Decentralized Asynchronous Multi-player Bandits

Jingqi Fan, Canzhe Zhao, Shuai Li, Siwei Wang*

https://arxiv.org/abs/2509.25824

Problem formulation:

- M players, K arms, T total steps.
- Let $[M] := \{1, ..., M\}$ and $[K] := \{1, ..., K\}$.
- Let $1 \leq T^j_{\text{start}} < T^j_{\text{end}} \leq T$. 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^j_{start} to T^j_{end}.
- Player j is only aware of T, but does not know T^j_{start} and T^j_{end}.
- At each step $t \in [T^j_{\text{start}}, T^j_{\text{end}}]$, player j pulls an arm $\pi^j(t) \in [K]$.
- She observes $< r^j(t), \eta^j(t) >$, where
  1. $r^j(t) := X^j(t)[1 - \eta^j(t)]$ is a reward, and $X^j(t) \sim \text{Bernoulli}(\mu_{\pi^j(t)})$;
  2. $\eta^j(t) := \mathbb{1}\left[\exists j' \neq j, j' \in [M] : \pi^j(t) = \pi^{j'}(t)\right]$ is a collision indicator.

Assumption:

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

Regret Definition:

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

where μ_k is the k-th biggest reward expectation. $\mu_1 > \mu_2 > \cdots > \mu_K$.

	Environment	Com	Async setting	Regret bound
Boursier and Perchet [2019]	Decentralized	No	Players arrive at different times but never leave.	$\mathcal{O}\left(\frac{KM\log T}{\Delta_{(1)}^2}+\frac{KM^2\log T}{\mu_M}\right)$
Dakdouk [2022]	Decentralized	Yes	Activation probability p	$\mathcal{O}\left(\max\left\{K^2, \frac{\log(KT)}{Mp(1-p/K)^M}\right\}T^{2/3}\right)$
Richard et al. [2024]	Centralized	Yes	Known activation probability p	$\mathcal{O}\left(\sqrt{KT\log(KT)\min\{K, Mp\}}\right)$
Richard et al. [2024]	Centralized	Yes	Known activation probability p	$\mathcal{O}\left(\frac{(K^2+(1+p)M^2)\log(KT)}{\Delta_{(2)}}\right)$
ACE	Decentralized	No	Players arrive and leave arbitrarily over time.	$\mathcal{O}\left(m^{3/2}M\sqrt{T\ln T}+\frac{mKM\log T}{\Delta_{(3)}^2}\right)$

Note:
Here “Com” column indicates whether direct communication (rather than via collision) is allowed.
Our setting is more general and the assumption is mild.

Challenge 1

Players do not know when others join the system.

Previous communication phase does not work. A player can join at any time and break the communication.

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.

Challenge 1:

difficult to avoid collisions

Solution 1:

- There is no `Communication` 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 occupied (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 ε.
- 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 $\mathcal{A}^j$.

Challenge 1:

difficult to avoid collisions

Solution 1:

- There is no `Communication` 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 occupied (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 ε.
- 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 of ACE

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.

Challenge 1:

difficult to avoid collisions

Solution 1:

- There is no `Communication` 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 occupied (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 ε.
- 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 of ACE

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.

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

DoubleSelection

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

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

DoubleSelection

- **Exploration phase:** player j samples $k \sim \mathrm{Uniform}([K] \setminus \mathcal{A}^j)$.
  - w.p. $1-\varepsilon$: pulls k twice;
  - w.p. ε: 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. ε: 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:

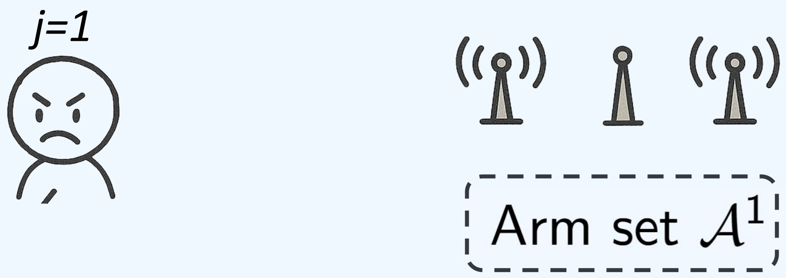
- **Condition 1:** $\eta_{k_1}(t-1) + \eta_{k_2}(t) = 0$, where $k_1 = k_2 = k$;
- **Condition 2:** $\mathrm{LCB}^j_k \geq \mathrm{UCB}^j_\ell$ for all $\ell \neq k$, $\ell \in [K] \setminus \mathcal{A}^j$.

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

DoubleSelection

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

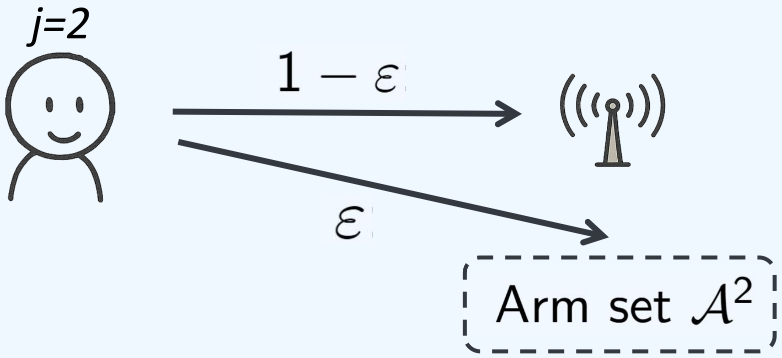
Exploration



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 $\ell \neq k$, $\ell \in [K] \setminus \mathcal{A}^j$.

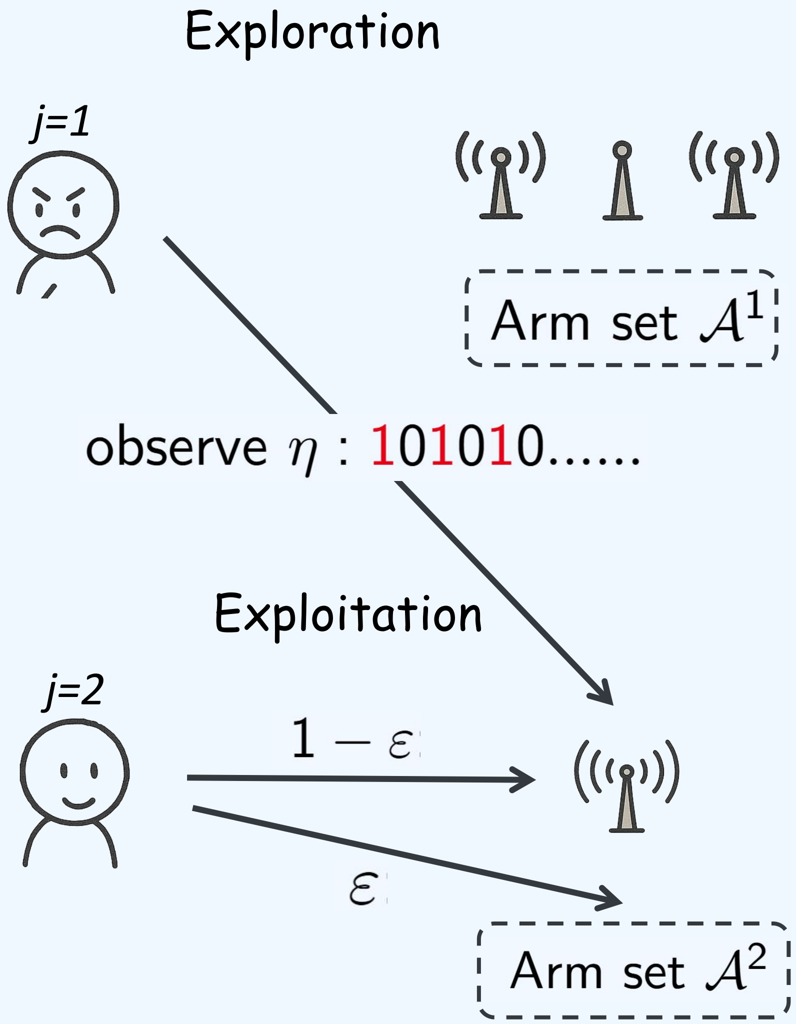
Exploitation



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

DoubleSelection

- **Exploration phase:** player j samples $k \sim \mathrm{Uniform}([K] \setminus \mathcal{A}^j)$.
  - w.p. $1-\varepsilon$: pulls k twice;
  - w.p. ε: 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. ε: 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:** $\mathrm{LCB}_k^j \geq \mathrm{UCB}_\ell^j$ for all $\ell \neq k$, $\ell \in [K] \setminus \mathcal{A}^j$.

Some Algorithmic Definition

- 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 number of time steps that are required for player j to identify an occupied arm k and a released arm k, respectively.
- We also define:

$$\hat{\mu}_k^j(t) := \frac{\sum_{t'=1}^{t} r_k^j(t')\,\mathbb{1}\{\eta_k(t') = 0\}}{N_k^j(t)}, \qquad N_k^j(t) := \sum_{t'=1}^{t} \mathbb{1}\{\pi^j(t') = k,\ \eta_k(t') = 0\},$$

$$\mathrm{UCB}_k^j(t) := \hat{\mu}_k^j(t) + \sqrt{\frac{6 \log T}{N_k^j(t)}}, \qquad \mathrm{LCB}_k^j(t) := \hat{\mu}_k^j(t) - \sqrt{\frac{6 \log T}{N_k^j(t)}}.$$

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 $A^j(t)$.

Proof Skectch: Distingush Events via Collison Probablity

Let $k \in \mathcal{A}^j$. player j pulls arm k. Then she receives a collision or non-collision.

For the Adding:

if arm $k \in \mathcal{A}^j$ is not occupied

if arm $k \in \mathcal{A}^j$ is occupied

0 $\quad 1 - \frac{1}{2e}$ $\quad 1 - \varepsilon$ $\quad 1$

collision probability axis

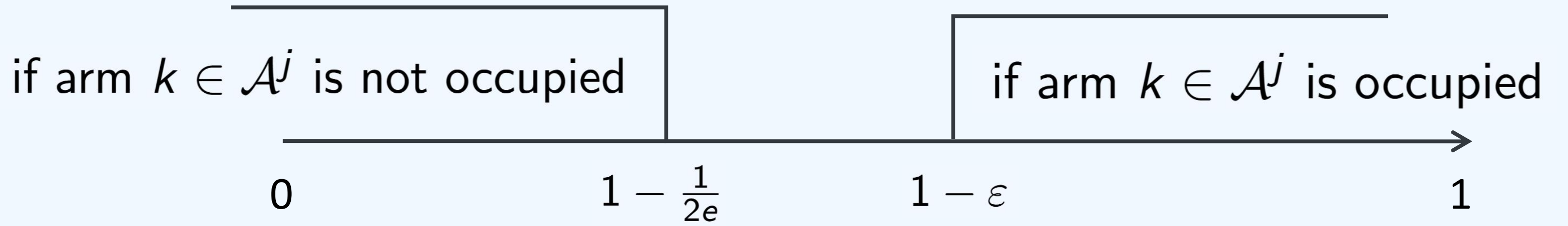
arm k is occupied
a player is exploiting it
the collision prob. ↑

Take at most $\mathcal{O}(K \ln T)$ steps to separate them w.p. $1 - \frac{1}{T^2}$.

Proof Skectch: Distingush Events via Collison Probablity

Let $k \in \mathcal{A}^j$. player j pulls arm k. Then she receives a collision or non-collision.

For the Adding:

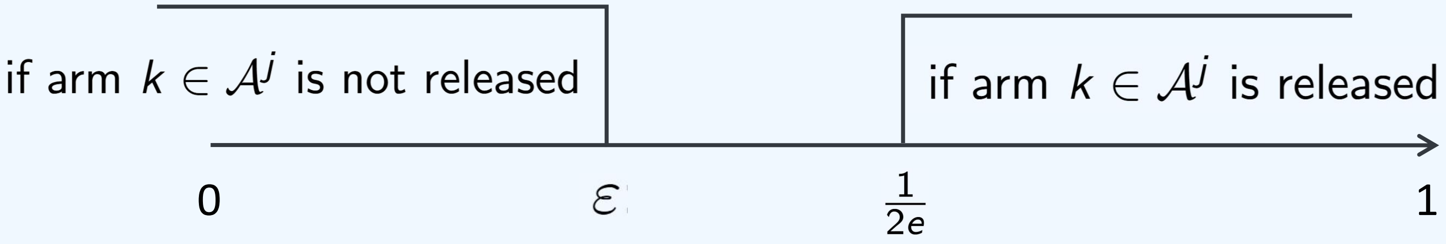


collision probability axis

arm k is occupied
a player is exploiting it
the collision prob. ↑

Take at most $\mathcal{O}(K \ln T)$ steps to separate them w.p. $1 - \frac{1}{T^2}$.

For the Removing:



non-collision probability axis

arm k is released
no player is exploiting it
the collision prob. ↓
the non-collision prob. ↑

Take at most $\mathcal{O}(\frac{m \ln T}{\varepsilon})$ steps to separate them w.p. $1 - \frac{1}{T^2}$.

Proof Skectch: Distingush Events via Collison Probablity

Let $k \in \mathcal{A}^j$. player j pulls arm k. Then she receives a collision or non-collision.

For the Adding:

if arm $k \in \mathcal{A}^j$ is not occupied

if arm $k \in \mathcal{A}^j$ is occupied

0 $\quad$ $\boxed{1 - \frac{1}{2e}}$ $\quad$ $1 - \varepsilon$ $\quad$ 1

collision probability axis

arm k is occupied
a player is exploiting it
the collision prob. ↑

Take at most $\mathcal{O}(K \ln T)$ steps to separate them w.p. $1 - \frac{1}{T^2}$.

For the Removing:

use the assumption that *m <= K/2*.

if arm $k \in \mathcal{A}^j$ is not released

if arm $k \in \mathcal{A}^j$ is released

0 $\quad$ ε $\quad$ $\boxed{\frac{1}{2e}}$ $\quad$ 1

non-collision probability axis

arm k is released
no player is exploiting it
the collision prob. ↓
the non-collision prob. ↑

Take at most $\mathcal{O}(\frac{m \ln T}{\varepsilon})$ steps to separate them w.p. $1 - \frac{1}{T^2}$.

Proof Skectch

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

the first m_t optimal arms' expectation — active players' rewards (definition)

$$\leq \sum_{t=1}^{T}\left(m_t - \mathbb{E}\left[\sum_{j:T^j_{\mathrm{start}}\leq t\leq T^j_{\mathrm{end}}} \mathbb{1}[\pi^j(t)\leq m_t, \eta^j(t)=0]\right]\right)$$

the number of active players — the number of active players who correctly select arms (select optimal arm and receive no collision)

Proof Skectch

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

$$\le \sum_{t=1}^{T} \left(m_t - \mathbb{E}\left[\sum_{j: T^j_{\text{start}} \le t \le T^j_{\text{end}}} \mathbb{1}[\pi^j(t) \le m_t, \eta^j(t) = 0] \right] \right)$$

$$\le \sum_{j \le M} |\text{adding arms to } \mathcal{A}^j| + |\text{remove arms from } \mathcal{A}^j| + |\text{exploration}| + |\text{bad events}|$$

the number of adding $\times$ the regret of one adding process

$\mathcal{O}(m^2 M \times K \ln T)$

the number of removing $\times$ the regret of one removing process

$\mathcal{O}(m^2 M \times \frac{m \ln T}{\varepsilon})$

successive elimination technique

$\mathcal{O}(\frac{mKM \log T}{\Delta^2} + \varepsilon MT)$

Proof Skectch

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

$$\leq \sum_{t=1}^{T}\left(m_t - \mathbb{E}\left[\sum_{j:T^j_{\text{start}}\leq t\leq T^j_{\text{end}}} \mathbb{1}[\pi^j(t)\leq m_t, \eta^j(t)=0]\right]\right)$$

$$\leq \sum_{j\leq M} |\text{adding arms to } \mathcal{A}^j| + |\text{remove arms from } \mathcal{A}^j| + |\text{exploration}| + |\text{bad events}|$$

the number of adding $\times$ the regret of one adding process

$\mathcal{O}(m^2 M \times K \ln T)$

the number of removing $\times$ the regret of one removing process

$\mathcal{O}(m^2 M \times \frac{m \ln T}{\varepsilon})$

successive elimination technique

$\mathcal{O}(\frac{mKM\log T}{\Delta^2} + \varepsilon MT)$

Why m^2M?

- Releasing arms can only happen due to a permanent departure of one player. There are m permanent departures.
- Each departure can cause at most (m-1) times of releasing.
- Sum over all players.

Proof Skectch

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

$$\leq \sum_{t=1}^{T}\left(m_t - \mathbb{E}\left[\sum_{j:T^j_{\text{start}}\leq t\leq T^j_{\text{end}}} \mathbb{1}[\pi^j(t)\leq m_t, \eta^j(t)=0]\right]\right)$$

$$\leq \sum_{j\leq M} |\text{adding arms to } \mathcal{A}^j| + |\text{remove arms from } \mathcal{A}^j| + |\text{exploration}| + |\text{bad events}|$$

the number of adding $\times$ the regret of one adding process

$\mathcal{O}(m^2M \times K\ln T)$

the number of removing $\times$ the regret of one removing process

$\mathcal{O}(m^2M \times \frac{m\ln T}{\varepsilon})$

successive elimination technique

$\mathcal{O}(\frac{mKM\log T}{\Delta^2} + \varepsilon MT)$

Why m^2M?

- Releasing arms can only happen due to a permanent departure of one player. There are m permanent departures.
- Each departure can cause at most (m-1) times of releasing.
- Sum over all players.

same for the adding process

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

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

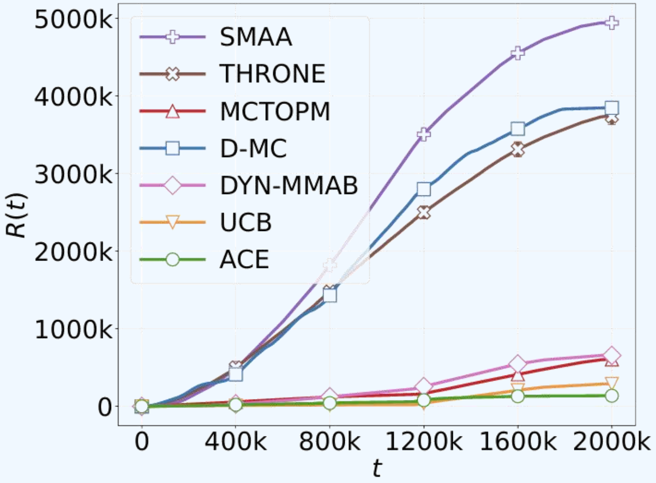
where $\Delta := \min_{k\leq m}(\mu_k - \mu_{k+1})$.

$\mathcal{O}(\log T/\Delta^2)$ **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 $\mathcal{O}(\sqrt{T\log T})$.

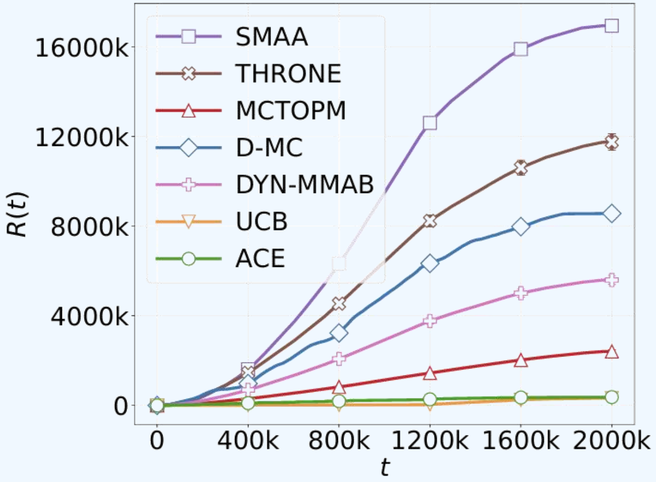
Corollory 1.

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

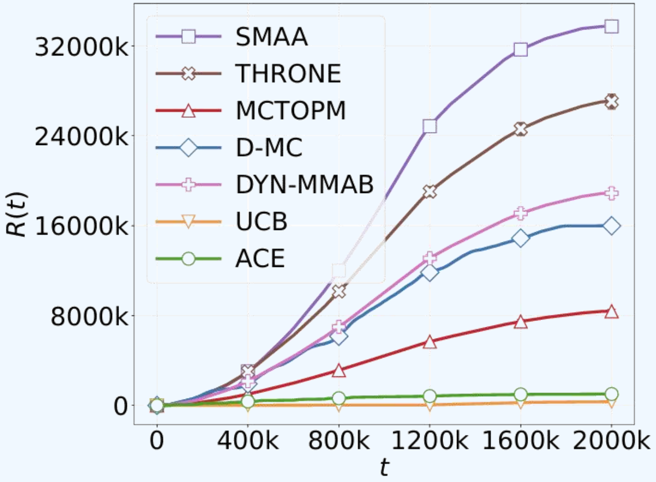
$$\mathbb{E}[R(T)] \leq \frac{288eK^2M\log(T)}{\Delta^2} + 34K^{3/2}M\sqrt{T\ln(T)} + 1926K^3M\ln(T) + (3eK^2M)^2 .$$



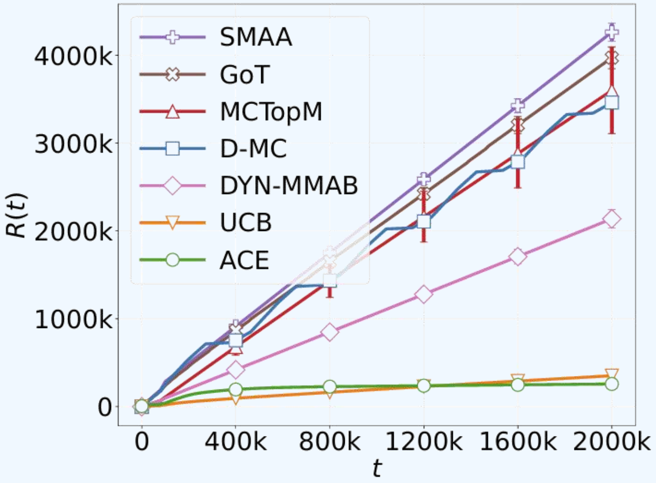
(a) K=20, random.



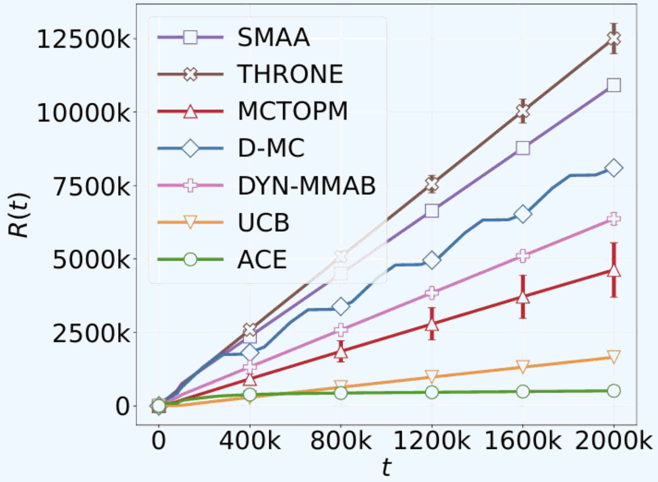
(b) K=50, random.



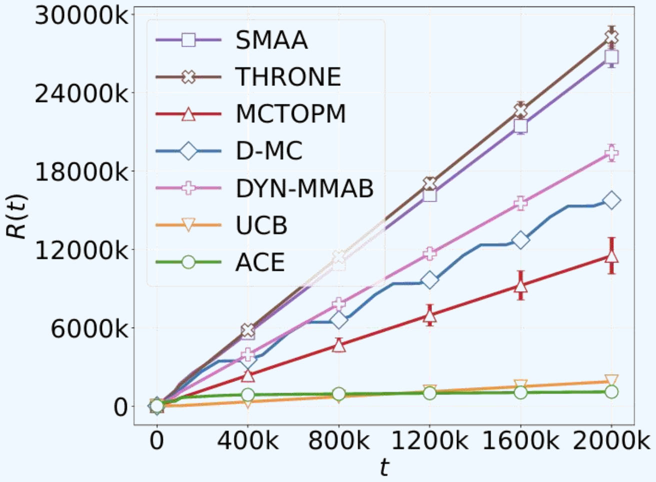
(c) K=100, random.



(d) K=20, synthetic.

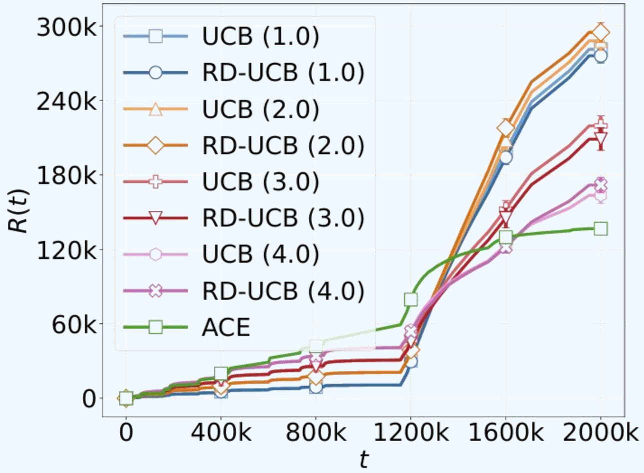


(e) K=50, synthetic.

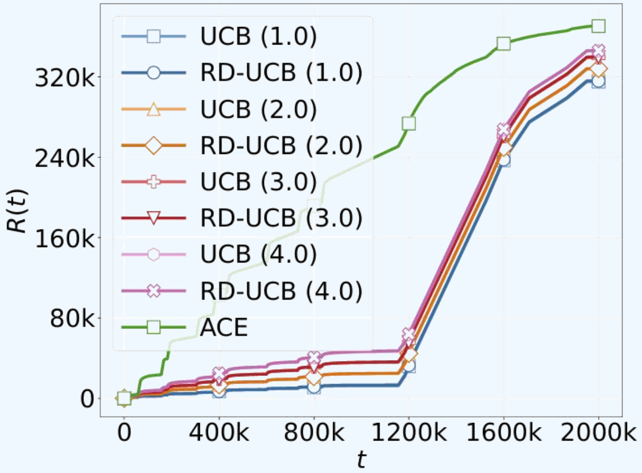


(f) K=100, synthetic.

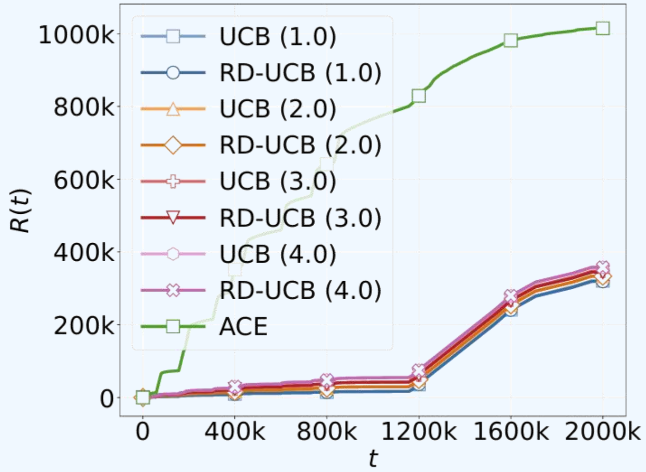
Figure 1: Comparison of cumulative regret for different numbers of arms K under different asynchronization settings.



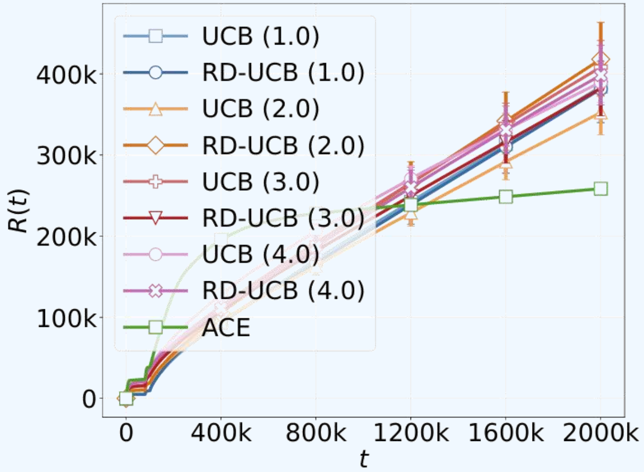
(a) K=20, random, with UCBs.



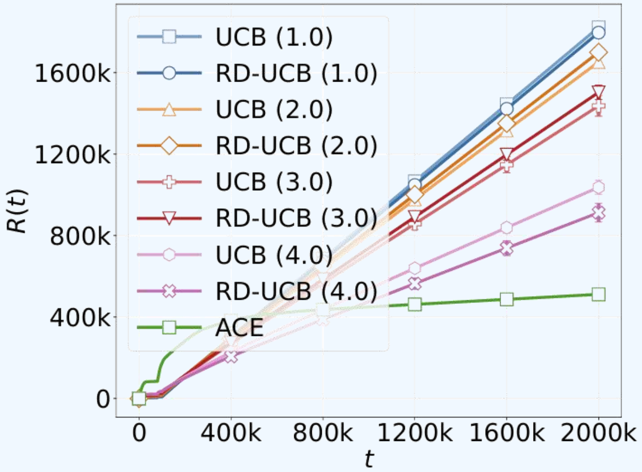
(b) K=50, random, with UCBs.



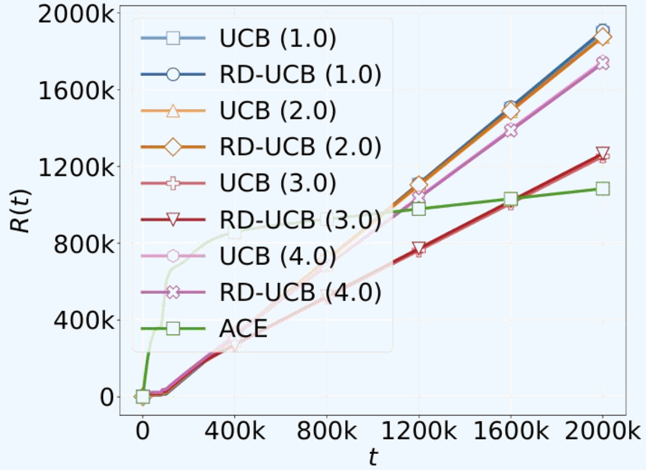
(c) K=100, random, with UCBs.



(d) K=20, synthetic, with UCBs.

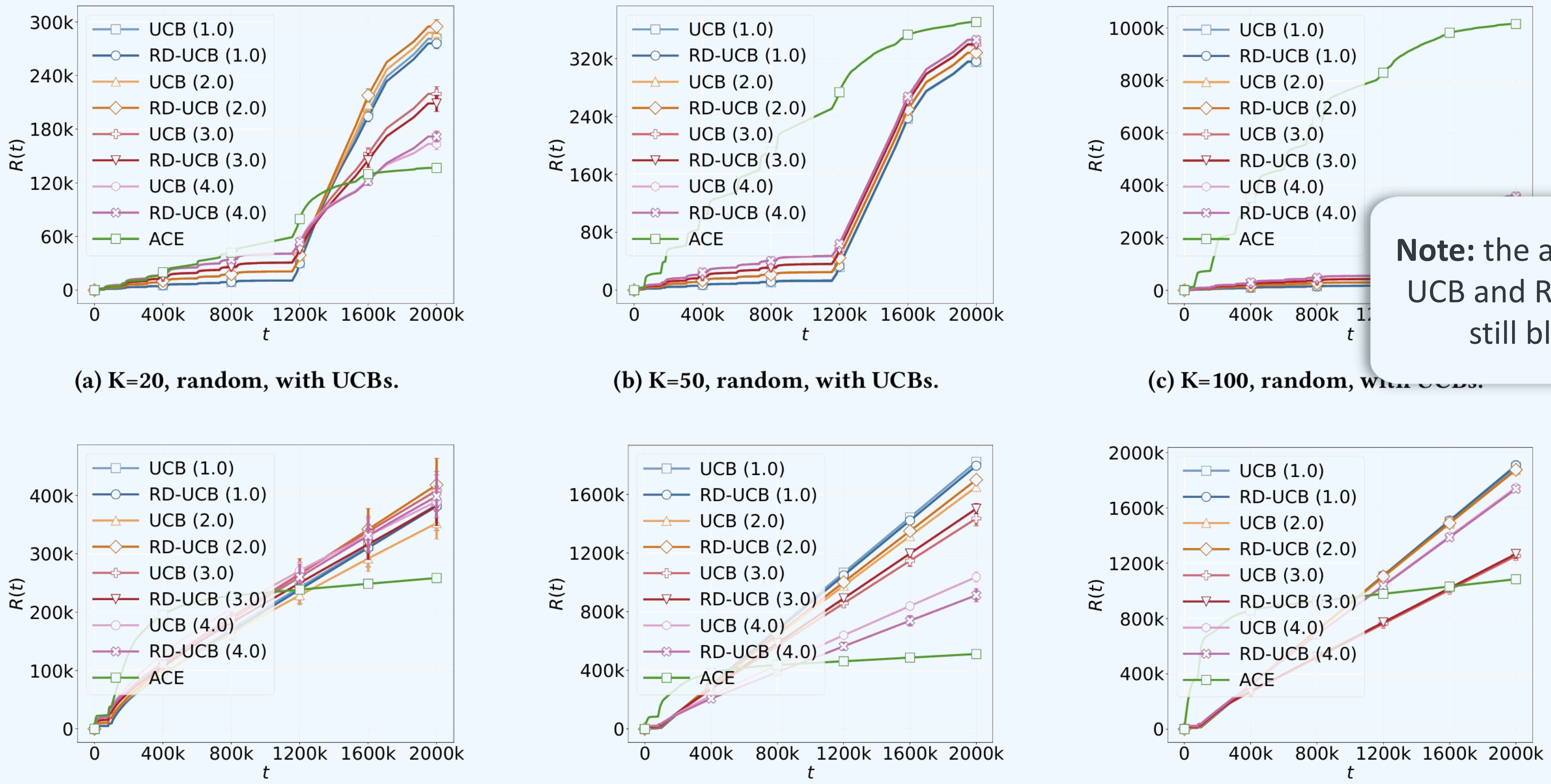


(e) K=50, synthetic, with UCBs.



(f) K=100, synthetic, with UCBs.

Figure 2: Comparison of cumulative regret between UCB with multiple parameters and ACE for different K under different asynchronous settings.



Note: the analysis of UCB and RD-UCB is still blank.

(a) K=20, random, with UCBs.

(b) K=50, random, with UCBs.

(c) K=100, random, with UCBs.

(d) K=20, synthetic, with UCBs.

(e) K=50, synthetic, with UCBs.

(f) K=100, synthetic, with UCBs.

Figure 2: Comparison of cumulative regret between UCB with multiple parameters and ACE for different K under different asynchronous settings.

Summary

SUMMARY

the first paper handling asynchronization in decentralized MP-MAB
with theoretical guarantee and good empirical performance
more general setting than previous works