# Dynamics and learning in online allocation problems

Maria Cherifa

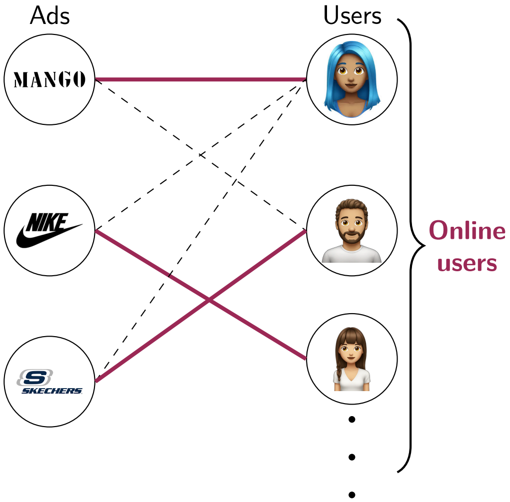
February 13, 2026

# 0 Table of contents

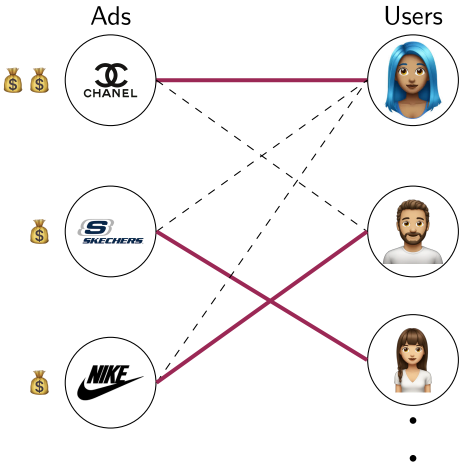


# 1 Motivation: online advertisement

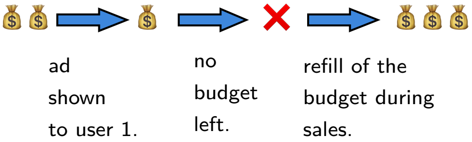
## Constraint 1: online arrivals



## Constraint 2: budget evolution

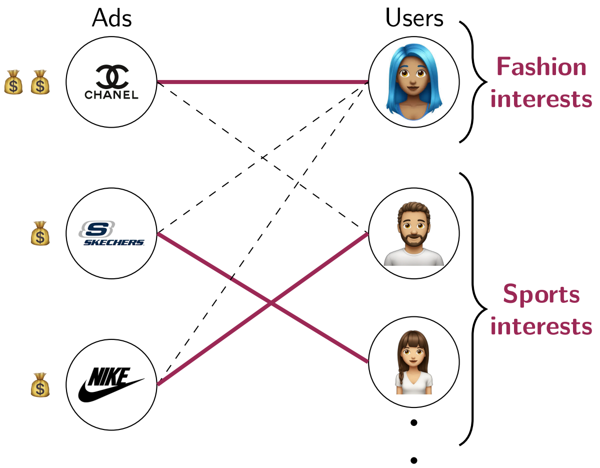



Budget evolution of an ad:



# 1 Motivation: online advertisement

## Constraint 3: community structure



# 1 From motivation to our problem setting

## Key elements of our setting

- **Online user arrivals:** decisions made sequentially.
- **Ad budgets:** limited and sometimes refilled.
- **Communities:** matching users to relevant ads.

# 1 Warm-up: standard online matching

Modeling:

- Natural structure: **bipartite graphs**.
- Online arrivals: **part of the graph is unknown**.

For $t = 1, \dots, |V|$:

- $v_t$ arrives with its edges.
- Each node $u \in U$ has a budget $b_u \in \{0, 1\}$ (degree of u).
- The algorithm can match $v_t$ to a neighbor in $U$.
- The matching decision is **irrevocable**.

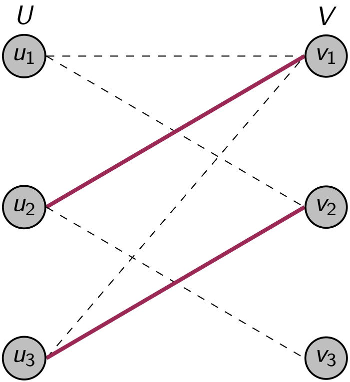



Figure: Online matching on a bipartite graph.

# 1 The performance of an algorithm

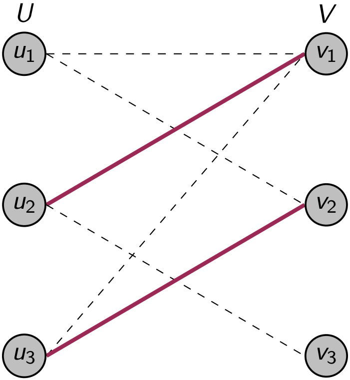



Figure: $ALG = 2$.

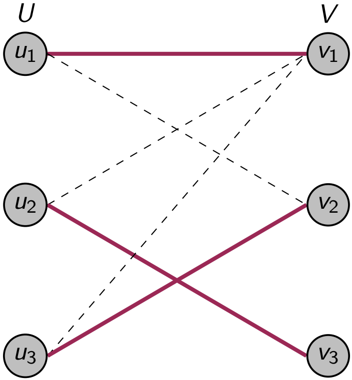



Figure: $OPT = 3$.

## 1 Competitive ratio

**Definition (informal)**

For $G \in \mathcal{G}$, where $\mathcal{G}$ is a family of graphs, the competitive ratio is defined as:

$$CR = \frac{\mathbb{E}(ALG(G))}{\mathbb{E}(OPT(G))}.$$

Note that $0 \leq CR \leq 1$.

# 1 The usual frameworks

- **Adversarial (Adv)**: $G$ can be any graph. The vertices of $V$ arrive in any order.
- **Stochastic (IID)**: The vertices of $V$ are drawn iid from a distribution.

Frameworks depend on the type of the graph!

# 1 Greedy Algorithm

**Algorithm 1:**

**Input:** a bipartite graph $G$
**Output:** a matching $M$
**for** $t = 1, \ldots, |V|$ **do**
  Match $v_t$ to any free neighbor chosen uniformly at random.
  Update $M$.
**end**

**Theorem:(informal)**

Adversarial setting (Mehta et al. 2013),

$$CR(\texttt{Greedy}) = \frac{1}{2}.$$

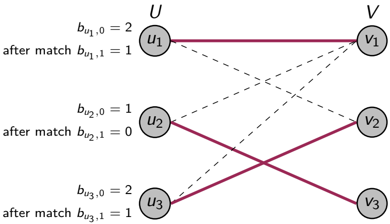
Stochastic setting (Mastin et al. 2013),

$$CR(\texttt{Greedy}) \geq 0.837.$$

# 1 Online $b$-matching problem: Balance algorithm

**Algorithm 2:**

**Input:** a bipartite graph $G$
**Output:** a matching $M$
**for** $t = 1, \dots, |V|$ **do**
  Match $v_t$ to a neighbor with highest remaining budget.
  Update $M$.
**end**




**Theorem: (informal)**

In the adversarial framework:
If $b_u = b$ (Kalyanasundaram et al. 2000),

$$CR(\texttt{Balance}) = 1 - \frac{1}{(1 + 1/b)^b}.$$

With different budgets (Albers et al. 2021),

$$CR(\texttt{Balance}) = 1 - \frac{1}{(1 + 1/b_{\min})^{b_{\min}}}.$$

$b_{\min} = \min_{u \in U} b_u$.

# 1 Differential equation method - the goal

In stochastic frameworks, **how does the matching process evolve over time?**

- Does it move smoothly toward equilibrium?
- Or does it fluctuate unpredictably due to randomness?

For $i = 0, 1, 2, \ldots$, we have a discrete time random process

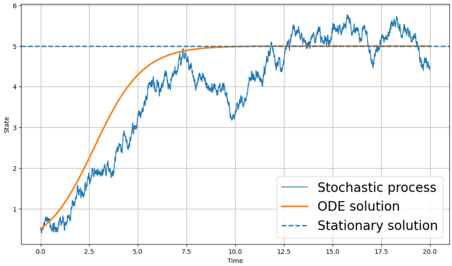
$$Y(i) = (Y_1(i), \ldots, Y_a(i)),$$

(e.g. numbers of edges, degrees, components in a random graph process).

Understand the *typical trajectory* of $Y$ as the system grows.

↓

Rely on ordinary differential equations.



# 1 Differential equation method - the flavour

Key idea → think of $\mathbb{E}[Y_k(i+1) - Y_k(i) \mid \mathcal{F}_i]$ as a gradient.

**Assumptions**

- **Approximate the drift.** $\mathbb{E}[Y_k(i+1) - Y_k(i) \mid \mathcal{F}_i] = F_k(i/n, Y_1/n, \dots, Y_a/n) + o(1)$, with $F$ "nice enough".
- **Check** $|Y_k(i+1) - Y_k(i)|$ is small enough.

nice enough = sufficiently smooth.

→ Then, with high probability, the random process $\frac{Y_k(i)}{n}$ stays very close to $y_k(t)$ the solution of $y_k{}'(t) = F_k(t, y_1(t), \dots, y_a(t))$.
→ Dynamic concentration of the process around its expected trajectory.

This result is known as the Wormald theorem (Wormald 1999; Warnke 2019).

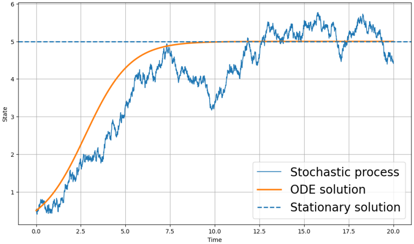
# 1 Differential equation method – the difficulty

⚠ In practice, the bottleneck is solving the ODE.

- The drift $F(\cdot)$ is often **non-linear and high-dimensional**.
- No closed-form solution in many real problems.
- Even numerically, the behavior may be tricky to understand.

Our strategy:

- **Study the stationary solution** (equilibrium).
- **Construct good approximations** of the ODE trajectory.
- **Solve the ODE** in closed form.



# More realistic setting: online matching with budget refills (Cherifa et al. 2024)

Clément Calauzènes

Vianney Perchet

## 2 From standard online matching to budget refills

**Constraint 1 + Constraint 2:** online arrivals and budget evolution.

### The standard online matching

- $G = (U, V, E)$ is a bipartite graph.
- Nodes in $U$ are offline and nodes in $V$ are online.
- Each node in $U$ has a budget $b_{u,t} \in \{0, 1\}$,

$$b_{u,t} = \begin{cases} b_{u,t-1} - 1 & \text{match}, \\ b_{u,t-1} & \text{no match}. \end{cases}$$

### Budget refills

- $G = (U, V, E)$ is a bipartite graph.
- Nodes in $U$ are offline and nodes in $V$ are online.
- Each node in $U$ has a budget $b_{u,t} \in \mathbb{N}$,

$$b_{u,t} = \begin{cases} b_{u,t-1} - 1 + r_t & \text{match}, \\ b_{u,t-1} + r_t & \text{no match}. \end{cases}$$

$r_t$ can be stochastic or deterministic.

# 2 Problem definition

Let G=(U,V,E) be a bipartite graph.

- $|U| = n, |V| = T$ with $T \geq n$.
- Nodes in $U$ are offline and nodes in $V$ are online.
- Each node in $U$ has a budget $b_{u,t} \geq 0$ at time $t \in [T]$.

A matching on G is a binary matrix $\mathbf{x} \in \{0,1\}^{n \times T}$ such that,

- $\forall (u,t) \in U \times V,\ (u,t) \notin E \Rightarrow x_{u,t} = 0$.
- $\forall t \in V,\ \sum_{u \in U} x_{u,t} \leq 1$.
- $\forall (u,t) \in U \times V,\ b_{u,t-1} < 1 \Rightarrow x_{u,t} = 0$.

**Two frameworks considered:**

- Stochastic framework.
- Adversarial framework.

## 2 The stochastic framework

$G$ is an Erdös - Rényi sparse random graph:

- Edges occurring independently with probability $p = a/n$.
- Each node in $U$ has a budget $b_{u,t} \in \mathbb{N}$. Budget dynamics:

$$b_{u,t} = \min(K,\, b_{u,t-1} - x_{u,t} + r_t), \qquad b_{u,0} = b_0,$$

$r_t$ is a realization of a Bernoulli random variable $\mathcal{B}(\frac{\beta}{n})$.
Budgets are capped by $K$.

## 2 The difficulty of the setting

- `Greedy` **in standard online matching**: the matching size built by `Greedy` satisfies,

$$\mathbb{E}\left[\texttt{Greedy}(G,t)-\texttt{Greedy}(G,t-1))|\texttt{Greedy}(G,t)\right]=1-\left(1-\frac{a}{n}\right)^{n-\texttt{Greedy}(G,t)}.$$

  - $\texttt{Greedy}(G,t)/n \xrightarrow{\text{w.h.p.}} z$, where $z$ is the solution of $\dot{z}(t)=1-e^{-a(1-z(t))}$.

- `Greedy` **in online matching matching with budget refill**: match only if budget $\geq 1$, thus,

$$\mathbb{E}\left[\texttt{Greedy}(G,t+1)-\texttt{Greedy}(G,t)|\texttt{Greedy}(G,t)\right]=1-\left(1-\frac{a}{n}\right)^{n-Y_0(t)}.$$

$\texttt{Greedy}(G,t)$ depends on $Y_k$, the number of nodes in $U$ with budget $k$.

  - **K dimensional problem!**

# 2 The matching size created by Greedy

**Theorem: (Cherifa et al. 2024) (informal)**

For $\frac{T}{n} \geq 1$, with high probability $\texttt{Greedy}(G, T)$ is given by,

$$\texttt{Greedy}(G, T) = nh(T/n) + o(n).$$

Where $h$ is solution of: $\dot{h}(\tau) = 1 - e^{-a(1-z_0(\tau))}$, and $z_0$ is the solution of,

$$\begin{cases} \dot{z}_0(\tau) = -z_0(\tau)\beta + z_1(\tau)g(z_0(\tau)) & \text{for } k = 0, \\ \dot{z}_k(\tau) = -\Delta z_k(\tau)\beta + \Delta z_{k+1}(\tau)g(z_0(\tau)) & \text{for } 1 \leq k \leq K-1, \\ \dot{z}_k(\tau) = \beta\, z_{k-1}(\tau) - z_k(\tau)g(z_0(\tau)) & \text{for } k = K, \\ \sum_{k=0}^{K} z_k(\tau) = 1. \end{cases} \tag{1}$$

---

$g(z_0(\tau)) = \frac{1-e^{-a+az_0(\tau)}}{1-z_0(\tau)}$, $\Delta z_k(\tau) = z_k(\tau) - z_{k-1}(\tau)$ and $0 \leq \tau \leq T/n$.

# 2 Solving the system

$$
\begin{cases}
\dot{h} = 1 - e^{-a(1-z_0)} \\
\dot{z}_0 = -z_0\,\beta + z_1\,\dfrac{1-e^{-a+az_0}}{1-z_0} \\
\dot{z}_k = (z_{k-1} - z_k)\beta + (z_{k+1} - z_k)\,\dfrac{1-e^{-a+az_0}}{1-z_0} \\
\dot{z}_K = \beta\, z_{K-1} - z_K\,\dfrac{1-e^{-a+az_0}}{1-z_0} \\
\sum_{k=0}^{K} z_k = 1
\end{cases}
$$

⚠ Hard to solve

- Non-linear and strongly coupled in $(z_0, \ldots, z_K)$.
- No closed-form even for small $K$.

💡 Our strategy

- Focus on the **stationary solution** $(z_0^*, \ldots, z_K^*)$: analyze its stability and derive performance guarantees.

# 2 Case $K = 1$

For $K = 1$, eq. (1) is reduced to

$$\begin{cases} \dot{z}_0(\tau) = -\beta\, z_0(\tau) + z_1(\tau) g(z_0(\tau)) \\ \dot{z}_1(\tau) = \beta z_0(\tau) - z_1(\tau) g(z_0(\tau)) \\ z_0(\tau) + z_1(\tau) = 1 \end{cases}$$

The stationary solution $({z_0}^*, {z_1}^*)$ is **exponentially stable**,

$$z_0^* = \frac{1}{\beta} - \frac{1}{a} W\left(\frac{a}{\beta} e^{-a\left(1-\frac{1}{\beta}\right)}\right),$$

$$z_1^* = z_0^* \frac{\beta}{g({z_0}^*)}.$$

---

$W$ is the Lambert function.

**Corollary: (Cherifa et al. 2024) (informal)**

For $\frac{T}{n} \geq 1$, with high probability,

$$|\texttt{Greedy}(G, T) - nh^*(T/n)| \leq C T^{1-\epsilon},$$

with,

$$h^*(x) = x\left(1 - e^{-a(1-{z_0}^*)}\right).$$

Here $\epsilon > 0$ and $C$ are known constants.

# 2 Case $K \geq 1$

The stationary solution of eq. (1) is **asymptotically stable** and is given by,

$$\left( {z_0}^*, {z_0}^* \frac{\beta}{g({z_0}^*)}, \dots, {z_0}^* \left( \frac{\beta}{g({z_0}^*)} \right)^K \right),$$

${z_0}^*$ is the unique solution of $\sum_{k=0}^{K} {z_0}^* \left( \frac{\beta}{g({z_0}^*)} \right)^k = 1.$

**Corollary: (Cherifa et al. 2024) (informal)**

For $\frac{T}{n} \geq 1$, with high probability,

$$|\texttt{Greedy}(G, T) - nh^*(T/n)| \leq o(T),$$

with $h^*(x) = x(1 - e^{-a(1-{z_0}^*)})$.

# 2 Difficulty of higher dimension

✔ Simple case: $K = 1$

- Only **two** budget levels: $\{0, 1\}$.
- Constraint $z_0 + z_1 = 1 \Rightarrow$ **one-dimensional ODE**.
- Every deviation is immediately pulled back.

Exponential stability
(fast convergence)

✘ General case: $K > 1$

- **Multiple** budget levels: $0, 1, \ldots, K$.
- Perturbations can **spread across levels**.
- In some directions pertubations decay **slowly**.

Only asymptotic stability
(slow convergence in some directions)

$\rightarrow$ **Stability strongly depends on** $K$

# 2 The technical reason behind asymptotic stability

Stability analysis of stationary solutions of an ODE system via the eigenvalues of the Jacobian matrix at the equilibrium point.

$$J = \begin{pmatrix} -\beta + {z_1}^* g'({z_0}^*) & g({z_0}^*) & 0 & \cdots & 0 \\ \beta + ({z_2}^* - {z_1}^*) g'({z_0}^*) & -\beta - g({z_0}^*) & g({z_0}^*) & \ddots & \vdots \\ ({z_3}^* - {z_2}^*) g'({z_0}^*) & \beta & -\beta - g({z_0}^*) & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & g({z_0}^*) \\ -{z_K}^* g'({z_0}^*) & \cdots & 0 & \beta & -g({z_0}^*). \end{pmatrix}.$$

The spectrum of $J$ lies in the left half-plane, apart from one eigenvalue located at zero.

# 2 Convergence of the CR

**Theorem: (Cherifa et al. 2024) (informal)**

$$CR(\texttt{Greedy}) \geq 1 - \frac{\beta T {z_0}^* \left(\frac{\beta}{g({z_0}^*)}\right)^K + n{z_0}^* \sum_{k=1}^K k \left(\frac{\beta}{g({z_0}^*)}\right)^k}{nb_0 + \beta T} + O(T^{-1/4}),$$

When $T, n, K$ tends to $+\infty$:

$$|CR(\texttt{Greedy}) - 1| \to 0,$$

where ${z_0}^*$ and $g$ defined as previously.

**Interpretation:**

- When $T \to \infty$: $CR(\texttt{Greedy}) \to \frac{g(z_0^*)(1-z_0^*)}{\beta}$ (the CR depends only on the stationary point).
- When $K, n \to \infty$: $\frac{g(z_0^*)(1-z_0^*)}{\beta} \to 1$ (Greedy wastes almost nothing when budgets grow).

## 2 The adversarial framework

$G = (U, V, E)$ is a bipartite graph generated by an oblivious adversary:

- $|U| = n$ and $|V| = T$ with $T \geq n$.
- Each node in $U$ has a budget $b_{u,t} \in \mathbb{N}$. Budget dynamics:

$$b_{u,t} = b_{u,t-1} - x_{u,t} + \mathbb{1}_{t \bmod m = 0}, \qquad b_{u,0} = b_0.$$

💡 $m$ is the parameter of the frequency of the refills.

## 2 Matching size of `Balance`

**Theorem: (Cherifa et al. 2024) (informal)**

For $m \geq \sqrt{T}$,

$$CR(\texttt{Balance}) \leq 1 - \frac{1}{\left(1 + \frac{1}{b_0}\right)^{b_0}}.$$

**Theorem:(Cherifa et al. 2024)(informal)**

For $m = o(\sqrt{T})$,

$$CR(\texttt{Balance}) \leq 1 - \underbrace{\frac{(1-\alpha)}{e^{(1-\alpha)}}}_{\simeq 0.73325\ldots}.$$

where $\alpha$ is defined by $\frac{1}{2} = \int_0^\alpha \frac{xe^x}{1-x} \mathrm{d}x$.

No deterministic algorithm can beat `Balance`.

# More realistic setting: online matching on stochastic block model (Cherifa et al. 2025)

Clément Calauzènes

Vianney Perchet

# 3 From Erdös - Rényi graph to stochastic block model

**Constraint 1 + Constraint 3**: online arrivals and community structure.

**Erdös - Rényi model**

- All ads are statistically identical.
- Every user connects to every ads with same probability $p$.
- Matching decisions depend only on availability (budget 0 or 1).
- ❌ no community structure.

**Stochastic block model**

- Users and ads are divided into communities.
- Edge probability depends on class pair $p_{c,d}$.
- Each class evolves differently and affects the others.

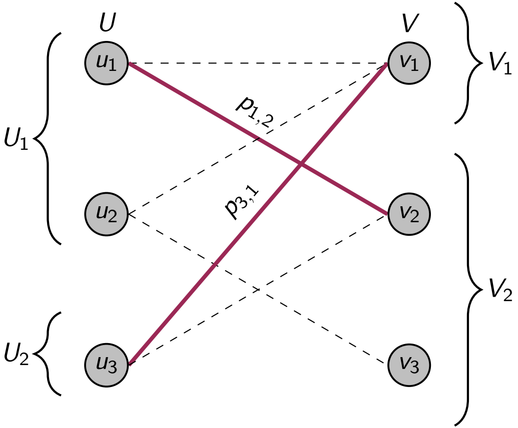
**Matching depends on class proportions and compatibility matrix.**

# 3 Problem setting

Let $G$ be a bipartite graph $G = (U, V, E)$.

- $|U| = n$ offline nodes, $|V| = T \geq n$ online arrivals.
- Offline and online nodes have classes in $\mathcal{C}$ and $\mathcal{D}$.
- Each $u \in U$ and $v \in V$ has a class $c(u) \sim \mu$ and $d(v) \sim \nu$.
- Conditional on classes,

$$\Pr[(u, v) \in E] = p_{c(u), d(v)}.$$




Generalization of Erdös - Rényi

# 3 Notations and assumptions

- The sparse regime: every $p_{c,d} = \frac{a_{c,d}}{n}$.
- $b_c$ is the proportion of nodes in class $c$.
- $M_c(t)$ is the number of matched nodes in class $c$.
- $M(t) := \sum_{c \in \mathcal{C}} M_c(t)$ is the size of the matching constructed.
- $\mathcal{F}_c(t)$ is the set of unmatched nodes of class $c$.

Two cases of study:

- The probabilities $p_{c,d}$ are **known**.
- The probabilities $p_{c,d}$ are **unknown**.

## 3 Known $p_{c,d}$ - Myopic strategy

**Goal.** When a node of class $d$ arrives, pick an offline class $c$ that yields a good chance of matching **without exhausting capacities**.

**Idea.** Pre-compute an **optimal plan** $Q^*$ such that:

$$Q^* \in \arg\max_{Q} \sum_{c,d} Q(c,d)p(c,d),$$

$$\text{s.t} \quad \sum_{d} Q(c,d)\nu(d) = b_c, \ \forall c \in \mathcal{C},$$

$$\text{and} \ \sum_{c} Q(c,d)\nu(d) = \nu(d), \forall d \in \mathcal{D}.$$

**Intuition.** $Q^*$ is a **smart allocation**: It sends each type-$d$ arrival to the most promising offline classes while respecting capacities.

# Known $p_{c,d}$ - `Myopic` strategy

**Algorithm 3:** Myopic algorithm

**Output:** Updated matching $M(t)$

Compute the optimal $Q^*$.

**for** $t \in [T]$ **do**

  Choose $c_t \in \mathcal{C}$ at random with probability $Q^*(c_t, d_t)/\nu(d_t)$.

  **if** $\mathcal{F}_{c_t}(t) = \emptyset$ **then**

    $M(t) = M(t-1)$.

  **else**

    $M(t) = M(t-1) \cup \{(u_t, t)\}$ for $u_t \sim \text{unif}(\mathcal{F}_{c_t}(t))$.

# 3 From Erdős–Rényi to SBM: difficulty for `Myopic`

- `Myopic` **in Erdős–Rényi (homogeneous case)**
  - One global matching process $M(t)$.
  - $\frac{M(t)}{n} \xrightarrow{\text{w.h.p.}} y(t), \qquad \dot{y}(t) = 1 - e^{-a(1-y(t))}$.
  - **Scalar ODE, explicit closed form for** $y$.

- `Myopic` **in SBM (structured, heterogeneous case)**
  - One matching process *per class* $c$: $M_c(t)$.
  - With high probability, for each class $c$: $\frac{M_c(t)}{n} \xrightarrow{\text{w.h.p.}} y_c(t)$, where $y_c$ solves $\dot{y}_c(t) = \sum_{d=1}^{D} \left(1 - e^{-a_{c,d}\left(b_c - y_c(t)\right)}\right) Q^*(c,d)$.
  - **No closed form:** depends on the *class* $c$, all neighbor classes $d$, and the optimal plan $Q^*$.

# 3 Known $p_{c,d}$ - the matching size of `Myopic` strategy

**Theorem: (Cherifa et al. 2025) (informal)**

Let $T = \alpha n$, and $y_c : [0, \alpha] \to \mathbb{R}$ be the solution of the following ODE,

$$\dot{y}_c(s) = \sum_{d=1}^{D} \left(1 - e^{-a_{c,d}(b_c - y_c(s))}\right) Q^*(c, d),$$
$$y_c(0) = 0.$$

Then, for each class $c \in \mathcal{C}$, $M_c(t)$ satisfies w.h.p,

$$|M_c(t)/n - y_c(t/n)| \leq \mathcal{O}_{L_c,\alpha}(n^{-1/3}).$$

Moreover,$y_c = \tilde{y}_c - e_c$, with $\tilde{y}_c(t) = b_c\left(1 - e^{-tL_c}\right)$, and $e_c$ satisfies,

$$e_c(t) \leq J_c(1 - e^{-L_c t})/L_c.$$

---

where, $J_c$ and $L_c$ are known constants.

# 3 Known $p_{c,d}$ - Balance algorithm

**Algorithm 4:** Balance

**Output:** Updated matching $M(t)$

**for** $t \in [T]$ **do**

Choose $c_t = \arg\max_{c\in[C]} \sum_{j=1}^{D} (1 - (1 - \frac{a_{c,j}}{n})^{nb_c - M_c(t)})\nu(j)$

**if** $\mathcal{F}_{c_t}(t) = \emptyset$ **then**

$M(t) = M(t-1)$.

**else**

$M(t) = M(t-1) \cup \{(u_t, t)\}$ for $u_t \sim \text{unif}(\mathcal{F}_{c_t}(t))$.

# 3 Known $p_{c,d}$ - Balance is harder than Greedy/Myopic

Greedy: smooth decisions

- Matching probability is a **continuous** function of the state.
- **ODE approximation works**.

Balance: switches to the class with **max availability**.

- Drift has an **indicator of the maximizer**:

$$\mathbb{E}\Big[M_c(t+1)-M_c(t) \mid n, \mathbf{M}(t)\Big] = H_{c,b_c,n}(M_c(t))\cdot\underbrace{\mathbf{1}\Big\{H_{c,b_c,n}(M_c(t)) = \max_{k\in[C]} H_{k,b_k,n}(M_k(t))\Big\}}_{\text{discontinuous at ties}}$$

🚫 **Drift is not Lipschitz** $\Rightarrow$ ODE method fails.

➔ A stronger tool is needed: Differential Inclusions.

---

$H_{c,b_c,n}(x) = \sum_{d=1}^{D}(1-(1-a_{c,d}/n)^{nb_c-x})\,\nu(d)$

# 3 Differential Inclusions

**ODE: one direction at each point**

$$\dot{x}(t) = f(x(t)).$$

**Differential inclusion: many possible directions**

$$\dot{x}(t) \in F(x(t)),$$

where $F(x)$ is a **set-valued map**.

**It is needed here because:**

- Balance **switches** between classes: drift is not continuous.
- DI **naturally handles switching** and multiple possible drifts.

# 3 Convergence to a Differential Inclusion (Gast et al. 2011)

**Does the matching process converge to a DI?**

**Setting** (Markov chain with small drift and noise)

$$Y^N(k+1) = Y^N(k) + g^N(Y^N(k)) + U^N(k+1)$$

**Assumptions**

- **Vanishing drift:** $g^N = \gamma^N f^N$, with $\gamma^N \to 0$.
- **Small noise:** $U^N$ is a martingale difference (no big jumps).

**Key idea:** If the drift is discontinuous, the limit is **set-valued**:

$$\dot{y}(t) \in F(y(t)).$$

→ With high probability, $Y^N$ stays **close** to a trajectory of $\dot{y}(t) \in F(y(t))$.

**Even with discontinuities, the stochastic process has a deterministic limit.**

# 3 Known $p_{c,d}$ – matching size of Balance

**Our contribution.** the matching process under Balance **converges to a deterministic differential inclusion**.

**Theorem (informal) (Cherifa et al. 2025)**

Let $T = \alpha n$, and let $m$ be the unique solution of the differential inclusion

$$\dot{m}(t) \in F(m(t)) := \mathrm{conv}\left\{ f_{c,b_c}(m_c(t))\, e_c \, : \;\; c \in \operatorname*{arg\,max}_{k \in [C]} f_{k,b_k}(m_k(t)) \right\},$$

where $f_{c,b_c}(x) = \sum_{d=1}^{D} \left(1 - e^{-a_{c,d}(b_c - x)}\right) \nu(d)$. Then, for all $t \in [T]$ and $c \in \mathcal{C}$, with high probability,

$$\left| \frac{M_c(t)}{n} - m_c(t/n) \right| \leq \mathcal{A}_{\alpha,c,L}/n,$$

Here, $m_c$ is **is known in closed form**.

---

$e_c$ is the $c$-th basis vector of $\mathbb{R}^{|\mathcal{C}|}$ and $L$, $\mathcal{A}_{\alpha,c,L}$, are known constants.

→ **First analysis of Balance in sparse random SBM.**

## 3 Unknown $p_{c,d}$ – from Matching to Bandit Learning

**In practice: the SBM parameters are unknown.**

- The connection rates $a_{c,d}$ are not given to the algorithm.
- Matching outcomes are the **only source of information**.
- The platform must decide **who to match** *and* **learn connection probabilities** at the same time.

**Decision-making + Statistical learning are coupled.**

### Bandit View

- Each class $c \in \mathcal{C}$ behaves like an **arm**.
- Playing arm $c_t$ at time $t$ reveals a **Bernoulli reward**:

$$\text{reward}_t = \mathbf{1}\{\text{match succeeds}\}$$

- We must balance:
  - **Exploration:** try classes to estimate $a_{c,d}$.
  - **Exploitation:** match with the best-estimated class.

**Our goal:** match with unknown $p_{c,d} = a_{c,d}/n$.

- Try each classes in $C$ to build estimates of $D_{c,d} = \left(1 - \frac{a_{c,d}}{n}\right)^{nb_c - M_c(t)}$.

**Explore then commit (ETC)**

For $t \leq T_{\text{explore}}$:

- try all classes uniformly,
- collect match outcomes (match/ no match),
- estimate all $D_{c,d}$.

- Freeze the estimates, and run Balance.

# 3 The regret

For each class $c \in \mathcal{C}$,

- $M_c(t)$ is the number of matches made by the $\texttt{Balance}$ up to time $t$.
- $\hat{M}_c(t)$ is the number of matches made by $\texttt{ETC} - \texttt{balance}$ up to time $t$.

**Theorem: (Cherifa et al. 2025) (informal)**

Let $R(T) = \sum_{c \in \mathcal{C}} M_c(T) - \hat{M}_c(T)$ denote the regret of $\texttt{ETC} - \texttt{balance}$.
Suppose the exploration phase lasts for $T_{\text{explore}} = T^{\frac{q+3}{4}}$, for some $0 < q < 1$.
Then the regret satisfies:

$$R(T) = \mathcal{O}_q(T^{\frac{q+3}{4}}).$$

# 3 ETC instead of UCB

**Goal:** Control the regret

$$R(T) = \sum_{c \in \mathcal{C}} \left( M_c(T) - \hat{M}_c(T) \right)$$

$$\leq \sum_{c \in \mathcal{C}} \Big( \underbrace{|M_c(T) - nm_c(T/n)|}_{DI\ approximation\ for\ \texttt{Balance}} + \underbrace{|nm_c(T/n) - n\hat{m}_c(T/n)|}_{DI\ learning\ error} - \underbrace{|\hat{M}_c(T) - n\hat{m}_c(t)(T/n)|}_{DI\ error\ for\ the\ learning\ algorithm} \Big)$$

If UCB is used:

- $\hat{m}_c$ is hard to solve.
- The **bonus term in estimation changes at every round** $\mathrm{UCB}_{c,d}(t) = \hat{D}_{c,d} + \sqrt{\frac{\alpha \log t}{T_{c,d}}}$.
- UCB **mixes exploration and exploitation**.

With ETC:

- $\hat{m}_c$ has the same structure as $m_c$ with estimated parameters.
- **Explore first**: collect unbiased information.
- Then **freeze estimates** $\Rightarrow$ Balance has **fixed parameters.**

# 4 Conclusion and future works

- More sophisticated refills dynamics.

In our setting:

$$b_{u,t} = b_{u,t-1} - x_{u,t} + r_t$$

Where $r_t$ is a Bernoulli random variable.

Generalization:

- Poisson Refills : $r_t$ is a realization of Poisson random variables.
- State dependant refills, nodes with low budgets get refills.

- Simple refills dynamics in Geometric random graphs and configuration models.
- Stochastic block model with budget refills.
  1. Matching will depend on budgets and on classes affinities.
  2. More coupled system of ODE to analyze.

# Thank you 😃🙏

# 5 Bibliography I


Albers, Susanne and Sebastian Schubert (2021). "Optimal Algorithms for Online b-Matching with Variable Vertex Capacities". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021)*. Ed. by Mary Wootters and Laura Sanità. Vol. 207. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2:1–2:18.

Cherifa, Maria, Clement Calauzenes, and Vianney Perchet (May 2024). "Dynamic online matching with budget refills". In: DOI: 10.48550/arXiv.2405.09920.

— (May 2025). "Online matching on stochastic block model". In: DOI: 10.48550/arXiv.2405.09920.

Gast, Nicolas and Bruno Gaujal (Apr. 2011). *Markov chains with discontinuous drifts have differential inclusions limits. Application to stochastic stability and mean field approximation.* Research Report RR-7315.

# 5 Bibliography II


Kalyanasundaram, Bala and Kirk R. Pruhs (2000). “An optimal deterministic algorithm for online b-matching”. In: *Theoretical Computer Science* 233.1, pp. 319–325.

Mastin, Andrew and Patrick Jaillet (2013). *Greedy Online Bipartite Matching on Random Graphs*. DOI: 10.48550/ARXIV.1307.2536. URL: https://arxiv.org/abs/1307.2536.

Mehta, Aranyak et al. (2013). *Online Matching and Ad Allocation*. Cambridge University Press.

Warnke, Lutz (2019). “On Wormald’s differential equation method”. In: *ArXiv* abs/1905.08928.

Wormald, Nicolas (1999). *The differential equation method for random graph processes and greedy algorithms*. URL: https://users.monash.edu.au/~nwormald/papers/de.pdf.