Local Limit of the Random Degree Constrained Process

Márton Szőke

Department of Stochastics, Budapest University of Technology and Economics

February 28, 2025

ELTE,
Probability Theory and Statistics Seminar

Joint work with Balázs Ráth and Lutz Warnke

1. Local limit of the RDCP
2. Critical time

Random Degree Constrained Process

RDCP on the complete graph:

Random Degree Constrained Process

RDCP on the complete graph:

- Let $\underline{p} = (p_k)_{k=2}^{\infty}$ be a distribution on $\{2, 3, \dots\}$

Random Degree Constrained Process

RDCP on the complete graph:

- Let $\underline{p} = (p_k)_{k=2}^{\infty}$ be a distribution on $\{2, 3, \dots\}$
- RDCP is a random process $\left(G_{\underline{p}}^{n}(t)\right)_{t \geqslant 0}$ on n vertices where t is a **continuous time** parameter

Random Degree Constrained Process

RDCP on the complete graph:

- Let $\underline{p} = (p_k)_{k=2}^{\infty}$ be a distribution on $\{2, 3, \dots\}$
- RDCP is a random process $\left(G^n_{\underline{p}}(t)\right)_{t \geqslant 0}$ on n vertices where t is a **continuous time** parameter
- $G^n_{\underline{p}}(0)$ is the empty graph on n vertices

Random Degree Constrained Process

RDCP on the complete graph:

- Let $\underline{p} = (p_k)_{k=2}^{\infty}$ be a distribution on $\{2, 3, \dots\}$
- RDCP is a random process $\left(G_{\underline{p}}^n(t)\right)_{t \geqslant 0}$ on n vertices where t is a **continuous time** parameter
- $G_{\underline{p}}^n(0)$ is the empty graph on n vertices
- Vertices have i.i.d. **random degree constraints** $d(v) \sim \underline{p}$, $v \in [n]$

Random Degree Constrained Process

RDCP on the complete graph:

- Let $\underline{p} = (p_k)_{k=2}^{\infty}$ be a distribution on $\{2, 3, \dots\}$
- RDCP is a random process $\left(G_{\underline{p}}^{n}(t)\right)_{t \geqslant 0}$ on n vertices where t is a **continuous time** parameter
- $G_{\underline{p}}^{n}(0)$ is the empty graph on n vertices
- Vertices have i.i.d. **random degree constraints** $d(v) \sim \underline{p}$, $v \in [n]$
- Each edge e has an **activation time** $X_e \sim \text{EXP}\left(\frac{1}{n}\right)$

Random Degree Constrained Process

RDCP on the complete graph:

- Let $\underline{p} = (p_k)_{k=2}^{\infty}$ be a distribution on $\{2, 3, \dots\}$
- RDCP is a random process $\left(G_{\underline{p}}^n(t)\right)_{t \geqslant 0}$ on n vertices where t is a **continuous time** parameter
- $G_{\underline{p}}^n(0)$ is the empty graph on n vertices
- Vertices have i.i.d. **random degree constraints** $d(v) \sim \underline{p}$, $v \in [n]$
- Each edge e has an **activation time** $X_e \sim \mathrm{EXP}\left(\frac{1}{n}\right)$
- $d(v)$'s and activation times are independent of each other

Random Degree Constrained Process

RDCP on the complete graph:

- Let $\underline{p} = (p_k)_{k=2}^{\infty}$ be a distribution on $\{2, 3, \dots\}$
- RDCP is a random process $\left(G_{\underline{p}}^n(t)\right)_{t \geqslant 0}$ on n vertices where t is a **continuous time** parameter
- $G_{\underline{p}}^n(0)$ is the empty graph on n vertices
- Vertices have i.i.d. **random degree constraints** $d(v) \sim \underline{p}$, $v \in [n]$
- Each edge e has an **activation time** $X_e \sim \text{EXP}\left(\frac{1}{n}\right)$
- $d(v)$'s and activation times are independent of each other
- Edge $e = \{v, w\}$ is added at time $t = X_e$ if both v, w have degree strictly less than their own degree constraint in $G_{\underline{p}}^n(t)$

Random Degree Constrained Process

RDCP on the complete graph:

- Let $\underline{p} = (p_k)_{k=2}^{\infty}$ be a distribution on $\{2, 3, \dots\}$
- RDCP is a random process $\left(G_{\underline{p}}^{n}(t)\right)_{t \geqslant 0}$ on n vertices where t is a **continuous time** parameter
- $G_{\underline{p}}^{n}(0)$ is the empty graph on n vertices
- Vertices have i.i.d. **random degree constraints** $d(v) \sim \underline{p}$, $v \in [n]$
- Each edge e has an **activation time** $X_e \sim \mathrm{EXP}\left(\frac{1}{n}\right)$
- $d(v)$'s and activation times are independent of each other
- Edge $e = \{v, w\}$ is added at time $t = X_e$ if both v, w have degree strictly less than their own degree constraint in $G_{\underline{p}}^{n}(t)$
  If e.g. v already has degree $d(v)$ at time X_e then e will never be added

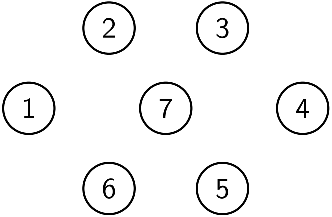
Random Degree Constrained Process

RDCP on the complete graph:

- Let $\underline{p} = (p_k)_{k=2}^{\infty}$ be a distribution on $\{2, 3, \dots\}$
- RDCP is a random process $\left(G_{\underline{p}}^n(t)\right)_{t \geqslant 0}$ on n vertices where t is a **continuous time** parameter
- $G_{\underline{p}}^n(0)$ is the empty graph on n vertices
- Vertices have i.i.d. **random degree constraints** $d(v) \sim \underline{p}$, $v \in [n]$
- Each edge e has an **activation time** $X_e \sim \mathrm{EXP}\left(\frac{1}{n}\right)$
- $d(v)$'s and activation times are independent of each other
- Edge $e = \{v, w\}$ is added at time $t = X_e$ if both v, w have degree strictly less than their own degree constraint in $G_{\underline{p}}^n(t)$
  If e.g. v already has degree $d(v)$ at time X_e then e will never be added
- Vertex v is **saturated** at time t if its degree in $G_{\underline{p}}^n(t)$ is equal to $d(v)$

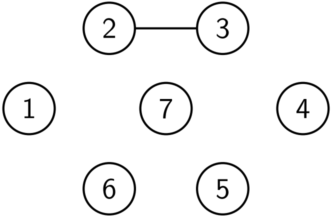
An example for RDCP

RDCP on K_7 with degree constraint 3 for each vertex:



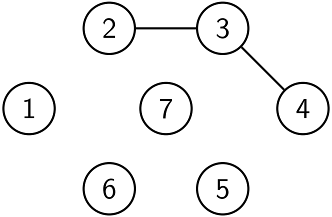
An example for RDCP

RDCP on K_7 with degree constraint 3 for each vertex:



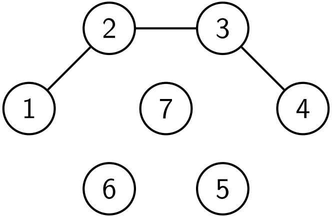
An example for RDCP

RDCP on K_7 with degree constraint 3 for each vertex:



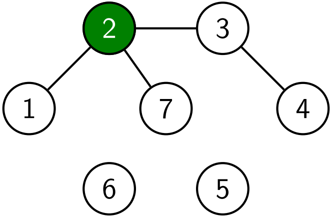
An example for RDCP

RDCP on K_7 with degree constraint 3 for each vertex:



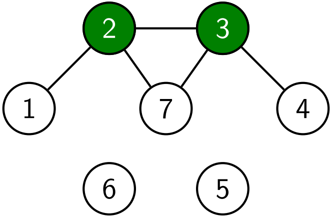
An example for RDCP

RDCP on K_7 with degree constraint 3 for each vertex:



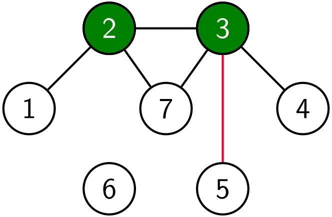
An example for RDCP

RDCP on K_7 with degree constraint 3 for each vertex:



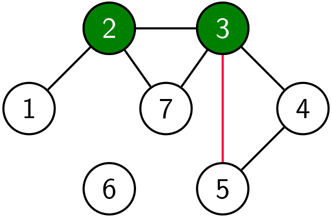
An example for RDCP

RDCP on K_7 with degree constraint 3 for each vertex:



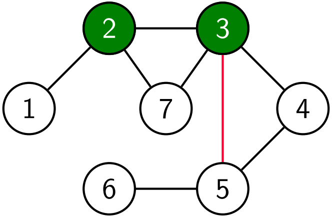
An example for RDCP

RDCP on K_7 with degree constraint 3 for each vertex:



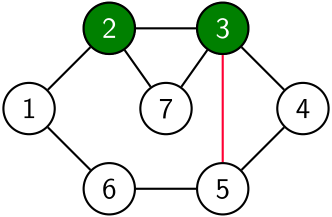
An example for RDCP

RDCP on K_7 with degree constraint 3 for each vertex:



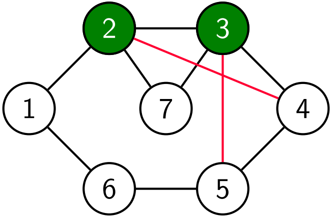
An example for RDCP

RDCP on K_7 with degree constraint 3 for each vertex:



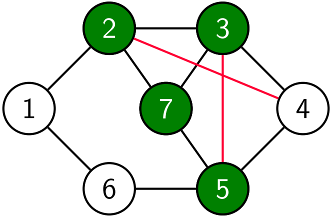
An example for RDCP

RDCP on K_7 with degree constraint 3 for each vertex:



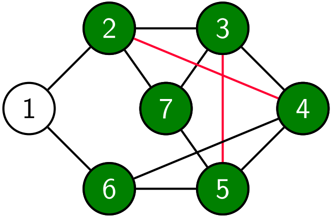
An example for RDCP

RDCP on K_7 with degree constraint 3 for each vertex:



An example for RDCP

RDCP on K_7 with degree constraint 3 for each vertex:



Configuration model

Benchmark model

- Random graph on n vertices with given degrees $\underline{d} = (d(i))_{i=1}^{n}$

Configuration model

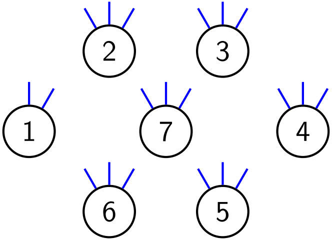
Benchmark model

- Random graph on n vertices with given degrees $\underline{d} = (d(i))_{i=1}^{n}$
- Construction: uniform matching of half-edges attached to the vertices

Configuration model

Benchmark model

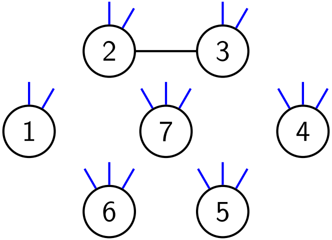
- Random graph on n vertices with given degrees $\underline{d} = (d(i))_{i=1}^{n}$
- Construction: uniform matching of half-edges attached to the vertices



Configuration model

Benchmark model

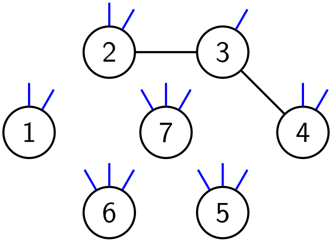
- Random graph on n vertices with given degrees $\underline{d} = (d(i))_{i=1}^{n}$
- Construction: uniform matching of half-edges attached to the vertices



Configuration model

Benchmark model

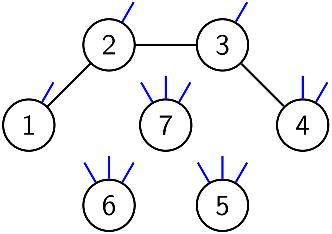
- Random graph on n vertices with given degrees $\underline{d} = (d(i))_{i=1}^n$
- Construction: uniform matching of half-edges attached to the vertices



Configuration model

Benchmark model

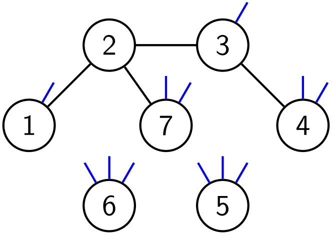
- Random graph on n vertices with given degrees $\underline{d} = (d(i))_{i=1}^{n}$
- Construction: uniform matching of half-edges attached to the vertices



Configuration model

Benchmark model

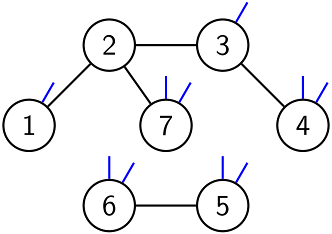
- Random graph on n vertices with given degrees $\underline{d} = (d(i))_{i=1}^{n}$
- Construction: uniform matching of half-edges attached to the vertices



Configuration model

Benchmark model

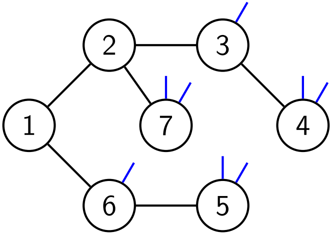
- Random graph on n vertices with given degrees $\underline{d} = (d(i))_{i=1}^{n}$
- Construction: uniform matching of half-edges attached to the vertices



Configuration model

Benchmark model

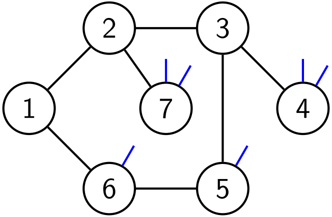
- Random graph on n vertices with given degrees $\underline{d} = (d(i))_{i=1}^{n}$
- Construction: uniform matching of half-edges attached to the vertices



Configuration model

Benchmark model

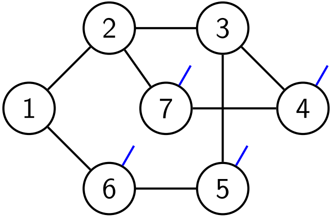
- Random graph on n vertices with given degrees $\underline{d} = (d(i))_{i=1}^n$
- Construction: uniform matching of half-edges attached to the vertices



Configuration model

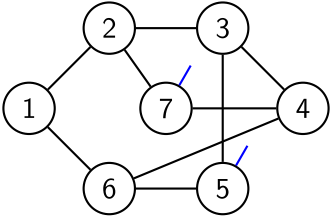
Benchmark model

- Random graph on n vertices with given degrees $\underline{d} = (d(i))_{i=1}^{n}$
- Construction: uniform matching of half-edges attached to the vertices



Configuration model
Benchmark model

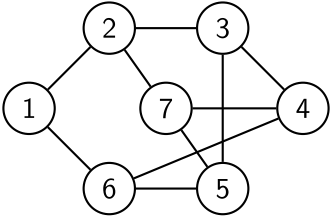
- Random graph on n vertices with given degrees $\underline{d} = (d(i))_{i=1}^{n}$
- Construction: uniform matching of half-edges attached to the vertices



Configuration model

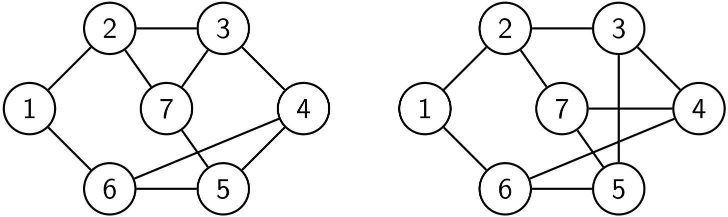
Benchmark model

- Random graph on n vertices with given degrees $\underline{d} = (d(i))_{i=1}^{n}$
- Construction: uniform matching of half-edges attached to the vertices



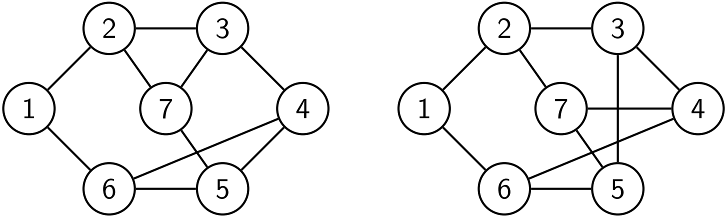
Comparison of the two models

- RDCP and configuration model both generate a random graph for a given degree sequence



Comparison of the two models

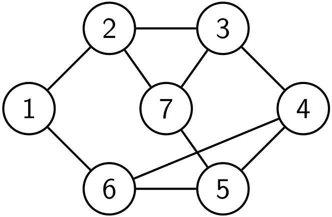
- RDCP and configuration model both generate a random graph for a given degree sequence

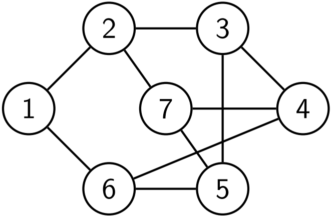


- Question: Is RDCP at time t a configuration model?

Comparison of the two models

- RDCP and configuration model both generate a random graph for a given degree sequence





- Question: Is RDCP at time t a configuration model? No!

Theorem (RDCP is far from uniform (Molloy, Surya, Warnke, 2022+))

If the degree constraints are not nearly regular then the final graph of the RDCP is not contiguous to the configuration model with the same degree sequence.

Local weak limit

Definition (Local weak convergence)

- $\{ \mathcal{G}^n \}$: a sequence of random graphs

Local weak limit

Definition (Local weak convergence)

- $\{\mathcal{G}^n\}$: a sequence of random graphs
- $\mathcal{B}^{n,R}(w)$: the R-neighbourhood of a vertex w in $\mathcal{G}^n$ with respect to graph distance, viewed as a graph rooted at w

Local weak limit

Definition (Local weak convergence)

- $\{ \mathcal{G}^n \}$: a sequence of random graphs
- $\mathcal{B}^{n,R}(w)$: the R-neighbourhood of a vertex w in $\mathcal{G}^n$ with respect to graph distance, viewed as a graph rooted at w
- u^n: a vertex that is independent of $\mathcal{G}^n$ and uniformly distributed in $V(\mathcal{G}^n)$

Local weak limit

Definition (Local weak convergence)

- $\{ \mathcal{G}^n \}$: a sequence of random graphs
- $\mathcal{B}^{n,R}(w)$: the R-neighbourhood of a vertex w in $\mathcal{G}^n$ with respect to graph distance, viewed as a graph rooted at w
- u^n: a vertex that is independent of $\mathcal{G}^n$ and uniformly distributed in $V(\mathcal{G}^n)$

$\mathcal{G}^n$ **converges in distribution in the local weak sense** to the random rooted graph $(\mathcal{G}, u)$, if for every rooted graph (H, v) and every integer $R \geqslant 1$, we have

$$\mathbb{P}\left(\mathcal{B}^{n,R}(u^n) \simeq (H, v)\right) \longrightarrow \mathbb{P}\left(\mathcal{B}^{R}(u) \simeq (H, v)\right) \text{ as } n \to \infty, \tag{1}$$

Local weak limit

Definition (Local weak convergence)

- $\{\mathcal{G}^n\}$: a sequence of random graphs
- $\mathcal{B}^{n,R}(w)$: the R-neighbourhood of a vertex w in $\mathcal{G}^n$ with respect to graph distance, viewed as a graph rooted at w
- u^n: a vertex that is independent of $\mathcal{G}^n$ and uniformly distributed in $V(\mathcal{G}^n)$

$\mathcal{G}^n$ **converges in distribution in the local weak sense** to the random rooted graph $(\mathcal{G}, u)$, if for every rooted graph (H, v) and every integer $R \geqslant 1$, we have

$$\mathbb{P}\left(\mathcal{B}^{n,R}(u^n) \simeq (H, v)\right) \longrightarrow \mathbb{P}\left(\mathcal{B}^{R}(u) \simeq (H, v)\right) \text{ as } n \to \infty, \qquad (1)$$

where $\mathcal{B}^R(u)$ is the R-neighbourhood of the root u in $\mathcal{G}$.

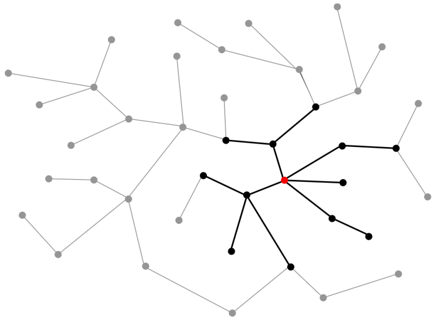
Local weak limit

Advantages of local weak convergence:

Local weak limit

Advantages of local weak convergence:

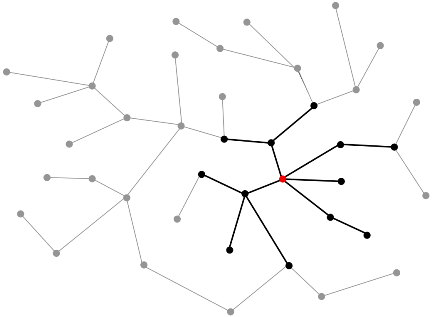
- It describes the limiting local structure of the neighbourhood of a typical vertex in large graphs



Local weak limit

Advantages of local weak convergence:

- It describes the limiting local structure of the neighbourhood of a typical vertex in large graphs

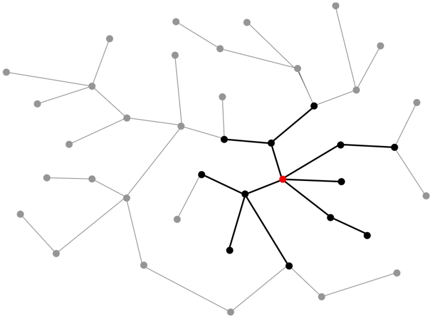


- It can be used to investigate numerous graph properties, parameters

Local weak limit

Advantages of local weak convergence:

- It describes the limiting local structure of the neighbourhood of a typical vertex in large graphs



- It can be used to investigate numerous graph properties, parameters
- The limiting objects (often branching processes) are much simpler to study than the original model

Local weak limit of the configuration model

Benchmark model

- **Classical** result

Local weak limit of the configuration model

Benchmark model

- **Classical** result
- Under some regularity conditions, the configuration model converges locally to a Galton–Watson (a.k.a. Bienaymé–Galton–Watson) tree

Local weak limit of the configuration model

Benchmark model

- **Classical** result
- Under some regularity conditions, the configuration model converges locally to a Galton–Watson (a.k.a. Bienaymé–Galton–Watson) tree
- The distribution of the number of the children of the **root** is the same as the given degree distribution $\underline{p} = (p_k)_{k=1}^{\infty}$

Local weak limit of the configuration model

Benchmark model

- **Classical** result
- Under some regularity conditions, the configuration model converges locally to a Galton–Watson (a.k.a. Bienaymé–Galton–Watson) tree
- The distribution of the number of the children of the **root** is the same as the given degree distribution $\underline{p} = (p_k)_{k=1}^{\infty}$
- The distribution of the number of the children of **other vertices** is the down-shifted size-biased version of $\underline{p}$

Local weak limit of the configuration model

Benchmark model

- **Classical** result
- Under some regularity conditions, the configuration model converges locally to a Galton–Watson (a.k.a. Bienaymé–Galton–Watson) tree
- The distribution of the number of the children of the **root** is the same as the given degree distribution $\underline{p} = (p_k)_{k=1}^{\infty}$
- The distribution of the number of the children of **other vertices** is the down-shifted size-biased version of $\underline{p}$

$$p_k^* = \frac{(k+1)p_{k+1}}{m},$$

Local weak limit of the configuration model

Benchmark model

- **Classical** result
- Under some regularity conditions, the configuration model converges locally to a Galton–Watson (a.k.a. Bienaymé–Galton–Watson) tree
- The distribution of the number of the children of the **root** is the same as the given degree distribution $\underline{p} = (p_k)_{k=1}^{\infty}$
- The distribution of the number of the children of **other vertices** is the down-shifted size-biased version of $\underline{p}$

$$p_k^* = \frac{(k+1)p_{k+1}}{m},$$

where m is the expected value of the degree distribution $\underline{p}$

Local weak limit of the configuration model

Benchmark model

- **Classical** result
- Under some regularity conditions, the configuration model converges locally to a Galton–Watson (a.k.a. Bienaymé–Galton–Watson) tree
- The distribution of the number of the children of the **root** is the same as the given degree distribution $\underline{p} = (p_k)_{k=1}^{\infty}$
- The distribution of the number of the children of **other vertices** is the down-shifted size-biased version of $\underline{p}$

$$p_k^* = \frac{(k+1)p_{k+1}}{m},$$

where m is the expected value of the degree distribution $\underline{p}$

In contrast to the configuration model, the **local weak limit of the RDCP was unknown** before our work

Local weak limit of the RDCP

Theorem (Local weak convergence of RDCP (Ráth, Sz., Warnke, 2024+))

Fix $t \in \mathbb{R}_+ \cup \{\infty\}$. Then the random degree constrained process on K_n at time t converges locally to a random rooted tree $\left(G^{\infty}_{\underline{p}}(t), \varnothing\right)$, i.e.,

$$G^n_{\underline{p}}(t) \longrightarrow \left(G^{\infty}_{\underline{p}}(t), \varnothing\right) \text{ locally as } n \to \infty. \tag{2}$$

Local weak limit of the RDCP

Theorem (Local weak convergence of RDCP (Ráth, Sz., Warnke, 2024+))

Fix $t \in \mathbb{R}_+ \cup \{\infty\}$. Then the random degree constrained process on K_n at time t converges locally to a random rooted tree $\left(G_{\underline{p}}^{\infty}(t), \varnothing\right)$, i.e.,

$$G_{\underline{p}}^{n}(t) \longrightarrow \left(G_{\underline{p}}^{\infty}(t), \varnothing\right) \text{ locally as } n \to \infty. \tag{2}$$

We even have a more general result:

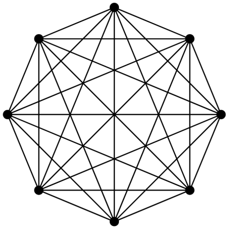
Local weak limit of the RDCP

Theorem (Local weak convergence of RDCP (Ráth, Sz., Warnke, 2024+))

Fix $t \in \mathbb{R}_+ \cup \{\infty\}$. Then the random degree constrained process on K_n at time t converges locally to a random rooted tree $\left(G^{\infty}_{\underline{p}}(t), \varnothing\right)$, i.e.,

$$G^n_{\underline{p}}(t) \longrightarrow \left(G^{\infty}_{\underline{p}}(t), \varnothing\right) \text{ locally as } n \to \infty. \tag{2}$$

We even have a more general result:
Instead of K_n



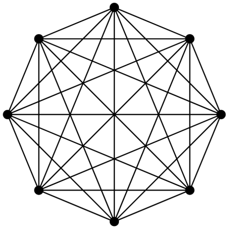
Local weak limit of the RDCP

Theorem (Local weak convergence of RDCP (Ráth, Sz., Warnke, 2024+))

Fix $t \in \mathbb{R}_+ \cup \{\infty\}$. Then the random degree constrained process on K_n at time t converges locally to a random rooted tree $\left(G^{\infty}_{\underline{p}}(t), \varnothing\right)$, i.e.,

$$G^{n}_{\underline{p}}(t) \longrightarrow \left(G^{\infty}_{\underline{p}}(t), \varnothing\right) \text{ locally as } n \to \infty. \tag{2}$$

We even have a more general result:
Instead of K_n, the sequence of host graphs can be any **high degree almost regular** graph sequence



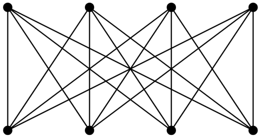
Local weak limit of the RDCP

Theorem (Local weak convergence of RDCP (Ráth, Sz., Warnke, 2024+))

Fix $t \in \mathbb{R}_+ \cup \{\infty\}$. Then the random degree constrained process on K_n at time t converges locally to a random rooted tree $\left(G^{\infty}_{\underline{p}}(t), \varnothing\right)$, i.e.,

$$G^{n}_{\underline{p}}(t) \longrightarrow \left(G^{\infty}_{\underline{p}}(t), \varnothing\right) \text{ locally as } n \to \infty. \tag{2}$$

We even have a more general result:
Instead of K_n, the sequence of host graphs can be any **high degree almost regular** graph sequence
E.g. regular complete bipartite graphs $K_{n,n}$



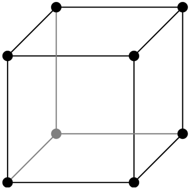
Local weak limit of the RDCP

Theorem (Local weak convergence of RDCP (Ráth, Sz., Warnke, 2024+))

Fix $t \in \mathbb{R}_+ \cup \{\infty\}$. Then the random degree constrained process on K_n at time t converges locally to a random rooted tree $\left(G^{\infty}_{\underline{p}}(t), \varnothing\right)$, i.e.,

$$G^n_{\underline{p}}(t) \longrightarrow \left(G^{\infty}_{\underline{p}}(t), \varnothing\right) \text{ locally as } n \to \infty. \tag{2}$$

We even have a more general result:
Instead of K_n, the sequence of host graphs can be any **high degree almost regular** graph sequence
E.g. n-dimensional hypercube



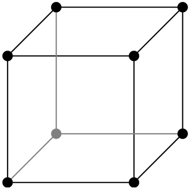
Local weak limit of the RDCP

Theorem (Local weak convergence of RDCP (Ráth, Sz., Warnke, 2024+))

Fix $t \in \mathbb{R}_+ \cup \{\infty\}$. Then the random degree constrained process on K_n at time t converges locally to a random rooted tree $\left(G^\infty_{\underline{p}}(t), \varnothing\right)$, i.e.,

$$G^n_{\underline{p}}(t) \longrightarrow \left(G^\infty_{\underline{p}}(t), \varnothing\right) \text{ locally as } n \to \infty. \tag{2}$$

We even have a more general result:
Instead of K_n, the sequence of host graphs can be any **high degree almost regular** graph sequence
E.g. n-dimensional hypercube



Question: How can we describe the local weak limit $G^\infty_{\underline{p}}(t)$?

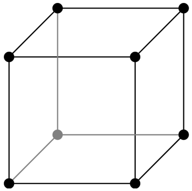
Local weak limit of the RDCP

Theorem (Local weak convergence of RDCP (Ráth, Sz., Warnke, 2024+))

Fix $t \in \mathbb{R}_+ \cup \{\infty\}$. Then the random degree constrained process on K_n at time t converges locally to a random rooted tree $\left(G^{\infty}_{\underline{p}}(t), \varnothing\right)$, i.e.,

$$G^{n}_{\underline{p}}(t) \longrightarrow \left(G^{\infty}_{\underline{p}}(t), \varnothing\right) \text{ locally as } n \to \infty. \quad (2)$$

We even have a more general result:
Instead of K_n, the sequence of host graphs can be any **high degree almost regular** graph sequence
E.g. n-dimensional hypercube



Question: How can we describe the local weak limit $G^{\infty}_{\underline{p}}(t)$?
Is it a Galton–Watson tree?

Poisson weighted infinite tree

If we decorate K_n with i.i.d. exponentially distributed edge weights:

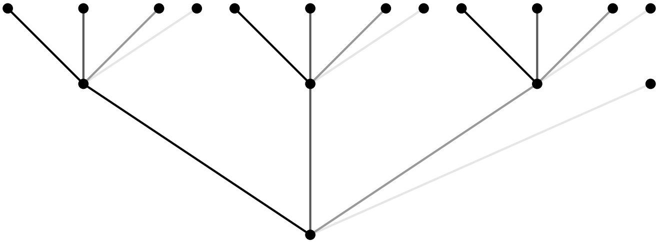
Poisson weighted infinite tree

If we decorate K_n with i.i.d. exponentially distributed edge weights:
Its local weak limit is the **Poisson weighted infinite tree** (PWIT)

Poisson weighted infinite tree

If we decorate K_n with i.i.d. exponentially distributed edge weights:
Its local weak limit is the **Poisson weighted infinite tree** (PWIT)

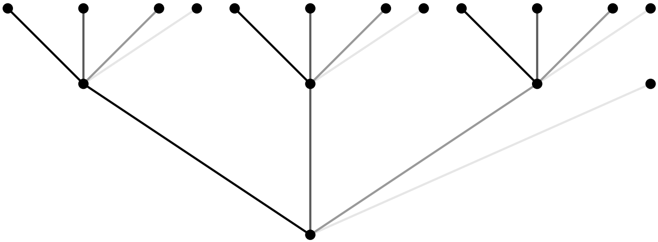
- PWIT is an infinite tree with random edge labels



Poisson weighted infinite tree

If we decorate K_n with i.i.d. exponentially distributed edge weights:
Its local weak limit is the **Poisson weighted infinite tree** (PWIT)

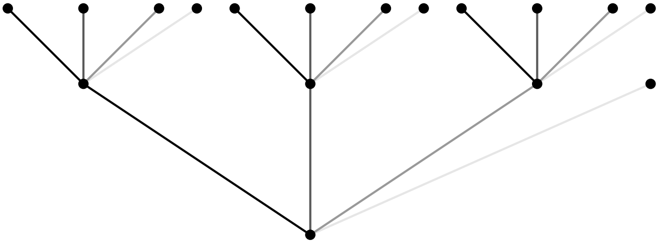
- PWIT is an infinite tree with random edge labels
- Each vertex has infinitely many children



Poisson weighted infinite tree

If we decorate K_n with i.i.d. exponentially distributed edge weights:
Its local weak limit is the **Poisson weighted infinite tree** (PWIT)

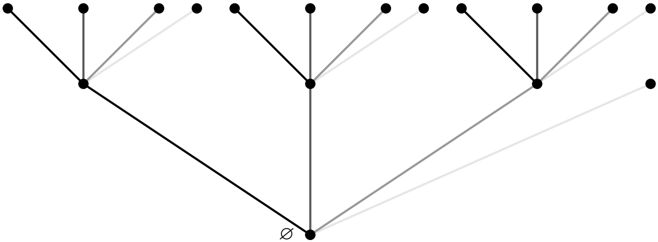
- PWIT is an infinite tree with random edge labels
- Each vertex has infinitely many children
- The vertices are denoted by finite strings of positive integers



Poisson weighted infinite tree

If we decorate K_n with i.i.d. exponentially distributed edge weights:
Its local weak limit is the **Poisson weighted infinite tree** (PWIT)

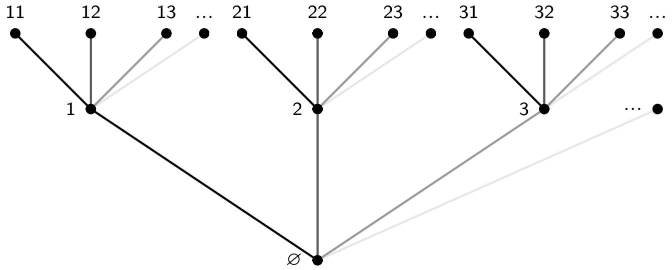
- PWIT is an infinite tree with random edge labels
- Each vertex has infinitely many children
- The vertices are denoted by finite strings of positive integers
    - The root of the tree is the empty string $\varnothing$



Poisson weighted infinite tree

If we decorate K_n with i.i.d. exponentially distributed edge weights:
Its local weak limit is the **Poisson weighted infinite tree** (PWIT)

- PWIT is an infinite tree with random edge labels
- Each vertex has infinitely many children
- The vertices are denoted by finite strings of positive integers
  - The root of the tree is the empty string $\varnothing$
  - If $\mathbf{i}$ is such a string and $j \in \mathbb{N}_+$ then the vertex $\mathbf{i}j$ is a child of vertex $\mathbf{i}$

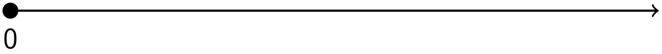


Poisson weighted infinite tree: edge labels

- For each vertex $\mathbf{i}$, take a unit intensity Poisson point process (PPP) $(\tau^{\mathbf{i}j})_{j=1}^{\infty}$ on $[0, \infty)$

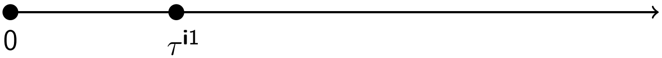
Poisson weighted infinite tree: edge labels

- For each vertex $\mathbf{i}$, take a unit intensity Poisson point process (PPP) $(\tau^{\mathbf{i}j})_{j=1}^{\infty}$ on $[0,\infty)$:



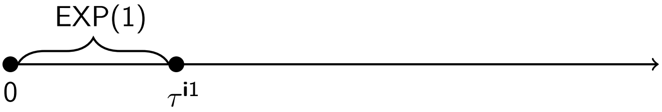
Poisson weighted infinite tree: edge labels

- For each vertex $\mathbf{i}$, take a unit intensity Poisson point process (PPP) $(\tau^{\mathbf{i}j})_{j=1}^{\infty}$ on $[0, \infty)$:



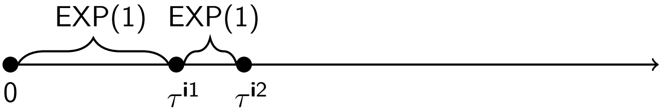
Poisson weighted infinite tree: edge labels

- For each vertex $\mathbf{i}$, take a unit intensity Poisson point process (PPP) $(\tau^{\mathbf{i}j})_{j=1}^{\infty}$ on $[0, \infty)$:



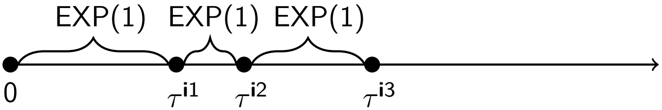
Poisson weighted infinite tree: edge labels

- For each vertex $\mathbf{i}$, take a unit intensity Poisson point process (PPP) $(\tau^{\mathbf{i}j})_{j=1}^{\infty}$ on $[0, \infty)$:



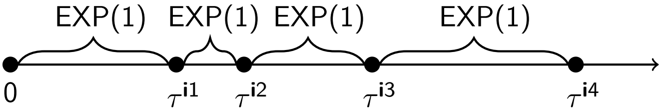
Poisson weighted infinite tree: edge labels

- For each vertex $\mathbf{i}$, take a unit intensity Poisson point process (PPP) $(\tau^{\mathbf{i}j})_{j=1}^{\infty}$ on $[0, \infty)$:



Poisson weighted infinite tree: edge labels

- For each vertex $\mathbf{i}$, take a unit intensity Poisson point process (PPP) $(\tau^{\mathbf{i}j})_{j=1}^{\infty}$ on $[0, \infty)$:

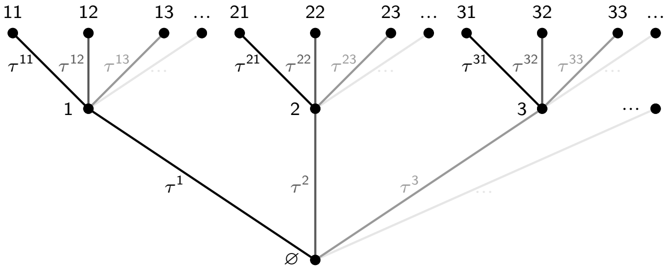


Poisson weighted infinite tree: edge labels

- For each vertex $\mathbf{i}$, take a unit intensity Poisson point process (PPP) $(\tau^{\mathbf{i}j})_{j=1}^{\infty}$ on $[0, \infty)$
- The PPP's are independent of each other

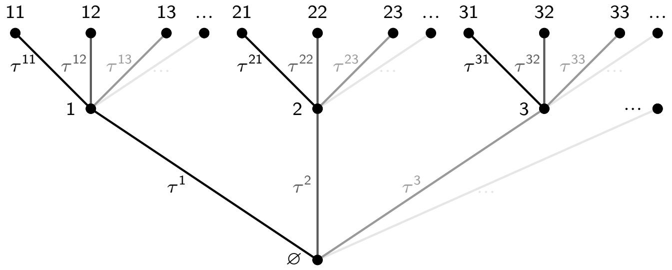
Poisson weighted infinite tree: edge labels

- For each vertex $\mathbf{i}$, take a unit intensity Poisson point process (PPP) $(\tau^{\mathbf{i}j})_{j=1}^{\infty}$ on $[0, \infty)$
- The PPP's are independent of each other
- $\tau^{\mathbf{i}j}$ is the label of the edge $\{\mathbf{i}, \mathbf{i}j\}$ that connects vertex $\mathbf{i}$ to its j^{th} child



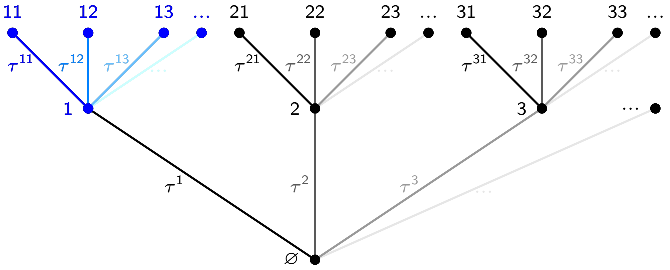
Poisson weighted infinite tree: edge labels

- For each vertex $\mathbf{i}$, take a unit intensity Poisson point process (PPP) $(\tau^{\mathbf{i}j})_{j=1}^{\infty}$ on $[0,\infty)$
- The PPP's are independent of each other
- $\tau^{\mathbf{i}j}$ is the label of the edge $\{\mathbf{i}, \mathbf{i}j\}$ that connects vertex $\mathbf{i}$ to its j^{th} child
- $\tau^{\mathbf{i}j}$ can be interpreted as the **arrival time** of the edge $\{\mathbf{i}, \mathbf{i}j\}$



Poisson weighted infinite tree: edge labels

- For each vertex $\mathbf{i}$, take a unit intensity Poisson point process (PPP) $(\tau^{\mathbf{i}j})_{j=1}^{\infty}$ on $[0,\infty)$
- The PPP's are independent of each other
- $\tau^{\mathbf{i}j}$ is the label of the edge $\{\mathbf{i},\mathbf{i}j\}$ that connects vertex $\mathbf{i}$ to its j^{th} child
- $\tau^{\mathbf{i}j}$ can be interpreted as the **arrival time** of the edge $\{\mathbf{i}, \mathbf{i}j\}$



- The subtree rooted at any vertex is a PWIT itself

RDCP on the PWIT

- We decorate the vertices of the PWIT with i.i.d. random degree constraints with distribution $\underline{p}$

RDCP on the PWIT

- We decorate the vertices of the PWIT with i.i.d. random degree constraints with distribution $\underline{p}$
- $d(\mathbf{i})$: the degree constraint of the vertex $\mathbf{i}$

RDCP on the PWIT

- We decorate the vertices of the PWIT with i.i.d. random degree constraints with distribution $\underline{p}$
- $d(\mathbf{i})$: the degree constraint of the vertex $\mathbf{i}$
- The RDCP construction carries over to the PWIT

RDCP on the PWIT

- We decorate the vertices of the PWIT with i.i.d. random degree constraints with distribution $\underline{p}$
- $d(\mathbf{i})$: the degree constraint of the vertex $\mathbf{i}$
- The RDCP construction carries over to the PWIT:
  - Try to add the edges in increasing order of their arrival times

RDCP on the PWIT

- We decorate the vertices of the PWIT with i.i.d. random degree constraints with distribution $\underline{p}$
- $d(\mathbf{i})$: the degree constraint of the vertex $\mathbf{i}$
- The RDCP construction carries over to the PWIT:
  - Try to add the edges in increasing order of their arrival times
  - Only add an edge if it does not violate any degree constraint

RDCP on the PWIT

- We decorate the vertices of the PWIT with i.i.d. random degree constraints with distribution $\underline{p}$
- $d(\mathbf{i})$: the degree constraint of the vertex $\mathbf{i}$
- The RDCP construction carries over to the PWIT:
  - Try to add the edges in increasing order of their arrival times
  - Only add an edge if it does not violate any degree constraint

Theorem (The limit as the RDCP on PWIT (Ráth, Sz., Warnke, 2024+))

Fix $t \in \mathbb{R}_+ \cup \{\infty\}$. Then the random rooted tree $\left(G^{\infty}_{\underline{p}}(t), \varnothing\right)$ that appears as the limit object of the RDCP on K_n is the connected component of the root in the RDCP on the PWIT at time t.

RDCP on the PWIT

- We decorate the vertices of the PWIT with i.i.d. random degree constraints with distribution $\underline{p}$
- $d(\mathbf{i})$: the degree constraint of the vertex $\mathbf{i}$
- The RDCP construction carries over to the PWIT:
    - Try to add the edges in increasing order of their arrival times
    - Only add an edge if it does not violate any degree constraint

Theorem (The limit as the RDCP on PWIT (Ráth, Sz., Warnke, 2024+))

Fix $t \in \mathbb{R}_+ \cup \{\infty\}$. Then the random rooted tree $\left(G^{\infty}_{\underline{p}}(t), \varnothing\right)$ that appears as the limit object of the RDCP on K_n is the connected component of the root in the RDCP on the PWIT at time t.

Question: Is it a Galton–Watson tree?

Phantom saturation times of the RDCP on the PWIT

- The **phantom saturation time** $T_{\mathbf{i}}$ of vertex **i** of the PWIT is the time when vertex **i** saturates in the RDCP on the PWIT **rooted at i**

Phantom saturation times of the RDCP on the PWIT

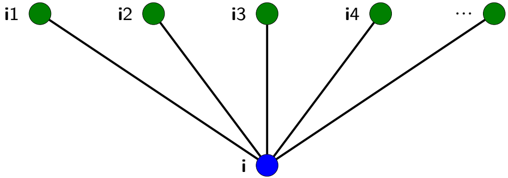
- The **phantom saturation time** $T_{\mathbf{i}}$ of vertex **i** of the PWIT is the time when vertex **i** saturates in the RDCP on the PWIT **rooted at i**
- The subtree rooted at any vertex is a PWIT

Phantom saturation times of the RDCP on the PWIT

- The **phantom saturation time** $T_{\mathbf{i}}$ of vertex **i** of the PWIT is the time when vertex **i** saturates in the RDCP on the PWIT **rooted at i**
- The subtree rooted at any vertex is a PWIT
  $\Rightarrow$ each $T_{\mathbf{i}}$ has the same distribution

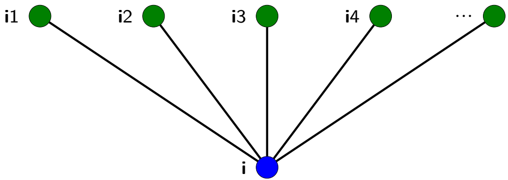
Phantom saturation times of the RDCP on the PWIT

- The **phantom saturation time** $T_{\mathbf{i}}$ of vertex **i** of the PWIT is the time when vertex **i** saturates in the RDCP on the PWIT **rooted at i**
- The subtree rooted at any vertex is a PWIT
  $\Rightarrow$ each $T_{\mathbf{i}}$ has the same distribution
- The distribution of phantom saturation times solves a **recursion**:



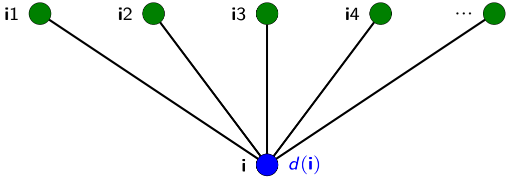
Phantom saturation times of the RDCP on the PWIT

- The **phantom saturation time** $T_{\mathbf{i}}$ of vertex $\mathbf{i}$ of the PWIT is the time when vertex $\mathbf{i}$ saturates in the RDCP on the PWIT **rooted at i**
- The subtree rooted at any vertex is a PWIT
  $\Rightarrow$ each $T_{\mathbf{i}}$ has the same distribution
- The distribution of phantom saturation times solves a **recursion**:
  If we know:



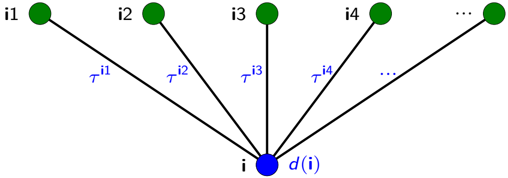
Phantom saturation times of the RDCP on the PWIT

- The **phantom saturation time** $T_{\mathbf{i}}$ of vertex **i** of the PWIT is the time when vertex **i** saturates in the RDCP on the PWIT **rooted at i**
- The subtree rooted at any vertex is a PWIT
  $\Rightarrow$ each $T_{\mathbf{i}}$ has the same distribution
- The distribution of phantom saturation times solves a **recursion**:
  If we know: the degree constraint of a given vertex: $d(\mathbf{i})$



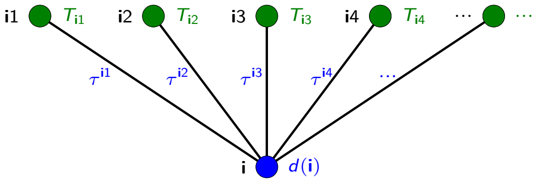
Phantom saturation times of the RDCP on the PWIT

- The **phantom saturation time** $T_{\mathbf{i}}$ of vertex **i** of the PWIT is the time when vertex **i** saturates in the RDCP on the PWIT **rooted at i**
- The subtree rooted at any vertex is a PWIT
  $\Rightarrow$ each $T_{\mathbf{i}}$ has the same distribution
- The distribution of phantom saturation times solves a **recursion**:
  If we know: the degree constraint of a given vertex: $d(\mathbf{i})$
  the arrival times of the edges to its children: $(\tau^{\mathbf{i}j})_{j=1}^{\infty}$



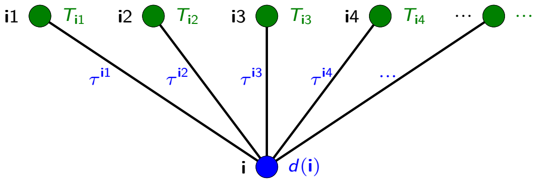
Phantom saturation times of the RDCP on the PWIT

- The **phantom saturation time** $T_{\mathbf{i}}$ of vertex **i** of the PWIT is the time when vertex **i** saturates in the RDCP on the PWIT **rooted at i**
- The subtree rooted at any vertex is a PWIT
  $\Rightarrow$ each $T_{\mathbf{i}}$ has the same distribution
- The distribution of phantom saturation times solves a **recursion**:
  If we know: the degree constraint of a given vertex: $d(\mathbf{i})$
  the arrival times of the edges to its children: $(\tau^{\mathbf{i}j})_{j=1}^{\infty}$
  the phantom saturation times of its children: $(T_{\mathbf{i}j})_{j=1}^{\infty}$



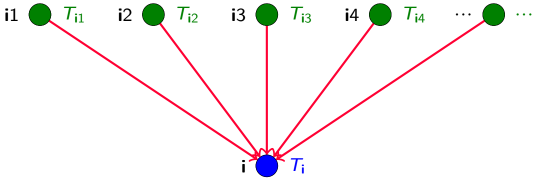
Phantom saturation times of the RDCP on the PWIT

- The **phantom saturation time** $T_{\mathbf{i}}$ of vertex **i** of the PWIT is the time when vertex **i** saturates in the RDCP on the PWIT **rooted at i**
- The subtree rooted at any vertex is a PWIT
  $\Rightarrow$ each $T_{\mathbf{i}}$ has the same distribution
- The distribution of phantom saturation times solves a **recursion**:
  If we know: the degree constraint of a given vertex: $d(\mathbf{i})$
  the arrival times of the edges to its children: $(\tau^{\mathbf{i}j})_{j=1}^{\infty}$
  the phantom saturation times of its children: $(T_{\mathbf{i}j})_{j=1}^{\infty}$
  $\Rightarrow$ we can determine the phantom saturation time of the vertex: $T_{\mathbf{i}}$



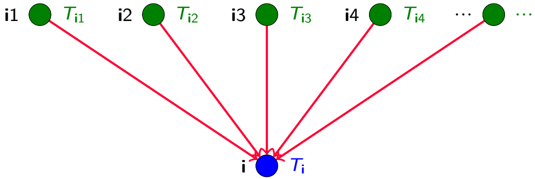
Phantom saturation times of the RDCP on the PWIT

- Recursion: generation $(r+1)$, $(T_{\mathbf{i}j})_{j=1}^{\infty}$ $\Rightarrow$ generation r, $T_{\mathbf{i}}$



Phantom saturation times of the RDCP on the PWIT

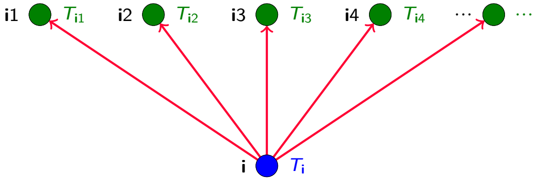
- Recursion: generation $(r+1)$, $(T_{\mathbf{i}j})_{j=1}^{\infty}$ $\Rightarrow$ generation r, $T_{\mathbf{i}}$



↳ Explicit fixed point equation for the distribution of $T_{\mathbf{i}}$

Phantom saturation times of the RDCP on the PWIT

- Recursion: generation $(r+1)$, $(T_{\mathbf{i}j})_{j=1}^{\infty}$ $\Rightarrow$ generation r, $T_{\mathbf{i}}$

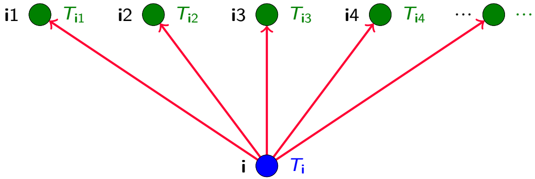


↳ Explicit fixed point equation for the distribution of $T_{\mathbf{i}}$

- Other recursion for the distribution of phantom saturation times: generation r, $T_{\mathbf{i}}$ $\Rightarrow$ generation $(r+1)$, $(T_{\mathbf{i}j})_{j=1}^{\infty}$

Phantom saturation times of the RDCP on the PWIT

- Recursion: generation $(r+1)$, $(T_{\mathbf{i}j})_{j=1}^{\infty}$ $\Rightarrow$ generation r, $T_{\mathbf{i}}$

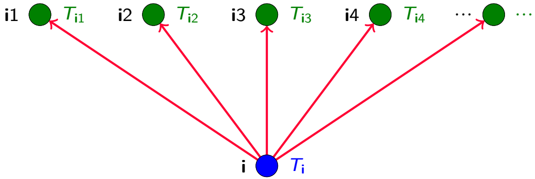


↳ Explicit fixed point equation for the distribution of $T_{\mathbf{i}}$

- Other recursion for the distribution of phantom saturation times: generation r, $T_{\mathbf{i}}$ $\Rightarrow$ generation $(r+1)$, $(T_{\mathbf{i}j})_{j=1}^{\infty}$
- The local weak limit is the family tree of a **multi-type branching process** (MTBP)

Phantom saturation times of the RDCP on the PWIT

- Recursion: generation $(r+1)$, $(T_{\mathbf{i}j})_{j=1}^{\infty} \Rightarrow$ generation r, $T_{\mathbf{i}}$

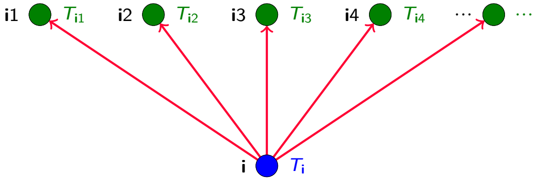


↳ Explicit fixed point equation for the distribution of $T_{\mathbf{i}}$

- Other recursion for the distribution of phantom saturation times: generation r, $T_{\mathbf{i}} \Rightarrow$ generation $(r+1)$, $(T_{\mathbf{i}j})_{j=1}^{\infty}$
- The local weak limit is the family tree of a **multi-type branching process** (MTBP)
  - The type of a vertex is its phantom saturation time

Phantom saturation times of the RDCP on the PWIT

- Recursion: generation $(r+1)$, $(T_{\mathbf{i}j})_{j=1}^{\infty} \Rightarrow$ generation r, $T_{\mathbf{i}}$



↳ Explicit fixed point equation for the distribution of $T_{\mathbf{i}}$

- Other recursion for the distribution of phantom saturation times: generation r, $T_{\mathbf{i}} \Rightarrow$ generation $(r+1)$, $(T_{\mathbf{i}j})_{j=1}^{\infty}$
- The local weak limit is the family tree of a **multi-type branching process** (MTBP)
  - The type of a vertex is its phantom saturation time
  - The type of a vertex determines the distribution of the number of its children and their types

Critical time t_c

Theorem (Phase transition of the RDCP (Warnke, Wormald, 2022+))

There exists a critical time $t_c = t_c(\underline{p})$ when a giant connected component emerges in the RDCP on K_n.

Critical time t_c

Theorem (Phase transition of the RDCP (Warnke, Wormald, 2022+))

There exists a critical time $t_c = t_c(\underline{p})$ when a giant connected component emerges in the RDCP on K_n.

Question: What can we say about the value of t_c?

Critical time t_c

Theorem (Phase transition of the RDCP (Warnke, Wormald, 2022+))

There exists a critical time $t_c = t_c(\underline{p})$ when a giant connected component emerges in the RDCP on K_n.

Question: What can we say about the value of t_c?

- We can characterize t_c using the local weak limit

Critical time t_c

Theorem (Phase transition of the RDCP (Warnke, Wormald, 2022+))

There exists a critical time $t_c = t_c(\underline{p})$ when a giant connected component emerges in the RDCP on K_n.

Question: What can we say about the value of t_c?

- We can characterize t_c using the local weak limit:

  Namely: t_c can be defined as the first time when the local weak limit has infinitely many vertices with positive probability

Eigenvalue criteria for t_c

Configuration model:

Eigenvalue criteria for t_c

Configuration model:

- The local weak limit is a **branching process** tree

Eigenvalue criteria for t_c

Configuration model:

- The local weak limit is a **branching process** tree
- μ: the expected value of the offspring distribution

Eigenvalue criteria for t_c

Configuration model:

- The local weak limit is a **branching process** tree
- μ: the expected value of the offspring distribution
- A branching process survives if $\mu > 1$

Eigenvalue criteria for t_c

Configuration model:

- The local weak limit is a **branching process** tree
- μ: the expected value of the offspring distribution
- A branching process survives if $\mu > 1$
- The critical time of the configuration model is classical (Molloy–Reed criterion)

Eigenvalue criteria for t_c

Configuration model:

- The local weak limit is a **branching process** tree
- μ: the expected value of the offspring distribution
- A branching process survives if $\mu > 1$
- The critical time of the configuration model is classical (Molloy–Reed criterion)

RDCP on K_n:

Eigenvalue criteria for t_c

Configuration model:

- The local weak limit is a **branching process** tree
- μ: the expected value of the offspring distribution
- A branching process survives if $\mu > 1$
- The critical time of the configuration model is classical (Molloy–Reed criterion)

RDCP on K_n:

- The local weak limit is the family tree of a **MTBP**

Eigenvalue criteria for t_c

Configuration model:

- The local weak limit is a **branching process** tree
- μ: the expected value of the offspring distribution
- A branching process survives if $\mu > 1$
- The critical time of the configuration model is classical (Molloy–Reed criterion)

RDCP on K_n:

- The local weak limit is the family tree of a **MTBP**
- μ: the principal eigenvalue of its so called branching operator

Eigenvalue criteria for t_c

Configuration model:

- The local weak limit is a **branching process** tree
- μ: the expected value of the offspring distribution
- A branching process survives if $\mu > 1$
- The critical time of the configuration model is classical (Molloy–Reed criterion)

RDCP on K_n:

- The local weak limit is the family tree of a **MTBP**
- μ: the principal eigenvalue of its so called branching operator
- It is an integral operator in our case

Eigenvalue criteria for t_c

Configuration model:

- The local weak limit is a **branching process** tree
- μ: the expected value of the offspring distribution
- A branching process survives if $\mu > 1$
- The critical time of the configuration model is classical (Molloy–Reed criterion)

RDCP on K_n:

- The local weak limit is the family tree of a **MTBP**
- μ: the principal eigenvalue of its so called branching operator
- It is an integral operator in our case
- A MTBP survives if $\mu > 1$

Fine properties of t_c

- We prove an asymptotic expansion of t_c

Fine properties of t_c

- We prove an asymptotic expansion of t_c
- If each degree constraint is d (**random d-process**) then $t_c = t_c(d)$ satisfies

$$t_c(d) = 1 + \frac{2 + o(1)}{\mathrm{e} \cdot d!} \text{ as } d \to \infty \tag{3}$$

Fine properties of t_c

- We prove an asymptotic expansion of t_c
- If each degree constraint is d (**random $\boldsymbol{d}$-process**) then $t_c = t_c(d)$ satisfies

$$t_c(d) = 1 + \frac{2 + o(1)}{\mathrm{e} \cdot d!} \text{ as } d \to \infty \tag{3}$$

Corollary:

Fine properties of t_c

- We prove an asymptotic expansion of t_c
- If each degree constraint is d (**random $\boldsymbol{d}$-process**) then $t_c = t_c(d)$ satisfies

$$t_c(d) = 1 + \frac{2 + o(1)}{\mathrm{e} \cdot d!} \text{ as } d \to \infty \tag{3}$$

Corollary:

- There exists d_0 such that for $d \geqslant d_0$, the random d-process has significantly more edges when the giant component emerges than the Erdős–Rényi graph process (where there are no degree constraints)

Fine properties of t_c

- We prove an asymptotic expansion of t_c
- If each degree constraint is d (**random *d*-process**) then $t_c = t_c(d)$ satisfies
$$t_c(d) = 1 + \frac{2 + o(1)}{\mathrm{e} \cdot d!} \text{ as } d \to \infty \tag{3}$$

Corollary:

- There exists d_0 such that for $d \geqslant d_0$, the random d-process has significantly more edges when the giant component emerges than the Erdős–Rényi graph process (where there are no degree constraints)
- This result answers a non-trivial open problem of Warnke and Wormald for large values of d

Fine properties of t_c

- We prove an asymptotic expansion of t_c
- If each degree constraint is d (**random d-process**) then $t_c = t_c(d)$ satisfies

$$t_c(d) = 1 + \frac{2 + o(1)}{\mathrm{e} \cdot d!} \text{ as } d \to \infty \tag{3}$$

Corollary:

- There exists d_0 such that for $d \geqslant d_0$, the random d-process has significantly more edges when the giant component emerges than the Erdős–Rényi graph process (where there are no degree constraints)
- This result answers a non-trivial open problem of Warnke and Wormald for large values of d
- For $d = 3, 4, \ldots, d_0 - 1$ it is still open

Thank you for your attention!

szokemp@math.bme.hu

https://arxiv.org/pdf/2409.11747