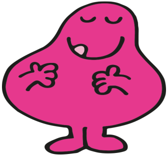
Greedy maximal independent sets via local limits

Peleg Michaeli

Tel Aviv University

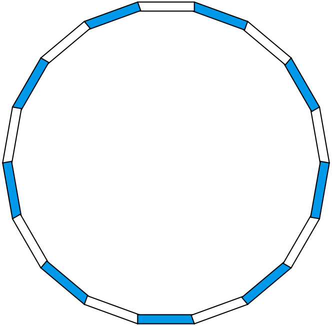
The 31st International Conference on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms (AofA2020)

September 2020

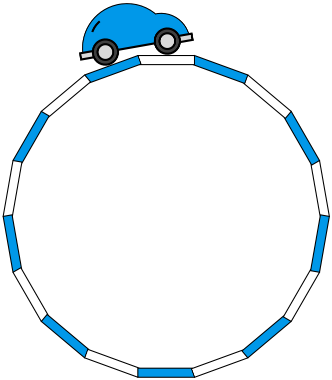


Joint work with Michael Krivelevich, Tamás Mészáros and Clara Shikhelman

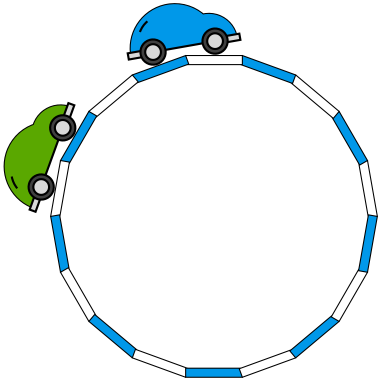
Parking cars on a cycle



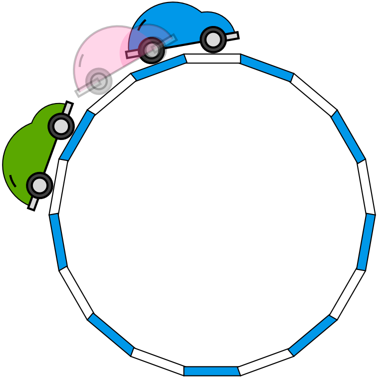
Parking cars on a cycle



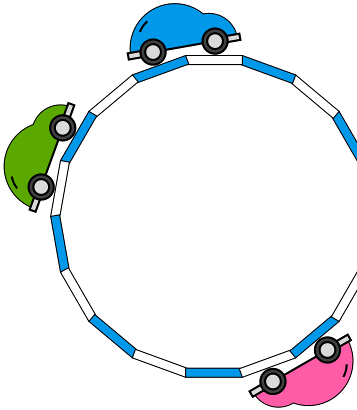
Parking cars on a cycle



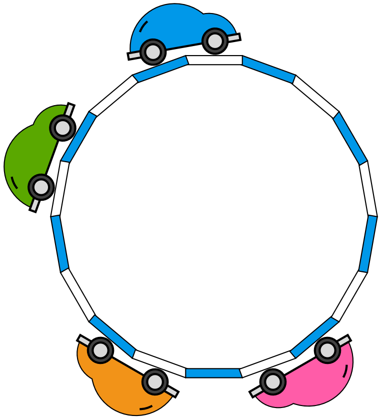
Parking cars on a cycle



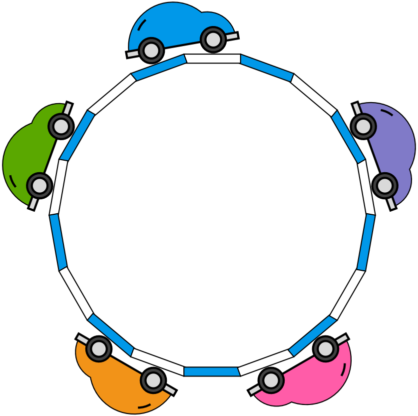
Parking cars on a cycle



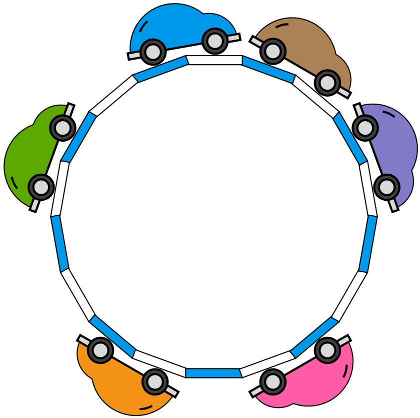
Parking cars on a cycle



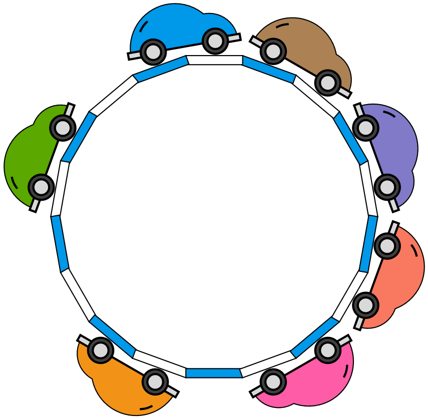
Parking cars on a cycle



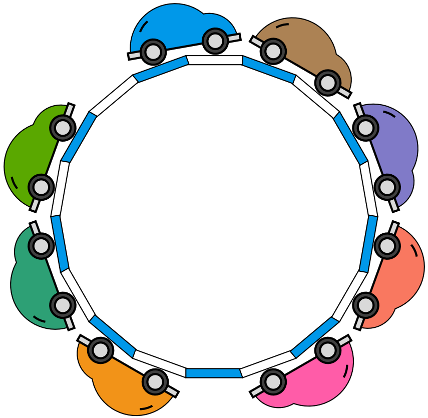
Parking cars on a cycle



Parking cars on a cycle



Parking cars on a cycle



Independent sets

An *independent set* is a set of vertices in a graph, no two of which are adjacent.

Independent sets

An *independent set* is a set of vertices in a graph, no two of which are adjacent.

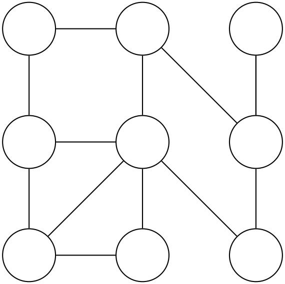
- Finding **maximum** independent sets is very hard 

Independent sets

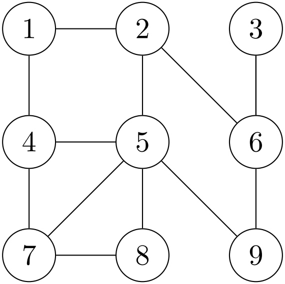
An *independent set* is a set of vertices in a graph, no two of which are adjacent.

- Finding **maximum** independent sets is very hard
- Finding **maximal** independent sets is very easy

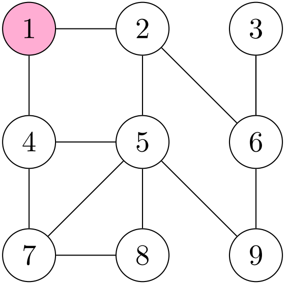
Greedy MIS



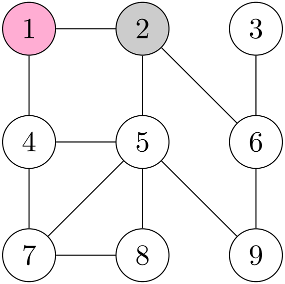
Greedy MIS



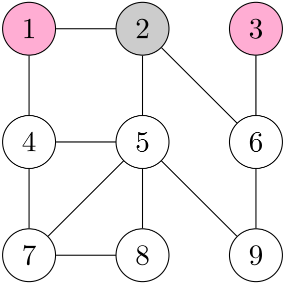
Greedy MIS



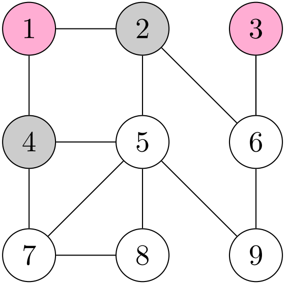
Greedy MIS



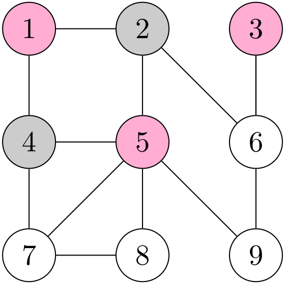
Greedy MIS



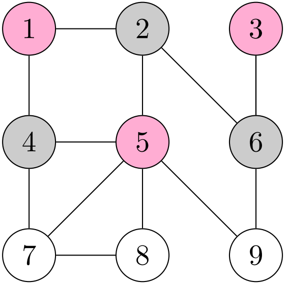
Greedy MIS



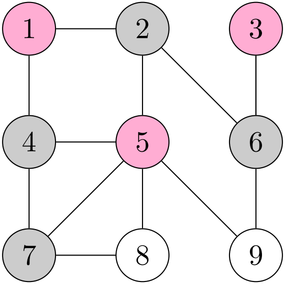
Greedy MIS



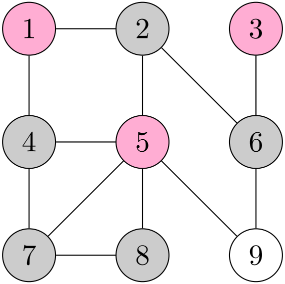
Greedy MIS



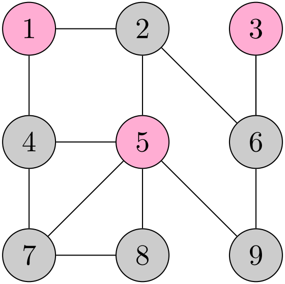
Greedy MIS



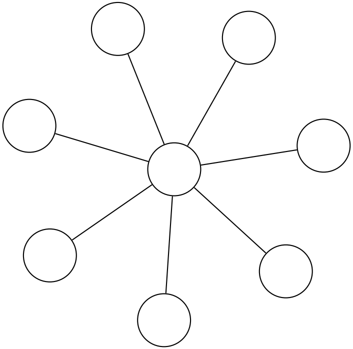
Greedy MIS



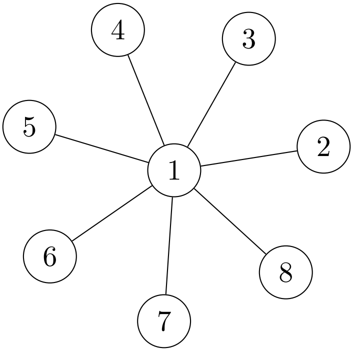
Greedy MIS



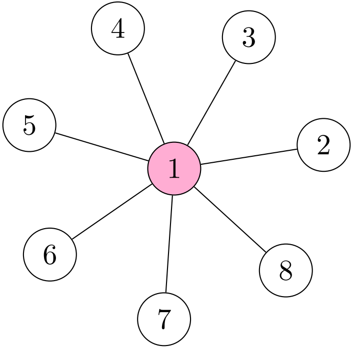
Greedy MIS — performance



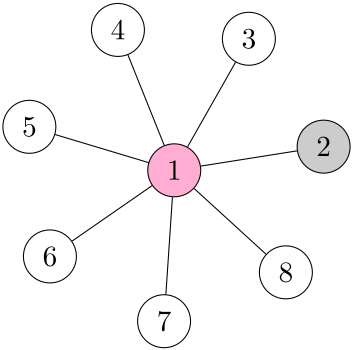
Greedy MIS — performance



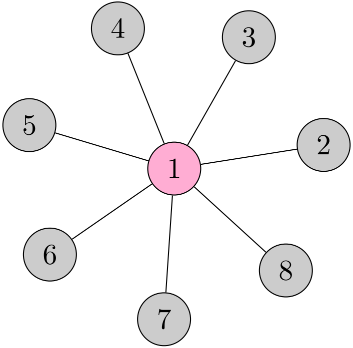
Greedy MIS — performance



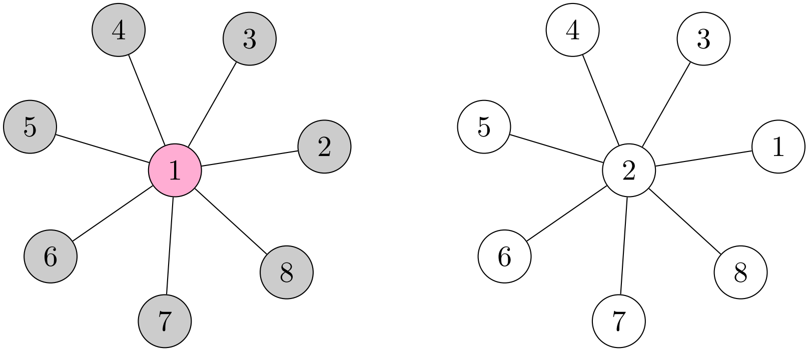
Greedy MIS — performance



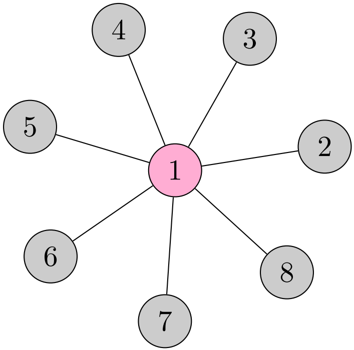
Greedy MIS — performance

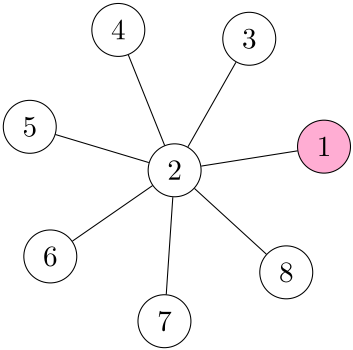


Greedy MIS — performance

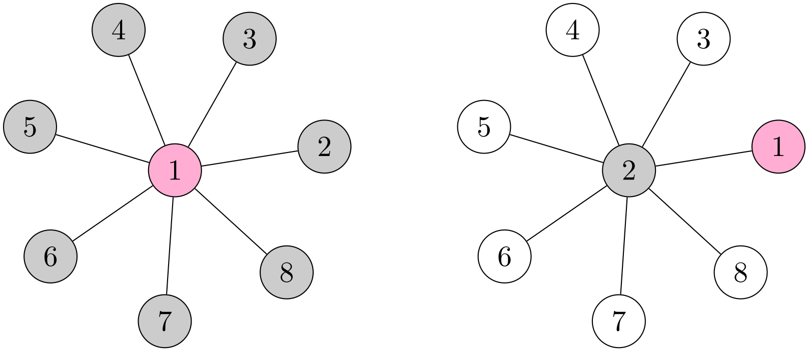


Greedy MIS — performance

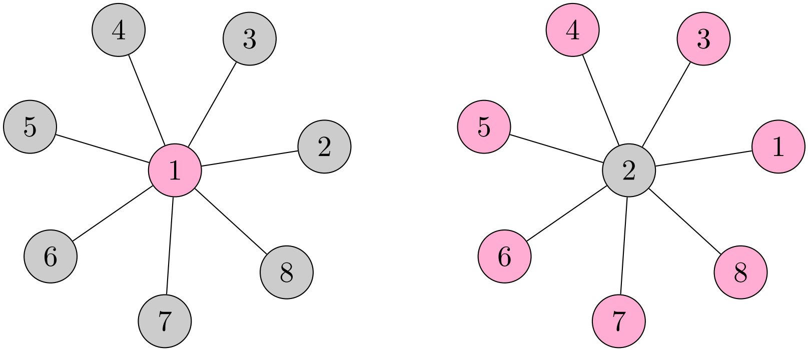




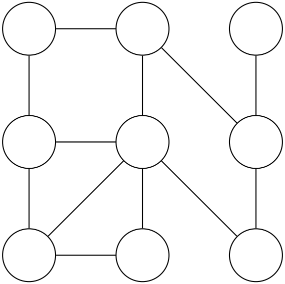
Greedy MIS — performance



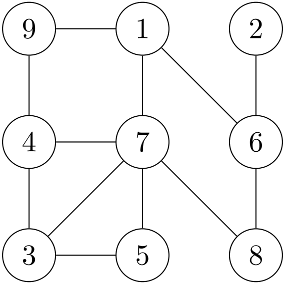
Greedy MIS — performance



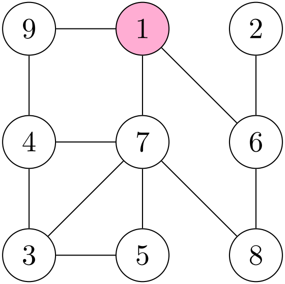
Random greedy MIS — sequential



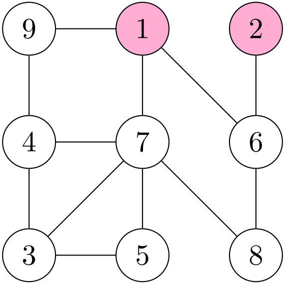
Random greedy MIS — sequential



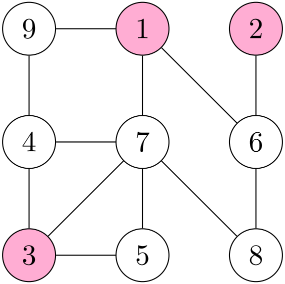
Random greedy MIS — sequential



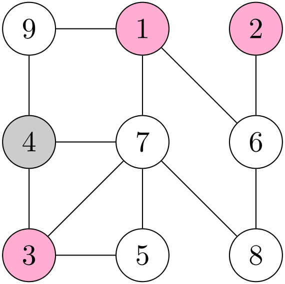
Random greedy MIS — sequential



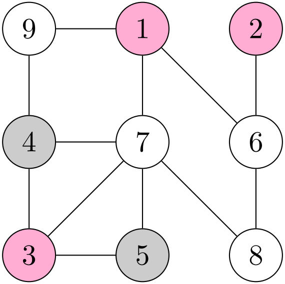
Random greedy MIS — sequential



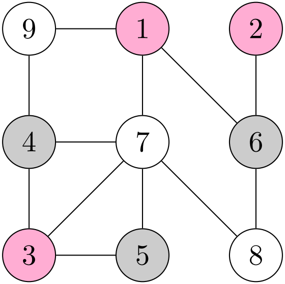
Random greedy MIS — sequential



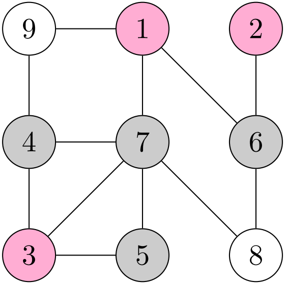
Random greedy MIS — sequential



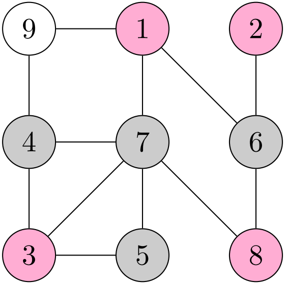
Random greedy MIS — sequential



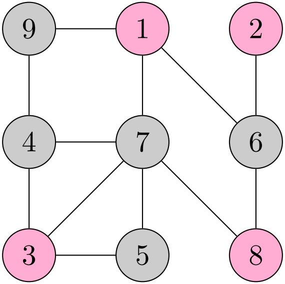
Random greedy MIS — sequential



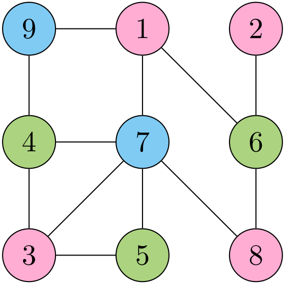
Random greedy MIS — sequential



Random greedy MIS — sequential



Random greedy MIS — sequential



Greedy independence ratio — previous work

Let $\mathbf{I}(G)$ be the yielded independent set, and let $\iota(G) = |\mathbf{I}(G)|/|V(G)|$.

Greedy independence ratio — previous work

random variable

Let $\mathbf{I}(G)$ be the yielded independent set, and let $\iota(G) = |\mathbf{I}(G)|/|V(G)|$.

Greedy independence ratio — previous work

random variable

Let $\mathbf{I}(G)$ be the yielded independent set, and let $\iota(G) = |\mathbf{I}(G)|/|V(G)|$.

Flory '39, Page '59 $\quad \iota(P_n) \to \frac{1}{2}(1 - e^{-2})$

Greedy independence ratio — previous work

random variable

Let $\mathbf{I}(G)$ be the yielded independent set, and let $\iota(G) = |\mathbf{I}(G)|/|V(G)|$.

Flory '39, Page '59 $\iota(P_n) \to \frac{1}{2}(1 - e^{-2})$

McDiarmid '84 $\iota(G(n, d/n)) \to \log(1 + d)/d$

Greedy independence ratio — previous work

random variable

Let $\mathbf{I}(G)$ be the yielded independent set, and let $\iota(G) = |\mathbf{I}(G)|/|V(G)|$.

Flory '39, Page '59	$\iota(P_n) \to \frac{1}{2}(1 - e^{-2})$
McDiarmid '84	$\iota(G(n, d/n)) \to \log(1 + d)/d$
Wormald '95	$\iota(\mathcal{G}_{n,d}) \to \frac{1}{2}\left(1 - (d-1)^{-2/(d-2)}\right)$

Greedy independence ratio — previous work

random variable

Let $\mathbf{I}(G)$ be the yielded independent set, and let $\iota(G) = |\mathbf{I}(G)|/|V(G)|$.

Flory '39, Page '59	$\iota(P_n) \to \frac{1}{2}(1 - e^{-2})$
McDiarmid '84	$\iota(G(n, d/n)) \to \log(1 + d)/d$
Wormald '95	$\iota(\mathcal{G}_{n,d}) \to \frac{1}{2}\left(1 - (d-1)^{-2/(d-2)}\right)$
Lauer–Wormald '07	(same for d-regular graphs with girth $\to \infty$)

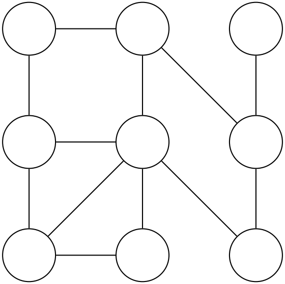
Greedy independence ratio — previous work

random variable →

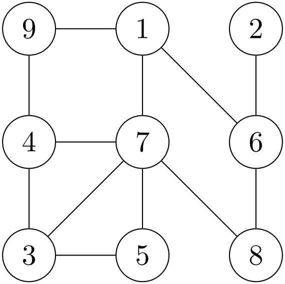
Let $\mathbf{I}(G)$ be the yielded independent set, and let $\iota(G) = |\mathbf{I}(G)|/|V(G)|$.

Flory '39, Page '59	$\iota(P_n) \to \frac{1}{2}(1 - e^{-2})$
McDiarmid '84	$\iota(G(n, d/n)) \to \log(1+d)/d$
Wormald '95	$\iota(\mathcal{G}_{n,d}) \to \frac{1}{2}\big(1 - (d-1)^{-2/(d-2)}\big)$
Lauer–Wormald '07	(same for d-regular graphs with girth $\to \infty$)
BJL '17, BJM '17	random graphs with given degree sequence

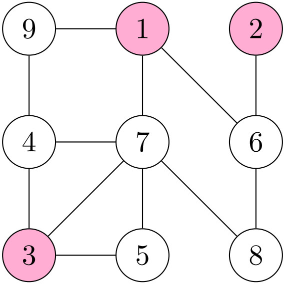
(Random) greedy MIS — parallel



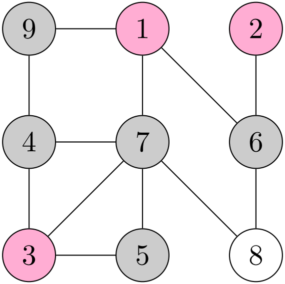
(Random) greedy MIS — parallel



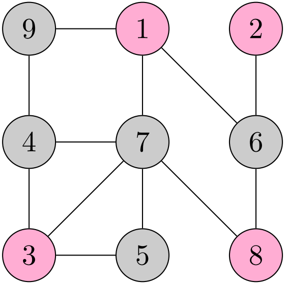
(Random) greedy MIS — parallel



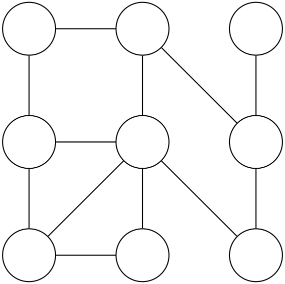
(Random) greedy MIS — parallel



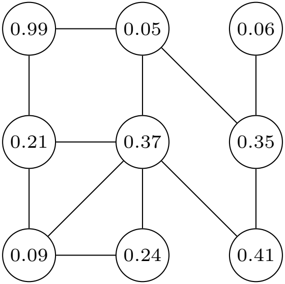
(Random) greedy MIS — parallel



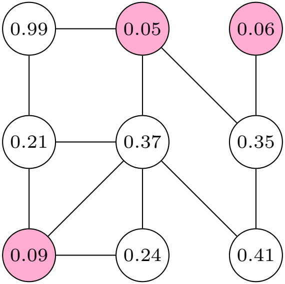
Random labelling



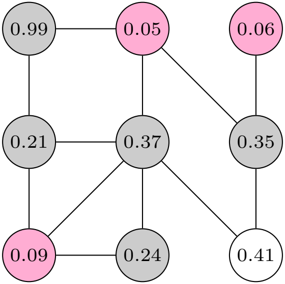
Random labelling



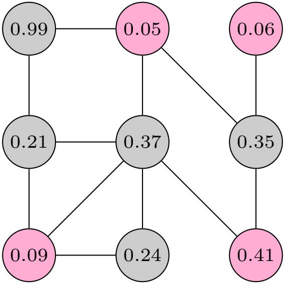
Random labelling



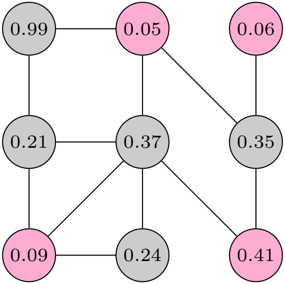
Random labelling



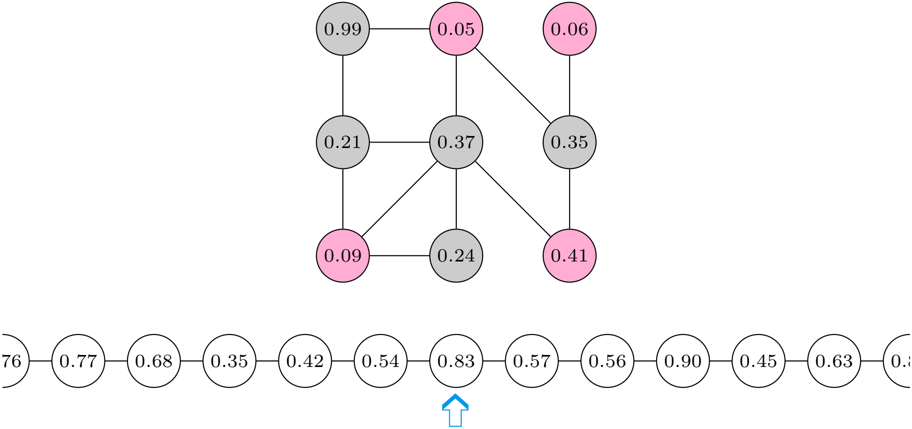
Random labelling



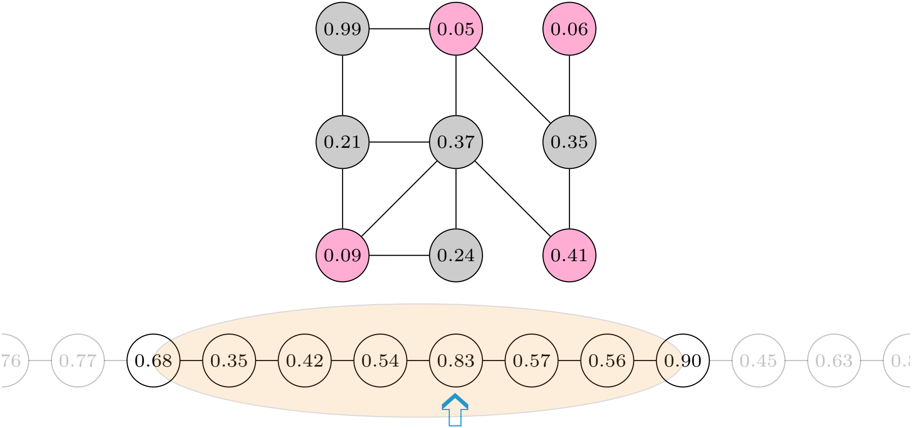
Random labelling



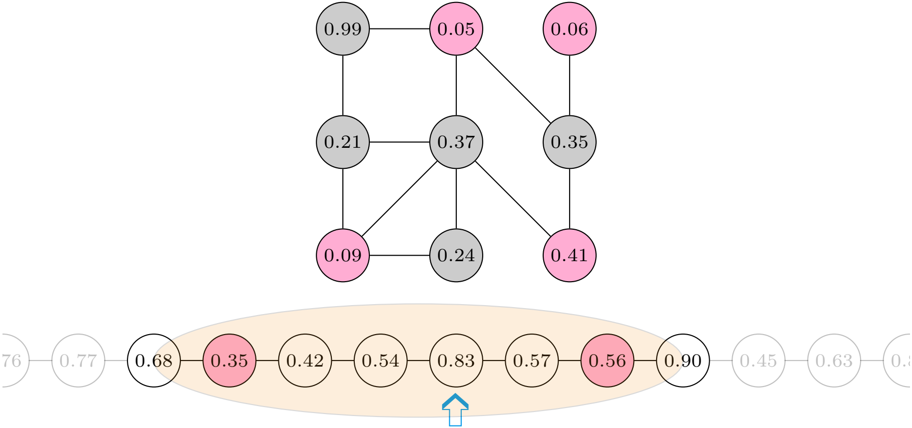
Random labelling



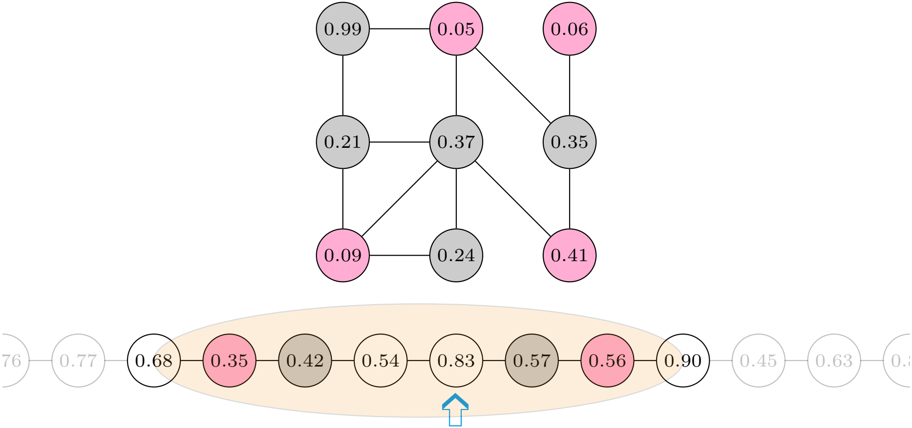
Random labelling



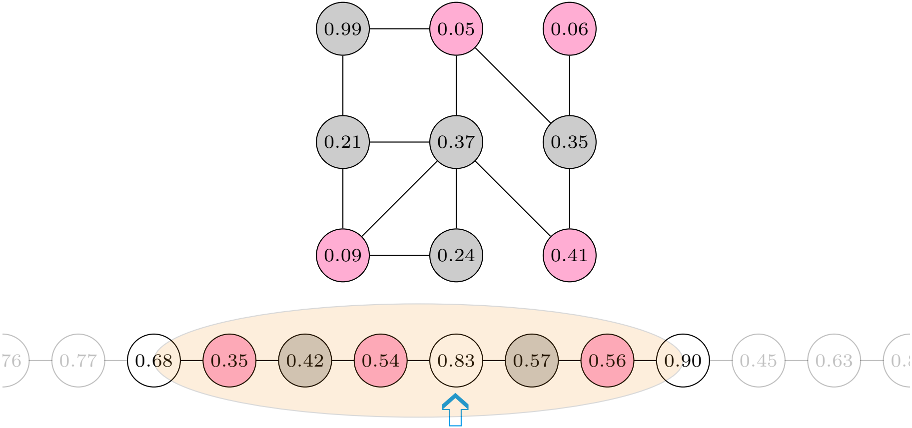
Random labelling



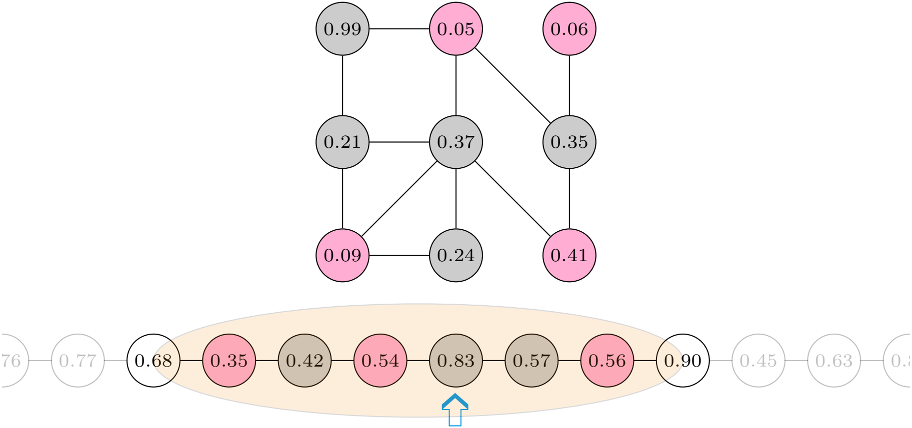
Random labelling



Random labelling



Random labelling



General framework

Let G_n be a graph sequence satisfying $|G_n| \to \infty$.

- We wish to calculate the asymptotics of $\iota(G_n)$.

General framework

Let G_n be a graph sequence satisfying $|G_n| \to \infty$.

- We wish to calculate the asymptotics of $\iota(G_n)$.
- We approximate $\mathbb{E}[\iota(G_n)] = \mathbb{P}(\rho_n \in \mathbf{I}(G_n))$ for ρ_n chosen uniformly.

General framework

Let G_n be a graph sequence satisfying $|G_n| \to \infty$.

- We wish to calculate the asymptotics of $\iota(G_n)$.
- We approximate $\mathbb{E}[\iota(G_n)] = \mathbb{P}(\rho_n \in \mathbf{I}(G_n))$ for ρ_n chosen uniformly.
- We hope that this is determined by a small neighbourhood of ρ_n.

General framework

Let G_n be a graph sequence satisfying $|G_n| \to \infty$.

- We wish to calculate the asymptotics of $\iota(G_n)$.
- We approximate $\mathbb{E}[\iota(G_n)] = \mathbb{P}(\rho_n \in \mathbf{I}(G_n))$ for ρ_n chosen uniformly.
- We hope that this is determined by a small neighbourhood of ρ_n.
- Decay of correlation $\implies$ $\iota(G_n) \sim \mathbb{E}[\iota(G_n)]$ a.a.s.

General framework

Let G_n be a graph sequence satisfying $|G_n| \to \infty$.

- We wish to calculate the asymptotics of $\iota(G_n)$.
- We approximate $\mathbb{E}[\iota(G_n)] = \mathbb{P}(\rho_n \in \mathbf{I}(G_n))$ for ρ_n chosen uniformly.
- We hope that this is determined by a small neighbourhood of ρ_n.
- Decay of correlation $\implies \iota(G_n) \sim \mathbb{E}[\iota(G_n)]$ a.a.s.
- This local view of ρ_n is captured by the *local limit* of G_n.

General framework

Let G_n be a graph sequence satisfying $|G_n| \to \infty$.

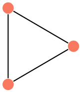
- We wish to calculate the asymptotics of $\iota(G_n)$.
- We approximate $\mathbb{E}[\iota(G_n)] = \mathbb{P}(\rho_n \in \mathbf{I}(G_n))$ for ρ_n chosen uniformly.
- We hope that this is determined by a small neighbourhood of ρ_n.
- Decay of correlation $\implies$ $\iota(G_n) \sim \mathbb{E}[\iota(G_n)]$ a.a.s.
- This local view of ρ_n is captured by the *local limit* of G_n.
- Develop a machinery to calculate the probability that the root of the local limit is red.

Local limits (a.k.a. Benjamini–Schramm Limits)

We say that a (random) graph sequence G_n *converges locally* to a (random) rooted graph (U, ρ) if for every $r \geq 0$ the ball $B_{G_n}(\rho_n, r)$ converges in distribution to $B_U(\rho, r)$, where ρ_n is a uniform vertex of G_n.

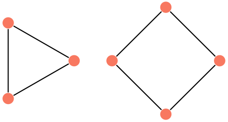
Local limits (a.k.a. Benjamini–Schramm Limits)

We say that a (random) graph sequence G_n *converges locally* to a (random) rooted graph (U, ρ) if for every $r \geq 0$ the ball $B_{G_n}(\rho_n, r)$ converges in distribution to $B_U(\rho, r)$, where ρ_n is a uniform vertex of G_n.



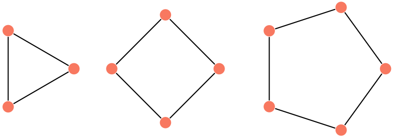
Local limits (a.k.a. Benjamini–Schramm Limits)

We say that a (random) graph sequence G_n *converges locally* to a (random) rooted graph (U, ρ) if for every $r \geq 0$ the ball $B_{G_n}(\rho_n, r)$ converges in distribution to $B_U(\rho, r)$, where ρ_n is a uniform vertex of G_n.



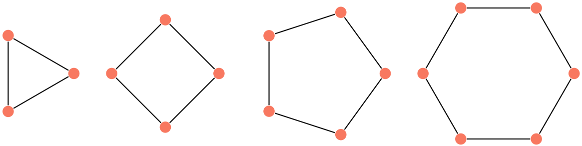
Local limits (a.k.a. Benjamini–Schramm Limits)

We say that a (random) graph sequence G_n *converges locally* to a (random) rooted graph (U, ρ) if for every $r \geq 0$ the ball $B_{G_n}(\rho_n, r)$ converges in distribution to $B_U(\rho, r)$, where ρ_n is a uniform vertex of G_n.



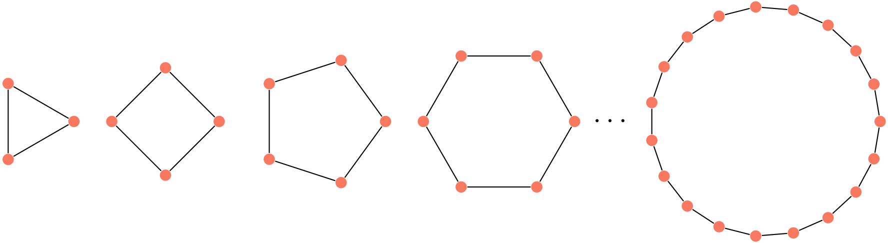
Local limits (a.k.a. Benjamini–Schramm Limits)

We say that a (random) graph sequence G_n *converges locally* to a (random) rooted graph (U, ρ) if for every $r \geq 0$ the ball $B_{G_n}(\rho_n, r)$ converges in distribution to $B_U(\rho, r)$, where ρ_n is a uniform vertex of G_n.



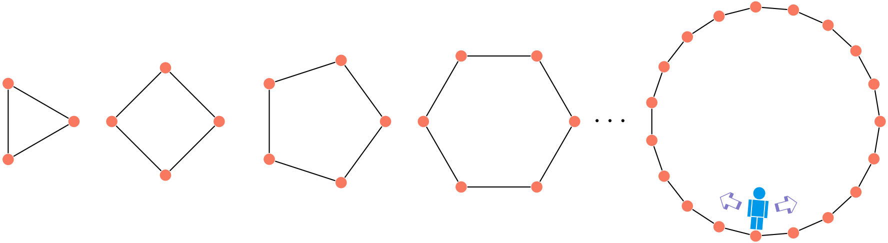
Local limits (a.k.a. Benjamini–Schramm Limits)

We say that a (random) graph sequence G_n *converges locally* to a (random) rooted graph (U, ρ) if for every $r \geq 0$ the ball $B_{G_n}(\rho_n, r)$ converges in distribution to $B_U(\rho, r)$, where ρ_n is a uniform vertex of G_n.



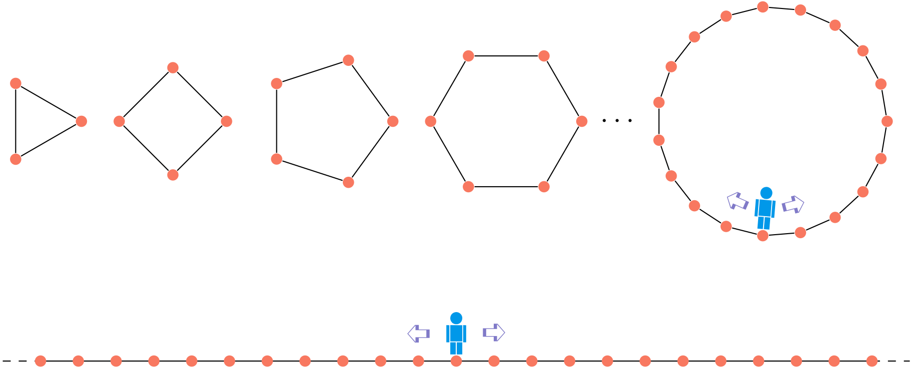
Local limits (a.k.a. Benjamini–Schramm Limits)

We say that a (random) graph sequence G_n *converges locally* to a (random) rooted graph (U, ρ) if for every $r \geq 0$ the ball $B_{G_n}(\rho_n, r)$ converges in distribution to $B_U(\rho, r)$, where ρ_n is a uniform vertex of G_n.



Local limits (a.k.a. Benjamini–Schramm Limits)

We say that a (random) graph sequence G_n *converges locally* to a (random) rooted graph (U, ρ) if for every $r \geq 0$ the ball $B_{G_n}(\rho_n, r)$ converges in distribution to $B_U(\rho, r)$, where ρ_n is a uniform vertex of G_n.



Local limits (a.k.a. Benjamini–Schramm Limits)

We say that a (random) graph sequence G_n *converges locally* to a (random) rooted graph (U, ρ) if for every $r \geq 0$ the ball $B_{G_n}(\rho_n, r)$ converges in distribution to $B_U(\rho, r)$, where ρ_n is a uniform vertex of G_n.

Examples

- $P_n, C_n \xrightarrow{\text{loc}} \mathbb{Z}$
- $[n]^d \xrightarrow{\text{loc}} \mathbb{Z}^d$
- $G(n, d/n) \xrightarrow{\text{loc}} \mathcal{T}_d$, a Galton–Watson Pois$(d)$ tree
- $G_{n,d} \xrightarrow{\text{loc}}$ the d-regular tree
- Uniform random tree $T_n \xrightarrow{\text{loc}} \hat{\mathcal{T}}_1$, a size-biased GW Pois$(1)$ tree
- Finite d-ary balanced tree $\xrightarrow{\text{loc}}$ the canopy tree

Convergence of the greedy independence ratio

Say that G_n has *subfactorial path growth* if the expected number of paths from a typical vertex is subfactorial in their length.

Convergence of the greedy independence ratio

Say that G_n has *subfactorial path growth* if the expected number of paths from a typical vertex is subfactorial in their length.

(bounded degree $\subsetneq$ subfactorial path growth)

Convergence of the greedy independence ratio

Say that G_n has *subfactorial path growth* if the expected number of paths from a typical vertex is subfactorial in their length.

(bounded degree $\subsetneq$ subfactorial path growth)

Theorem (Krivelevich, Mészáros, M., Shikhelman '20)

Suppose G_n has subfactorial path growth.
If $G_n \xrightarrow{\text{loc}} (U,\rho)$ then $\iota(G_n) \to \iota(U,\rho)$ a.a.s.

Convergence of the greedy independence ratio

Say that G_n has *subfactorial path growth* if the expected number of paths from a typical vertex is subfactorial in their length.

(bounded degree $\subsetneq$ subfactorial path growth)

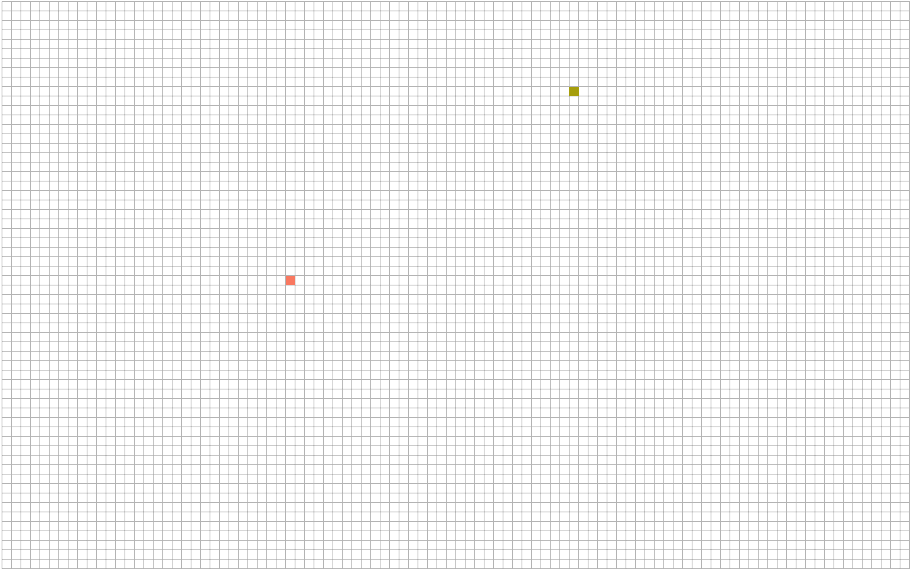
Theorem (Krivelevich, Mészáros, M., Shikhelman '20)

Suppose G_n has subfactorial path growth.
If $G_n \xrightarrow{\text{loc}} (U, \rho)$ then $\iota(G_n) \to \iota(U, \rho)$ a.a.s.

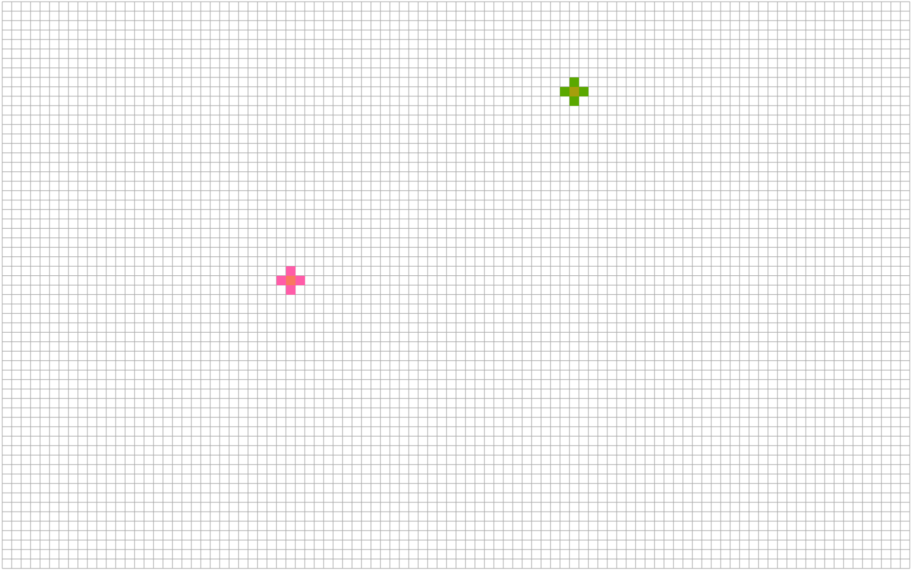
$\mathbb{P}(\rho \text{ is red})$

Exploration algorithms / decay of correlation

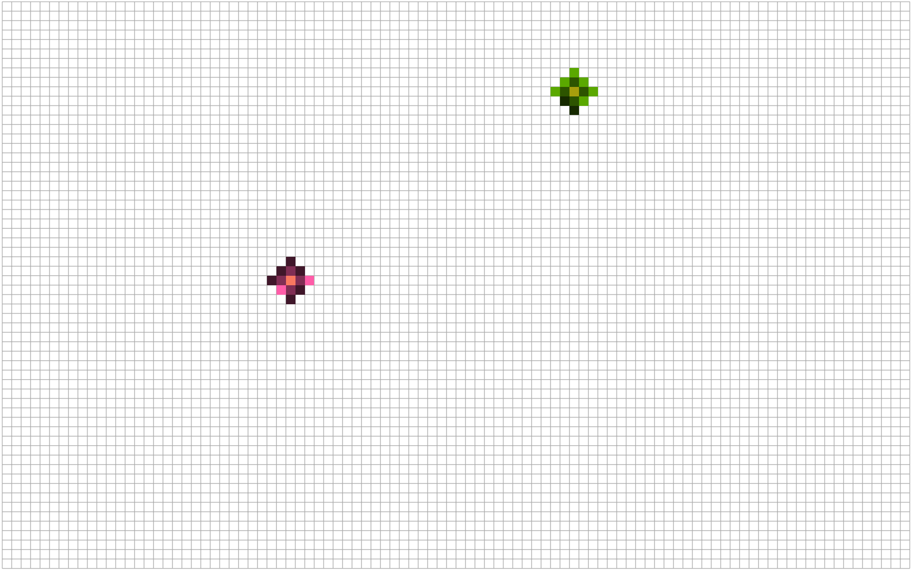
Exploration algorithms / decay of correlation



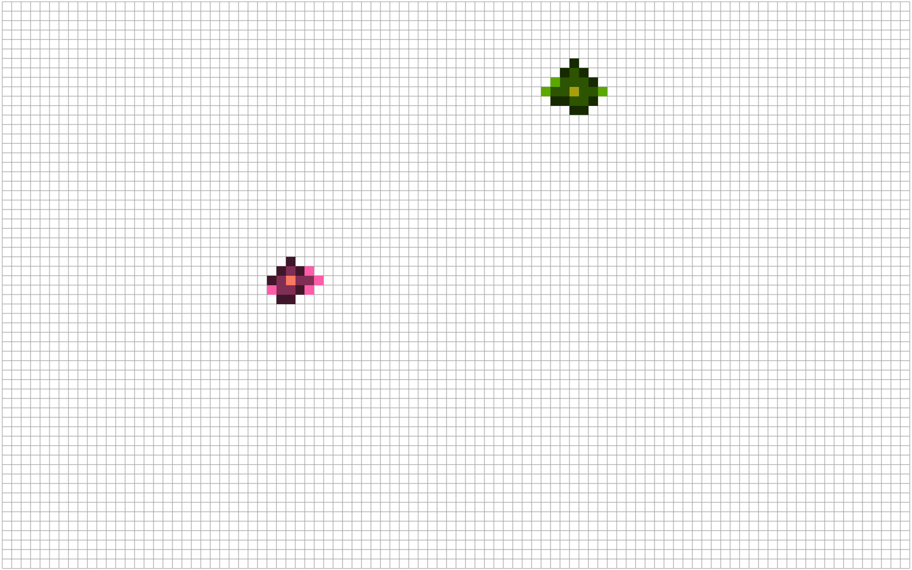
Exploration algorithms / decay of correlation



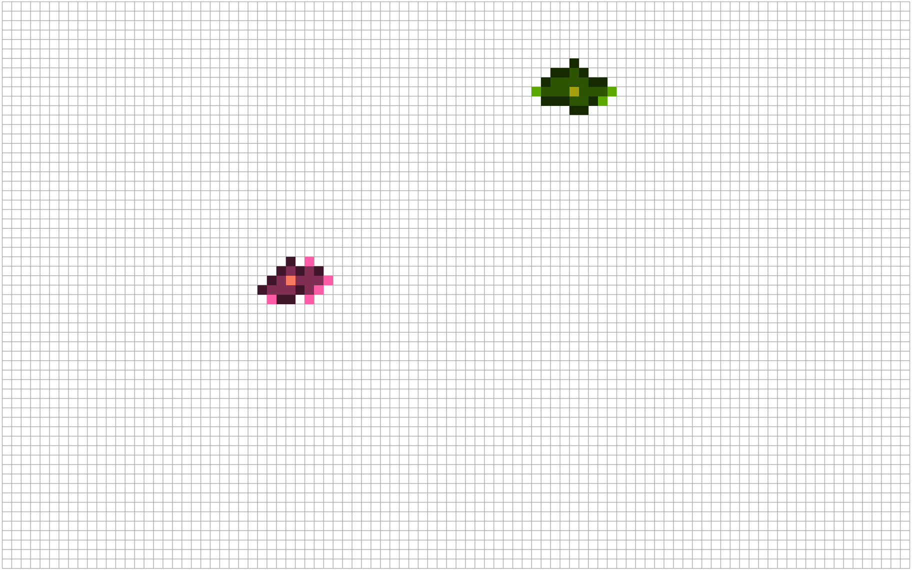
Exploration algorithms / decay of correlation



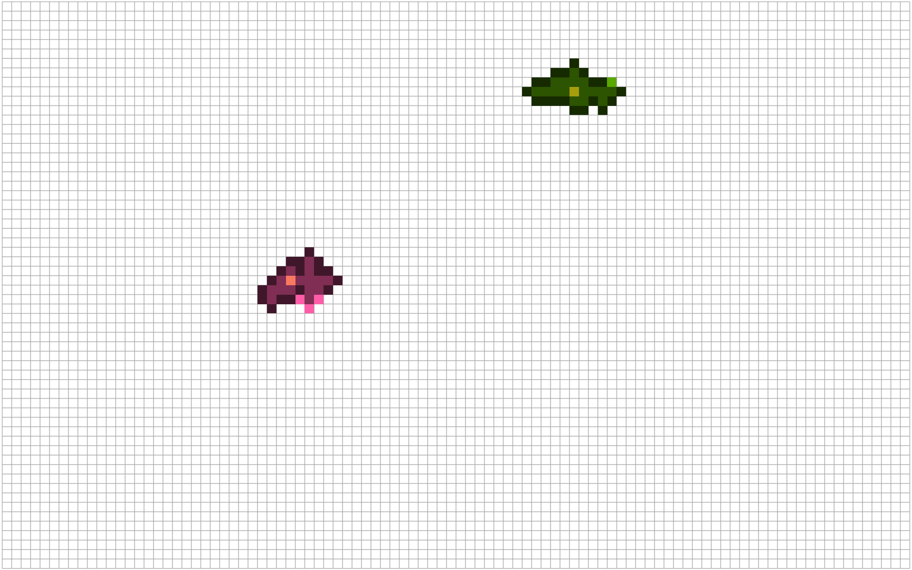
Exploration algorithms / decay of correlation



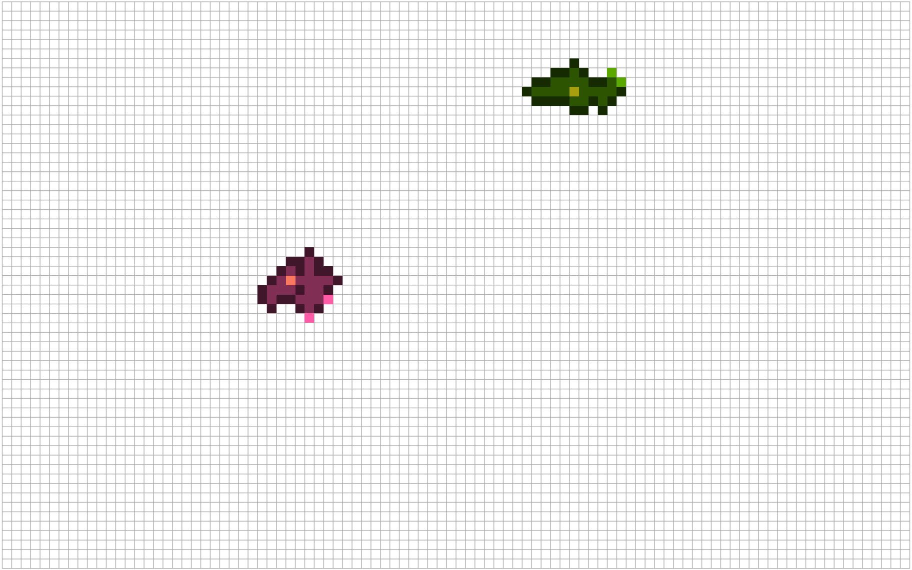
Exploration algorithms / decay of correlation



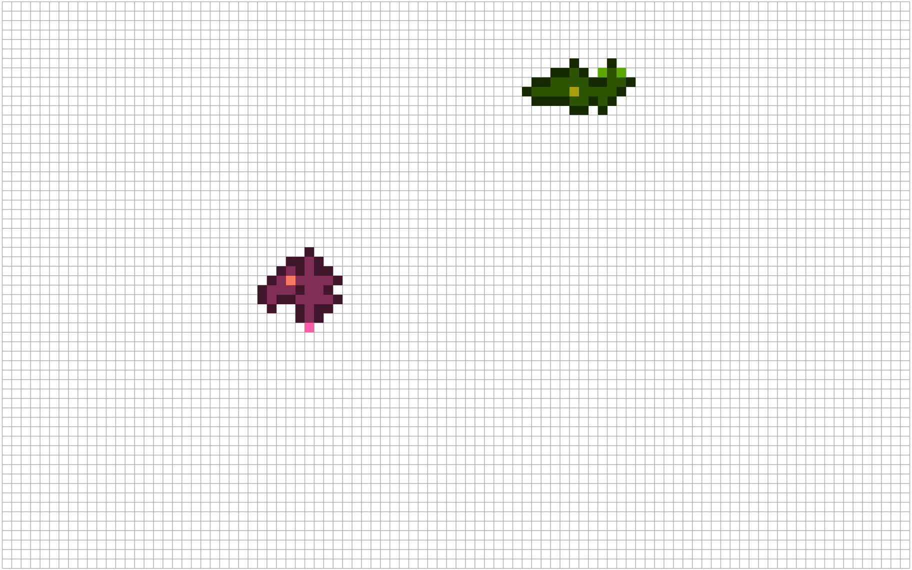
Exploration algorithms / decay of correlation



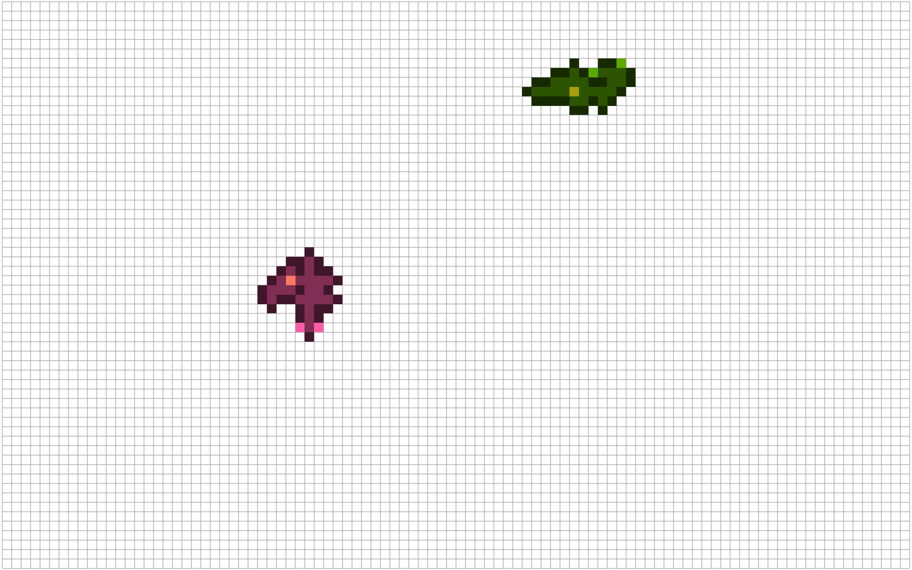
Exploration algorithms / decay of correlation



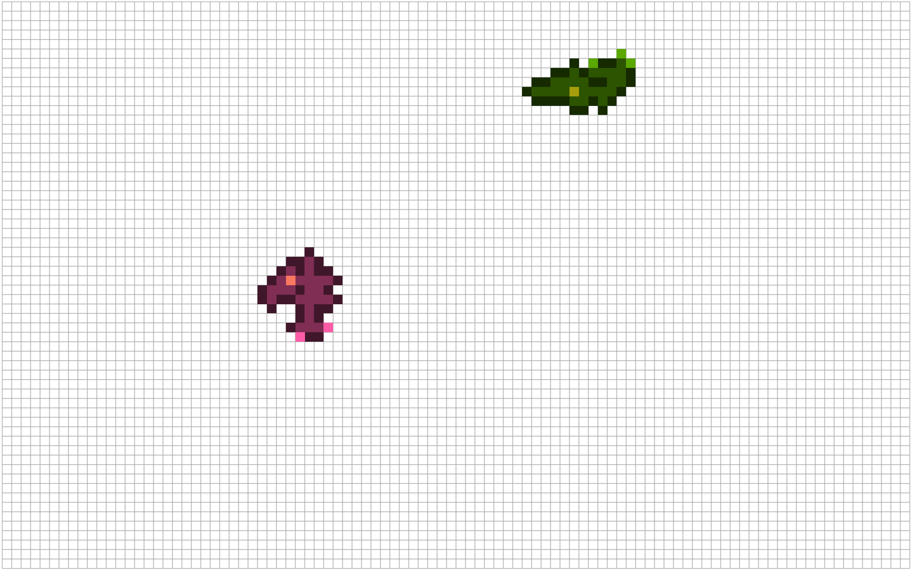
Exploration algorithms / decay of correlation



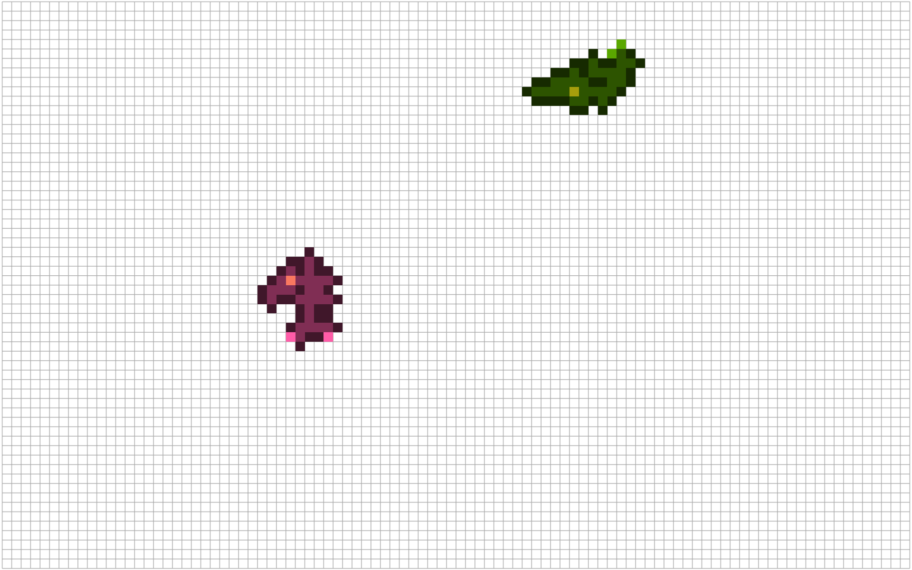
Exploration algorithms / decay of correlation



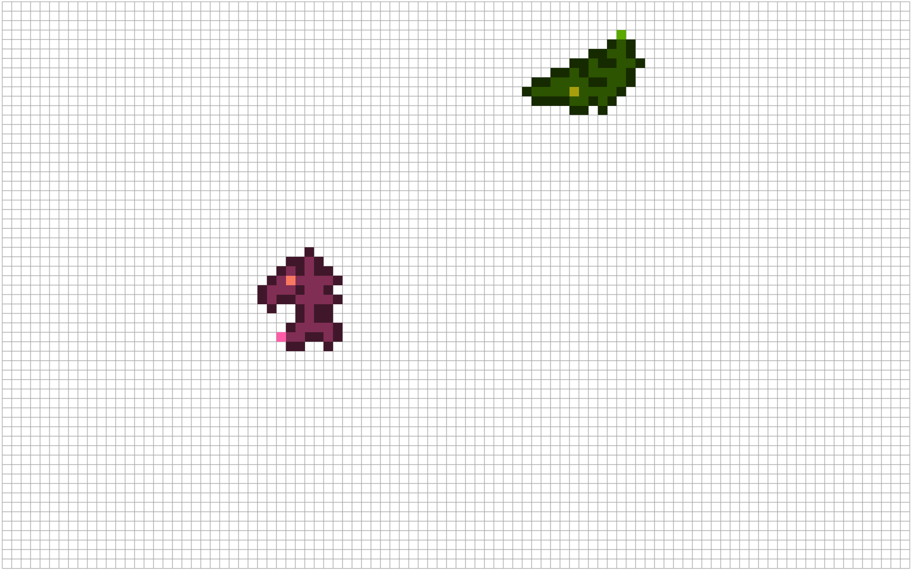
Exploration algorithms / decay of correlation



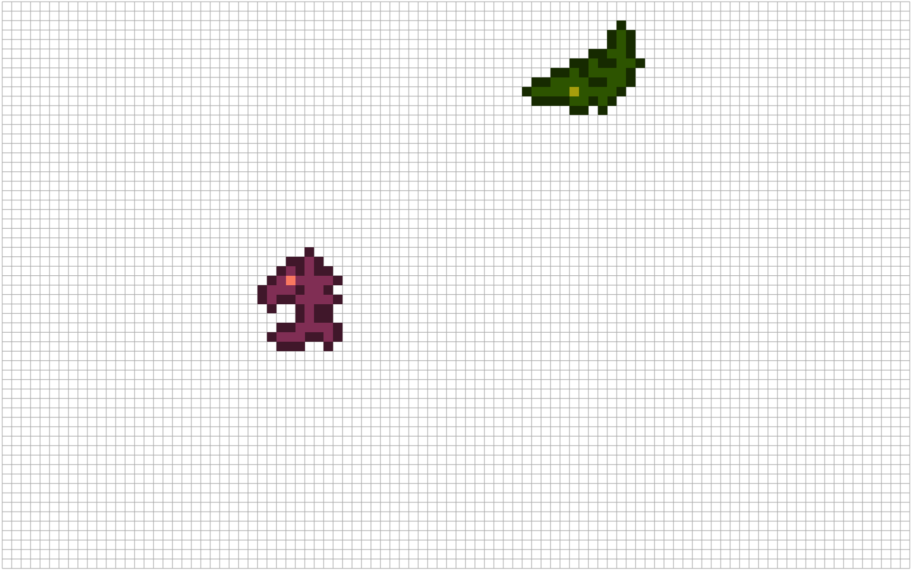
Exploration algorithms / decay of correlation



Exploration algorithms / decay of correlation



Exploration algorithms / decay of correlation



Locally tree-like

We need to calculate $\iota(U, \rho)$,

Locally tree-like

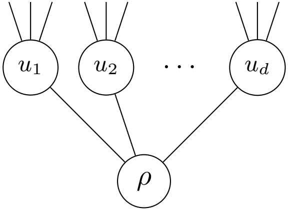
We need to calculate $\iota(U, \rho)$, but even $\iota(\mathbb{Z}^2)$ is still unknown...

Locally tree-like

We need to calculate $\iota(U, \rho)$, but even $\iota(\mathbb{Z}^2)$ is still unknown...
Let us therefore restrict ourselves to *locally tree-like* graph sequences, i.e., graph sequences for which (U, ρ) is almost surely a tree.

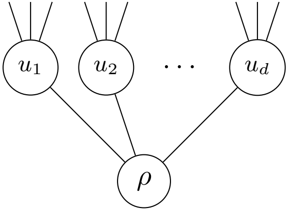
Locally tree-like

We need to calculate $\iota(U, \rho)$, but even $\iota(\mathbb{Z}^2)$ is still unknown...
Let us therefore restrict ourselves to *locally tree-like* graph sequences, i.e., graph sequences for which (U, ρ) is almost surely a tree.



Locally tree-like

We need to calculate $\iota(U, \rho)$, but even $\iota(\mathbb{Z}^2)$ is still unknown...
Let us therefore restrict ourselves to *locally tree-like* graph sequences, i.e., graph sequences for which (U, ρ) is almost surely a tree.



Assuming the children of ρ are roots to independent subtrees, and conditioning on the label of ρ, children of the *past* are roots to independent processes.

Systems of ordinary differential equations

Let (U, ρ) be a single-type branching process.

Systems of ordinary differential equations

Let (U, ρ) be a single-type branching process.

$$y(x) = \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \wedge \sigma_\rho < x)$$

Systems of ordinary differential equations

Let (U, ρ) be a single-type branching process.

$$\begin{aligned} y(x) &= \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \wedge \sigma_\rho < x) \\ &= x \cdot \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \mid \sigma_\rho < x) \\ &= \int_0^x \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \mid \sigma_\rho = z) dz \end{aligned}$$

Systems of ordinary differential equations

Let (U, ρ) be a single-type branching process.

$$\begin{aligned} y(x) &= \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \wedge \sigma_\rho < x) \\ &= x \cdot \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \mid \sigma_\rho < x) \\ &= \int_0^x \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \mid \sigma_\rho = z) dz \\ y'(x) &= \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \mid \sigma_\rho = x) \end{aligned}$$

Systems of ordinary differential equations

Let (U, ρ) be a single-type branching process.

$$\begin{aligned} y(x) &= \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \wedge \sigma_\rho < x) \\ &= x \cdot \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \mid \sigma_\rho < x) \\ &= \int_0^x \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \mid \sigma_\rho = z) dz \\ y'(x) &= \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \mid \sigma_\rho = x) \end{aligned}$$

Thus, if y is a unique solution of

$$y'(x) = \sum_{\ell \in \mathbb{N}} \mathbb{P}(\xi[< x] = \ell) \left(1 - \frac{y(x)}{x}\right)^\ell, \qquad y(0) = 0,$$

then, $\iota(U, \rho) = y(1)$.

Systems of ordinary differential equations

Let (U, ρ) be a single-type branching process.

$$\begin{aligned} y(x) &= \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \wedge \sigma_\rho < x) \\ &= x \cdot \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \mid \sigma_\rho < x) \\ &= \int_0^x \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \mid \sigma_\rho = z) dz \\ y'(x) &= \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \mid \sigma_\rho = x) \end{aligned}$$

Thus, if y is a unique solution of

$$y'(x) = \sum_{\ell \in \mathbb{N}} \mathbb{P}(\xi[< x] = \ell) \left(1 - \frac{y(x)}{x}\right)^\ell, \qquad y(0) = 0,$$

then, $\iota(U, \rho) = y(1)$.

Systems of ordinary differential equations

Let (U, ρ) be a (simple) multitype branching process.

$$\begin{aligned} y_k(x) &= \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \wedge \sigma_\rho < x \mid \tau = k) \\ &= x \cdot \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \mid \sigma_\rho < x, \tau = k) \\ &= \int_0^x \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \mid \sigma_\rho = z, \tau = k) dz \\ y_k'(x) &= \mathbb{P}(\rho \in \mathbf{I}(U[\mathcal{P}_\rho]) \mid \sigma_\rho = x, \tau = k) \end{aligned}$$

Thus, if y is a unique solution of

$$y_k'(x) = \sum_{\ell \in \mathbb{N}^{\mathcal{T}}} \prod_{j \in \mathcal{T}} \mathbb{P}\Big(\xi^{k \to j}[< x] = \ell_j\Big) \left(1 - \frac{y_j(x)}{x}\right)^{\ell_j}, \qquad y_k(0) = 0,$$

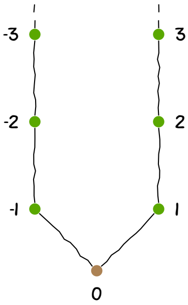
then, $\iota(U, \rho) = \mathbb{E}[y_k(1)]$.

Application: paths and cycles

P_n and C_n converge locally to $\mathbb{Z}$, which can be thought of as a 2-type branching process.

Application: paths and cycles

P_n and C_n converge locally to $\mathbb{Z}$, which can be thought of as a 2-type branching process.

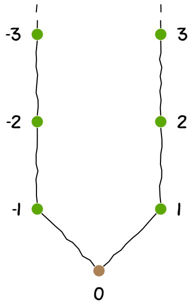


Application: paths and cycles

P_n and C_n converge locally to $\mathbb{Z}$, which can be thought of as a 2-type branching process.

$$y_b'(x) = 1 - y_b(x) \implies y_b(x) = 1 - e^{-x},$$

$$y_r'(x) = (1 - y_b(x))^2 = e^{-2x} \implies y_r(x) = \frac{1}{2}\left(1 - e^{-2x}\right).$$



Application: paths and cycles

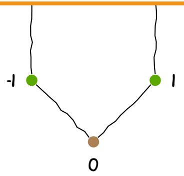
P_n and C_n converge locally to $\mathbb{Z}$, which can be thought of as a 2-type branching process.

$$y'_b(x) = 1 - y_b(x) \implies y_b(x) = 1 - e^{-x},$$

$$y'_r(x) = (1 - y_b(x))^2 = e^{-2x} \implies y_r(x) = \frac{1}{2}\left(1 - e^{-2x}\right).$$

Thus

$$\iota(P_n),\ \iota(C_n) \to \iota(\mathbb{Z}) = y_2(1) = \frac{1}{2}\left(1 - e^{-2}\right).$$



Application: paths and cycles

P_n and C_n converge locally to $\mathbb{Z}$, which can be thought of as a 2-type branching process.

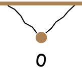
$$y_b'(x) = 1 - y_b(x) \implies y_b(x) = 1 - e^{-x},$$

$$y_r'(x) = (1 - y_b(x))^2 = e^{-2x} \implies y_r(x) = \frac{1}{2}\big(1 - e^{-2x}\big).$$

Thus

$$\iota(P_n),\ \iota(C_n) \to \iota(\mathbb{Z}) = y_2(1) = \frac{1}{2}\big(1 - e^{-2}\big).$$

$$\alpha(P_n)/n,\ \alpha(C_n)/n \to 1/2.$$



Application: binomial random graphs

Easy fact: $G(n, d/n)$ converges locally to the Pois(d) branching process.

$$y'(x) = \sum_{\ell=0}^{\infty} \frac{(dx)^\ell}{e^{dx}\ell!}\left(1 - \frac{y(x)}{x}\right)^\ell = e^{-dy(x)}.$$

hence $y(x) = \log(1 + dx)/d$.

Application: binomial random graphs

Easy fact: $G(n, d/n)$ converges locally to the Pois(d) branching process.

$$y'(x) = \sum_{\ell=0}^{\infty} \frac{(dx)^\ell}{e^{dx}\ell!}\left(1 - \frac{y(x)}{x}\right)^\ell = e^{-dy(x)}.$$

hence $y(x) = \log(1 + dx)/d$. Thus

$$\iota(G(n, d/n)) \to \iota(\mathcal{T}_d) = y(1) = \frac{\log(1+d)}{d}.$$

Application: binomial random graphs

Easy fact: $G(n, d/n)$ converges locally to the Pois(d) branching process.

$$y'(x) = \sum_{\ell=0}^{\infty} \frac{(dx)^\ell}{e^{dx}\ell!}\left(1 - \frac{y(x)}{x}\right)^\ell = e^{-dy(x)}.$$

hence $y(x) = \log(1 + dx)/d$. Thus

$$\iota(G(n, d/n)) \to \iota(\mathcal{T}_d) = y(1) = \frac{\log(1+d)}{d}.$$

$$\alpha(G(n, d/n))/n \to 2\log d/d \cdot (1 + o_d(1)).$$

Size-biased Galton–Watson branching processes

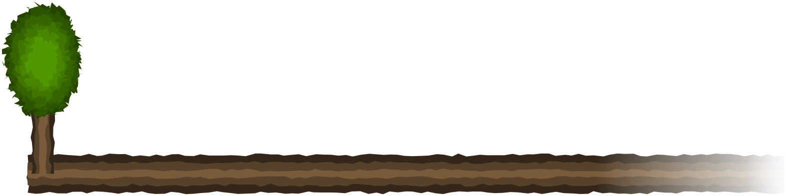
Grimmett '80: the sequence of uniform random trees converges locally to the *size-biased Galton–Watson* Pois(1) tree.

Size-biased Galton–Watson branching processes

Grimmett '80: the sequence of uniform random trees converges locally to the *size-biased Galton–Watson* Pois(1) tree.

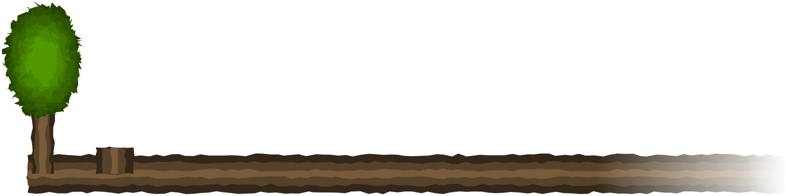
Size-biased Galton–Watson branching processes

Grimmett '80: the sequence of uniform random trees converges locally to the *size-biased Galton–Watson* Pois(1) tree.



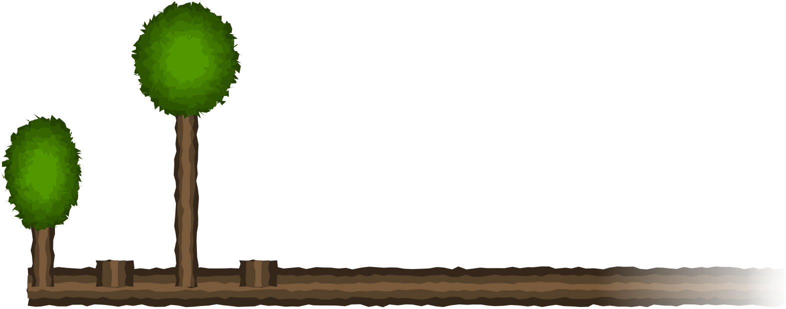
Size-biased Galton–Watson branching processes

Grimmett '80: the sequence of uniform random trees converges locally to the *size-biased Galton–Watson* Pois(1) tree.



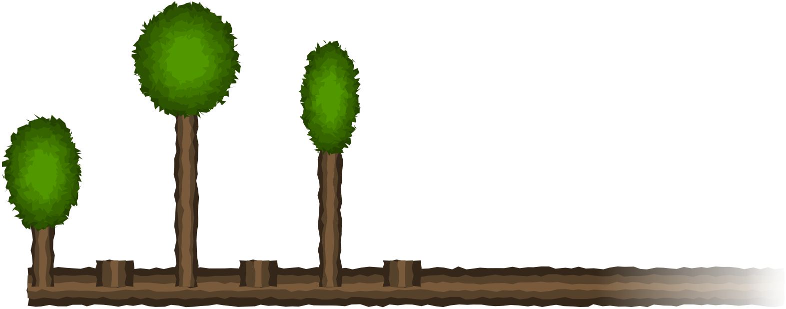
Size-biased Galton–Watson branching processes

Grimmett '80: the sequence of uniform random trees converges locally to the *size-biased Galton–Watson* Pois(1) tree.



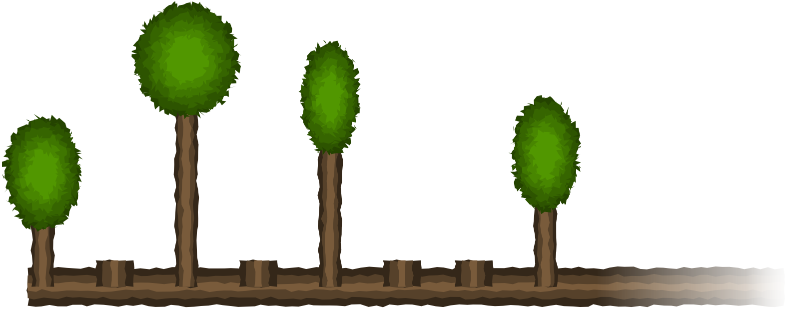
Size-biased Galton–Watson branching processes

Grimmett '80: the sequence of uniform random trees converges locally to the *size-biased Galton–Watson* Pois(1) tree.



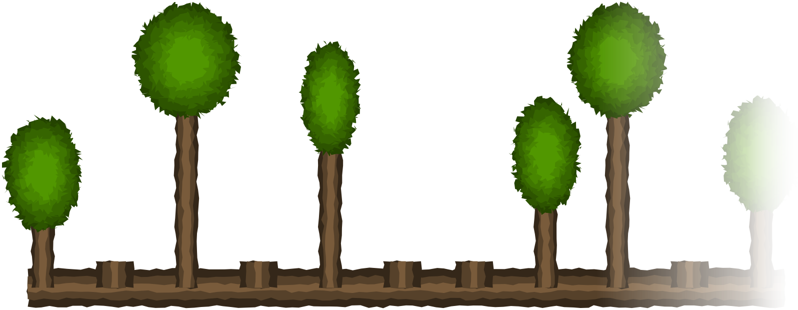
Size-biased Galton–Watson branching processes

Grimmett '80: the sequence of uniform random trees converges locally to the *size-biased Galton–Watson* Pois(1) tree.



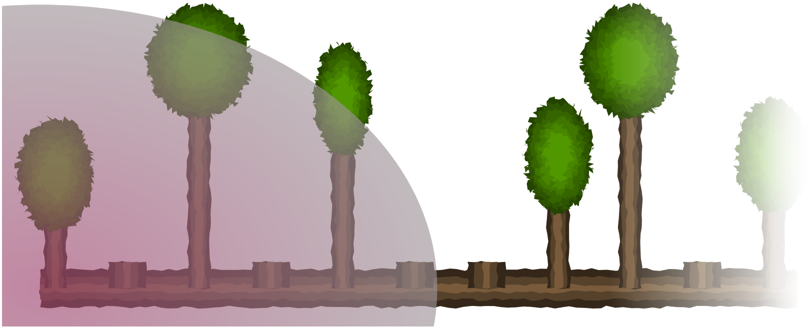
Size-biased Galton–Watson branching processes

Grimmett '80: the sequence of uniform random trees converges locally to the *size-biased Galton–Watson* Pois(1) tree.



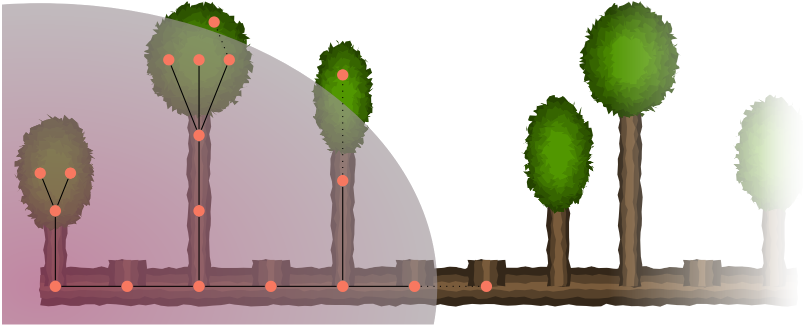
Size-biased Galton–Watson branching processes

Grimmett '80: the sequence of uniform random trees converges locally to the *size-biased Galton–Watson* Pois(1) tree.



Size-biased Galton–Watson branching processes

Grimmett '80: the sequence of uniform random trees converges locally to the *size-biased Galton–Watson* Pois(1) tree.



Application: uniform random trees

Let s be the type of a vertex on the spine, and t be the type of a vertex on one of the hanging trees. We have already seen

$$y_{\mathsf{t}}(x) = \log(1 + x),$$

Application: uniform random trees

Let s be the type of a vertex on the spine, and t be the type of a vertex on one of the hanging trees. We have already seen

$$y_{\mathsf{t}}(x) = \log(1+x),$$

and

$$y_{\mathsf{s}}'(x) = (1 - y_{\mathsf{s}}(x))y_{\mathsf{t}}'(x) = \frac{1 - y_{\mathsf{s}}(x)}{1+x},$$

hence $y_{\mathsf{s}}(x) = 1 - (1+x)^{-1}$, and we get

$$\iota(T_n) \to \iota(\hat{\mathcal{T}}_1) = y_{\mathsf{s}}(1) = \frac{1}{2}.$$

Application: uniform random trees

Let s be the type of a vertex on the spine, and t be the type of a vertex on one of the hanging trees. We have already seen

$$y_{\mathsf{t}}(x) = \log(1+x),$$

and

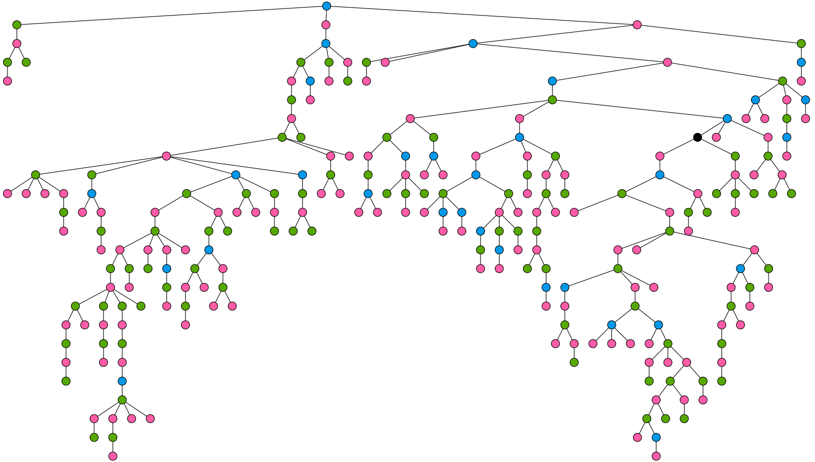
$$y_{\mathsf{s}}'(x) = (1 - y_{\mathsf{s}}(x)) y_{\mathsf{t}}'(x) = \frac{1 - y_{\mathsf{s}}(x)}{1+x},$$

hence $y_{\mathsf{s}}(x) = 1 - (1+x)^{-1}$, and we get

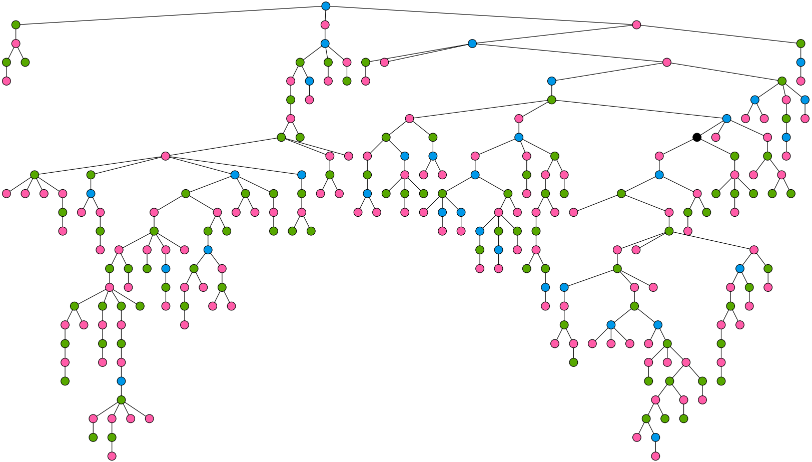
$$\iota(T_n) \to \iota(\hat{\mathcal{T}}_1) = y_{\mathsf{s}}(1) = \frac{1}{2}.$$

$$\alpha(T_n)/n \to W_0(1) \approx 0.56714...$$

Simulations don't lie

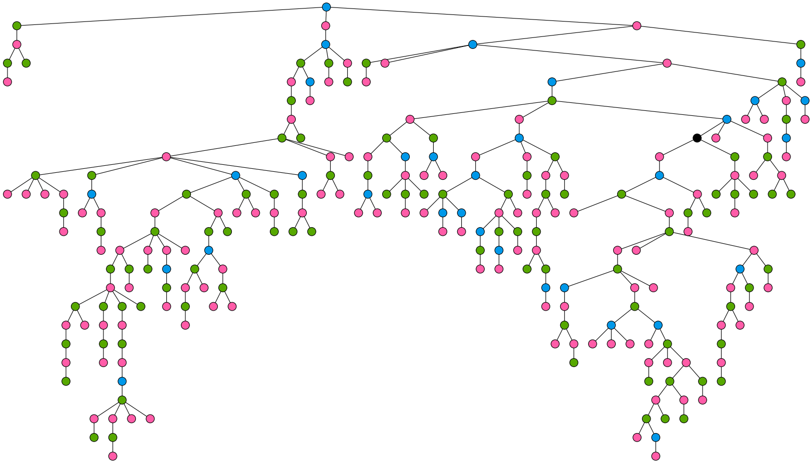


Simulations don't lie



red: 125 (50%), green: 92 ($\approx$ 37%), blue: 32 ($\approx$ 13%), black: 1

Simulations don't lie (but I do)



red: 125 (50%), green: 92 ($\approx$ 37%), blue: 32 ($\approx$ 13%), black: 1

Greedy independence ratio — results

Flory '39, Page '59 $\iota(P_n) \to \frac{1}{2}(1 - e^{-2})$

McDiarmid '84 $\iota(G(n, d/n)) \to \log(1 + d)/d$

Wormald '95 $\iota(\mathcal{G}_{n,d}) \to \frac{1}{2}\big(1 - (d-1)^{-2/(d-2)}\big)$

Lauer–Wormald '07 (d-regular graphs with girth $\to \infty$)

Greedy independence ratio — results

Flory '39, Page '59	$\iota(P_n) \to \frac{1}{2}(1 - e^{-2})$	✓
McDiarmid '84	$\iota(G(n, d/n)) \to \log(1 + d)/d$	
Wormald '95	$\iota(\mathcal{G}_{n,d}) \to \frac{1}{2}\big(1 - (d-1)^{-2/(d-2)}\big)$	
Lauer–Wormald '07	(d-regular graphs with girth $\to \infty$)	

Greedy independence ratio — results

Flory '39, Page '59	$\iota(P_n) \to \frac{1}{2}(1 - e^{-2})$	✓
McDiarmid '84	$\iota(G(n, d/n)) \to \log(1 + d)/d$	✓
Wormald '95	$\iota(\mathcal{G}_{n,d}) \to \frac{1}{2}\big(1 - (d-1)^{-2/(d-2)}\big)$	
Lauer–Wormald '07	(d-regular graphs with girth $\to \infty$)	

Greedy independence ratio — results

Flory '39, Page '59	$\iota(P_n) \to \frac{1}{2}(1 - e^{-2})$	✓
McDiarmid '84	$\iota(G(n, d/n)) \to \log(1 + d)/d$	✓
Wormald '95	$\iota(\mathcal{G}_{n,d}) \to \frac{1}{2}\big(1 - (d-1)^{-2/(d-2)}\big)$	✓
Lauer–Wormald '07	(d-regular graphs with girth $\to \infty$)	

Greedy independence ratio — results

Flory '39, Page '59	$\iota(P_n) \to \frac{1}{2}(1 - e^{-2})$	✓
McDiarmid '84	$\iota(G(n, d/n)) \to \log(1 + d)/d$	✓
Wormald '95	$\iota(\mathcal{G}_{n,d}) \to \frac{1}{2}\big(1 - (d-1)^{-2/(d-2)}\big)$	✓
Lauer–Wormald '07	(d-regular graphs with girth $\to \infty$)	✓

Greedy independence ratio — results

Flory '39, Page '59	$\iota(P_n) \to \frac{1}{2}(1 - e^{-2})$	✓
McDiarmid '84	$\iota(G(n, d/n)) \to \log(1+d)/d$	✓
Wormald '95	$\iota(\mathcal{G}_{n,d}) \to \frac{1}{2}\left(1 - (d-1)^{-2/(d-2)}\right)$	✓
Lauer–Wormald '07	(d-regular graphs with girth $\to \infty$)	✓
KMMS '20	$\iota(T_n) \to \frac{1}{2}$	✸

Greedy independence ratio — results

Flory '39, Page '59	$\iota(P_n) \to \frac{1}{2}(1 - e^{-2})$	✓
McDiarmid '84	$\iota(G(n, d/n)) \to \log(1+d)/d$	✓
Wormald '95	$\iota(\mathcal{G}_{n,d}) \to \frac{1}{2}\big(1 - (d-1)^{-2/(d-2)}\big)$	✓
Lauer–Wormald '07	(d-regular graphs with girth $\to \infty$)	✓
KMMS '20	$\iota(T_n) \to \frac{1}{2}$	✷
	(same for functional digraphs)	✷

Paths are the worst trees

Paths are the worst trees

- $\iota(P_n) \to \frac{1}{2}\left(1 - e^{-2}\right) \approx 0.43233...$

Paths are the worst trees

- $\iota(P_n) \to \frac{1}{2}\left(1 - e^{-2}\right) \approx 0.43233...$
- $\iota(S_n) \to 1$

Paths are the worst trees

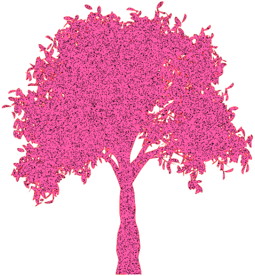
- $\iota(P_n) \to \frac{1}{2}\left(1 - e^{-2}\right) \approx 0.43233...$
- $\iota(S_n) \to 1$

$\leq$

Paths are the worst trees

- $\iota(P_n) \to \frac{1}{2}\left(1 - e^{-2}\right) \approx 0.43233...$
- $\iota(S_n) \to 1$

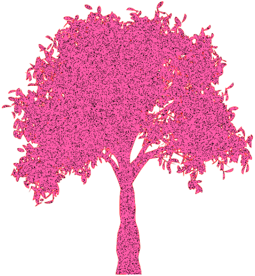
$\leq$



Paths are the worst trees

- $\iota(P_n) \to \frac{1}{2}\left(1 - e^{-2}\right) \approx 0.43233...$
- $\iota(S_n) \to 1$

$\leq$

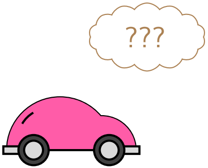


Theorem (Krivelevich, Mészáros, M., Shikhelman '20)

If T is a tree on n vertices, then $\mathbb{E}[\iota(P_n)] \leq \mathbb{E}[\iota(T)]$.

What's next?

- Graph sequences that are not locally tree-like
- Better/other local rules
- Other colours



Thank You!