Oldest-Edge Deletion in a Preferential Attachment Graph

Tony Johansson

Uppsala University

RS&A 2017
Gniezno

Preferential Attachment Graphs

Random graph model popular for evolving real-world networks

Based on concept “the rich get richer”

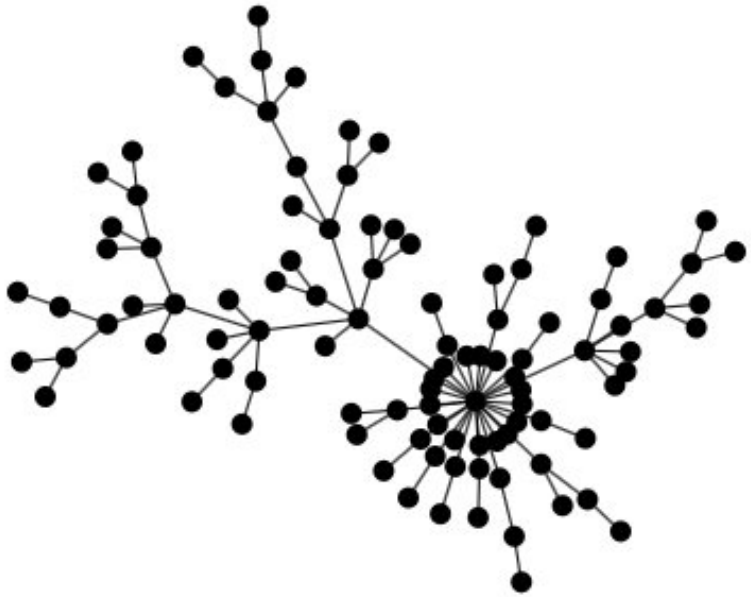
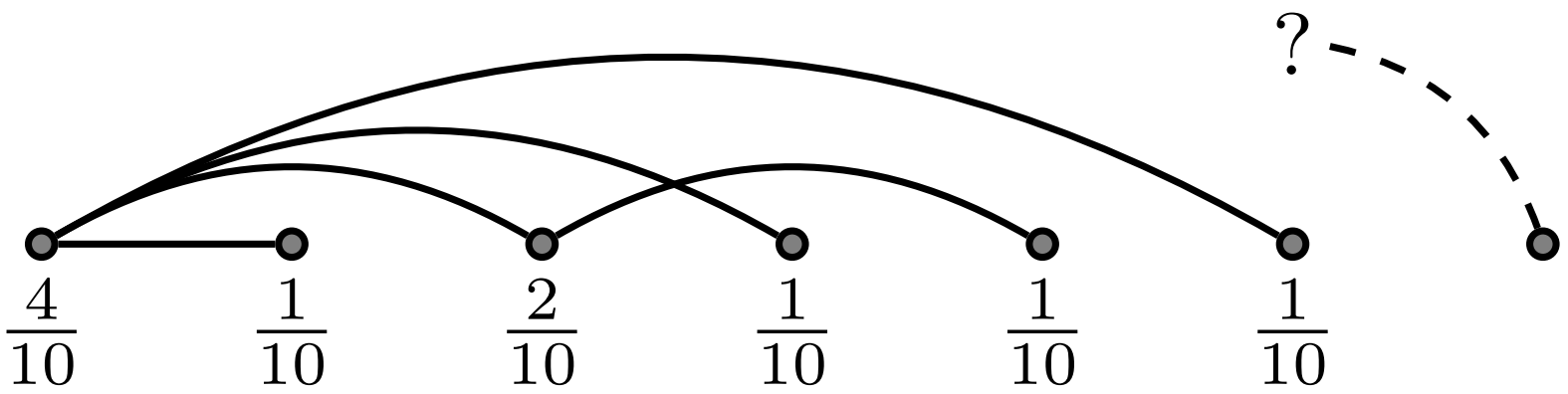


Figure: isomorphis.wordpress.com

Preferential Attachment Graphs

Barabási-Albert model: Given a graph G_t on $[t]$, form G_{t+1} by adding vertex $t+1$ along with $m \geq 1$ edges.

Probability of edge choosing v is proportional to degree of v in G_t.



Preferential Attachment Graphs

Barabási-Albert model popular since it follows a power law.

Theorem (Bollobás, Riordan, Spencer, Tusnády 2001)

Let X_k denote the number of vertices of degree k in G_n. There exists an $\alpha > 0$ such that for large k,

$$\lim_{n\to\infty} \frac{X_k}{n} = k^{-3}(\alpha + o_k(1)).$$

Preferential Attachment Graphs

Barabási-Albert model popular since it follows a power law.

Theorem (Bollobás, Riordan, Spencer, Tusnády 2001)

Let X_k denote the number of vertices of degree k in G_n. There exists an $\alpha > 0$ such that for large k,

$$\lim_{n\to\infty} \frac{X_k}{n} = k^{-3}(\alpha + o_k(1)).$$

Real-world networks often follow power laws.

Preferential Attachment Graphs

Barabási-Albert model popular since it follows a power law.

Theorem (Bollobás, Riordan, Spencer, Tusnády 2001)

Let X_k denote the number of vertices of degree k in G_n. There exists an $\alpha > 0$ such that for large k,

$$\lim_{n\to\infty} \frac{X_k}{n} = k^{-3}(\alpha + o_k(1)).$$

Real-world networks often follow power laws.

Most random graph models, on the contrary, follow an exponential law:

$$\frac{X_k}{n} \longrightarrow c^k(\alpha + o_k(1)), \quad 0 < c < 1$$

Edge Deletion

We consider a variation of the Barabási-Albert model with online oldest-edge deletion, to be defined later.

We will consider the following questions:

- What is the probability distribution of $\deg v$ for a fixed vertex v?
- What is the degree sequence of the graph?
- What is the component structure of the graph?

Alternative View ($m = 1$)

Suppose an edge e is added to a graph G. Then

e chooses a vertex in G with probability proportional to degree

$\Longleftrightarrow$

e chooses an edge f uniformly at random from G, and chooses an endpoint of f uniformly at random

Indeed,

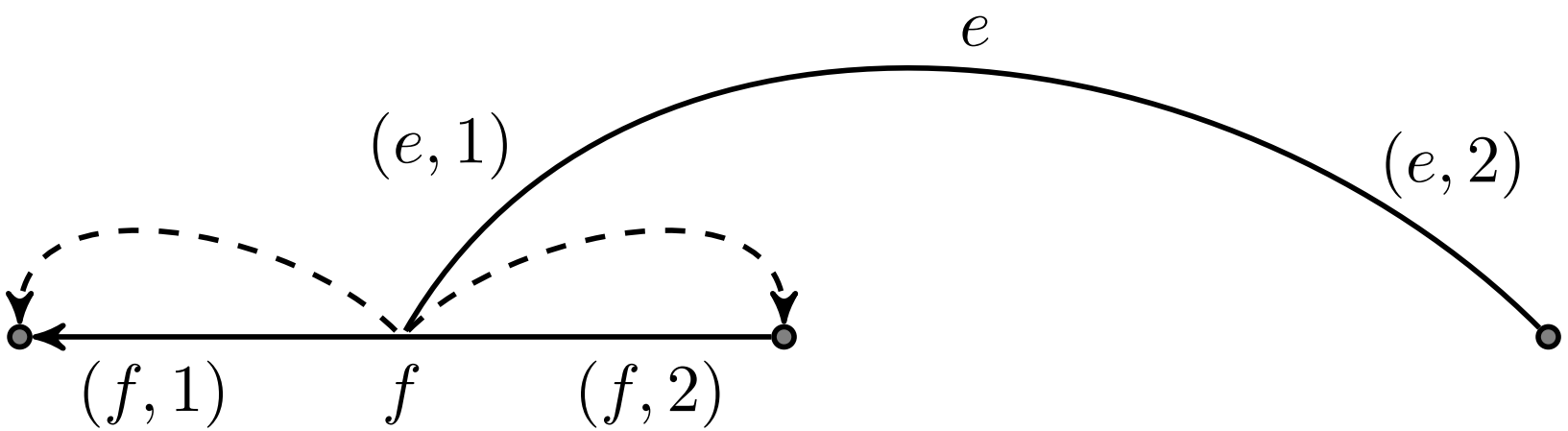
$$\frac{\deg v}{\sum_u \deg u} = \frac{1}{2|E(G)|} \sum_{f \in E(G)} [v \in f]$$

Alternative View ($m = 1$)

Edges are oriented from right to left. The base is fixed while the tip is random. If e chooses (f, j) we say that

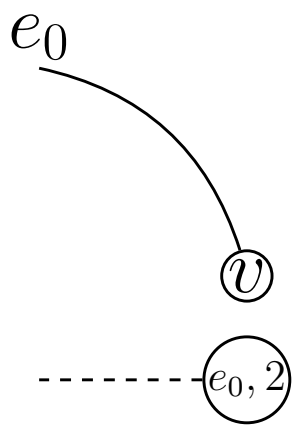
- $j = 1$ corresponds to choosing the tip of f,
- $j = 2$ to choosing the base of f.

Then $(e, 1)$ and (f, j) are incident to the same vertex, while $(e, 2)$ is incident to the vertex e was added with



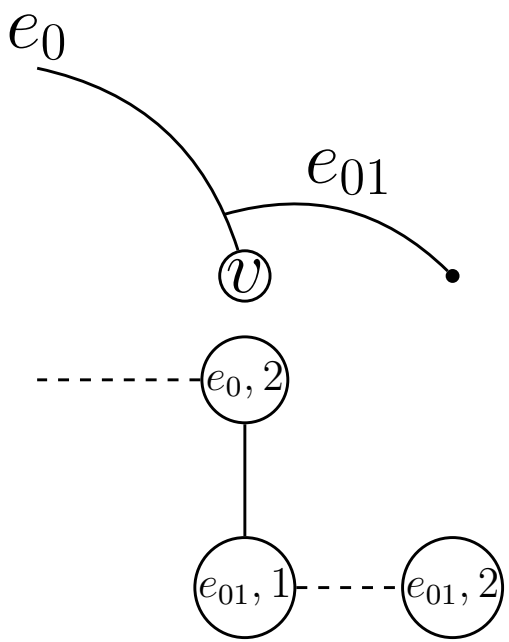
Alternative View ($m = 1$)

Edges choosing edges sets up partition of pairs (e, j)



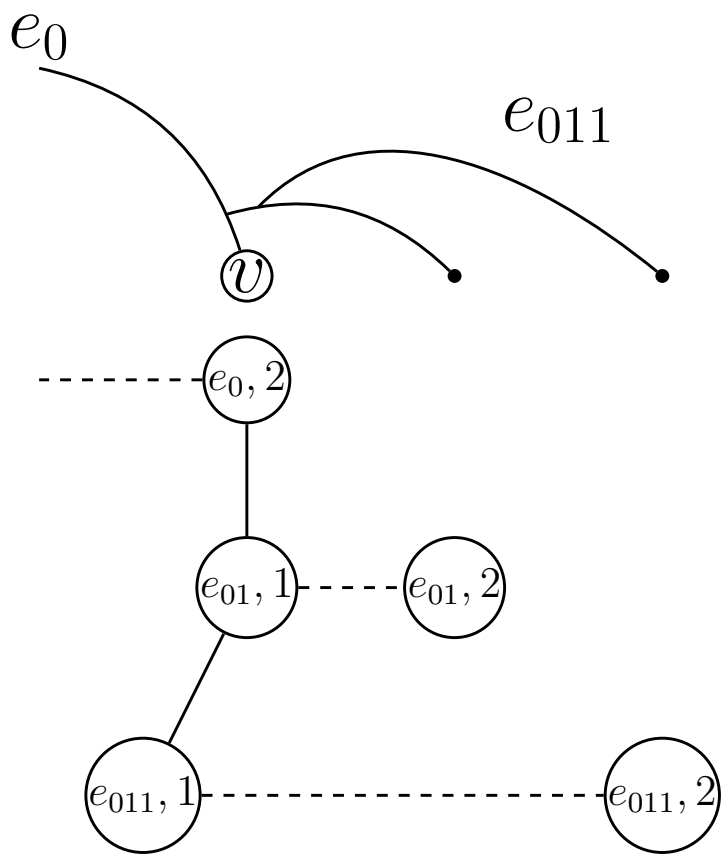
Alternative View ($m = 1$)

Edges choosing edges sets up partition of pairs (e, j)



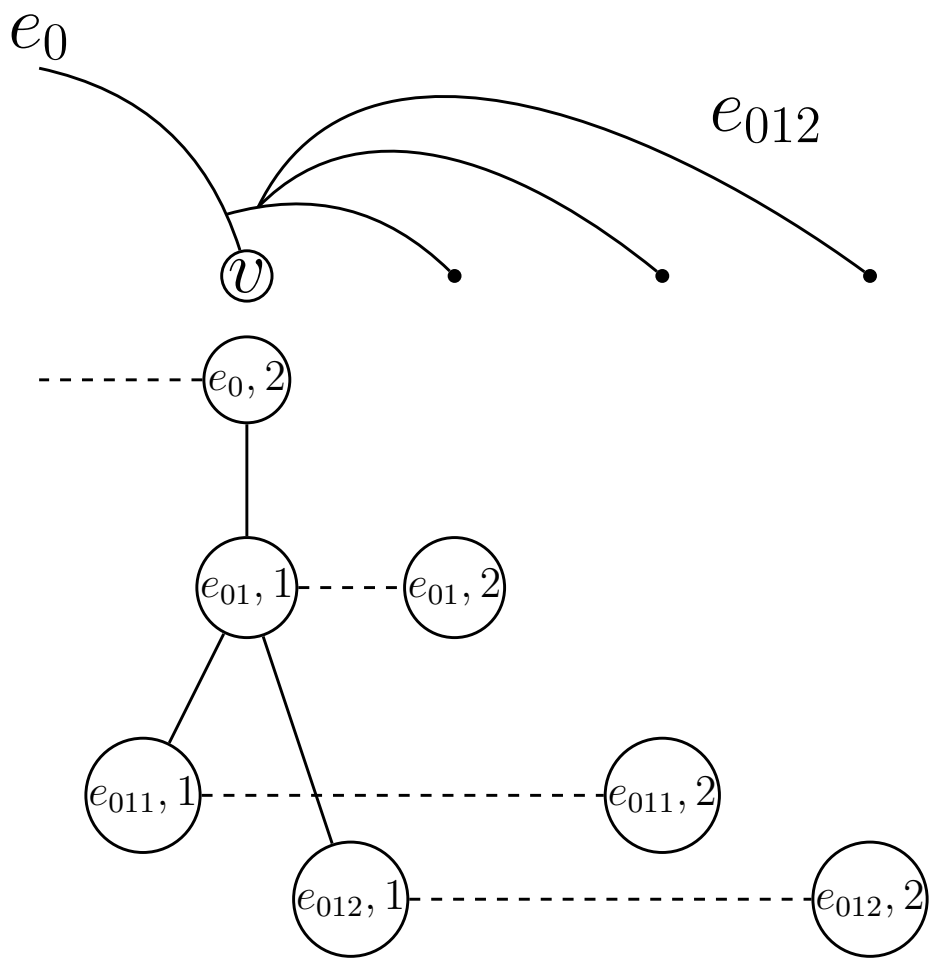
Alternative View ($m = 1$)

Edges choosing edges sets up partition of pairs (e, j)



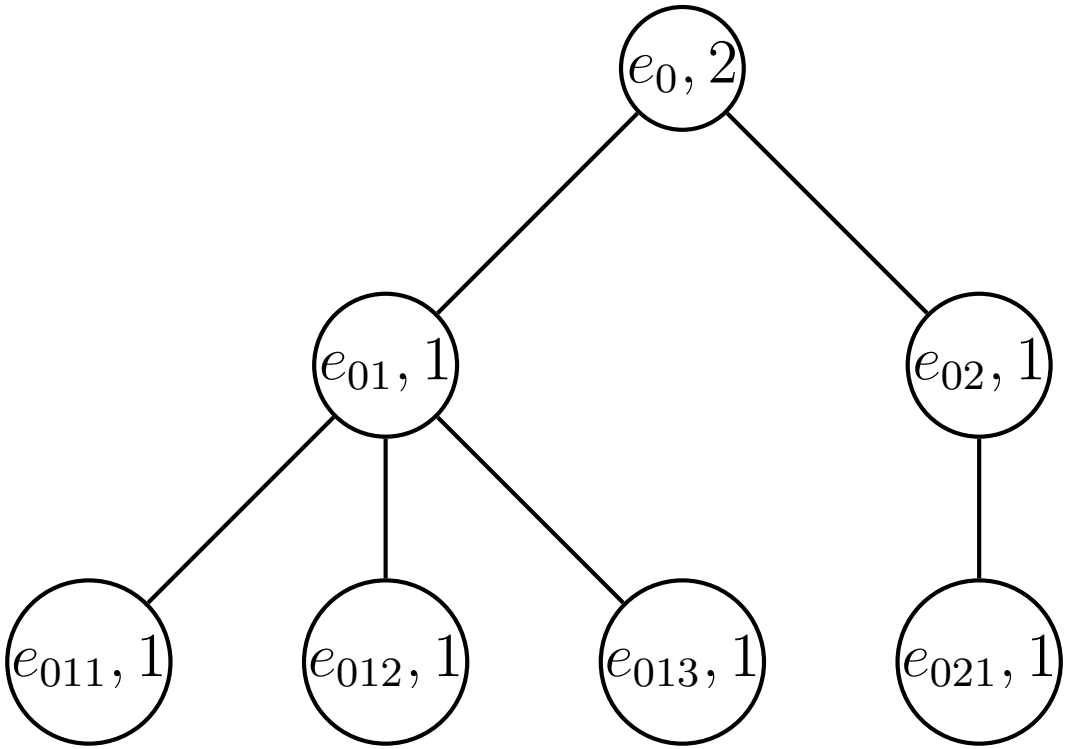
Alternative View ($m = 1$)

Edges choosing edges sets up partition of pairs (e, j)



Alternative View ($m = 1$)

Degree of v equals size of tree $T(v)$



Alternative View ($m = 1$)

Edge e is added at time t attaches to any (f, j) with probability $\frac{1}{t}$

In time range $[t, (1 + \varepsilon)t]$, any (f, j) gains a child with probability $\varepsilon + O(\varepsilon^2)$

Alternative View ($m = 1$)

Edge e is added at time t attaches to any (f, j) with probability $\frac{1}{t}$

In time range $[t, (1 + \varepsilon)t]$, any (f, j) gains a child with probability $\varepsilon + O(\varepsilon^2)$

Re-scaling time logarithmically, each (f, j) attracts nodes according to a Poisson process

Alternative View ($m = 1$)

Edge e is added at time t attaches to any (f, j) with probability $\frac{1}{t}$

In time range $[t, (1+\varepsilon)t]$, any (f, j) gains a child with probability $\varepsilon + O(\varepsilon^2)$

Re-scaling time logarithmically, each (f, j) attracts nodes according to a Poisson process

Tree $T(v)$ can be described as a Crump-Mode-Jagers process (branching Poisson process)

Alternative View ($m = 1$)

Edge e is added at time t attaches to any (f, j) with probability $\frac{1}{t}$

In time range $[t, (1 + \varepsilon)t]$, any (f, j) gains a child with probability $\varepsilon + O(\varepsilon^2)$

Re-scaling time logarithmically, each (f, j) attracts nodes according to a Poisson process

Tree $T(v)$ can be described as a Crump-Mode-Jagers process (branching Poisson process)

Size of tree is geometrically distributed, average depending on age of v

Edge Deletion

Introduce parameter $1/2 < p < 1$.

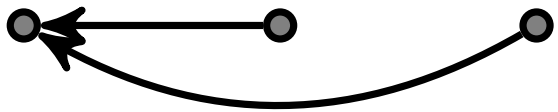
With probability p add vertex with m edges like before,
with probability $1 - p$ remove m oldest edges.

Vertex set $\{1, \ldots, \nu_t\}$ for some random ν_t.

Edge Deletion

Introduce parameter $1/2 < p < 1$.

With probability p add vertex with m edges like before, with probability $1 - p$ remove m oldest edges.



Vertex set $\{1, \ldots, \nu_t\}$ for some random ν_t.

Edge Deletion

Introduce parameter $1/2 < p < 1$.

With probability p add vertex with m edges like before, with probability $1 - p$ remove m oldest edges.

Vertex set $\{1, \ldots, \nu_t\}$ for some random ν_t.

Edge Deletion

Introduce parameter $1/2 < p < 1$.

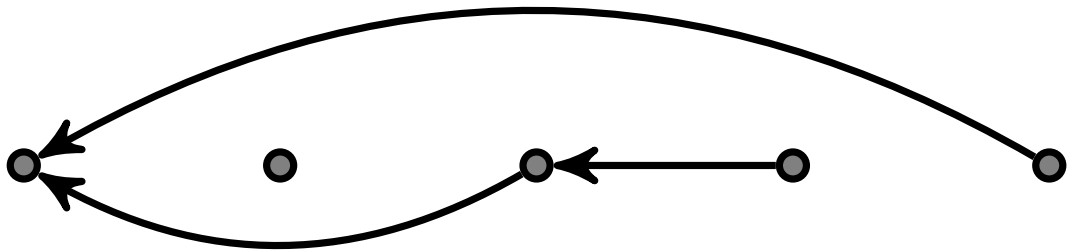
With probability p add vertex with m edges like before,
with probability $1 - p$ remove m oldest edges.

Vertex set $\{1, \ldots, \nu_t\}$ for some random ν_t.

Edge Deletion

Introduce parameter $1/2 < p < 1$.

With probability p add vertex with m edges like before, with probability $1 - p$ remove m oldest edges.

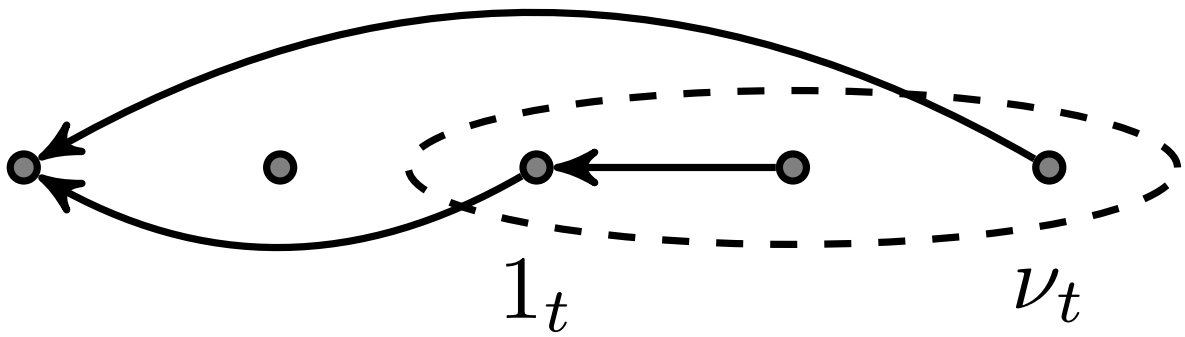


Vertex set $\{1, \ldots, \nu_t\}$ for some random ν_t.

Edge Deletion

Introduce parameter $1/2 < p < 1$.

With probability p add vertex with m edges like before, with probability $1 - p$ remove m oldest edges.



Vertex set $\{1, \ldots, \nu_t\}$ for some random ν_t.

Edge Deletion

Goals:

1. Determine degree sequence. Power law?
2. Is there a giant component?

Edge Deletion

Goals:

1. Determine degree sequence. Power law?
2. Is there a giant component?

Tools:

1. Couple degree of a vertex to Crump-Mode-Jagers process (branching Poisson process)
2. Generate graph off-line

Edge Deletion

Suppose $m = 1$. We will consider $m > 1$ later.

Theorem (J.)

There exist $p_{n,v}$ $q_{n,v}$ such that for any v and $k \geq 0$,

$$Pr\{\deg v = k\} \approx \begin{cases} 1 - q_{n,v}, & k = 0, \\ q_{n,v} p_{n,v} (1 - p_{n,v})^{k-1}, & k > 0. \end{cases}$$

In other words, $\deg v$ is either zero or geometrically distributed.

The proof explicitly provides $p_{n,v}$, $q_{n,v}$.

Edge Deletion

Suppose $m = 1$. We will consider $m > 1$ later.

Theorem (J.)

There exist $p_{n,v}$ $q_{n,v}$ such that for any v and $k \geq 0$,

$$Pr\{\deg v = k\} \approx \begin{cases} 1 - q_{n,v}, & k = 0, \\ q_{n,v} p_{n,v} (1 - p_{n,v})^{k-1}, & k > 0. \end{cases}$$

In other words, $\deg v$ is either zero or geometrically distributed.

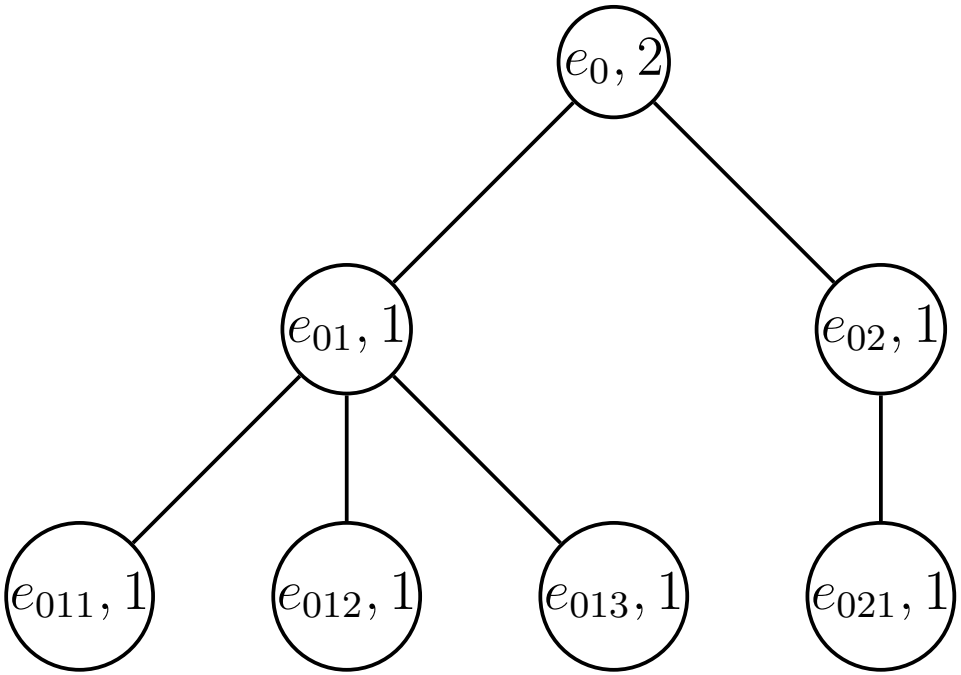
The proof explicitly provides $p_{n,v}$, $q_{n,v}$.

Theorem (J.)

There exists a $p_0 \approx 0.83$ such that if $p > p_0$ then G_n follows a power law whp, and if $p < p_0$ then G_n follows an exponential law whp.

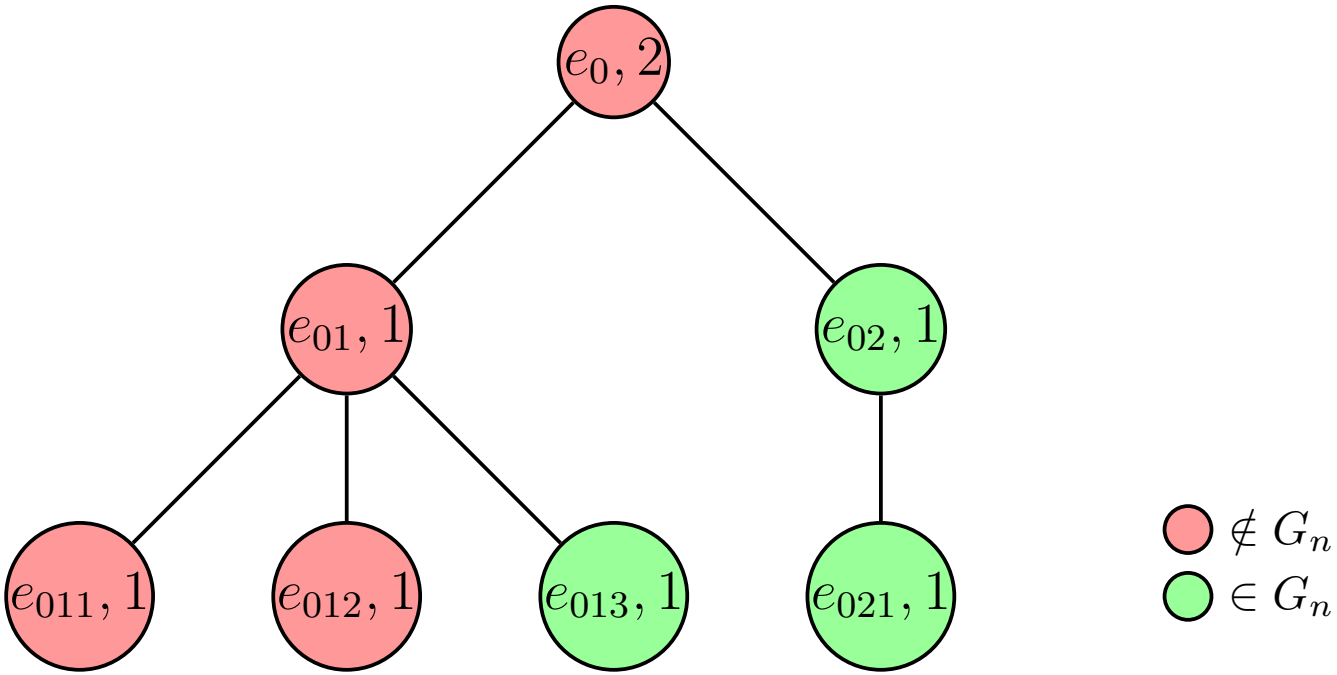
Degree Distribution

Need to keep track of which edges are still in graph



Degree Distribution

Need to keep track of which edges are still in graph



This vertex has degree 3 in G_n

Degree Distribution

The sequence $\sigma \in \{\pm 1\}^n$ of additions/removals is essentially a simple random walk

Degree Distribution

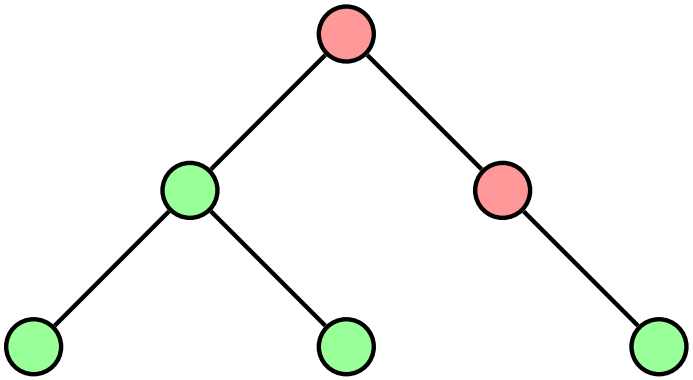
The sequence $\sigma \in \{\pm 1\}^n$ of additions/removals is essentially a simple random walk

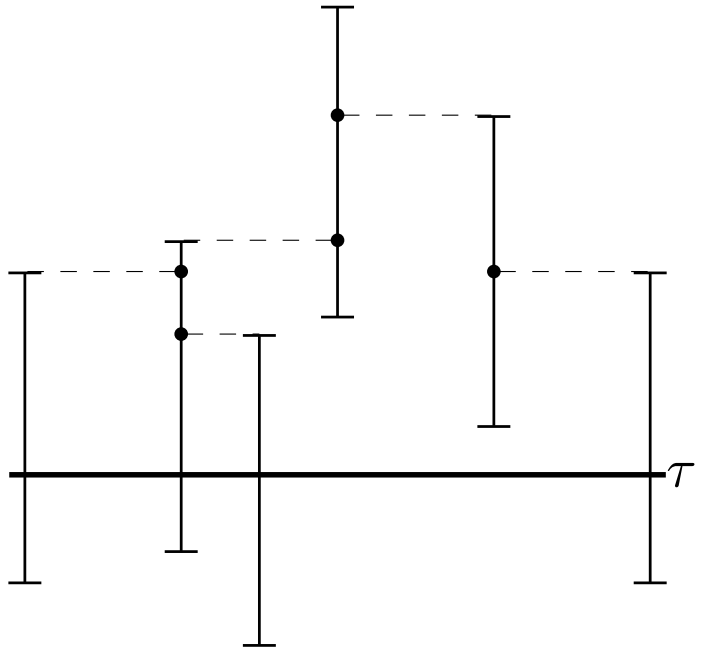
An edge e added at time t is removed at time $\approx \frac{p}{1-p} t$ whp

Determining existence of an edge is almost deterministic as function of its age!

Degree Distribution

Rescaling time logarithmically, $\tau = \log_{\frac{p}{1-p}} t$, an edge added at time τ is removed at time $\approx \tau + 1$





Branching Poisson processes now have life-time 1. Degree is number of active processes at time τ

Degree Distribution

Crump, Mode (1969): Number of active processes in this Crump-Mode-Jagers process at time τ has probability generating function $F(s,\tau)$ given by

$$F(s,\tau) = s \exp\left\{\alpha \int_0^\tau (F(s,u)-1)du\right\}, 0 \leq \tau \leq 1$$

$$F(s,\tau) = \exp\left\{\alpha \int_{\tau-1}^\tau (F(s,u)-1)du\right\}, \tau > 1.$$

Degree Distribution

Crump, Mode (1969): Number of active processes in this Crump-Mode-Jagers process at time τ has probability generating function $F(s, \tau)$ given by

$$F(s,\tau) = s \exp\left\{\alpha \int_0^\tau (F(s,u) - 1)du\right\}, 0 \leq \tau \leq 1$$

$$F(s,\tau) = \exp\left\{\alpha \int_{\tau-1}^\tau (F(s,u) - 1)du\right\}, \tau > 1.$$

Solution is zero-or-geometric distribution, parameters depending on τ

Degree Sequence

G_n behaves differently depending on p.

Theorem (J.)

Let X_k denote the number of vertices of degree k in G_n. Then

$$\frac{X_k}{n} \sim \begin{cases} \alpha^{k(1+o_k(1))}, & \alpha < 1, \\ k^{-\eta-1}(1+o_k(1)), & \alpha > 1, \end{cases}$$

where $\eta = \eta(p) \in (2, \infty)$.

As a continous function of $p \in (1/2, 1)$, α increases from $1/2$ to ∞ with $\alpha = 1$ at $p \approx 0.83$.

Why a Phase Transition?

- When $\alpha < 1$, the CMJ process is ultimately extinct whp. Old vertices are isolated.

- When $\alpha > 1$, the process survives with some positive probability, and the old vertices have high expected degree.

Larger m

If $m > 1$, we can generate the $m = 1$ graph and merge groups of m adjacent nodes

Degrees are sums of m iid random variables

Degree sequence is essentially unaffected, in particular phase transition occurs at $p \approx 0.83$ independently of m

Larger m

If $m > 1$, we can generate the $m = 1$ graph and merge groups of m adjacent nodes

Degrees are sums of m iid random variables

Degree sequence is essentially unaffected, in particular phase transition occurs at $p \approx 0.83$ independently of m

Larger m lead to a giant component, as we will now see

Intermission

So far we have

- Found probability distribution of $\deg v$, and
- determined the degree sequence of G_n.

Intermission

So far we have

- Found probability distribution of deg v, and
- determined the degree sequence of G_n.

Remainder of talk:

- Show how to generate graph off-line,
- Find conditions for unique giant component

Giant Component

Theorem (J.)

If $m = 1$ the largest component of G_n has size $O(\Delta(G_n) \log n)$ whp. If $m > 1$ the largest component has size $\Omega(n)$ while all other components have size $O(\log n)$ whp.

Furthermore, if $\alpha > 1$ then whp the largest component contains all but a $(13/14)^m$ proportion of the edges, and the number of isolated vertices is $\xi n(1 + o_n(1))$ for some $0 < \xi < p$.

Giant Component

Theorem (J.)

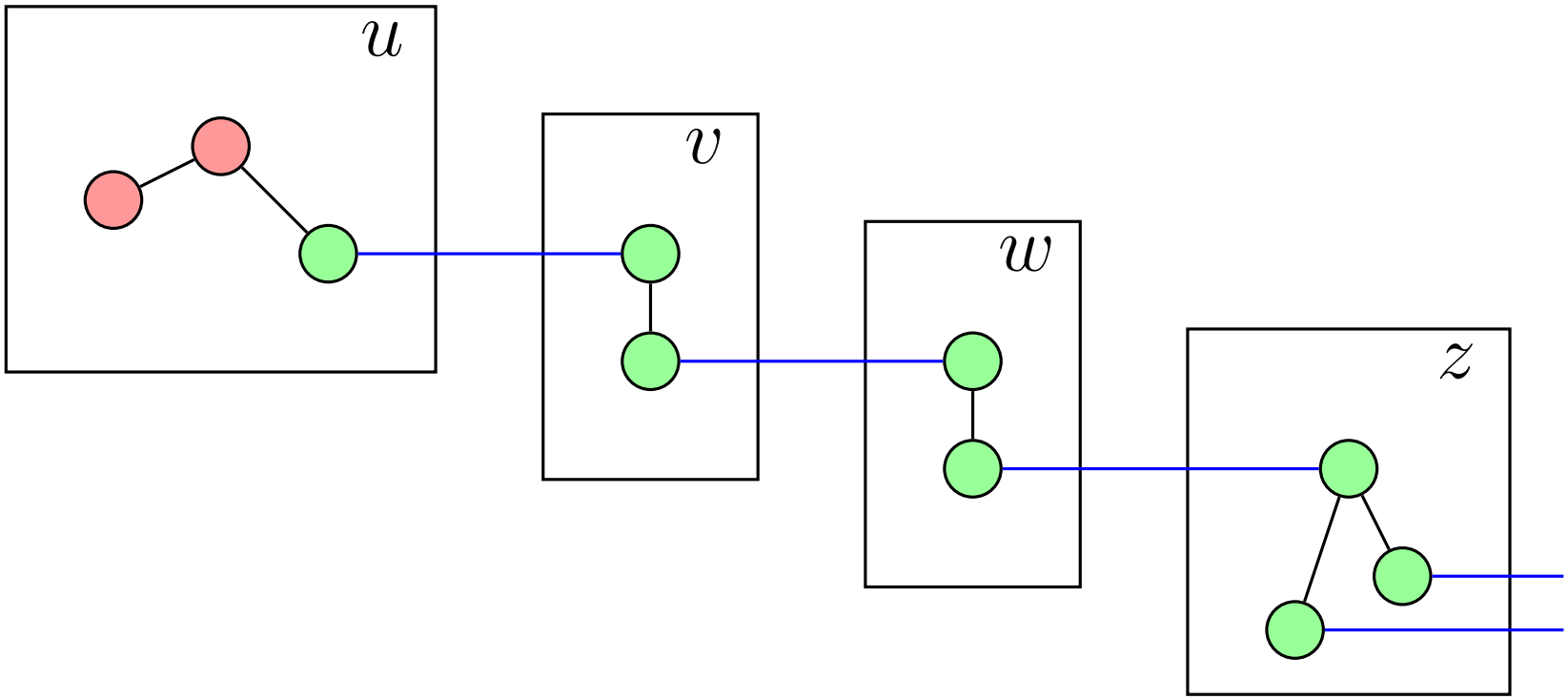
If $m = 1$ the largest component of G_n has size $O(\Delta(G_n) \log n)$ whp. If $m > 1$ the largest component has size $\Omega(n)$ while all other components have size $O(\log n)$ whp.

Furthermore, if $\alpha > 1$ then whp the largest component contains all but a $(13/14)^m$ proportion of the edges, and the number of isolated vertices is $\xi n(1 + o_n(1))$ for some $0 < \xi < p$.

- With high probability maximum degree $\Delta(G_n)$ is $O(\log n)$ when $\alpha < 1$ and $O(n^{1/\eta}) = o(n^{1/2})$ when $\alpha > 1$.
- Existence of linear-sized component is independent of p.

Exploring the Graph

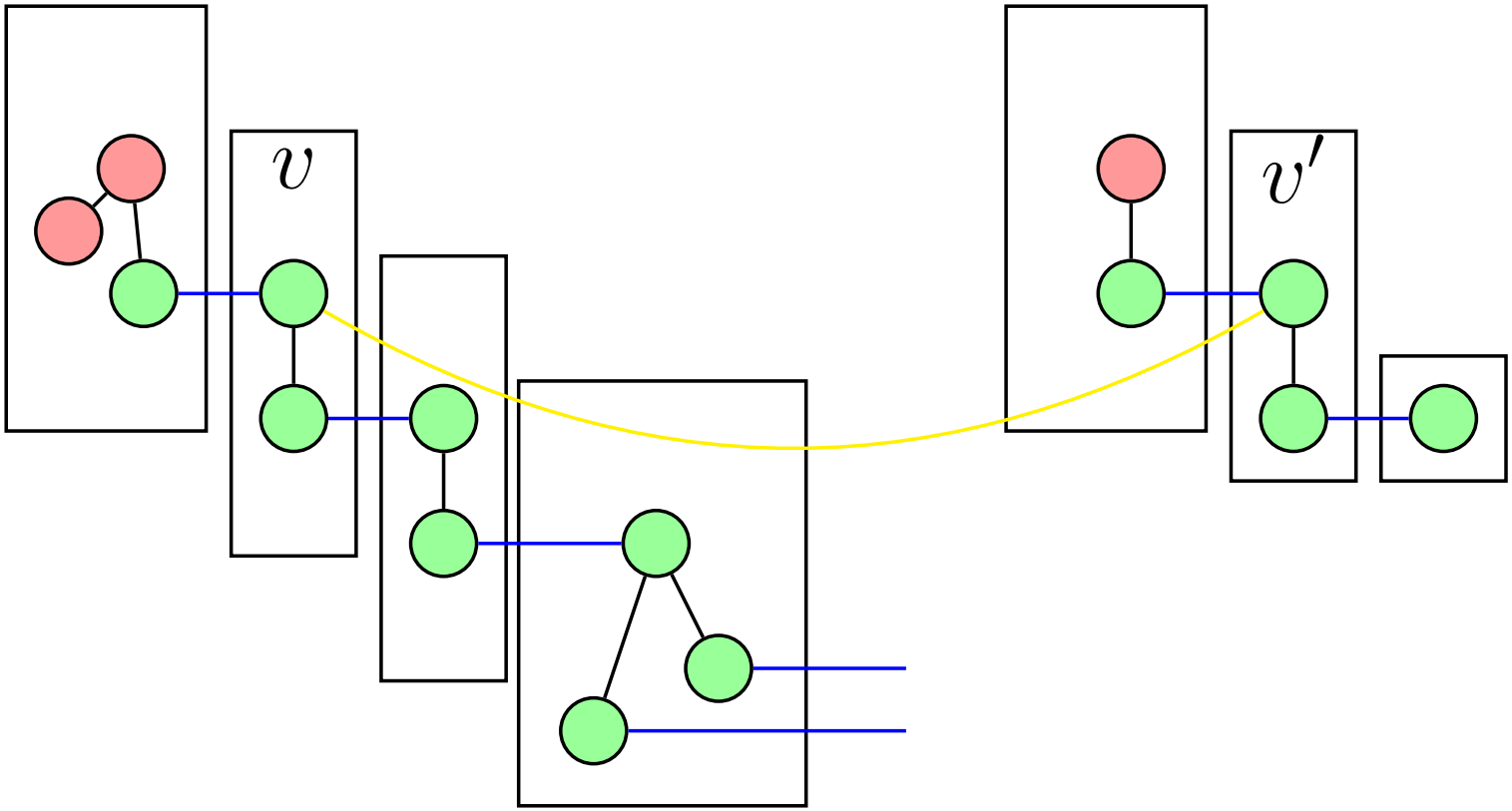
When $m = 1$, the component structure is as below



- Box — all half-edges incident to vertex
- Blue edge — connects two halves of a living edge

Exploring the Graph

When $m = 2$, we merge v with a vertex v' (yellow edge)



Larger m gives more yellow edges

Exploring the Graph

Starting with some edge e, the exploration process is:

1. Find all other edges incident to e, (black edges)
2. Follow all new-found edges to their other endpoint (blue edges)
3. Explore components of vertices to be merged with current component (yellow edges)

Exploring the Graph

Starting with some edge e, the exploration process is:

1. Find all other edges incident to e, (black edges)
2. Follow all new-found edges to their other endpoint (blue edges)
3. Explore components of vertices to be merged with current component (yellow edges)

Every time we follow a yellow edge, we expect > 1 additional yellow edges ($m \geq 2$)

Exploring the Graph

Starting with some edge e, the exploration process is:

1. Find all other edges incident to e, (black edges)
2. Follow all new-found edges to their other endpoint (blue edges)
3. Explore components of vertices to be merged with current component (yellow edges)

Every time we follow a yellow edge, we expect > 1 additional yellow edges ($m \geq 2$)

This can be turned into a supercritical Galton-Watson process, and the giant component follows in a standard way

Online to Offline

Generating the tree $T(v)$ online was enough to find the degree of a vertex (only need to go down the tree)

Online to Offline

Generating the tree $T(v)$ online was enough to find the degree of a vertex (only need to go down the tree)

To explore the tree $T(v)$ from any starting point, we need an offline generation scheme (need to go up the tree)

Online to Offline

Generating the tree $T(v)$ online was enough to find the degree of a vertex (only need to go down the tree)

To explore the tree $T(v)$ from any starting point, we need an offline generation scheme (need to go up the tree)

Conditioning on the sequence $\sigma \in \{\pm\}^n$ of additions/removals allows us to do this

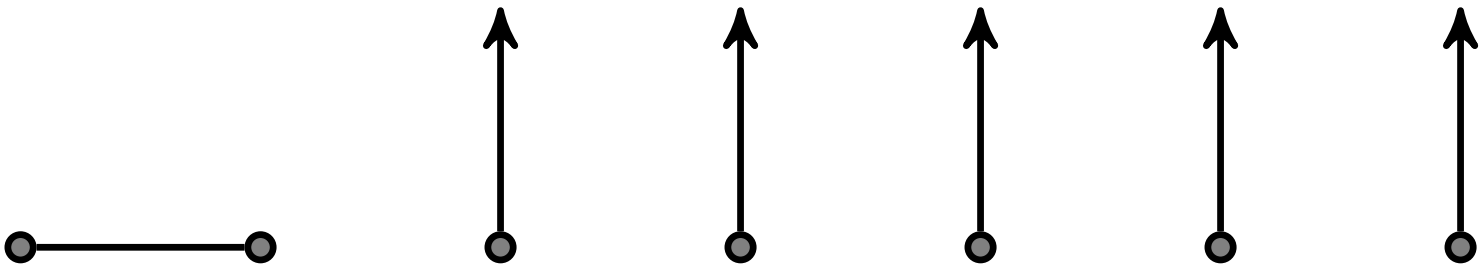
Online to Offline

Under σ, when an edge is to be assigned it knows exactly which edges it may choose – even if those edges are not yet assigned.

Online to Offline

Under σ, when an edge is to be assigned it knows exactly which edges it may choose – even if those edges are not yet assigned.

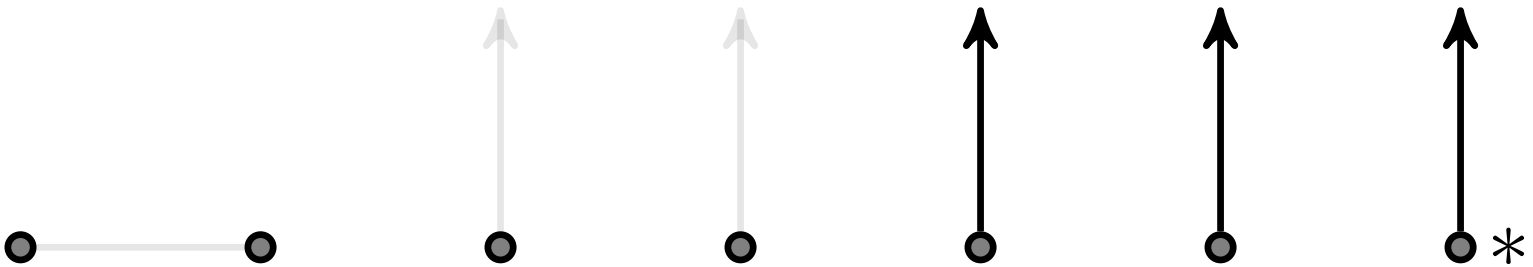
Allow edges to commit to the random endpoint of other edges.



Online to Offline

Under σ, when an edge is to be assigned it knows exactly which edges it may choose – even if those edges are not yet assigned.

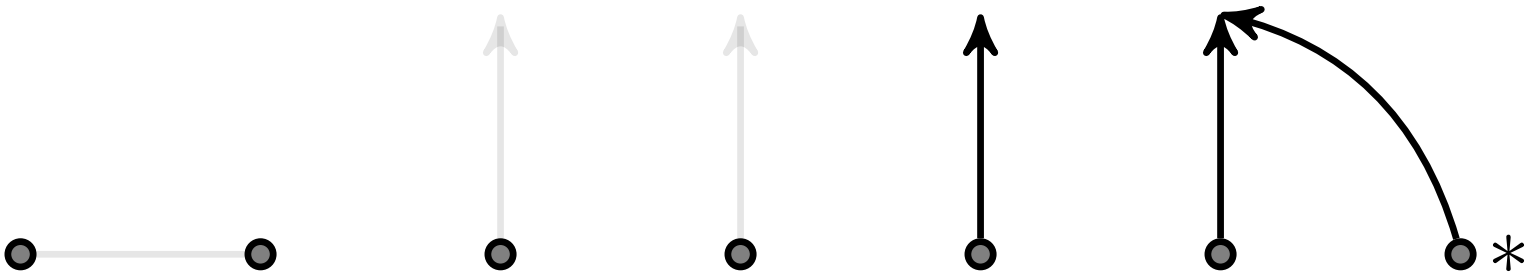
Allow edges to commit to the random endpoint of other edges.



Online to Offline

Under σ, when an edge is to be assigned it knows exactly which edges it may choose – even if those edges are not yet assigned.

Allow edges to commit to the random endpoint of other edges.

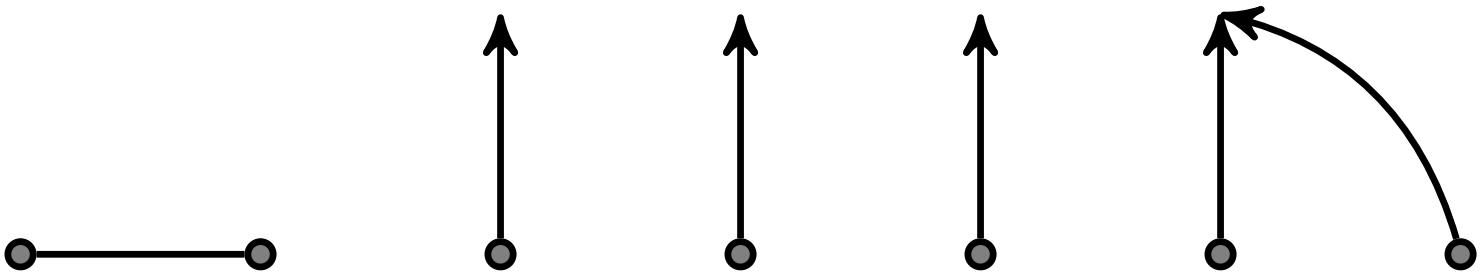


e_6 chooses $(e_5, 1)$

Online to Offline

Under σ, when an edge is to be assigned it knows exactly which edges it may choose – even if those edges are not yet assigned.

Allow edges to commit to the random endpoint of other edges.

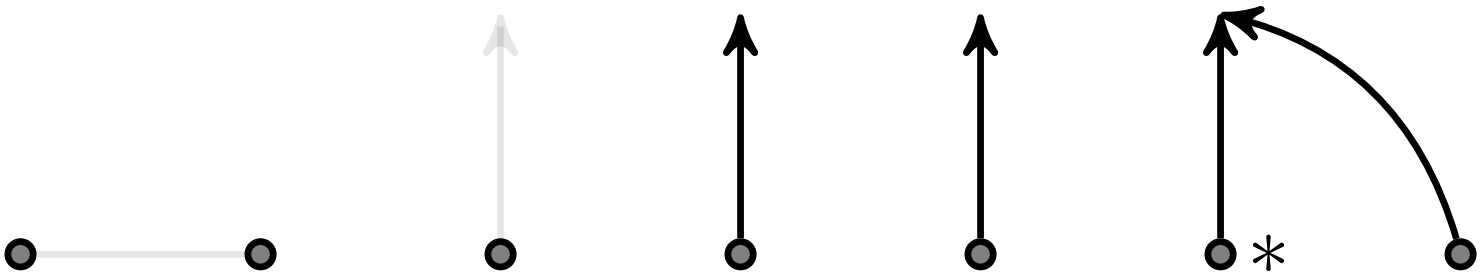


e_6 chooses $(e_5, 1)$

Online to Offline

Under σ, when an edge is to be assigned it knows exactly which edges it may choose – even if those edges are not yet assigned.

Allow edges to commit to the random endpoint of other edges.

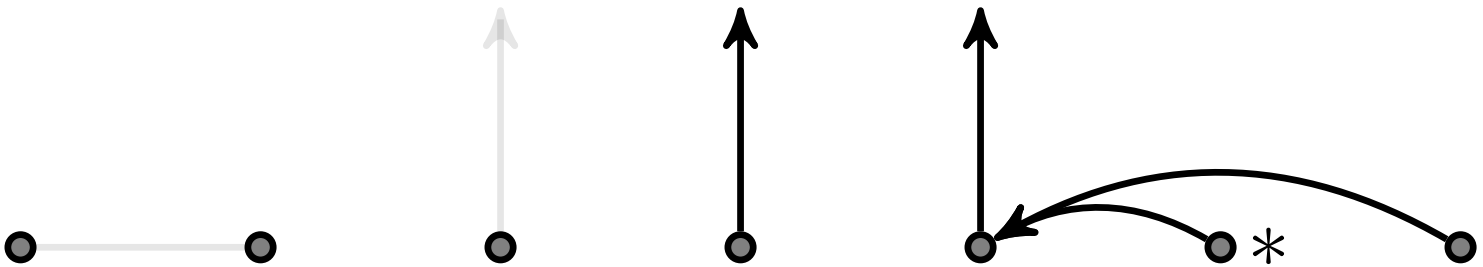


e_6 chooses $(e_5, 1)$

Online to Offline

Under σ, when an edge is to be assigned it knows exactly which edges it may choose – even if those edges are not yet assigned.

Allow edges to commit to the random endpoint of other edges.

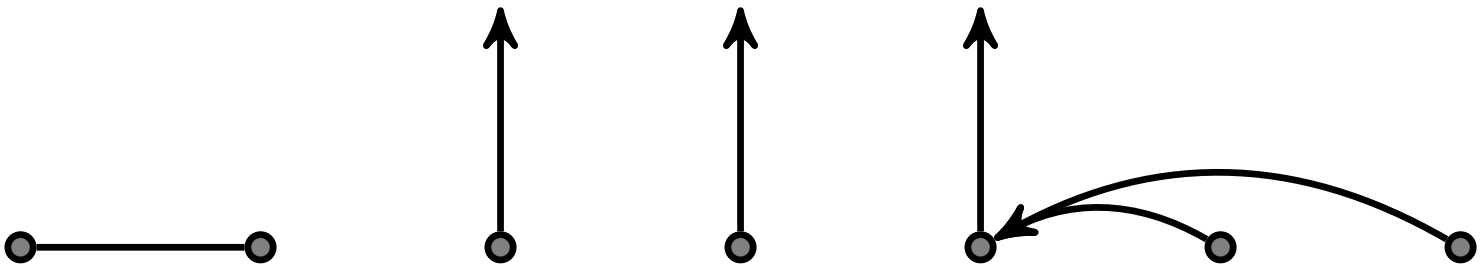


e_6 chooses $(e_5, 1)$, e_5 chooses $(e_4, 2)$.

Online to Offline

Under σ, when an edge is to be assigned it knows exactly which edges it may choose – even if those edges are not yet assigned.

Allow edges to commit to the random endpoint of other edges.



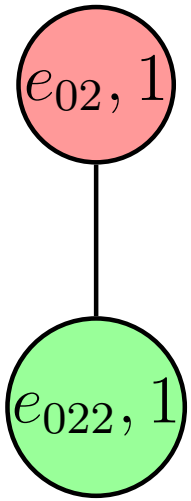
e_6 chooses $(e_5, 1)$, e_5 chooses $(e_4, 2)$.

Online to Offline

Offline generation allows us to find the tree of any pair (e, j)

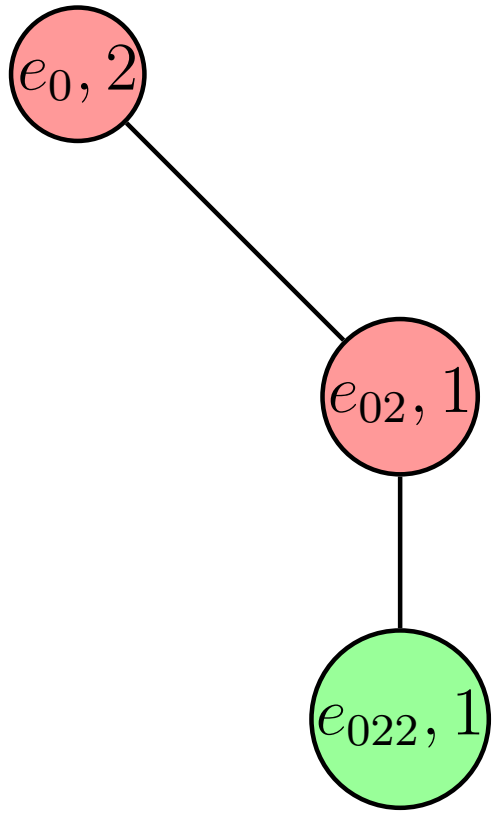
Online to Offline

Offline generation allows us to find the tree of any pair (e, j)



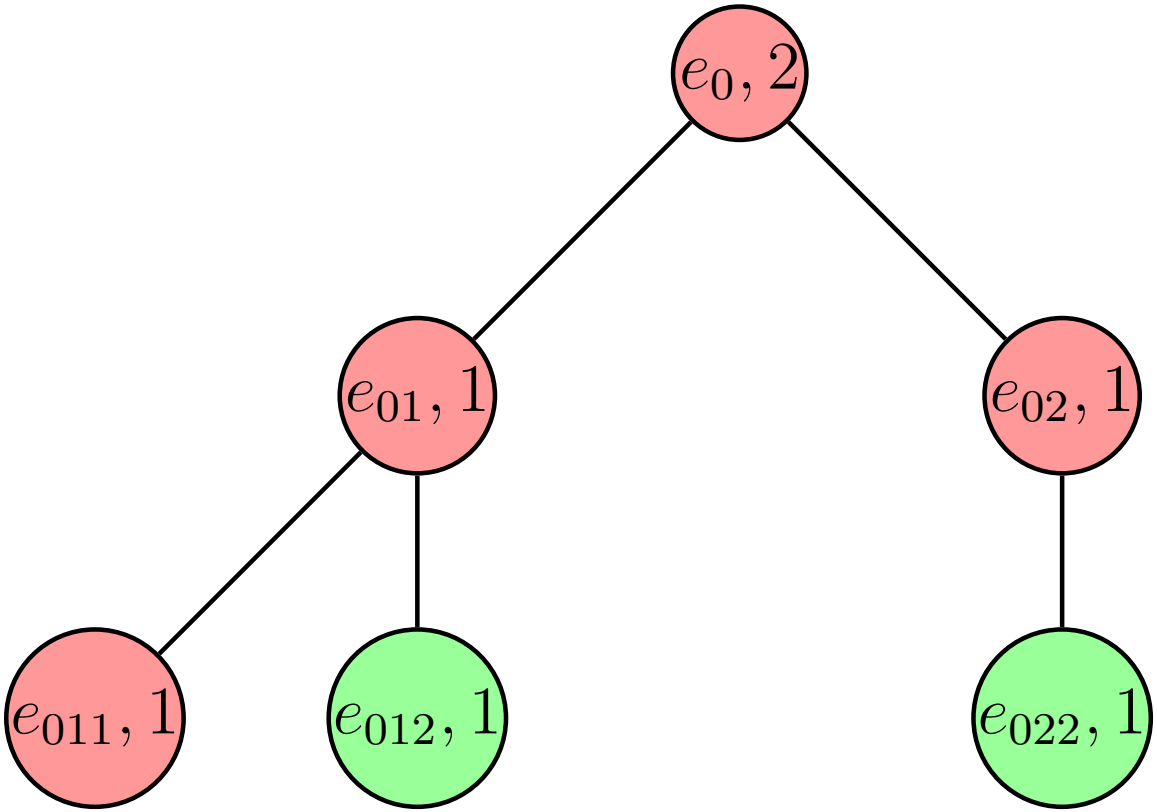
Online to Offline

Offline generation allows us to find the tree of any pair (e, j)



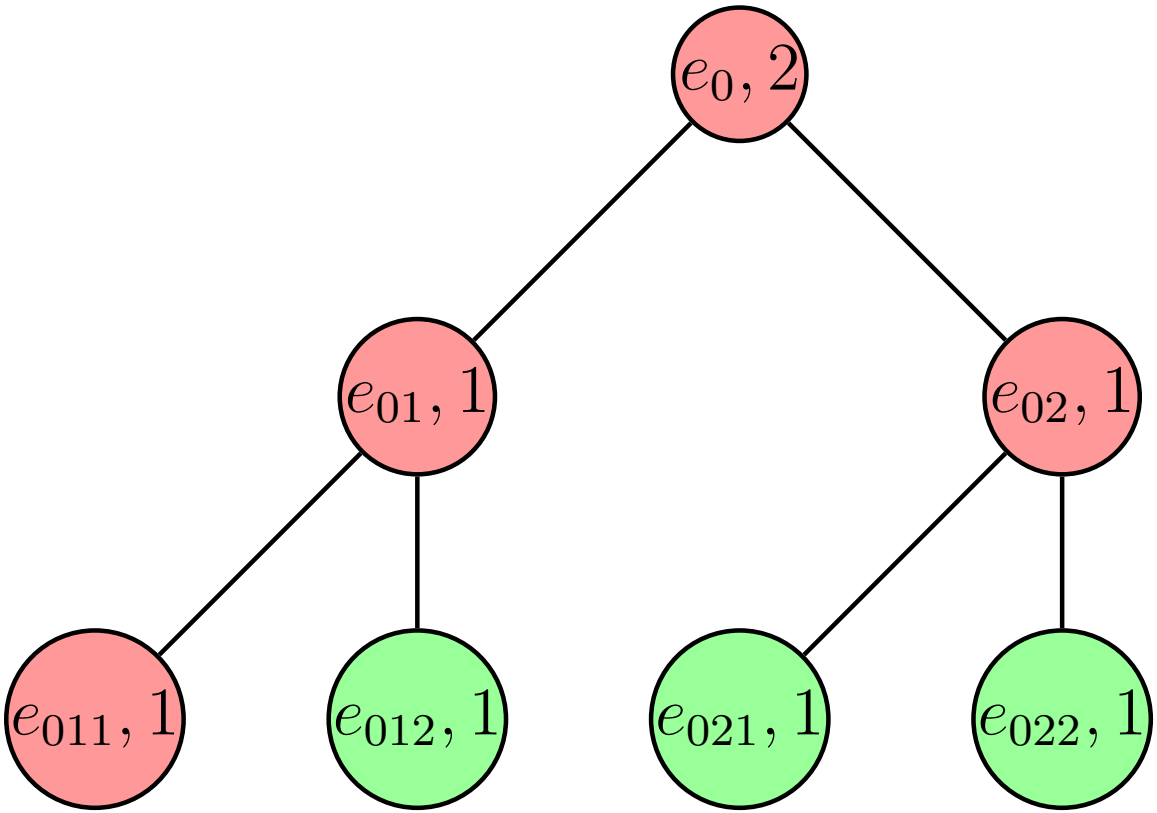
Online to Offline

Offline generation allows us to find the tree of any pair (e, j)



Online to Offline

Offline generation allows us to find the tree of any pair (e, j)



Future Questions

- Keep exploring this graph. What is the diameter?
- What is the exact size of the giant component? Tree sizes depend on the age of the root edge, making the problem difficult.
- Consider deleting the oldest vertex rather than the oldest edges.
- Uniform attachment graphs: new vertices choose old vertices uniformly at random.

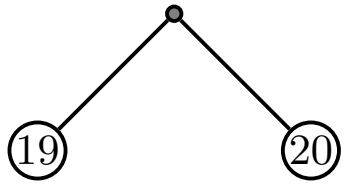
Thank You!

Thank you!

Giant Component

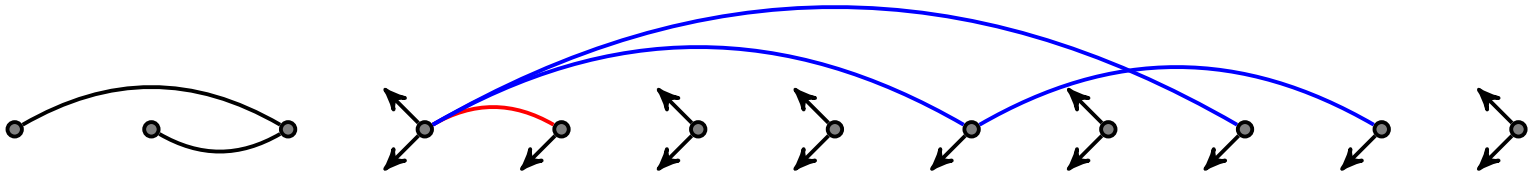
In Γ, this algorithm looks like this.

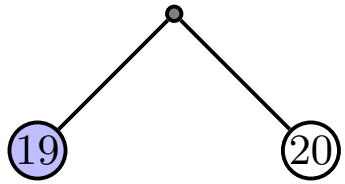
$S = \{19, 20\}$ 1_n v ν_n



Giant Component

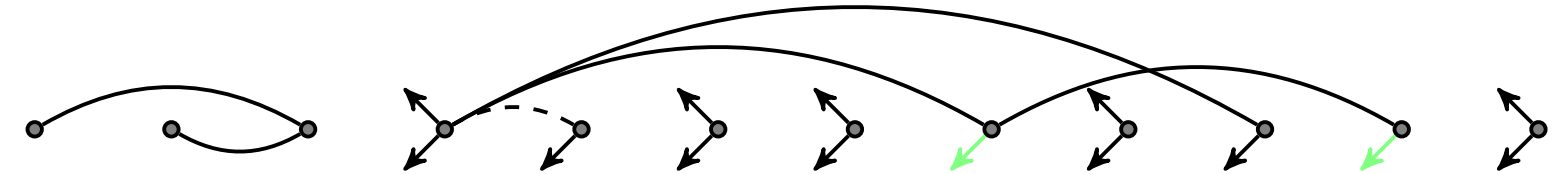
In Γ, this algorithm looks like this.

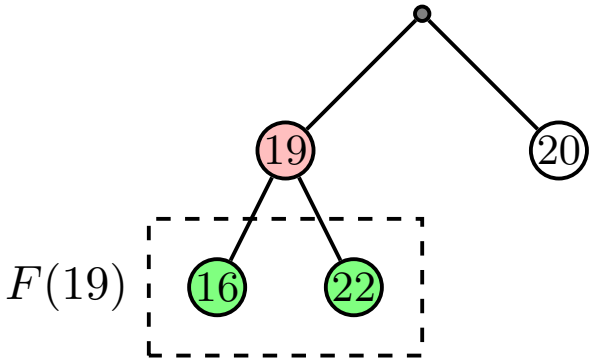




Giant Component

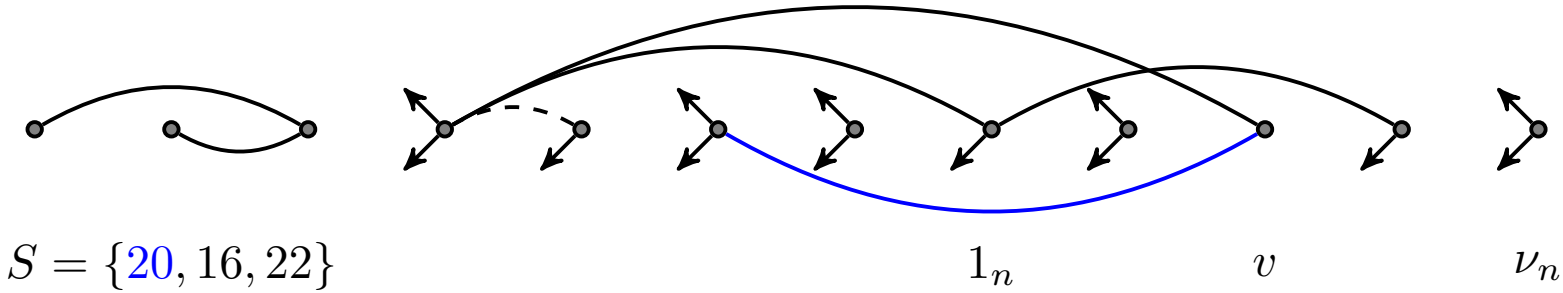
In Γ, this algorithm looks like this.

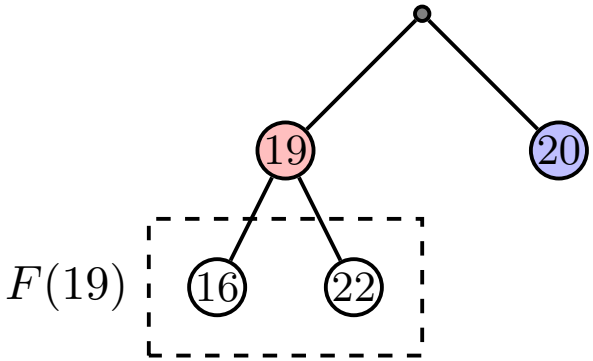




Giant Component

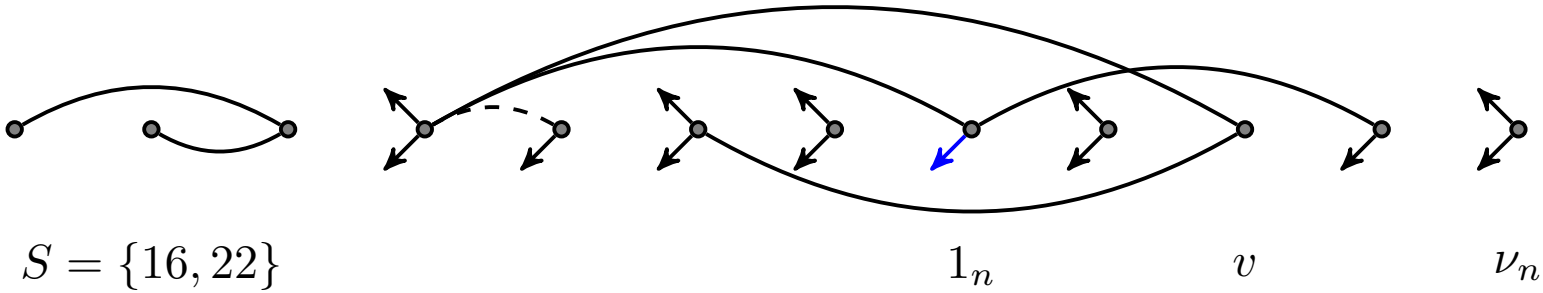
In Γ, this algorithm looks like this.

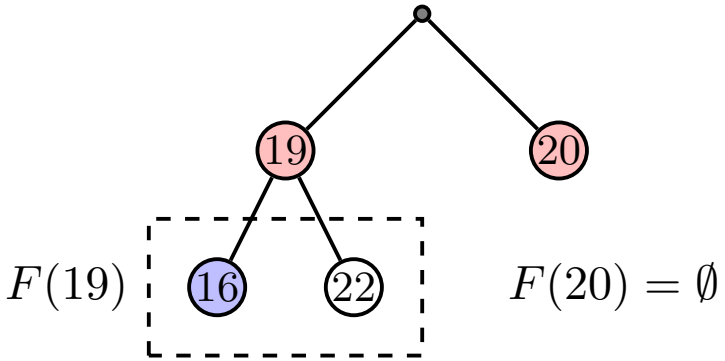




Giant Component

In Γ, this algorithm looks like this.





Giant Component

The size of the random tree can be bounded below by a Galton-Watson branching process.

Lemma

Let $m > 1$ and suppose $O(n^{1/2+\varepsilon})$ edges have been exposed. For any edge e, the size of $F(e)$ is stochastically bounded below by a random variable Z with distribution

$$Pr\{Z = k\} = \begin{cases} 0.26, & k = 0, \\ 0.46, & k = 1, \\ 0.28, & k = 2. \end{cases}$$

Giant Component

The size of the random tree can be bounded below by a Galton-Watson branching process.

Lemma

Let $m > 1$ and suppose $O(n^{1/2+\varepsilon})$ edges have been exposed. For any edge e, the size of $F(e)$ is stochastically bounded below by a random variable Z with distribution

$$Pr\{Z = k\} = \begin{cases} 0.26, & k = 0, \\ 0.46, & k = 1, \\ 0.28, & k = 2. \end{cases}$$

Since $\mathbf{E}[Z] > 1$, this implies that the tree reaches size $\Omega(n^{1/2+\varepsilon})$ with some constant probability.

Giant Component

Proof now follows a standard template.

1. Number of large vertices: There are $\Omega(n)$ vertices in components of size $\Omega(n^{1/2+\varepsilon})$ (call them large), since the branching process survives with constant probability.
2. Unique giant: Whp there are no two disjoint large components.
3. Small components: Whp no tree terminates with size in range $[C \log n, n^{1/2+\varepsilon}]$.

Giant Component

Proof now follows a standard template.

1. Number of large vertices: There are $\Omega(n)$ vertices in components of size $\Omega(n^{1/2+\varepsilon})$ (call them large), since the branching process survives with constant probability.
2. Unique giant: Whp there are no two disjoint large components.
3. Small components: Whp no tree terminates with size in range $[C \log n, n^{1/2+\varepsilon}]$.

This shows that there is one unique component of size $\Omega(n)$ while all other components have size $O(\log n)$.