# Decentralized Asynchronous Multi-player Bandits

**Setting:**
Players join or leave the system at arbitrary time steps.

**Difficulty:**
Players do not know when others come → unavoidable collisions
Players do not know when others leave → The optimal arms can change

**Algorithm:**
Adaptive change between exploration and exploitation → players can detect the change of optimal arms
Double Selection (Players pull arms at two consecutive steps) → players do not exploit the same optimal arms with others.

**Analysis:**
an upper bound

# Setting

**Problem formulation:**

- Let $1 \leq T_{\text{start}}^j < T_{\text{end}}^j \leq T$. A player is active at step $t$ means that she needs to pull an arm at this step. Let $m_t$ denote the number of active players at step $t$.
- Each player $j \in [M]$ is only active from $T_{\text{start}}^j$ to $T_{\text{end}}^j$.
- Player $j$ is only aware of $T$, but does not know $T_{\text{start}}^j$ and $T_{\text{end}}^j$.
- At each step $t \in [T_{\text{start}}^j, T_{\text{end}}^j]$, player $j$ pulls an arm $\pi^j(t) \in [K]$ and observes $< r^j(t), \eta^j(t) >$.

**Regret Definition:**

$$\mathbb{E}[R(T)] := \sum_{t \leq T} \sum_{k \leq m_t} \mu_k - \mathbb{E}\left[\sum_{t \leq T} \sum_{j : T_{\text{start}}^j \leq t \leq T_{\text{end}}^j} r^j(t)\right].$$

**Assumption:**

- There exists a constant $m$ such that for any $t$, $m_t \leq m \leq K/2$.

# Challenge

## Challenge 1

Players do not know when others join the system.

Previous Init or Com phase does not work. However, I do not want to assume a known *j* here. And the assumption is not reasonable in the setting.

Thus, it is difficult to avoid collisions.

## Challenge 2

Players do not know when others leave the system

The optimal arms depend on the number of active players. It can change.

When a player who is exploiting her optimal arm leaves the system, the left arms that are still exploited by players may become sub-optimal.

# Algorithm

**Challenge 1:**

difficult to avoid collisions

**Solution 1:**

- There is no `Init` or `Com` phase; each player independently executes her own policy.
- Player $j$ maintains a set $\mathcal{A}^j$, representing the arms believed to be occupied by other players.
- Player $j$ explores arms in $[K] \setminus \mathcal{A}^j$ uniformly at random.
- If arms in $[K] \setminus \mathcal{A}^j$ frequently result in collisions, player $j$ infers that those arms are likely being exploited by others and adds them to $\mathcal{A}^j$.

**Challenge 2:**

change of optimal arms

**Solution 2:**

- Player $j$ always pulls arms in $\mathcal{A}^j$ with a small probability $\varepsilon$.
- If arms in $\mathcal{A}^j$ frequently result in non-collisions, player $j$ infers that those arms are likely being released by others and removes them from $\mathcal{A}^j$.

## Algorithmic Framework

Player $j$ adaptively changes between an exploration phase and an exploitation phase:

- **Exploration phase:** If there exists an arm $k$ such that $\mathrm{LCB}_k^j \geq \mathrm{UCB}_\ell^j$ for all $\ell \neq k$, $\ell \in [K] \setminus \mathcal{A}^j$, then player $j$ transitions to the exploitation phase and pulls arm $k$ with probability $1 - \varepsilon$.
- **Exploitation phase:** If player $j$ detects that an arm in $\mathcal{A}^j$ has been released, she switches back to the exploration phase.

# Algorithm

**Challenge 1:**

difficult to avoid collisions

**Solution 1:**

- There is no `Init` or `Com` phase; each player independently executes her own policy.
- Player $j$ maintains a set $\mathcal{A}^j$, representing the arms believed to be occupied by other players.
- Player $j$ explores arms in $[K] \setminus \mathcal{A}^j$ uniformly at random.
- If arms in $[K] \setminus \mathcal{A}^j$ frequently result in collisions, player $j$ infers that those arms are likely being exploited by others and adds them to $\mathcal{A}^j$.

**Challenge 2:**

change of optimal arms

**Solution 2:**

- Player $j$ always pulls arms in $\mathcal{A}^j$ with a small probability $\varepsilon$.
- If arms in $\mathcal{A}^j$ frequently result in non-collisions, player $j$ infers that those arms are likely being released by others and removes them from $\mathcal{A}^j$.

Since player *j* does not always exploit *k*, others may also set *k* as exploitation arm!

## Algorithmic Framework

Player $j$ adaptively changes between an exploration phase and an exploitation phase:

- **Exploration phase:** If there exists an arm $k$ such that $\mathrm{LCB}_k^j \geq \mathrm{UCB}_\ell^j$ for all $\ell \neq k$, $\ell \in [K] \setminus \mathcal{A}^j$, then player $j$ transitions to the exploitation phase and pulls arm $k$ with probability $1 - \varepsilon$.
- **Exploitation phase:** If player $j$ detects that an arm in $\mathcal{A}^j$ has been released, she switches back to the exploration phase.

# Algorithm

**DoubleSelection**

- **Exploration phase:** player $j$ samples $k \sim \mathrm{Uniform}([K] \setminus \mathcal{A}^j)$.
  - w.p. $1 - \varepsilon$: pulls $k$ twice;
  - w.p. $\varepsilon$: pull arm $k$ once, then pull an arm $k' \sim \mathrm{Uniform}(\mathcal{A}^j)$.
- **Exploitation phase:** let $\hat{k}^j$ denote player $j$'s exploitation arm.
  - w.p. $1 - \varepsilon$: pulls $\hat{k}^j$ twice;
  - w.p. $\varepsilon$: pulls $\hat{k}^j$ once, then pull an arm $k' \sim \mathrm{Uniform}(\mathcal{A}^j)$.

Therefore, when a player wants to enter the exploitation phase, she needs to find an arm k satisfying:

- **Condition 1:** $\eta_{k_1}(t-1) + \eta_{k_2}(t) = 0$, where $k_1 = k_2 = k$;
- **Condition 2:** $\mathrm{LCB}^j_k \geq \mathrm{UCB}^j_\ell$ for all $\ell \neq k$, $\ell \in [K] \setminus \mathcal{A}^j$.

# Algorithm

Let $\mathcal{P}_k^j, \mathcal{Q}_k^j$ denote two queues with fixed length $L_p = 866 \ln T$ and $L_q = 570 \ln T$, respectively.

Let $T_o^j, T_r^j$ denote the numbers of time steps that are required for player $j$ to identify an occupied arm $k$ and a released arm $k$, respectively.

**To solve Challenge 1**

- At step $t$, if $k1 = k_2$ and they are both sampled from $[K] \setminus \mathcal{A}^j$, then player $j$ adds $[\eta_{k_1}(t-1) \cdot \eta_{k_2}(t)]$ into a queue $\mathcal{P}_k^j$.
- If there exists an arm $k$ s.t. $\sum_{i \in \mathcal{P}_k^j} i \geq \lceil 0.85 L_p \rceil$, then player $j$ adds $k$ to $\mathcal{A}^j$.

**Lemma 1.**
With probability at least $1 - 1/T^2$:
i) If arm k is occupied and remains occupied thereafter, player $j$ will add $k$ to $\mathcal{A}^j(t)$ with $E[T_o^j] \leq 1926 K ln T$ time steps;
ii) If arm $k$ is not occupied and remains not occupied thereafter, player $j$ will not add $k$ to $A^j(t)$.

**To solve Challenge 2**

- At step $t$, if $k$ is sampled from $\mathcal{A}^j$, then player $j$ adds $[1 - \eta_k(t)]$ into a queue $\mathcal{Q}_k^j$.
- If there exists an arm $k$ s.t. $\sum_{i \in \mathcal{Q}_k^j} i \geq \lceil 0.142 L_q \rceil$, then player $j$ removes $k$ from $\mathcal{A}^j$.

**Lemma 2.**
With probability at least $1 - 1/T^2$:
i) If arm $k$ is released and never occupied again, player $j$ will remove $k$ from $\mathcal{A}^j(t)$ with $E[T_r^j] \leq 1141 m ln T/\varepsilon$ time steps;
ii) If arm $k$ is not released and remains not released thereafter, player $j$ will not remove $k$ from $A^j(t)$.

# Analysis

**Theorem 1.**

Given $K$ arms and $M$ players, and let $\varepsilon = \min\{\sqrt{\frac{1141m^3 \ln(T)}{2T}}, \frac{1}{K}, \frac{1}{10}\}$, the regret of Algorithm 1 is bounded by

$$\mathbb{E}[R(T)] \leq \frac{576emKM\log(T)}{\Delta^2} + 96m^{3/2}M\sqrt{T\ln(T)} + 7704m^2KM\ln(T) + (4emKM)^2 ,$$

where $\Delta := \min_{k \leq m}(\mu_k - \mu_{k+1})$.

$\mathcal{O}(\log T/\Delta^2)$ **arises from Challenge 1:**
Players cannot completely avoid collisions, leading to a regret of $\mathcal{O}(\log T/\Delta^2)$ instead of the standard $\mathcal{O}(\log T/\Delta)$.
$\mathcal{O}(\sqrt{T}\log T)$ **incurs from Challenge 2:**
The set of optimal arms may change over time, so players must pull occupied arms with a small probability. This persistent exploration contributes a regret of $\mathcal{O}(\sqrt{T}\log T)$.

**Corollary 1.**

Given $K$ arms and $M$ players, $\varepsilon = \min\{\sqrt{\frac{1141K^3 \ln(T)}{16T}}, \frac{1}{K}, \frac{1}{10}\}$, the regret of Algorithm 1 is bounded by

$$R(T) \leq \frac{288eK^2M\log(T)}{\Delta^2} + 34K^{3/2}M\sqrt{T\ln(T)} + 1926K^3M\ln(T) + (3eK^2M)^2 ,$$