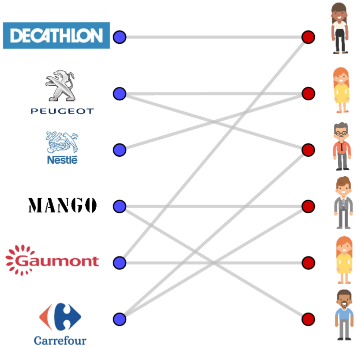
Online Bipartite Matching in Random Graphs

CIRM 2023

F. Sentenac, with V. Perchet, N. Noiry, L. Ménard and M. Lerasle.

Ad allocation on the internet

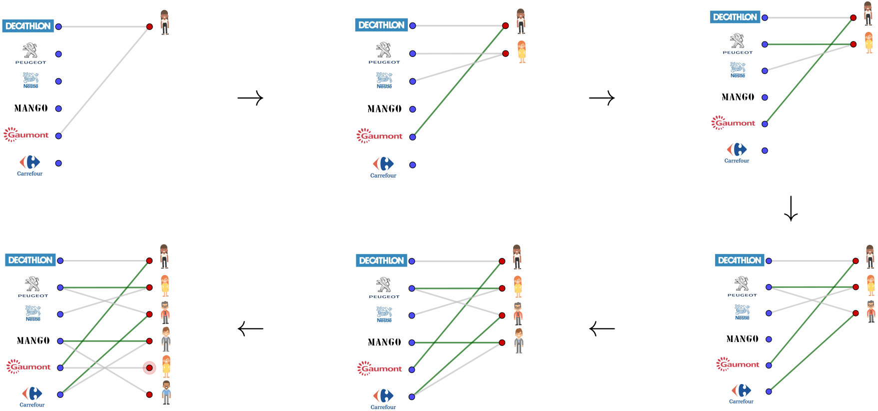


For the advertizing company:

- Different companies, each with a limited budget
- Ad-User compatibility based on criteria such as visited website, location...

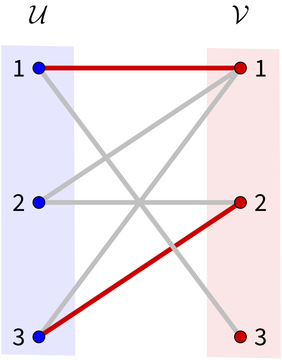
$\rightarrow$ Ad-User bipartite graph

Users are revealed sequentially and the matching is built on the fly: Online Matching



Matching on a Bipartite graph

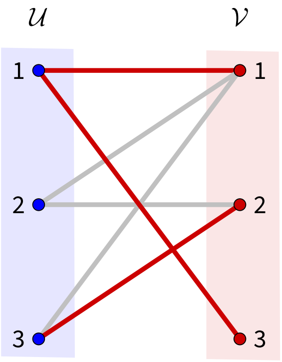
A matching is a set of edges with no common vertices.



A matching

Matching on a Bipartite graph

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

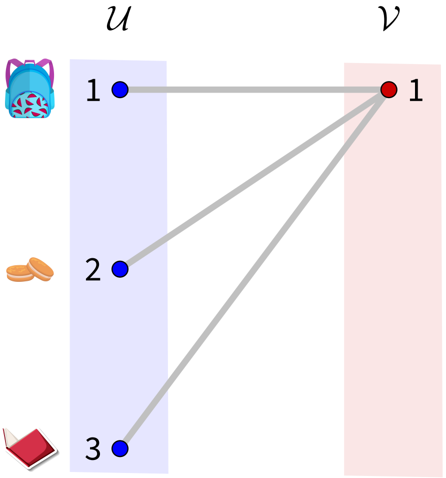


Not a matching

Online Matching

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

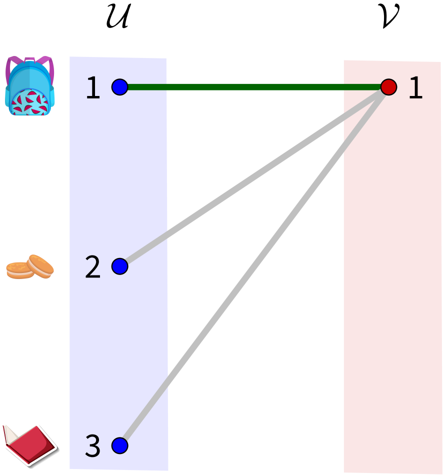
- v_t arrives along with its edges
- the algorithm can match it to a free vertex in $\mathcal{U}$
- the decision is final



Online Matching

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

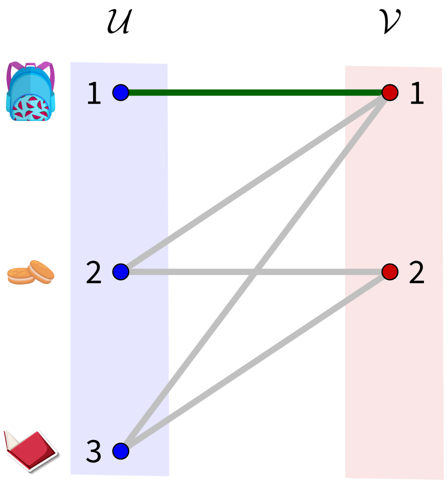
- v_t arrives along with its edges
- the algorithm can match it to a free vertex in $\mathcal{U}$
- the decision is final



Online Matching

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

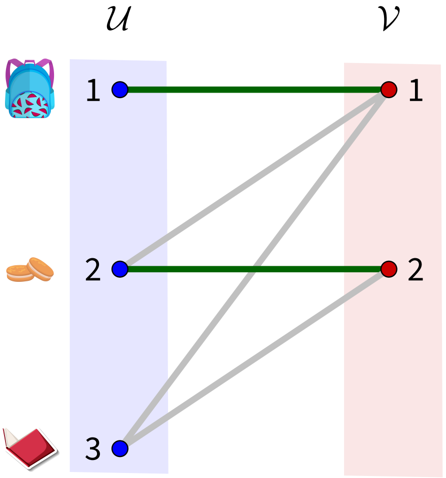
- v_t arrives along with its edges
- the algorithm can match it to a free vertex in $\mathcal{U}$
- the decision is final



Online Matching

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

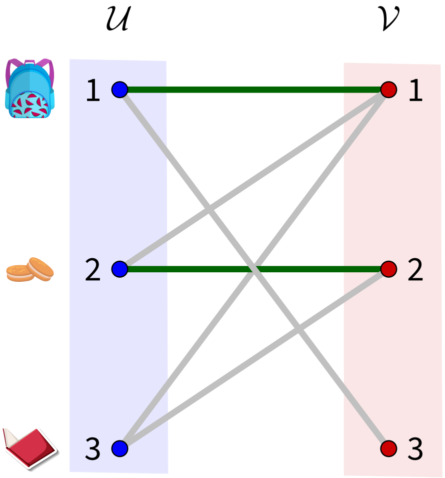
- v_t arrives along with its edges
- the algorithm can match it to a free vertex in $\mathcal{U}$
- the decision is final



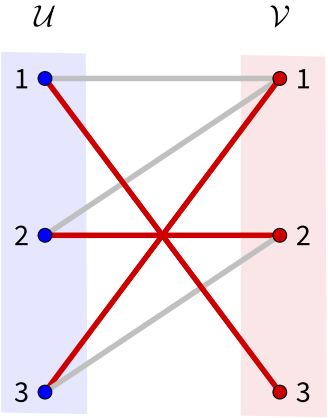
Online Matching

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

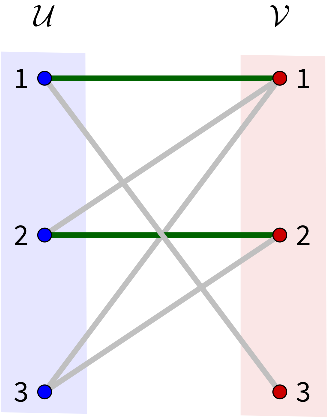
- v_t arrives along with its edges
- the algorithm can match it to a free vertex in $\mathcal{U}$
- the decision is final



Evaluating the performance



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



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

Our objective: Maximize algorithms' 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.

Users' arrival: the usual assumptions

- **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)

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

Algorithms and pre-existing litterature

GREEDY Algorithm in the case of Adversarial user arrivals

Algorithm 1: GREEDY Algorithm

for $t = 1, .., |\mathcal{V}|$ **do**

  Match v_t to any free neighbor;

end

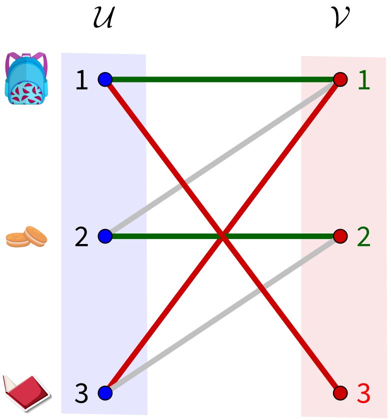
Theorem

In the Adversarial setting,

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

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

Proof: For every "missed" match, there is at least one "successful" match.



Adversarial users' arrival: a better algorithm, (Ad) RANKING

Algorithm 2: (Ad) RANKING Algorithm

Draw a random permutation π;

for $i = 1, .., |\mathcal{U}|$ **do**

  Assign to u_i rank $\pi(i)$;

end

for $t = 1, .., |\mathcal{V}|$ **do**

  Match v_t to its lowest ranked free neighbor;

end

Ad RANKING

Theorem [Karp et al., 1990]

In the Adversarial setting,

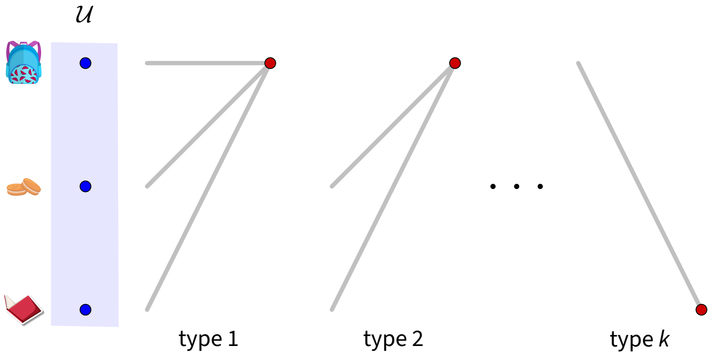
$$\text{C.R.(RANKING)} \geq 1 - \frac{1}{e}.$$

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

Adversarial is the worst case, but in real situations, the users' characteristics are not adversarial !

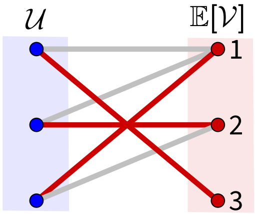
A stochasticity assumption: Known IID

There is a distribution over k fixed users' known types from which the incoming vertices are drawn i.i.d. [Feldman et al., 2009].

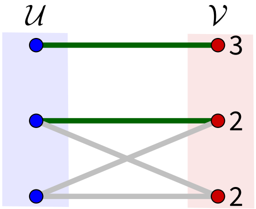


A first naive solution:

- Compute an optimal matching on the expected graph
- 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.

In a nutshell :

- (Involved) algorithms get a better competitive ratio (≥ 0.716), [Manshadi et al., 12, Jaillet and Lu, 14, Huang et al., 22] ,
- CR upper bounded by 0.823 [Manshadi et al., 12],
- CR of GREEDY still $1 - \frac{1}{e}$,
- Experiments tell a different story [Borodin et al., 2018].

Our work: Online Matching in Random Graphs

In the Configuration Model

- CM is a classical random graph model introduced by Bollobas in the 80s,
- A graph from the CM is close to a graph taken uniformly at random among those with a fixed degree distribution.

Theorem [Noiry, Perchet, S. ; 2021], informal

The size of the matching produced by GREEDY can be approximated w.h.p. by the solution of an ODE that depends on the degree distribution.

Related works: [Mastin and Jaillet, 2013, Aoudi et al., 2022, Aamand et al., 2022]

No correlations between the connections!

Handling correlation: The 1-D Geometric Model

1-D Random Geometric graph

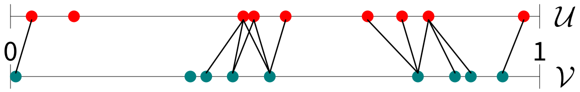
Motivation

- Campaigns & users have features $U_i \in \mathbb{R}^d$ and $V_j \in \mathbb{R}^d$
- Connected if "close enough" (for some Kernel)

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

- N points in $\mathcal{U}$ drawn iid uniformly in $[0, 1]$,
- N points in $\mathcal{V}$ drawn iid uniformly in $[0, 1]$,
- Edge between $u \in \mathcal{U}$ and $v \in \mathcal{V}$ iff:

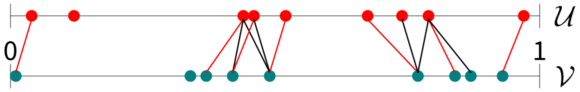
$$|u - v| \leq \frac{c}{N}.$$



Offline Maximum matching

Proposition

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



Remark:

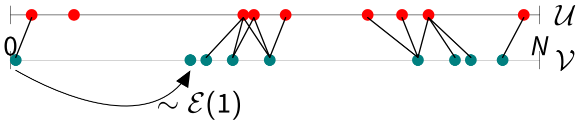
- **not** a direct consequence of 1D metric OT,
- **Similar** arguments: there exits a maximum matching with no crossing rays that matches every vertex to its leftmost free neighbor.

Computing the Maximum Matching Size

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 $[0, N]$,
- Edge between $u \in \mathcal{U}$ and $v \in \mathcal{V}$ iff $|u - v| \leq c$.



Expected matching sizes in the two model

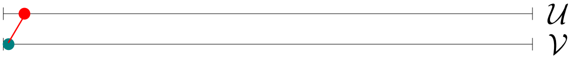
With $\gamma^*(c, N)$ and $M^*(c, N)$ the exp. sizes of the matchings in the two models:

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

Step 2: Generate the graph together with the matching.

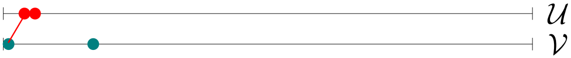
Three situations possible:

- Successful match !



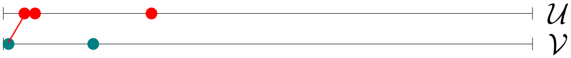
$\Longrightarrow$ generate next points in $\mathcal{U}$ and $\mathcal{V}$.

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



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

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



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

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

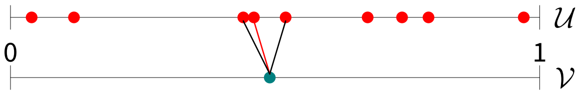
Proposition

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

Online Matching in the 1-D Geometric Model

Match to the closest point algorithm

The incoming point is matched to its closest available neighbor.



Theorem

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

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

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

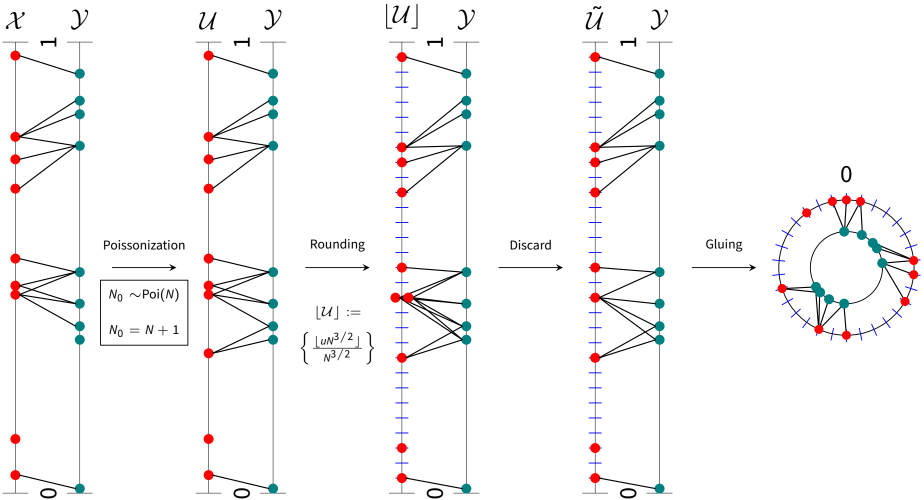
$$\begin{aligned} \frac{\partial f(x, t)}{\partial t} = & - \min(x, 2c) f(x, t) - \int_0^{+\infty} \frac{\min(x', 2c) f(x', t) f(x, t)}{\int_0^{+\infty} f(x', t) dx'} dx' \\ & + \frac{1}{\int_0^{+\infty} f(x', t) dx'} \int_0^x \min(x', 2c) f(x', t) f(x - x', t) dx' \end{aligned}$$

with the following initial conditions

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

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

The matching algorithm is studied on a modified graph:



Track the gap sizes between remaining free vertices

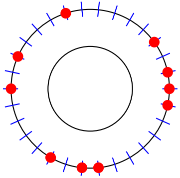
N_t = number of free vertices at iteration t.

$u_t(i)$ = coordinate of the (remaining) i^{th} free vertex

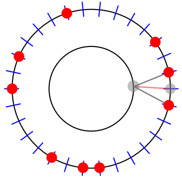
For $\ell \in [N^{3/2}]$, define

$$F_{k,N}(\ell, t) := \left| \left\{ i \in [N_t] \text{ s.t. } \left(u_t(i+1) - u_t(i)\right) = \frac{\ell}{kN} \right\} \right|,$$

On an example



$\longrightarrow$



$F_9(1,0) = 3$

$F_9(2,0) = 3$

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

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 $\ell \in [30]$:

$F_9(\ell,1) = 0$

Why are the gaps nice quantities?

Relation with matching size

$$M(t) = N_0 - \sum_{\ell} F_{k,N}(\ell, t).$$

Relation with probability of matching

With p_t the probability of getting a match at iteration t.

$$p_t = \frac{1}{kN} \sum_{\ell} \min(2c, \ell) F_{k,N}(\ell, t).$$

Are they tractable?

- Thanks to the discretization, $F_{k,N}(\ell, 0)$ concentrates,
- At every iteration, the gaps are ordered uniformly at random $\rightarrow$ explicit expression for the expected evolution.

Markov discrete process with initial condition concentration and "nice" expected evolution:

Differential Equation Method, Wormald,95

1. Discrete system: There exists Φ_N such that:

$$\mathbb{E}[F_{k,N}(\ell, t+1) - F_{k,N}(\ell, t) \,|\, \mathcal{F}_t] = \Phi_{k,N}\left(F_{k,N}(0,t), \dots, F_{k,N}(kN,t)\right) + o(1).$$

2. PDEs. Solutions asymptotically close to

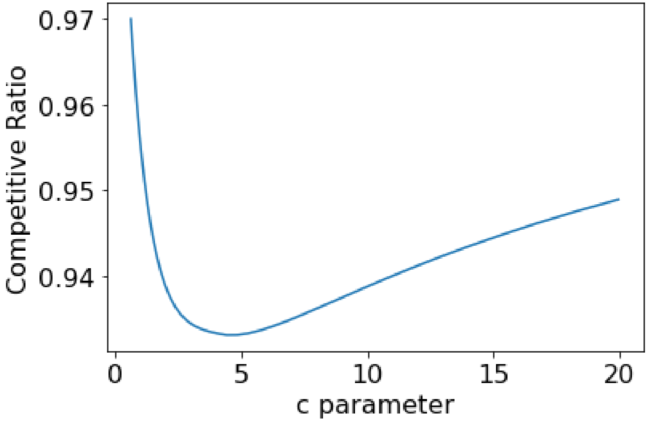
$$\frac{\partial f_k(\ell, t)}{\partial t} = \Phi_k\left(f_k(0,t)_{\ell=0}^{+}\infty, \ell\right).$$

   with initial conditions: $f_k(\ell, 0) = k(1 - e^{-\frac{1}{k}})^2 e^{-\frac{\ell}{k}}$.

3. Control of errors. With f the function of the theorem,

$$\forall t \in [0,1], \quad ||\, f(.,t) - f_k(.,t)\, ||_{L_1} \lesssim \frac{1}{k}.$$

Competitive ratio



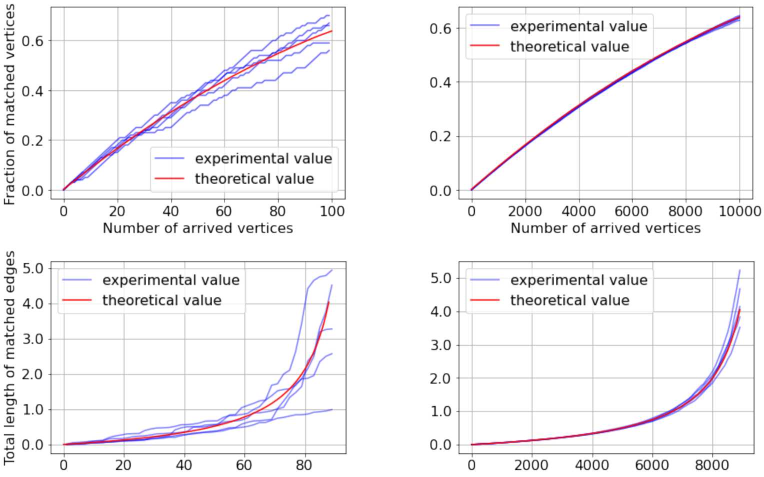
Link with metric matching on the line

- U_i and V_j are iid uniformly on $[0, 1]$
- V_t must be matched to some $U_{m(t)}$ with cost $d(U_{m(t)}, V_t)$
- OPT is OT, cost $\mathcal{O}(\sqrt{N})$
- After $(1-\delta)N$ points, the total cost of Match to the Closest [Akbarpour, 21, Balkanski, 22] is $\mathcal{O}\left(\frac{1}{\delta}\right)$

Performance of Match to the Closest

After $(1-\delta)N$ points, the total cost of Match to the Closest converges in probability to $0.5\left(\frac{1}{\delta}-1\right)$.

Experimental Results



Thank you for listening !

Aamand, A., Chen, J., and Indyk, P. (2022).
(optimal) online bipartite matching with degree information.
Advances in Neural Information Processing Systems, 35:5724–5737.

Aoudi, M. H. A. D., Moyal, P., and Robin, V. (2022).
Markovian online matching algorithms on large bipartite random graphs.
Methodology and Computing in Applied Probability, 24(4):3195–3225.

Borodin, A., Karavasilis, C., and Pankratov, D. (2018).
An experimental study of algorithms for online bipartite matching.

Feldman, J., Mehta, A., Mirrokni, V., and Muthukrishnan, S. (2009).
Online stochastic matching: Beating 1-1/e.
In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 117–126. IEEE.

Karp, R. M., Vazirani, U. V., and Vazirani, V. V. (1990).

An optimal algorithm for on-line bipartite matching.
In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358.

Mastin, A. and Jaillet, P. (2013).
Greedy online bipartite matching on random graphs.
arXiv preprint arXiv:1307.2536.