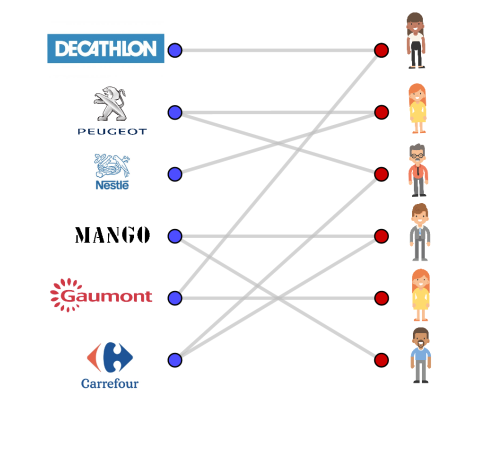
Online Matching in Bipartite Graphs

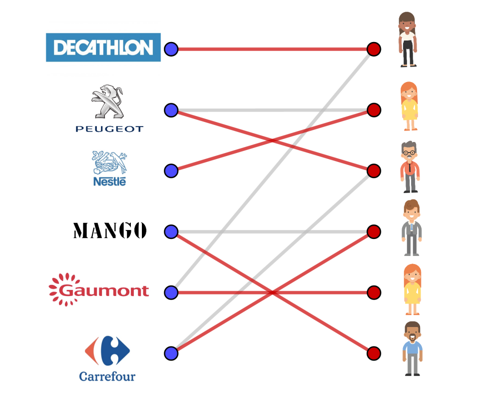
Flore Sentenac

Motivations: Dynamic allocation

Ad - User allocation



Ad - User allocation

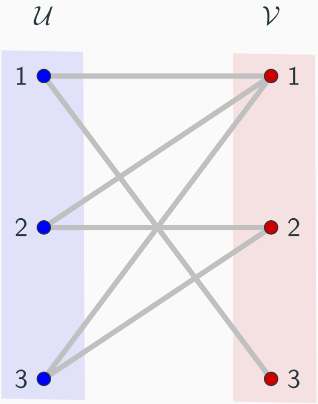


Problem definition

Matching on a Bipartite graph

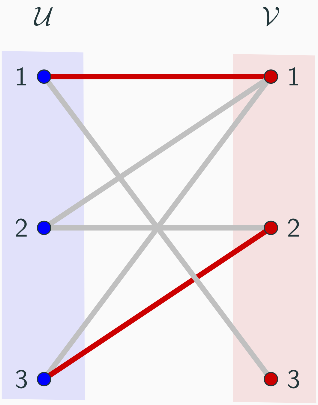
Graph $\mathcal{G} = ((\mathcal{U}, \mathcal{V}), \mathcal{E})$ bipartite if:

- 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

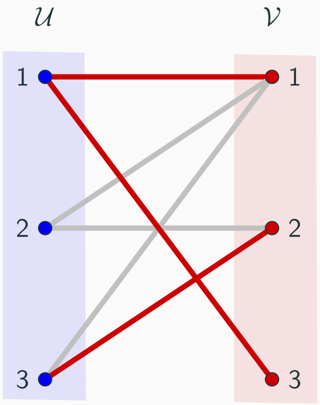
A matching is a set of edges with no common vertices.



A matching

Matching on a Bipartite graph

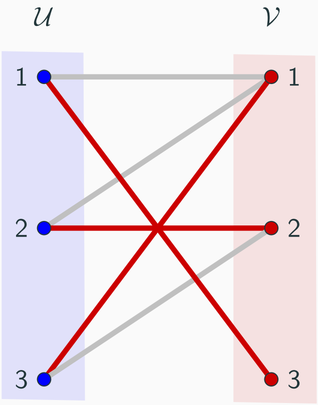
A matching is a set of edges with no common vertices.



Not a matching

Matching on a Bipartite graph

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

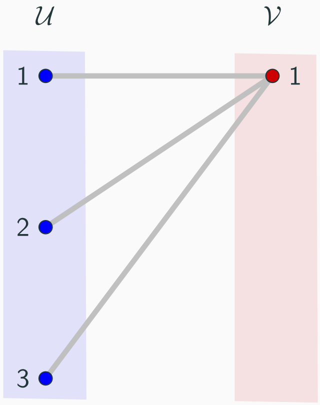


A maximum matching

Online Matching

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

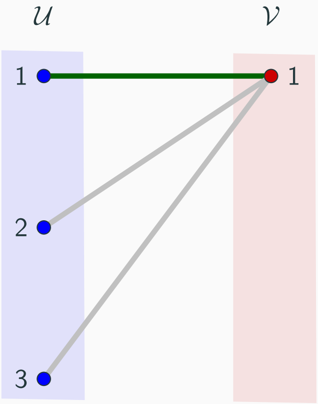
- v_t arrives along with its edges
- the algorithm can match it to a free vertex in $\mathcal{U}$
- the decision is final



Online Matching

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

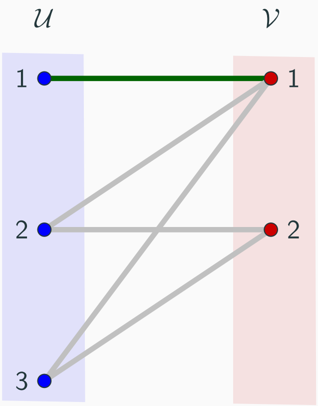
- v_t arrives along with its edges
- the algorithm can match it to a free vertex in $\mathcal{U}$
- the decision is final



Online Matching

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

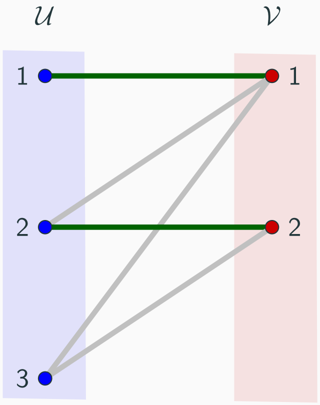
- v_t arrives along with its edges
- the algorithm can match it to a free vertex in $\mathcal{U}$
- the decision is final



Online Matching

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

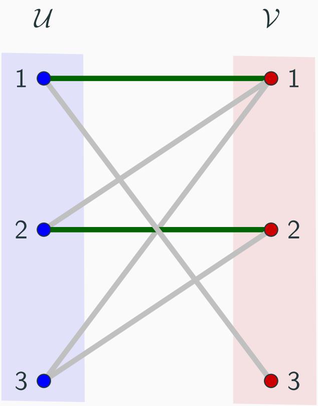
- v_t arrives along with its edges
- the algorithm can match it to a free vertex in $\mathcal{U}$
- the decision is final



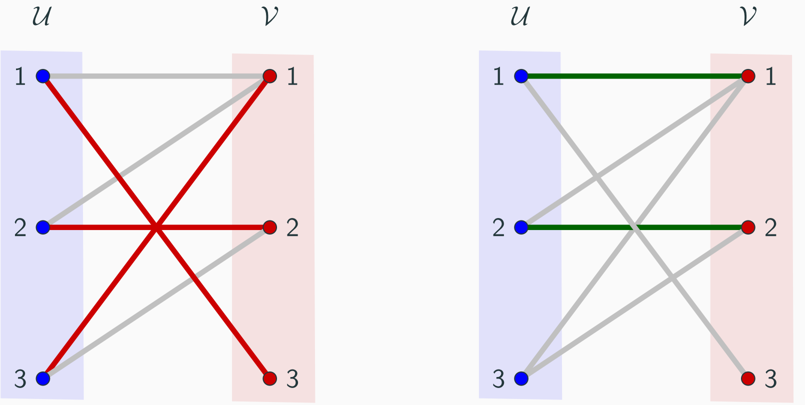
Online Matching

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



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

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

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)

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

The simplest algorithm : GREEDY

GREEDY

Algorithm 1: GREEDY Algorithm

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

  Match v_t to any free neighbor;

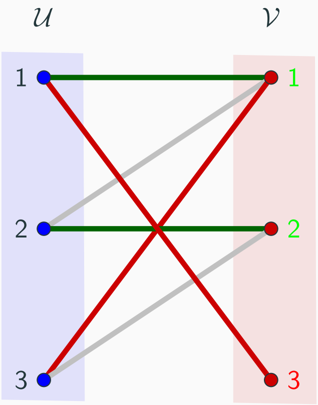
end

Theorem

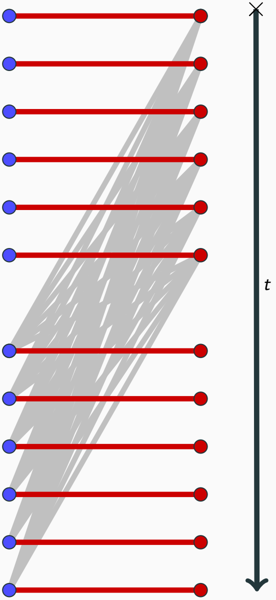
In the Adversarial setting,

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

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



GREEDY with Adversarial Arrivals: A difficult situation



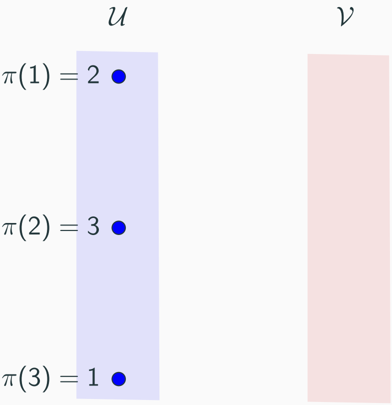
Using correlated randomness : RANKING

RANKING

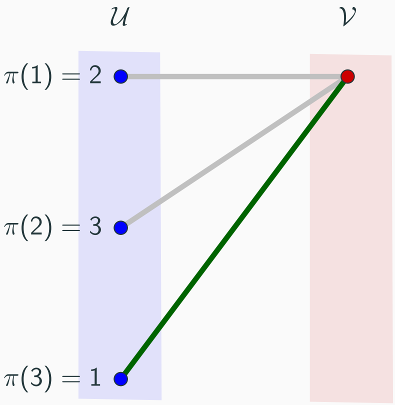
Algorithm 2: RANKING Algorithm

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

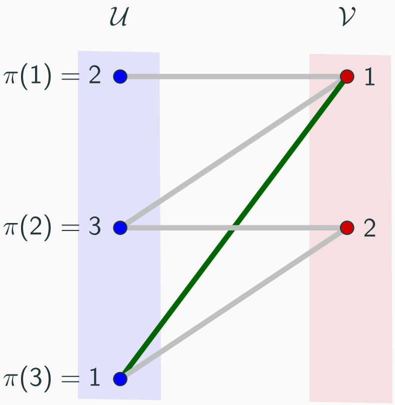
RANKING



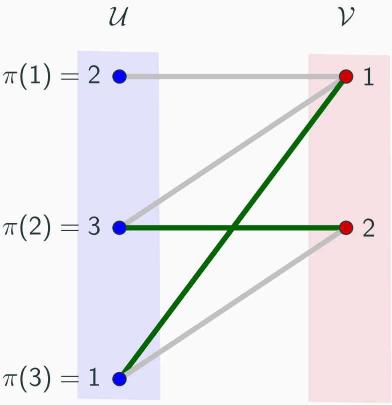
RANKING



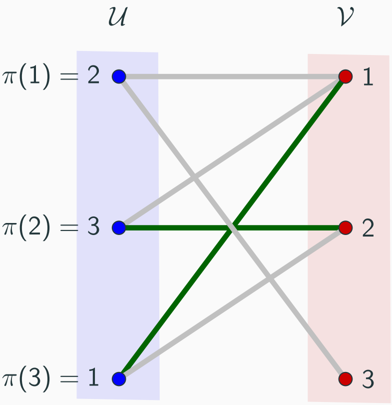
RANKING



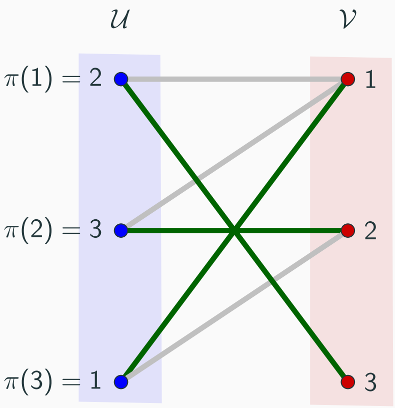
RANKING



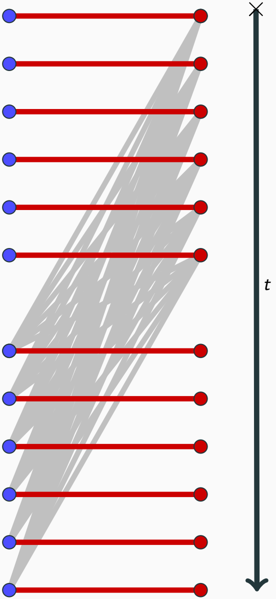
RANKING



RANKING



Back to GREEDY's difficult situation



RANKING

Theorem

In the Adversarial setting,

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

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

In our toolbox : Primal-Dual Analysis

Maximum Matching problem as an LP

Finding a maximum matching in the graph $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$ is equivalent to finding a solution of the following I-LP:

$$\begin{aligned} \text{maximize} \sum_{(u,v)\in\mathcal{E}} & x_{uv} \\ \text{s.t.} \quad \sum_{v:(u,v)\in\mathcal{E}} & x_{uv} \leq 1, \forall u \in \mathcal{U} \\ \sum_{u:(u,v)\in\mathcal{E}} & x_{uv} \leq 1, \forall v \in \mathcal{V} \\ & x_{uv} \in \{0,1\}, \forall (u,v) \in \mathcal{E} \end{aligned}$$

Maximum Matching problem as an LP

Matching linear program (P)

$$
\begin{aligned}
\text{maximize} \sum_{(u,v)\in\mathcal{E}} x_{uv} & \\
\text{s.t.} \sum_{v:(u,v)\in\mathcal{E}} x_{uv} &\leq 1, \forall u \in \mathcal{U} \\
\sum_{u:(u,v)\in\mathcal{E}} x_{uv} &\leq 1, \forall v \in \mathcal{V} \\
x_{uv} &\geq 0, \forall (u,v) \in \mathcal{E}
\end{aligned}
$$

Note: On bipartite graphs, the value of the relaxed program and the original one match.

The dual problem

Dual to the Matching linear program (D)

$$
\begin{aligned}
\text{minimize} \sum_{u \in \mathcal{U}} \alpha_u + \sum_{v \in \mathcal{V}} \beta_v \\
\text{s.t. } \alpha_u + \beta_v \geq 1, \forall (u, v) \in \mathcal{E} \\
\alpha_u \geq 0,\ \beta_v \geq 0,\ \forall u \in \mathcal{U}, v \in \mathcal{V}
\end{aligned}
$$

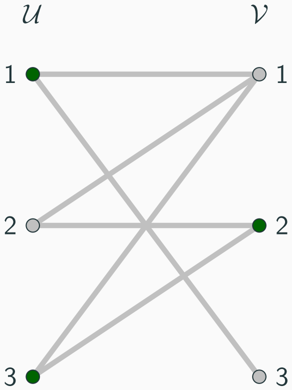
Note: This LP corresponds to the vertex cover problem.

Matching on a Bipartite graph

Dual to the Matching linear program (D)

$$\text{minimize} \sum_{u \in \mathcal{U}} \alpha_u + \sum_{v \in \mathcal{V}} \beta_v$$
$$\text{s.t. } \alpha_u + \beta_v \geq 1, \forall (u,v) \in \mathcal{E}$$
$$\alpha_u \geq 0,\ \beta_v \geq 0$$

Note : this LP corresponds to the vertex cover problem.



Overcomplicating the analysis of GREEDY

Algorithm 3: Primal Dual update for GREEDY

for $t = 1, .., |\mathcal{V}|$ **do**
  if *v has a free neighbor u* **then**
    Add (u, v) to $\mathcal{M}$;
    $\hat{x}_{uv} \leftarrow 1$; // primal update
    $\hat{\beta}_v \leftarrow \frac{1}{2}$,$\hat{\alpha}_u \leftarrow \frac{1}{2}$; // dual update
  end
end

Overcomplicating the analysis of GREEDY

$\forall (u, v) \in \mathcal{E}$, $2(\hat{\alpha}_u + \hat{\beta}_v) \geq 1 \implies (2\hat{\boldsymbol{\alpha}}, 2\hat{\boldsymbol{\beta}})$ is an admissible sol. of (D):

Overcomplicating the analysis of GREEDY

$\forall (u,v) \in \mathcal{E}$, $2(\hat{\alpha}_u + \hat{\beta}_v) \geq 1 \implies (2\hat{\boldsymbol{\alpha}}, 2\hat{\boldsymbol{\beta}})$ is an admissible sol. of (D):

$$2\mathsf{ALG}(\mathcal{G}) = 2 \sum_{(u,v)} \hat{x}_{uv}$$

Overcomplicating the analysis of GREEDY

$\forall (u,v) \in \mathcal{E}$, $2(\hat{\alpha}_u + \hat{\beta}_v) \geq 1 \implies (2\hat{\boldsymbol{\alpha}}, 2\hat{\boldsymbol{\beta}})$ is an admissible sol. of (D):

$$\begin{aligned} 2\mathrm{ALG}(\mathcal{G}) &= 2 \sum_{(u,v)} \hat{x}_{uv} \\ &= 2 \sum_{u \in \mathcal{U}} \hat{\alpha}_u + 2 \sum_{v \in \mathcal{V}} \hat{\beta}_v \end{aligned}$$

Overcomplicating the analysis of GREEDY

$\forall (u,v) \in \mathcal{E}$, $2(\hat{\alpha}_u + \hat{\beta}_v) \geq 1 \implies (2\hat{\boldsymbol{\alpha}}, 2\hat{\boldsymbol{\beta}})$ is an admissible sol. of (D):

$$\begin{aligned} 2\mathrm{ALG}(\mathcal{G}) &= 2 \sum_{(u,v)} \hat{x}_{uv} \\ &= 2 \sum_{u \in \mathcal{U}} \hat{\alpha}_u + 2 \sum_{v \in \mathcal{V}} \hat{\beta}_v \\ &\geq \sum_{u \in \mathcal{U}} \alpha^*_u + \sum_{v \in \mathcal{V}} \beta^*_v \end{aligned}$$

Overcomplicating the analysis of GREEDY

$\forall (u,v) \in \mathcal{E}$, $2(\hat{\alpha}_u + \hat{\beta}_v) \geq 1 \implies (2\hat{\boldsymbol{\alpha}}, 2\hat{\boldsymbol{\beta}})$ is an admissible sol. of (D):

$$\begin{aligned} 2\mathsf{ALG}(\mathcal{G}) &= 2 \sum_{(u,v)} \hat{x}_{uv} \\ &= 2 \sum_{u \in \mathcal{U}} \hat{\alpha}_u + 2 \sum_{v \in \mathcal{V}} \hat{\beta}_v \\ &\geq \sum_{u \in \mathcal{U}} \alpha^*_u + \sum_{v \in \mathcal{V}} \beta^*_v \\ &= \mathsf{OPT}(\mathcal{G}) \end{aligned}$$

What about RANKING ?

Algorithm 4: Primal Dual update for RANKING

for $u \in \mathcal{U}$ **do**
    Draw $r_u \sim \mathcal{U}([0,1])$
end
for $v = 1, .., |\mathcal{V}|$ **do**
    $u = \arg\min\{r_u | u \text{ unmatched}, (u,v) \in \mathcal{E}\}$;
    if $u \neq \emptyset$ **then**
        Add (u,v) to $\mathcal{M}$;
        $\hat{x}_{uv} \leftarrow 1$; `// primal update`
        $\hat{\beta}_v \leftarrow (1 - g(r_u))/c$, $\hat{\alpha}_u \leftarrow g(r_u)/c$; `// dual update`
    end
end

Primal-Dual Analysis of RANKING

Lemma

If $g(x) = e^{x-1}$ and $c = 1 - \frac{1}{e}$, then, $\forall (u, v) \in \mathcal{E}$:

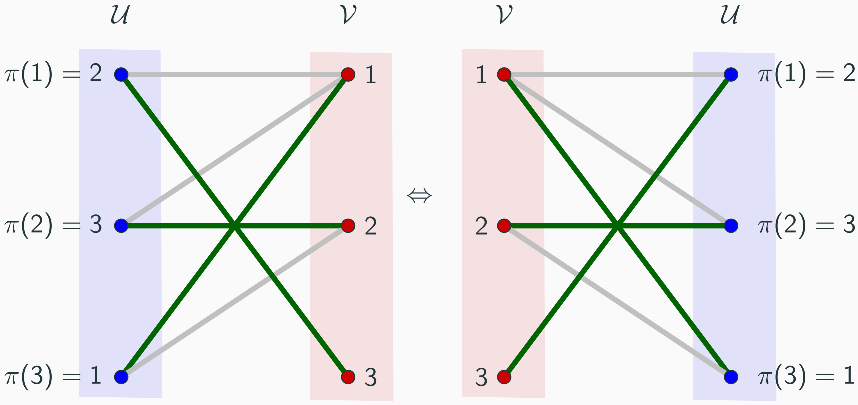
$$\mathbb{E}[\hat{\alpha}_u + \hat{\beta}_v] \geq 1$$

$$\begin{aligned}
\mathbb{E}[\mathsf{ALG}(G)] &= \left(1 - \frac{1}{e}\right) \mathbb{E}\left[\sum_{u \in \mathcal{U}} \hat{\alpha}_u + \sum_{v \in \mathcal{V}} \hat{\beta}_v\right] \\
&\geq \left(1 - \frac{1}{e}\right) \sum_{u \in \mathcal{U}} \alpha^*_u + \sum_{v \in \mathcal{V}} \beta^*_v \\
&= \left(1 - \frac{1}{e}\right) \mathsf{OPT}(\mathcal{G})
\end{aligned}$$

When you've got a hammer...

- We can study algorithms on weighted graphs.
- Other problems: Online Set Cover, Online Caching...

GREEDY Random Order



Theorem

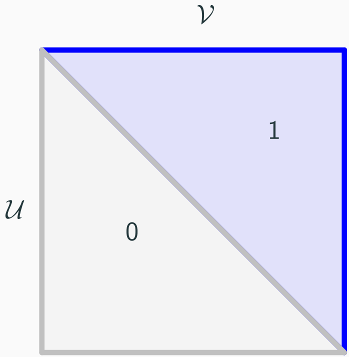
In the Random Order setting,

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

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

Worse case for RANKING

Upper triangular matrix:



Stochastic arrivals

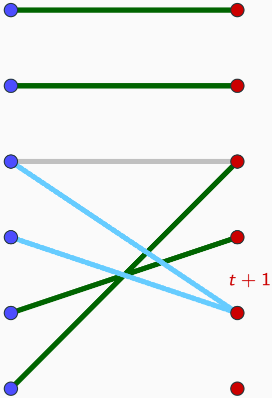
The Erdos-Renyi bipartite graph

Definition of $\mathcal{G}(N, N, c)$

- $|\mathcal{U}| = |\mathcal{V}| = N$
- $\mathbb{P}\left((u, v) \in \mathcal{E}\right) = \frac{c}{N}$

What is the performance of GREEDY on $\mathcal{G}(N, N, c)$?

In our toolbox :
The Differential Equation Method



$M_t =$ number of matched vertices at t,

$$\mathbb{P}(v_{t+1} \text{ matched } |M_t) = 1 - (1 - \frac{c}{N})^{N-M_t}$$
$$= \mathbb{E}[M_{t+1} - M_t | M_t]$$

Turning the discrete process into an ODE

Define the normalized random variable:

$$Z(\tau) = \frac{M(N\tau)}{N}, \quad 0 \le \tau \le 1.$$

Turning the discrete process into an ODE

Define the normalized random variable:

$$Z(\tau) = \frac{M(N\tau)}{N}, \quad 0 \le \tau \le 1.$$

We have:

$$\begin{aligned} \frac{\mathbb{E}[Z(\tau + 1/N) - Z(\tau) \mid Z(\tau)]}{1/N} &= 1 - \left(1 - \frac{c}{N}\right)^{N(1-Z(\tau))} \\ &= 1 - e^{-c(1-Z(\tau))} + o(1). \end{aligned}$$

Turning the discrete process into an ODE

Define the normalized random variable:

$$Z(\tau) = \frac{M(N\tau)}{N}, \quad 0 \leq \tau \leq 1.$$

We have:

$$\begin{aligned}\frac{\mathbb{E}[Z(\tau + 1/N) - Z(\tau) \mid Z(\tau)]}{1/N} &= 1 - \left(1 - \frac{c}{N}\right)^{N(1-Z(\tau))} \\ &= 1 - e^{-c(1-Z(\tau))} + o(1).\end{aligned}$$

As $N \to \infty$, we arrive at the differential equation:

$$\frac{dz(\tau)}{d\tau} = 1 - e^{-c(1-z(\tau))}.$$

Wormald's Theorem

Under the following conditions:

- the increments of the discrete random process are bounded a.s. by a constant.
- the function in the ODE is regular enough (Lipschitz), $(1 - e^{-c(1-z(\tau))}$ in the example).
- the approximation between the expectation and the function is small enough.

Then the difference between the discrete process M_t and the solution of the ODE $Nz(t/N)$ is $o(N)$ w.h.p..[1]

Final result

$$\frac{\text{GREEDY}\left(\mathcal{G}(N,N,c)\right)}{N} \xrightarrow[N\to+\infty]{\mathbb{P}} 1 - e^{\left(e^{-c}-1\right)}$$

Theorem

The asymptotic C.R. of the GREEDY algorithm on any Erdos-Renyi graph is lower bounded as:

$$C.R.(GREEDY(\mathcal{G}(N,N,c))) \geq 0.837$$

When you've got a hammer..

- GREEDY can be studied on a larger class of graphs (configuration model).
- Study random graph processes: find the size of the k-core of a graph, the largest independent set in a d-regular graph...

The Configuration Model

Introduced by Bollobás in 1980.

Consider two degree sequences

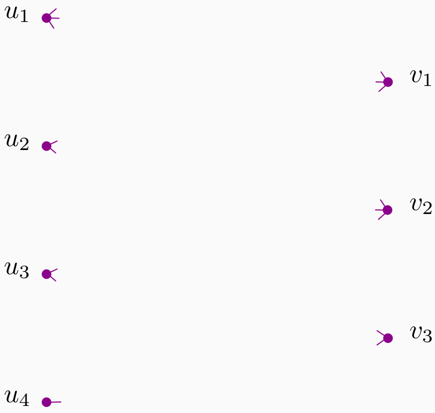
$$\begin{cases} \mathrm{d}^U = (d_1^U, \ldots, d_N^U) \in \mathbb{N}^N, & N \geq 1, \\ \mathrm{d}^V = (d_1^V, \ldots, d_T^V) \in \mathbb{N}^T, & T \geq 1, \end{cases} \qquad \text{s.t.} \ \sum_{i=1}^{N} d_i^U = \sum_{i=1}^{T} d_i^V.$$

Interpretation: d_i^U is the degree of the i-th vertex of U.

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

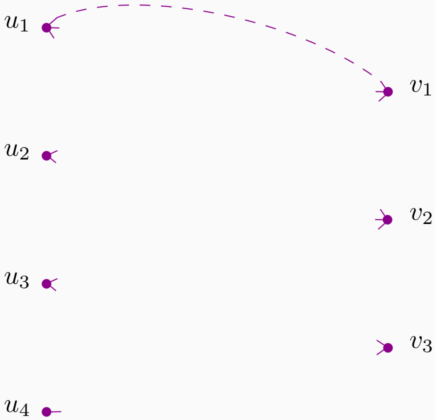
Definition

Example: $d_1^U = 3, d_2^U = 2, d_3^U = 2, d_4^U = 1$ and $d_1^V = 3, d_2^V = 3, d_3^V = 2$.



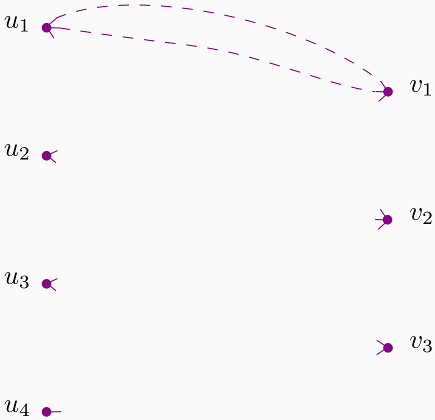
Definition

Example: $d_1^U = 3, d_2^U = 2, d_3^U = 2, d_4^U = 1$ and $d_1^V = 3, d_2^V = 3, d_3^V = 2$.



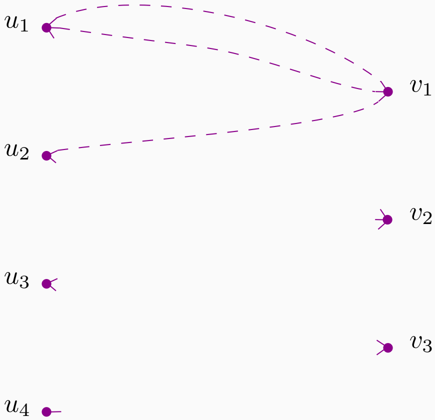
Definition

Example: $d_1^U = 3, d_2^U = 2, d_3^U = 2, d_4^U = 1$ and $d_1^V = 3, d_2^V = 3, d_3^V = 2$.



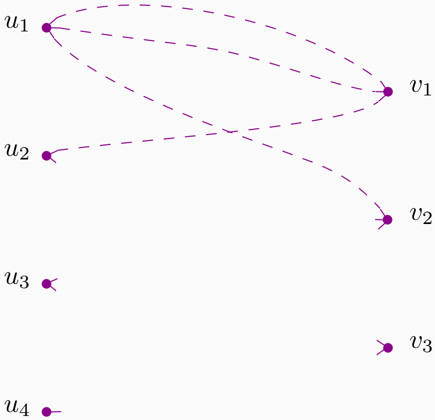
Definition

Example: $d_1^U = 3, d_2^U = 2, d_3^U = 2, d_4^U = 1$ and $d_1^V = 3, d_2^V = 3, d_3^V = 2$.



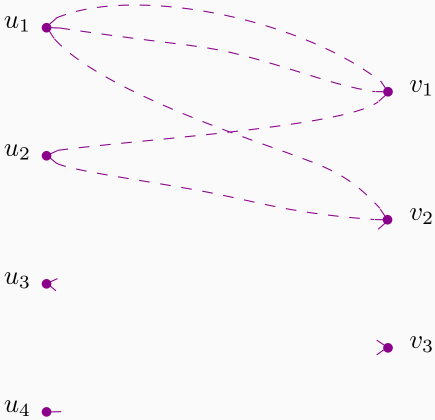
Definition

Example: $d_1^U = 3, d_2^U = 2, d_3^U = 2, d_4^U = 1$ and $d_1^V = 3, d_2^V = 3, d_3^V = 2$.



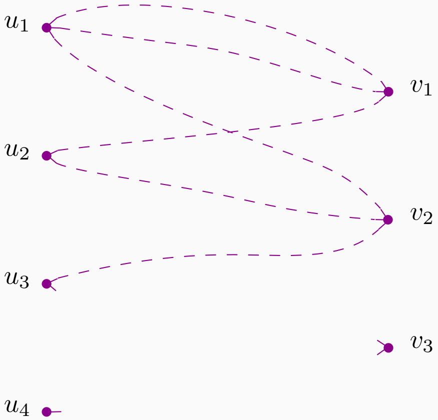
Definition

Example: $d_1^U = 3, d_2^U = 2, d_3^U = 2, d_4^U = 1$ and $d_1^V = 3, d_2^V = 3, d_3^V = 2$.



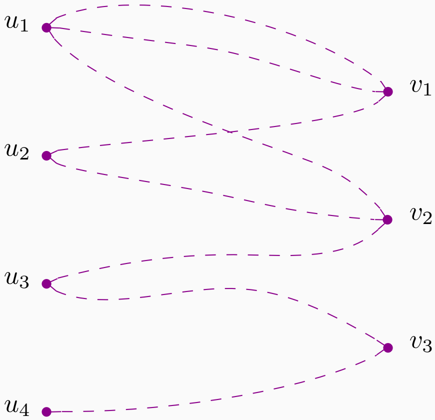
Definition

Example: $d_1^U = 3, d_2^U = 2, d_3^U = 2, d_4^U = 1$ and $d_1^V = 3, d_2^V = 3, d_3^V = 2$.



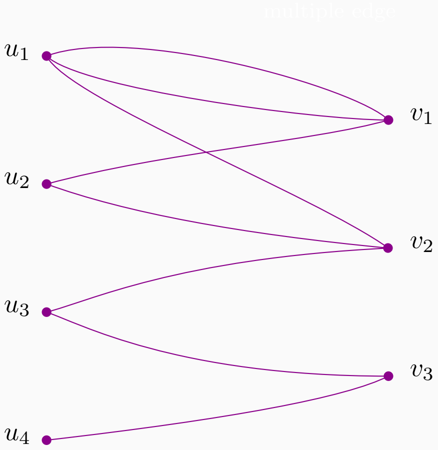
Definition

Example: $d_1^U = 3, d_2^U = 2, d_3^U = 2, d_4^U = 1$ and $d_1^V = 3, d_2^V = 3, d_3^V = 2$.



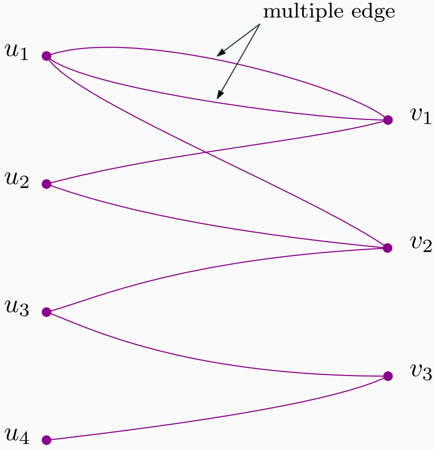
Definition

Example: $d_1^U = 3, d_2^U = 2, d_3^U = 2, d_4^U = 1$ and $d_1^V = 3, d_2^V = 3, d_3^V = 2$.



Definition

Example: $d_1^U = 3, d_2^U = 2, d_3^U = 2, d_4^U = 1$ and $d_1^V = 3, d_2^V = 3, d_3^V = 2$.



Random degree sequences

- π_U, π_V: two proba on $\mathbb{N}$ with expectations and finite 2nd moment.

$$\mu_U := \sum_{i \geq 0} i\pi_U(i) \quad \text{and} \quad \mu_V := \sum_{i \geq 0} i\pi_V(i).$$

- $d_1^U, \dots, d_N^U \overset{i.i.d.}{\sim} \pi_U, \qquad \sum_{i=1}^{N} d_i^U \approx \mu_U N.$
- $d_1^V, \dots, d_T^V \overset{i.i.d.}{\sim} \pi_V, \qquad \sum_{i=1}^{T} d_i^V \approx \mu_V T.$

Construction of configuration model : sequentially match half-edges.

Random degree sequences

- π_U, π_V: two proba on $\mathbb{N}$ with expectations and finite 2nd moment.

$$\mu_U := \sum_{i \geq 0} i\pi_U(i) \quad \text{and} \quad \mu_V := \sum_{i \geq 0} i\pi_V(i).$$

- $d_1^U, \dots, d_N^U \overset{i.i.d.}{\sim} \pi_U, \qquad \sum_{i=1}^{N} d_i^U \approx \mu_U N.$
- $d_1^V, \dots, d_T^V \overset{i.i.d.}{\sim} \pi_V, \qquad \sum_{i=1}^{T} d_i^V \approx \mu_V T.$

Construction of configuration model : sequentially match half-edges.

1 - Compatibility condition: $\mu_U N = \mu_V T$.

Discard $o(N) + o(T)$ unpaired half-edges in $\mathcal{U}$ or $\mathcal{V}$.

Random degree sequences

- π_U, π_V: two proba on $\mathbb{N}$ with expectations and finite 2nd moment.

$$\mu_U := \sum_{i \geq 0} i\pi_U(i) \quad \text{and} \quad \mu_V := \sum_{i \geq 0} i\pi_V(i).$$

- $d_1^U, \ldots, d_N^U \overset{i.i.d.}{\sim} \pi_U, \qquad \sum_{i=1}^{N} d_i^U \approx \mu_U N.$
- $d_1^V, \ldots, d_T^V \overset{i.i.d.}{\sim} \pi_V, \qquad \sum_{i=1}^{T} d_i^V \approx \mu_V T.$

Construction of configuration model : sequentially match half-edges.

1 - Compatibility condition: $\mu_U N = \mu_V T$.

Discard $o(N) + o(T)$ unpaired half-edges in $\mathcal{U}$ or $\mathcal{V}$.

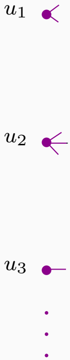
2 - Sparsity condition: $\mu_U = o(T)$.

Discard $o(T + N)$ multiple edges.

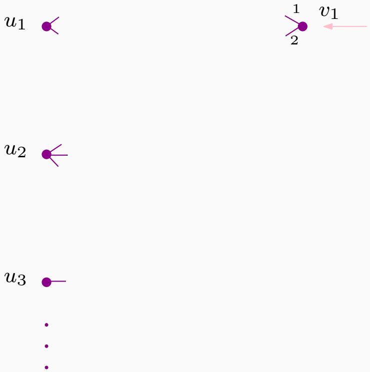
$\rightsquigarrow$ Sparse random bipartite graph $\mathrm{CM}(d^U, d^V)$ with asymptotic degree sequences given by π_U and π_V.

Greedy Online Matching Algorithm on a Bipartite Configuration Model

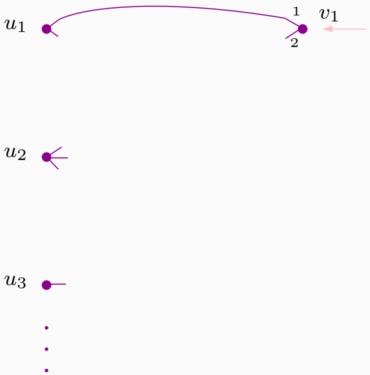
Definition with an example



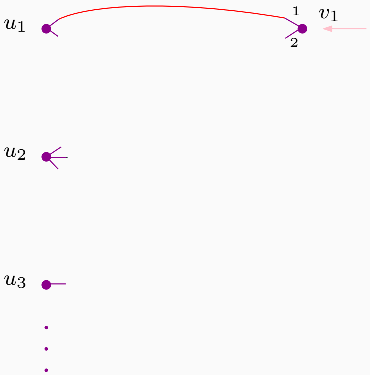
Definition with an example



Definition with an example

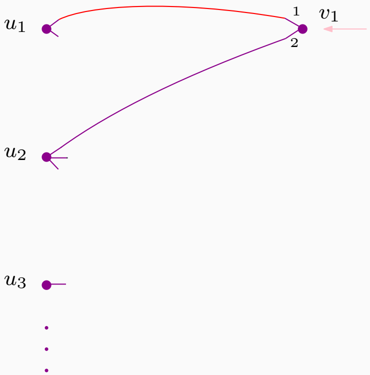


Definition with an example



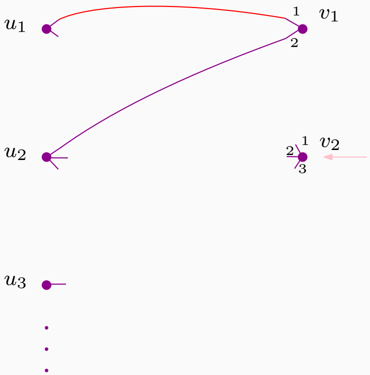
$$\mathcal{M}_1 = \big\{\{u_1, v_1\}\big\}$$

Definition with an example



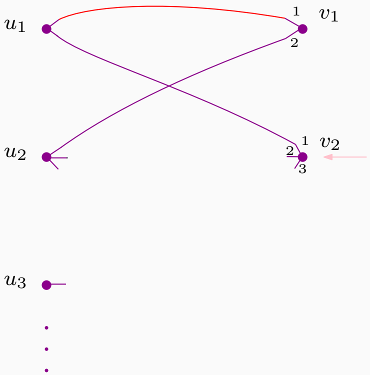
$$\mathcal{M}_1 = \{\{u_1, v_1\}\}$$

Definition with an example



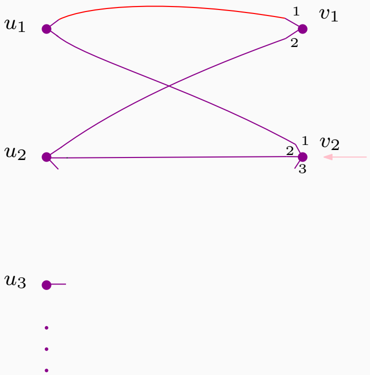
$$\mathcal{M}_1 = \{\{u_1, v_1\}\}$$

Definition with an example



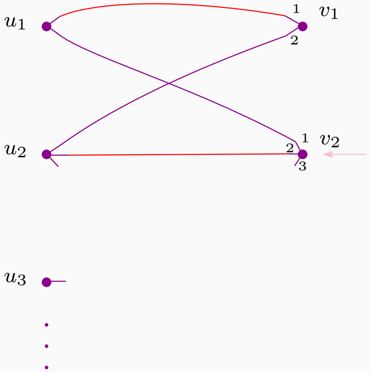
$$\mathcal{M}_1 = \{\{u_1, v_1\}\}$$

Definition with an example



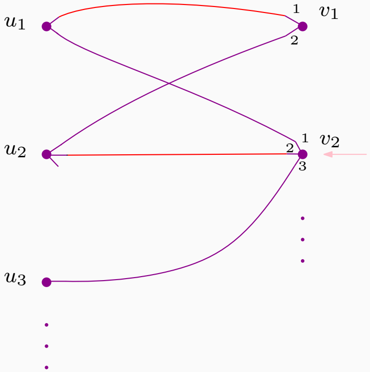
$$\mathcal{M}_1 = \{\{u_1, v_1\}\}$$

Definition with an example



$$\mathcal{M}_2 = \big\{\{u_1, v_1\}, \{u_2, v_2\}\big\}$$

Definition with an example



$$\mathcal{M}_2 = \big\{\{u_1, v_1\}, \{u_2, v_2\}\big\}$$

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

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

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

Then, the following convergence holds in probability:

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

And also...

- Non-asymptotic bounds: $\mathcal{M}(s)/N$ concentrates around G (with additional assumptions on the tails of π_U and π_V).
- Generalization to **weighted matching** where each vertex $u \in U$ has a capacity ω_u.

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

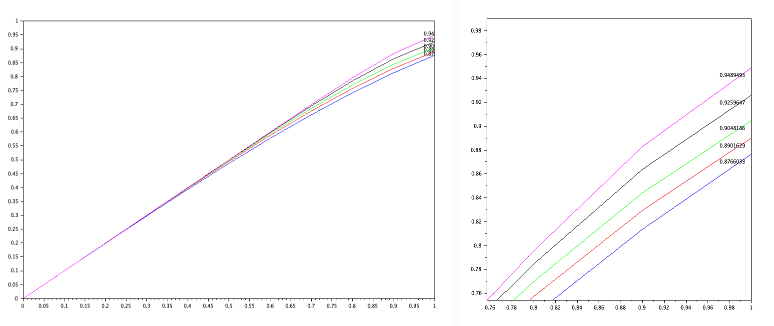


Figure 1: Numerical computations (on Scilab, results are almost instantaneous) of GREEDY performances for $d = 2$ (blue), $d = 3$ (red), $d = 4$ (green), $d = 6$ (black) and $d = 10$ (magenta).

The d-regular case

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

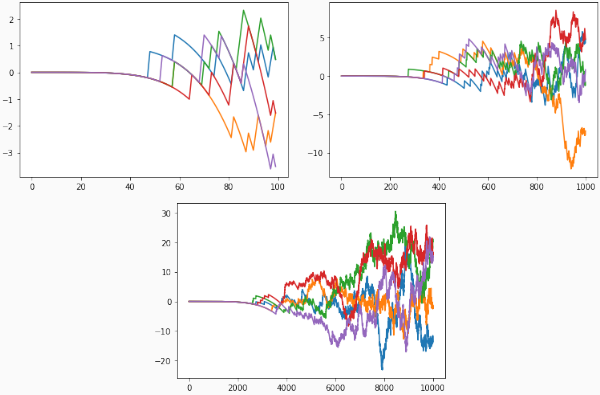


Figure 2: Difference between the theoretical performances and simulated performances of the GREEDY algorithm on the d-regular graph ($d = 4$) on 5 independent runs, with $N = 100, 1000, 10000$.

A last result

GREEDY vs. RANKING

GREEDY asymptotically outperforms RANKING in some configuration models.

Example: the 2-regular graphs.

- In 2-regular graphs, if the incoming vertex has a free neighbor of degree 1 and another free neighbor of degree 2, Ranking picks the free vertex
    - of degree **2** with proba $2/3$; [Greedy w.p. 1/2]
    - of degree **1** with proba $1/3$; [Greedy w.p. 1/2]
- If v has degree 1, it was not picked before, hence its rank is high.
- Ranking takes the **wrong decision** more frequently

Thank you!

References

[1] Nathanaël Enriquez, Gabriel Faraud, Laurent Ménard, and Nathan Noiry. Depth first exploration of a configuration model. *arXiv preprint arXiv:1911.10083*, 2019.