $\pi(p)$ is uniform in T.

Towards cutting down lattice paths

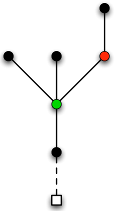


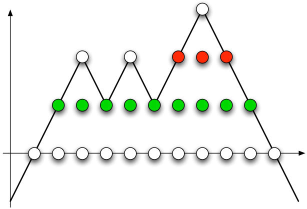
Let $A = \{(x, y) : 0 \leq x \leq 2n \text{ and } 0 \leq y \leq C_T(x)\}$
Say $\pi(p) = i$ if $p \in A$ correspond to the point $u \in T$
For $p \in A$ let $\ell_p = \#\{p' : \pi(p') = \pi(p)\}$

Sampling nodes uniformly at random

- pick $p = (x, y)$ with probability proportional to ℓ_p^{-1}
- then, $\pi(p)$ is uniform in T.

Towards cutting down lattice paths





Let $A = \{(x,y) : 0 \le x \le 2n \text{ and } 0 \le y \le C_T(x)\}$
Say $\pi(p) = i$ if $p \in A$ correspond to the point $u \in T$
For $p \in A$ let $\ell_p = \#\{p' : \pi(p') = \pi(p)\}$

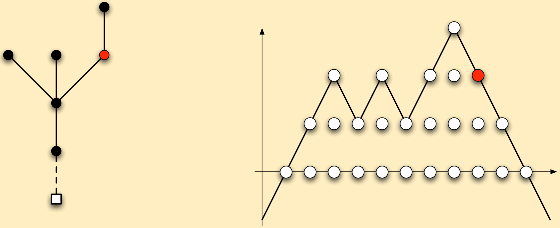
Sampling nodes uniformly at random

- pick $p = (x,y)$ with probability proportional to ℓ_p^{-1}
- then, $\pi(p)$ is uniform in T.

Cutting down lattice paths

Every cut inhibates a region of A:

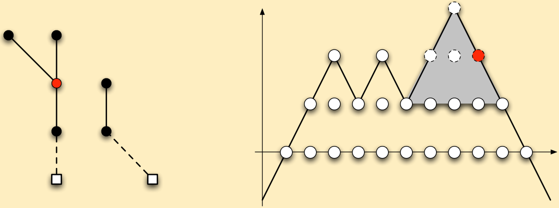
$2\#\{\text{nodes left in the tree}\} = \mathit{length}\{\text{fully active region}\}$



Cutting down lattice paths

Every cut inhibates a region of A:

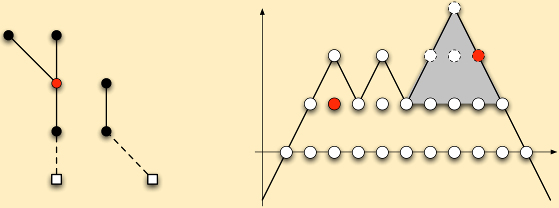
$2\#\{\text{nodes left in the tree}\} = \mathit{length}\{\text{fully active region}\}$



Cutting down lattice paths

Every cut inhibates a region of A:

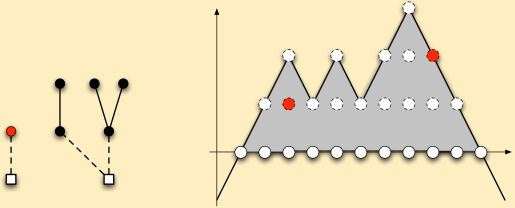
$2\#\{\text{nodes left in the tree}\} = \mathit{length}\{\text{fully active region}\}$



Cutting down lattice paths

Every cut inhibates a region of A:

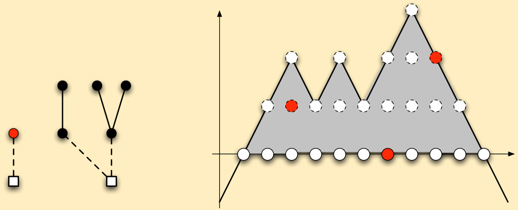
$2\#\{\text{nodes left in the tree}\} = \mathit{length}\{\text{fully active region}\}$



Cutting down lattice paths

Every cut inhibates a region of A:

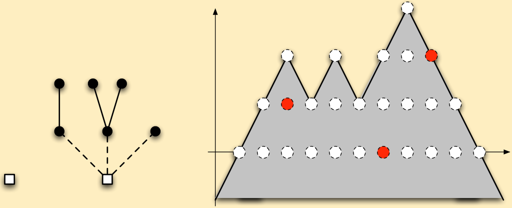
$2\#\{\text{nodes left in the tree}\} = \mathit{length}\{\text{fully active region}\}$



Cutting down lattice paths

Every cut inhibates a region of A:

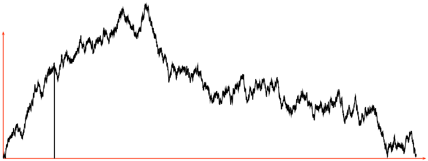
$2\#\{\text{nodes left in the tree}\} = \mathit{length}\{\text{fully active region}\}$



Cutting down the mightiest tree with … a Herring!

Correct scaling:

- mass by n
- distances by $\sqrt{n}$
- time by $\sqrt{n}$

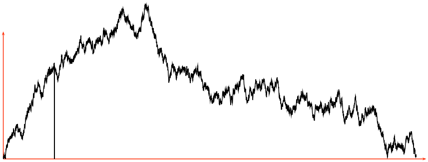


Setting: e an excursion

- $A_e = \{p = (x, y) : 0 \leq y \leq e(x)\}$
- $p \in A_e$, let $\ell_p = |\{p' : \pi(p) = \pi(p')\}|$
- $\mathcal{P}$ be a Poisson point process on $A_e \times [0, \infty)$ with intensity $\ell_{(x,y)}^{-1} dxdydt$
- $\mathcal{P}^\star \subseteq \mathcal{P}$ the points landing in the active region
- $\mu(s)$ the length of the fully active region (**mass left**) at time s

Cutting down the mightiest tree with ... a Herring!

Poisson Point Process



Correct scaling:

- mass by n
- distances by $\sqrt{n}$
- time by $\sqrt{n}$

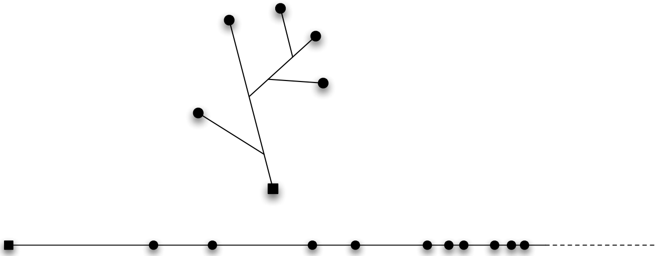
Setting: e an excursion

- $A_e = \{p = (x, y) : 0 \leq y \leq e(x)\}$
- $p \in A_e$, let $\ell_p = |\{p' : \pi(p) = \pi(p')\}|$
- $\mathcal{P}$ be a Poisson point process on $A_e \times [0, \infty)$ with intensity $\ell^{-1}_{(x,y)} dxdydt$
- $\mathcal{P}^\star \subseteq \mathcal{P}$ the points landing in the active region
- $\mu(s)$ the length of the fully active region (**mass left**) at time s

An other point of view

An ode to Poisson processes

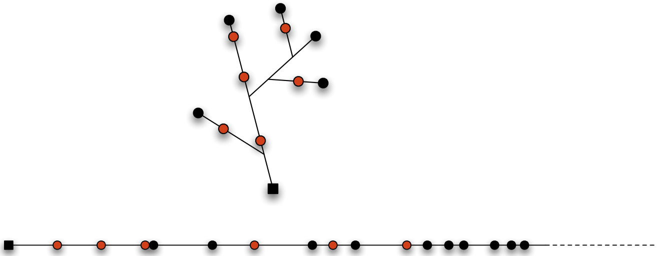
- Build the CRT from the PPP on $[0, \infty)$ with intensity xdx
- Cut the CRT with an other PPP on $[0, \infty) \times \mathbb{R}_{\geq 0}$ with intensity $dxdt$



An other point of view

An ode to Poisson processes

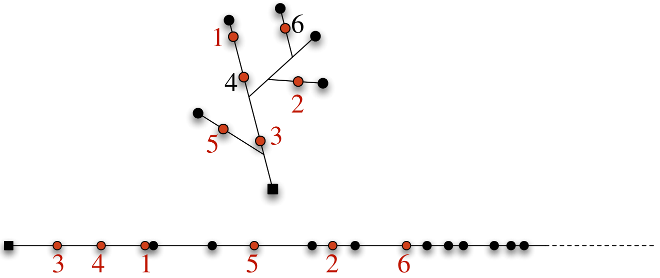
- Build the CRT from the PPP on $[0, \infty)$ with intensity xdx
- Cut the CRT with an other PPP on $[0, \infty) \times \mathbb{R}_{\geq 0}$ with intensity $dxdt$



An other point of view

An ode to Poisson processes

- Build the CRT from the PPP on $[0, \infty)$ with intensity xdx
- Cut the CRT with an other PPP on $[0, \infty) \times \mathbb{R}_{\geq 0}$ with intensity $dxdt$



Number of cuts in terms of the mass left

Lemma

Let T_n be a Cayley tree:

$$\frac{\kappa(T_n)}{\sqrt{n}} \xrightarrow[n\to\infty]{d} \int_0^\infty \mu(s)ds \stackrel{d}{=} R$$

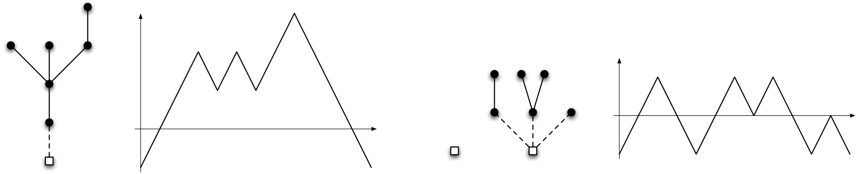
Proof.

As $n \to \infty$

$$\frac{\kappa(T_n)}{\sqrt{n}} = \frac{1}{\sqrt{n}} \sum_{i\geq 0} \mathbf{1}[p_i \in P^\star] \approx \lim_{t\to\infty} \frac{1}{\sqrt{n}} \sum_{i=0}^{t\sqrt{n}} \mathbf{1}[p_i \in P^\star]$$

$$\approx \lim_{t\to\infty} \frac{1}{\sqrt{n}} \sum_{i=0}^{t\sqrt{n}} \frac{m(i)}{n} \approx \int_0^\infty \mu(s)ds \qquad \square$$

From an excursion to a reflected bridge



Theorem

In the limit, the bijection turns a Brownian excursion e *into a Brownian bridge* B *and*

$$\frac{\kappa(T_n)}{\sqrt{n}} \xrightarrow[n\to\infty]{d} L_1(B) \overset{d}{=} R,$$

where $(L_s(B), 0 \leq s \leq 1)$ *is the local time process of* B *at* 0.

Extension to all Galton–Watson trees with finite variance

ξ r.v. such that $\mathbf{E}\xi = 1$ and $\mathbf{Var}[\xi] = \sigma^2 < \infty$

Theorem

Let T_n be a $GW(\xi)$ conditioned to be of size n:

$$\frac{\kappa(T_n)}{\sigma\sqrt{n}} \xrightarrow[n\to\infty]{d} R$$

Proof.

- at the scale $(n, \sqrt{n}, \sqrt{n})$ the process looks the same, but time flows more slowly by factor σ
- show that when the tree has size $o(n)$ the number of records is $o(\sqrt{n})$

$\square$

Isolating the randomness of the cut process

Two levels of randomness

- Random tree/excursion
- Random cutting process

Theorem

Conditional on $n^{-1/2}T_n \to \mathcal{T}$ *encoded by* $2e$,

$$\mathbb{P}\left(\frac{\kappa(T_n)}{\sqrt{n}} \leq x \;\middle|\; \mathcal{T}_n\right) \xrightarrow[n\to\infty]{d} \mathbb{P}\left(\int_0^\infty \mu_{2e}(s)ds \leq x \;\middle|\; e\right)$$