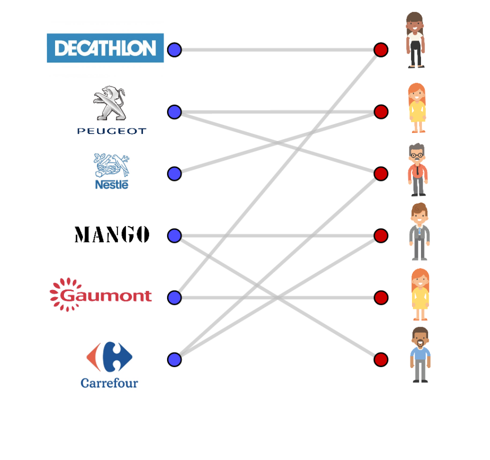
Online Matching in Random Bipartite Graphs

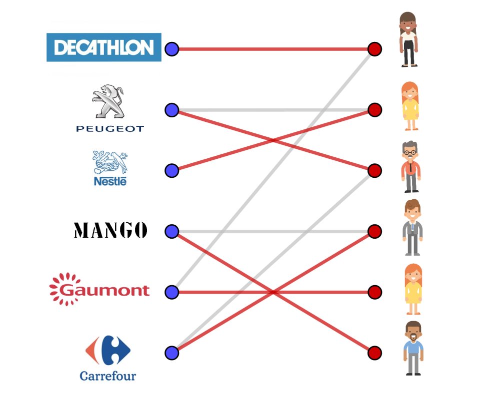
Flore Sentenac, joint work with Nathan Noiry, Vianney Perchet, Laurent Ménard and Matthieu Lerasle.

Motivation: Dynamic allocation

Ad - User allocation



Ad - User allocation

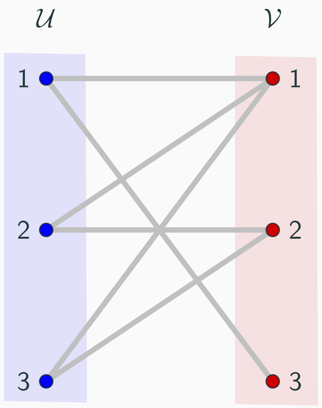


Problem definition

Matching on a Bipartite graph

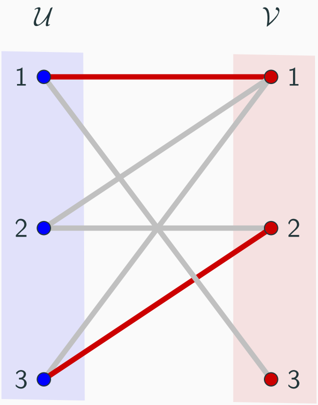
Graph $\mathcal{G} = ((\mathcal{U}, \mathcal{V}), \mathcal{E})$ bipartite if:

- Set of vertices is $\mathcal{U} \cup \mathcal{V}$,
- Only edges between $\mathcal{U}$ and $\mathcal{V}$: $\mathcal{E} \subset \mathcal{U} \times \mathcal{V}$.



Matching on a Bipartite graph

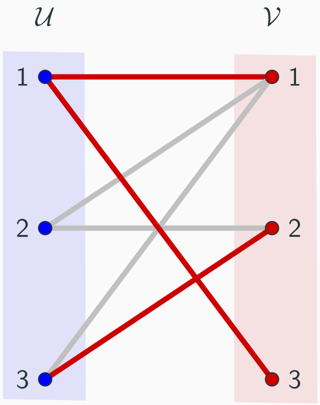
A matching is a set of edges with no common vertices.



A matching

Matching on a Bipartite graph

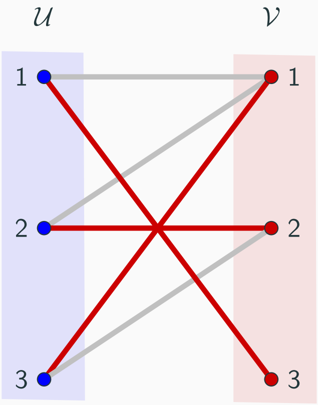
A matching is a set of edges with no common vertices.



Not a matching

Matching on a Bipartite graph

A matching is a set of edges with no common vertices.

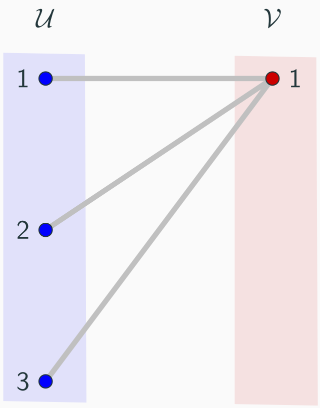


A maximum matching

Online Matching

For $t = 1, ..., |\mathcal{V}|$:

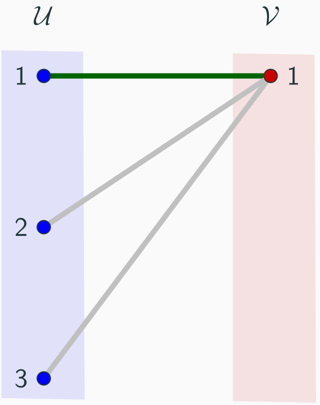
- v_t arrives along with its edges
- the algorithm can match it to a free vertex in $\mathcal{U}$
- the decision is final



Online Matching

For $t = 1, ..., |\mathcal{V}|$:

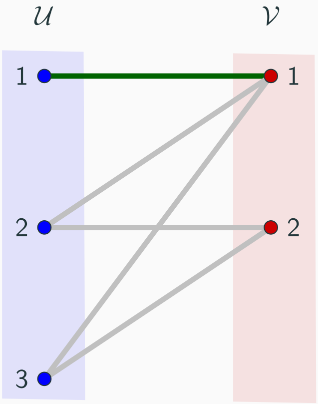
- v_t arrives along with its edges
- the algorithm can match it to a free vertex in $\mathcal{U}$
- the decision is final



Online Matching

For $t = 1, ..., |\mathcal{V}|$:

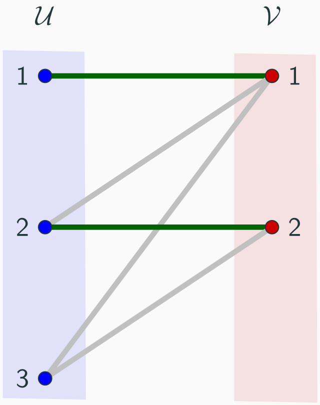
- v_t arrives along with its edges
- the algorithm can match it to a free vertex in $\mathcal{U}$
- the decision is final



Online Matching

For $t = 1, ..., |\mathcal{V}|$:

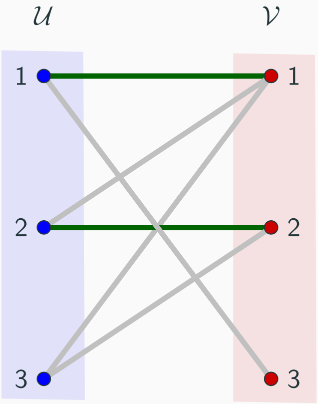
- v_t arrives along with its edges
- the algorithm can match it to a free vertex in $\mathcal{U}$
- the decision is final



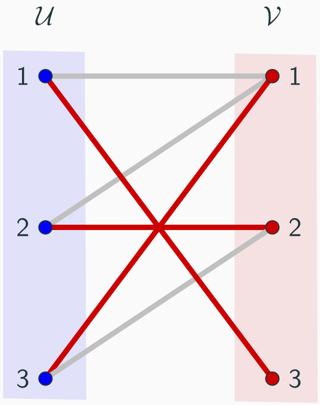
Online Matching

For $t = 1, ..., |\mathcal{V}|$:

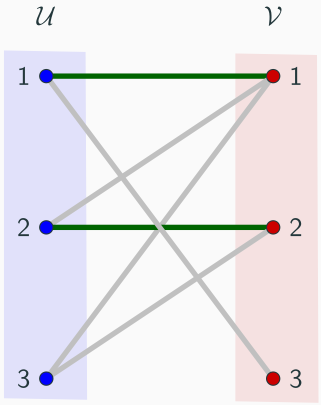
- v_t arrives along with its edges
- the algorithm can match it to a free vertex in $\mathcal{U}$
- the decision is final



Evaluating the performance



$\text{OPT}(\mathcal{G}) = 3$



$\text{ALG}(\mathcal{G}) = 2$

Competitive ratio

Definition

The competitive ratio is defined as:

$$\text{C.R.} = \min_{\mathcal{G}} \frac{\mathbb{E}[\text{ALG}(\mathcal{G})]}{\text{OPT}(\mathcal{G})}$$

Note that $0 \leq \text{C.R.} \leq 1$, and the higher the better.

The usual frameworks

- **Adversarial** (Adv): $\mathcal{G}$ can be any graph, the vertices of $\mathcal{V}$ arrive in any order.
- **Random Order** (RO): $\mathcal{G}$ can be any graph, the vertices of $\mathcal{V}$ arrive in random order.
- **Stochastic** (IID): The vertices of $\mathcal{V}$ are drawn iid from a distribution. (precise definition given latter)

The usual frameworks

- **Adversarial** (Adv): $\mathcal{G}$ can be any graph, the vertices of $\mathcal{V}$ arrive in any order.
- **Random Order** (RO): $\mathcal{G}$ can be any graph, the vertices of $\mathcal{V}$ arrive in random order.
- **Stochastic** (IID): The vertices of $\mathcal{V}$ are drawn iid from a distribution. (precise definition given latter)

$$\text{C.R.(Adv)} \leq \text{C.R.(RO)} \leq \text{C.R.(IID)}$$

GREEDY in the Adversarial framework

Algorithm 1: GREEDY Algorithm

```
for t = 1, .., |V| do
    Match v_t to any free neighbor;
end
```

Theorem

In the Adversarial setting,

$$\text{C.R.(GREEDY)} \geq \frac{1}{2}.$$

RANKING in the Adversarial framework

Algorithm 2: RANKING Algorithm

```
Draw a random permutation π;
for i = 1, .., |U| do
    Assign to u_i rank π(i);
end
for t = 1, .., |V| do
    Match v_t to its lowest ranked free neighbor;
end
```

Theorem (Karp, Vazirani, Vazirani, 1990)

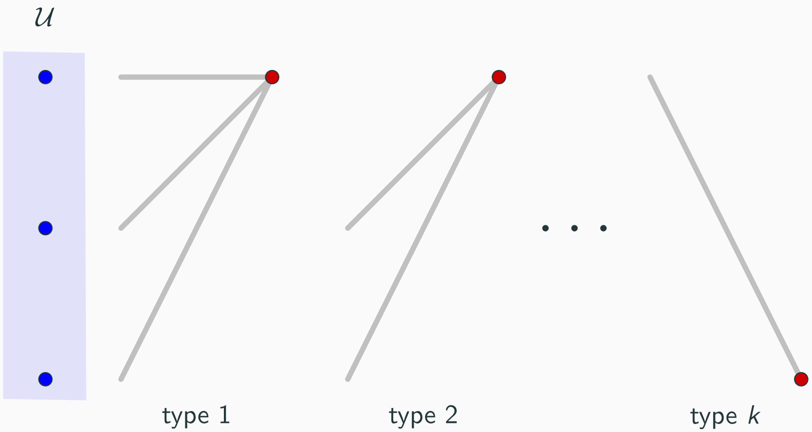
In the Adversarial setting,

$$\text{C.R.(RANKING)} \geq 1 - \frac{1}{e}.$$

Note : $1 - \frac{1}{e} \approx 0.63$

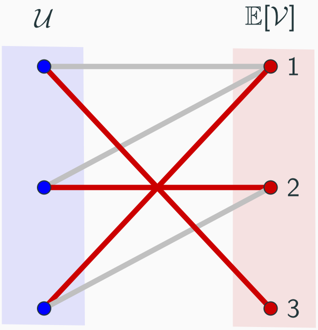
Known IID

Model : There is a distribution over k fixed known types from which the incoming vertices are drawn i.i.d..

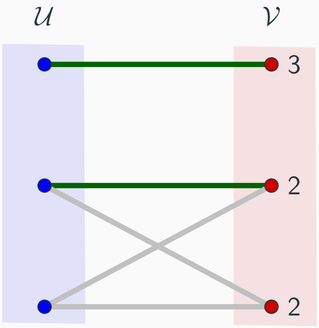


A first naive solution :

- Compute an optimal matching on the expected graph (assume integral expected arrival rates fro simplicity)
- Match the first incoming vertex of each type according to that matching.



Guide



Constructed Matching

$$CR = 1 - \frac{1}{e}$$

A better one : Compute an alternative matching on the expected graph and use it as a graph in case of a second arrival.

Theorem (Jaillet, Lu, 2013)

In the Known IID model, the 2-suggested matching algorithms as a CR lower bounded as:

$$C.R. \geq 0.702.$$

Main issues :

- CR upper bounded by 0.823,
- No flexibility in the model.

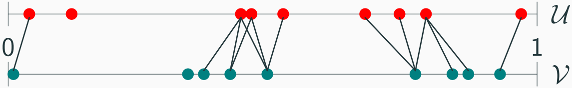
Online Matching in Random Graphs: The 1-D Geometric Model

1-D Random Geometric graph

Model : Random geometric graph Geom($\mathcal{U}, \mathcal{V}, c$):

- the points in $\mathcal{U}$ are N points drawn iid uniformly in $[0, 1]$,
- the points in $\mathcal{V}$ are N points drawn iid uniformly in $[0, 1]$,
- there is an edge between $u \in \mathcal{U}$ and $v \in \mathcal{V}$ iif:

$$|u - v| \leq \frac{c}{N}.$$

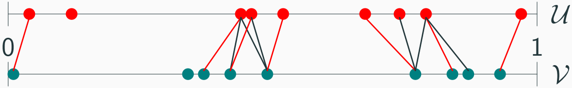


Motivation: the position of the points $\sim$ features.

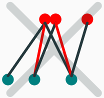
Offline Maximum matching

Proposition

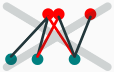
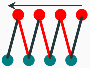
The algorithm matching free vertices from left to right produces a maximum matching.



Alternative formulation: the algorithm creates no augmenting path.



Proof: Consider the augmenting path of shortest length.

- No edges in the matching cross: 
- Points on both sides of the path have decreasing coordinates: 
- No position possible for the end point of the path:

The point is matched by the algorithm.

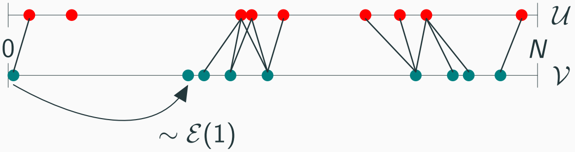
There exists a path of shorter length.

Size of the Maximum Matching

Step 1: Modify the graph generating process.

Random geometric graph Geom'$(\mathcal{U}, \mathcal{V}, c)$:

- $\mathcal{U}$ and $\mathcal{V}$ drawn from a Poisson Point Process of intensity 1 in $[0, N]$,
- there is an edge between $u \in \mathcal{U}$ and $v \in \mathcal{V}$ iif: $|u - v| \leq c$.



Expected matching sizes in the two model

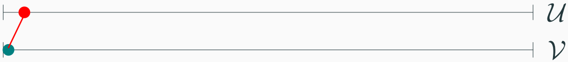
With $\gamma^*(c, N)$ and $M^*(c, N)$ the expected sizes of the matchings in the original and modified models:

$$|\gamma^*(c, N) - M^*(c, N)| \leq 4(1 + \sqrt{N \ln N}).$$

Step 2: Generate the graph together with the matching.

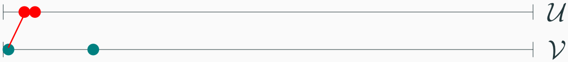
Three situations possible:

- Successful match !



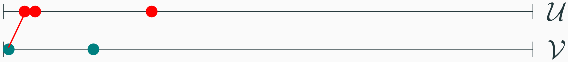
$\implies$ generate next points in $\mathcal{U}$ and $\mathcal{V}$.

- Last point in $\mathcal{U}$ too far behind.



$\implies$ generate next point in $\mathcal{U}$.

- Last point in $\mathcal{V}$ too far behind.



$\implies$ generate next point in $\mathcal{V}$.

The size of the gap between the two last generated points at time t is a random walk $\psi(t)$ s.t. :

$$\psi(t+1) - \psi(t) \sim \begin{cases} \text{Lap}(0,1) \text{ if } |\psi(t)| \leq c \\ \text{Exp}(1) \text{ if } \psi(t) \leq -c \\ -\text{Exp}(1) \text{ if } \psi(t) \geq c \end{cases}$$

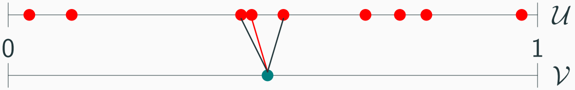
Proposition

$$\lim_{N \to \infty} \frac{M^*(c, N)}{N} = \frac{c}{c + \frac{1}{2}}.$$

Online Matching in the 1-D Geometric Model

Match to the closest point algorithm

The incoming point is matched to its closest available neighbor.



Our Result

Theorem (S., Noiry, Perchet, Ménard, Lerasle, 2022)

Let $\kappa(c, N)$ be the size of the matching obtained by *match to the closest point* algorithm on $G(\mathcal{X}, \mathcal{Y}, c/N)$. We have

$$\kappa(c, N) \xrightarrow[N \to +\infty]{\mathsf{P}} 1 - \int_0^{+\infty} f(x, 1)dx$$

with $f(x, t)$ the solution of the following differential equation

$$\frac{\partial f(x,t)}{\partial t} = -\min(x, 2c)f(x,t) - \int_0^{+\infty} \frac{\min(x', 2c)f(x',t)f(x,t)}{\int_0^{+\infty} f(x',t)dx'}dx' + \frac{1}{\int_0^{+\infty} f(x',t)dx'} \int_0^{x} \min(x', 2c)f(x',t)f(x-x',t)dx'$$

with the following initial conditions

$$f(x, 0) = e^{-x}.$$

Experimental results

Simulations with $c = 4$.

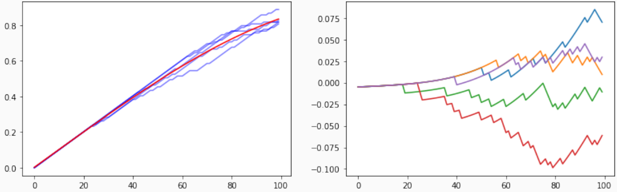


Figure 1: Difference between the theoretical performances and simulated performances of the GREEDY algorithm on the geometric graph ($c = 4$) on 5 independent runs, with $N = 100$.

Key to obtaining the PDEs: Finding the right quantities to track.

The matching algorithm is studied on a modified graph:

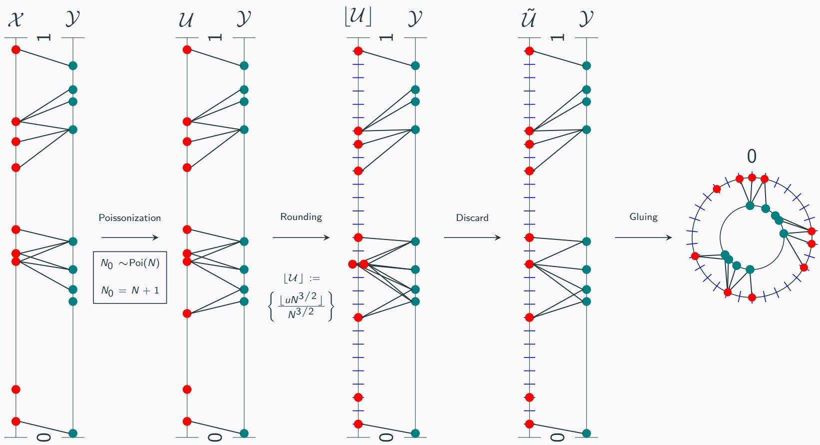


Figure 2: Graph Rounding

We track the value of the gaps between the remaining free vertices

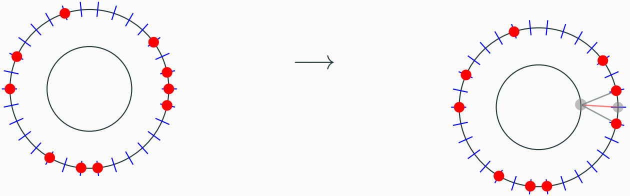
N_t is the number of free vertices at iteration t.

$u_t(i)$ is the coordinate of the i^{th} free vertex, with the (vertices enumerated according to their coordinates).

For $\ell \in [N^{3/2}]$, define

$$F_N(\ell, t) := \left| \left\{ N \left(u_t(i+1) - u_t(i) \right) = \frac{\ell}{\sqrt{N}} \,\middle|\, i \in [N_t] \right\} \right|,$$

On an example



$$F_9(1,0) = 3$$

$$F_9(2,0) = 3$$

$$F_9(4,0) = F_9(5,0) = F_9(6,0) = 1$$

For all other $\ell \in [30]$:

$$F_9(\ell,0) = 0$$

$$F_9(1,1) = 1$$

$$F_9(2,1) = 4$$

$$F_9(4,1) = F_9(5,1) = F_9(6,1) = 1$$

For all other $\ell \in [30]$:

$$F_9(\ell,1) = 0$$

→ Related to the number of matched vertices at time t,

$$M(t) = N_0 - \sum_{\ell} F_N(\ell, t).$$

→ There exists Φ such that:

$$\mathbb{E}[F_N(\ell, t+1) - F_N(\ell, t) \,|\, \mathcal{F}_t] = \Phi_N \left(F_N(0, t), \ldots, F_N(N^{3/2}, t) \right) + o(1).$$

Differences can be seen as discrete derivatives...

The differential equation method

For all $s \in (0,1)$, w.h.p.:

$$\left| \sum_{\ell} \frac{F(\ell, \lfloor sT \rfloor)}{N} - f_N(\ell, s) \right| \leq O(N^{-1/8}),$$

with $\forall \ell$:

$$\frac{\partial f_N(\ell, t)}{\partial t} = \Phi_N \left(f_N(0,t), \ldots, \ldots, f_N(N^{3/2}, t), \ell \right).$$

with the initial conditions:

$$f_N(\ell, 0) = \frac{1}{\sqrt{N}} e^{-\frac{\ell}{\sqrt{N}}}.$$

Differential equations for random processes and random graphs, Wormald ; 1995.

Last step

For any $t \in [0, 1]$, we have:

$$|| \, f(., t) - f_N(., t) \, ||_{L_1} \leq \frac{10}{\sqrt{N}}.$$

with f the function defined in the theorem.

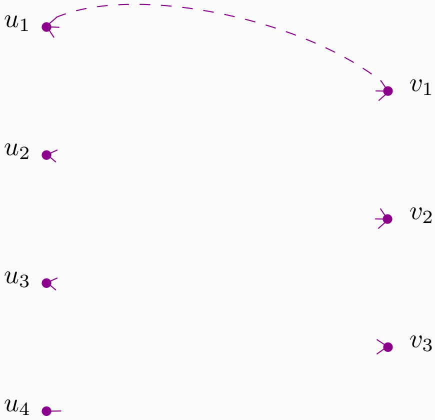
The Configuration Model

The Configuration Model

Introduced by Bollobás in 1980.

Consider two degree sequences $\mathsf{d}^U = (d_1^U, \ldots, d_N^U)$, $\mathsf{d}^\mathsf{V} = (d_1^V, \ldots, d_T^V)$.

The associated bipartite configuration model $\mathrm{CM}(\mathsf{d}^U, \mathsf{d}^V)$ is obtained through a uniform pairing of the half-edges.



Our result

- $\mathcal{M}(s)$: matching obtained after seeing a proportion s of V-vertices.
- Generating series:

$$\phi_U(s) := \sum_{i\geq 0} \pi_U(i) s^i \qquad \text{and} \qquad \phi_V(s) := \sum_{i\geq 0} \pi_V(i) s^i.$$

Theorem (S.,Noiry, Perchet, 2021)

Let G be the unique solution of the following ordinary differential equation:

$$G'(s) = \frac{1 - \phi_V\left(1 - \frac{1}{\mu_U}\phi_U'\left(1 - G(s)\right)\right)}{\frac{\mu_V}{\mu_U}\phi_U'(1 - G(s))}; \quad G(0) = 0.$$

Then, the following convergence holds in probability:

$$\frac{|\mathcal{M}(s)|}{N} \xrightarrow[N\to+\infty]{\mathbb{P}} 1 - \phi_U(1 - G(s)).$$

The d-regular case

Take $\pi_U = \pi_V = \delta_d$: all vertices have degree d.

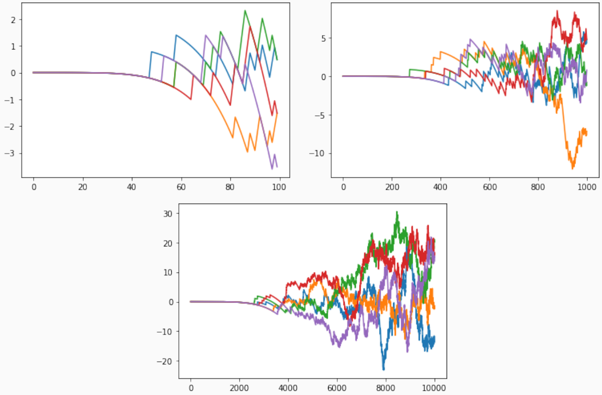


Figure 3: Difference between the theoretical performances and simulated performances of the GREEDY algorithm on the d-regular graph ($d = 4$) on 5 independent runs, with $N = 100, 1000, 10000$.

Thank you!

Evolution Law

Define:

$$M_N(\ell_-, \ell_+, t) := \left| \left\{ (u_t(i+1) - u_t(i)) = \frac{\ell_-}{N^{3/2}} \text{ and } (u_t(i+1) - u_t(i)) = \frac{\ell_+}{N^{3/2}} \right\} \right|.$$

And $\mathcal{F}_t$ the filtration associated with the values $(F_N(\ell, t'))_{\ell, t' \leq t}$ up to time t.

Lemma

For all $t \in [N]$, for all $\ell_-, \ell_+ \in (N^{3/2})^2$,

$$\mathbb{E}\left[M_N(\ell_-, \ell_+, t) \middle| \mathcal{F}_t \right] = \mathbb{1}\{\ell_- \neq \ell_+\} \frac{F_N(\ell_+, t) F_N(\ell_-, t)}{N_t - 1} + \mathbb{1}\{\ell_- = \ell_+\} \frac{F_N(\ell_-, t)(F_N(\ell_+, t) - 1)}{N_t - 1}.$$