Multi-Player Multi-Armed Bandits with Finite Shareable Resources Arms: Learning Algorithms & Applications*

Xuchuang Wang[1] , **Hong Xie**[2] , **John C.S. Lui**[1]
[1]Department of Computer Science & Engineering, The Chinese University of Hong Kong
[2]College of Computer Science, Chongqing University, China
[1]{xcwang, cslui}@cse.cuhk.edu.hk, [2]xiehong2018@foxmail.com

Abstract

Multi-player multi-armed bandits (MMAB) study how decentralized players cooperatively play the same multi-armed bandit so as to maximize their total cumulative rewards. Existing MMAB models mostly assume when more than one player pulls the same arm, they either have a collision and obtain zero rewards, or have no collision and gain independent rewards, both of which are usually too restrictive in practical scenarios. In this paper, we propose an MMAB with *shareable resources* as an extension to the collision and non-collision settings. Each shareable arm has finite shareable resources and a "per-load" reward random variable, both of which are unknown to players. The reward from a shareable arm is equal to the "per-load" reward multiplied by the minimum between the number of players pulling the arm and the arm's maximal shareable resources. We consider two types of feedback: sharing demand information (SDI) and sharing demand awareness (SDA), each of which provides different signals of resource sharing. We design the `DPE-SDI` and `SIC-SDA` algorithms to address the shareable arm problem under these two cases of feedback respectively and prove that both algorithms have logarithmic regrets that are tight in the number of rounds. We conduct simulations to validate both algorithms' performance and show their utilities in wireless networking and edge computing.

1 Introduction

Multi-armed bandits (MAB) [Lai and Robbins, 1985] is a canonical sequential decision making model for studying the *exploration-exploitation* trade-off. In a stochastic MAB, a player pulls one arm among $K \in \mathbb{N}_+$ arms per time slot and receives this pulled arm's reward generated by a random variable. To maximize the total reward (i.e., minimize *regret* which is the cumulative reward differences between the optimal arm and the chosen arms), the player needs to balance between choosing arms with high uncertainty in their reward means (exploration) and choosing the empirical good arms so far (exploitation). Anantharam *et al.* [1987] considered the multi-play MAB (MP-MAB) model in which the player can pull $M \in \{2, \dots, K-1\}$ arms from K arms per time slot.

Recently, a decentralized *multi-player* multi-armed bandits (MMAB) model was proposed [Liu and Zhao, 2010; Anandkumar *et al.*, 2011]. Instead of one player choosing M arms as in MP-MAB, there are M players in MMAB, and each player *independently* pulls one arm from K arms per time slot. One particular feature of MMAB is how to define the reward when several players choose the same arm at the same time slot. One typical setting is that all of them get zero reward, i.e., these players experience a *collision*, and this model was motivated by the cognitive radio network application [Jouini *et al.*, 2009]. Another setting is that each player gets an independent reward from the arm without influencing each other, i.e., the *non-collision* setting, which models applications in a distributed system [Landgren *et al.*, 2016].

However, to model many real world applications, we need to consider that arms may have *finite shareable resources* (or capacities), e.g., every channel (or arm) in a cognitive radio network can support a finite traffic demand, instead of the restrictive collision assumption. Similarly, an edge computing node (or arm) in a distributed system can only serve a finite number of tasks, and the over-simplified non-collision assumption cannot be applied. More concretely, consider a cognitive radio network consisting of K channels (arms) and M users (players). Each user chooses one channel to transmit information per time slot. If the chosen channel is available, the user can transmit his information. Some channels with high (low) bandwidth can support larger (smaller) number of users per time slot. To maximize the total transmission throughput, *some players can share these high-bandwidth channels*. But if too many users choose the same channel (i.e., channel competition occurs), then the channel can only transmit information at its maximum capacity. Another example is, in mobile edge computing, each edge computing node (arm) may have multiple yet finite computing units (resources), and thus can also be shared by multiple users (players). Similar scenarios also appear in many distributed systems, online advertisement placements, and many other applications.

To realistically model many of these applications, we propose the *M̲ulti-Player M̲ulti-A̲rmed B̲andits with Finite S̲hareable Resources A̲rms* (`MMAB-SA`) model. It introduces the *shareable* arms setting into MMAB as an extension to the collision and non-collision formulations. This setting not only allows several players to share an arm, but their rewards are dependent and limited by the arm's total shareable resources

*A full technical version of this paper is available on arXiv.

(quasi-collision). Formally, each arm is associated with both a "per-load" reward random variable X_k and a finite resources capacity $m_k \in \mathbb{N}_+$. When a_k players choose arm k, they acquire a total reward which is equal to $\min\{a_k, m_k\}X_k$: if the number of players pulling the arm does not exceed the arm's shareable resource capacity, i.e., $a_k \leqslant m_k$, then only a_k out of these m_k resources are utilized, e.g., one unit of resources will be given to each player; or otherwise, all m_k resources would be utilized. The reward X_k can model whether a channel is available or not, or the instantaneous computing power of an edge node, etc. Note that both the "per-load" reward mean $\mu_k := \mathbb{E}[X_k]$ and the resources m_k are unknown to players.

When sharing an arm, players not only observe their reward attained from the arm, but also some addition feedback about the degree of competition. In some applications, the arm can interact with players. For example, in a mobile edge computing system, an arm models a computing node. When serving players, the computing node can give players feedback about its loading condition. One type of feedback is the number of players who are sharing the arm in the current time slot, i.e., a_k. We call this kind of feedback as *"Sharing Demand Information"* (SDI). For some other applications, an arm can only provide some limited forms of feedback. Take a cognitive radio network as an example. Here, an arm corresponding to a wireless channel cannot send its loading information to users. Users can only sense whether there exists any sharing of the channel by other players or not, i.e., $\mathbb{1}\{a_k > 1\}$. We name this type of feedback as *"Sharing Demand Awareness"* (SDA). Note that the SDI feedback is more informative than SDA, since knowing a_k determines $\mathbb{1}\{a_k > 1\}$. In SDA, on the other hand, each player only needs to procure 1-bit feedback information, which is easier to realize in practice.

Recently, several algorithms for MMAB were proposed, e.g., in [Rosenski *et al.*, 2016; Besson and Kaufmann, 2018; Boursier and Perchet, 2019; Wang *et al.*, 2020]. However, none of them can address MMAB-SA. The reason is that the unknown resource capacity of each arm and the sharing mechanism complicate the decentralized learning problem: **(1) learn resources capacities while sharing arms:** instead of avoiding collisions for maximizing total reward as in MMAB, players in our setting *often have to* play the same arm not only to gain higher rewards, but also to infer the appropriate number of players to pull the arm since it depends on the arm's resources m_k; **(2) information extraction while sharing arms:** instead of taking infrequent collisions only as signals in MMAB (e.g., [Wang *et al.*, 2020]), players in our setting needs to *extract information* from the collision events of arm sharing. None of the known MMAB algorithms can address these two challenges. For the first challenge, we propose decentralized algorithms that can explore (learn) arm's maximum resources capacities and exploit (share) good arms while addressing the classic exploration-exploitation trade-off. For the second challenge, we design two types of communications (and their corresponding algorithms) to extract information from the degree of resource sharing under the SDI and SDA feedback respectively. Our contributions are as follows:

- We propose the MMAB-SA model which significantly expands the application scope of MMAB. It allows several players to share an arm with finite resources. We propose the SDI and SDA feedback mechanisms, both of which can be mapped nicely to many real world applications.
- We design the *Decentralized Parsimonious Exploration for SDI* (DPE-SDI) algorithm and the *Synchronization Involved Communication for SDA* (SIC-SDA) algorithm to address MMAB-SA under the SDI and SDA feedback respectively. In MMAB-SA, apart from estimating arm's reward means μ_k, players also need to estimate arms' resources capacity m_k. This new estimation task and the shareable arm setting requires more information exchange between players than MMAB. To address the issue, we devise two different communication protocols for DPE-SDI and SIC-SDA algorithms. We also rigorously prove that *both algorithms have logarithmic regrets* that are tight in term of time horizon.
- We conduct simulations to validate and compare the performance of DPE-SDI and SIC-SDA, and apply them in edge computing and wireless networking scenarios respectively.

2 Related Work

Our work falls into the research line of multi-player multi-armed bandits (MMAB) first studied by [Liu and Zhao, 2010; Anandkumar *et al.*, 2011]. Recently, Rosenski *et al.* [2016] proposed the musical chair algorithm which can distribute players to different arms. Utilizing collisions, Boursier and Perchet [2019] introduced a communication protocol between players. DPE1 [Wang *et al.*, 2020] was the first algorithm achieving the optimal regret bound. However, applying above algorithms to MMAB-SA would result in poor performance (i.e., linear regrets) because of the two challenges stated in Section 1's last paragraph (before contributions). Besides the collision setting, the non-collision MMAB model was also studied, e.g., in [Landgren *et al.*, 2016; Martínez-Rubio *et al.*, 2019]. However, their algorithms relied on the independent reward assumption and thus were not applicable in our dependent rewards of shareable arm setting.

There were several variants of MMAB. One variant considered the heterogenous reward setting in which players face different reward environments in pulling arms, e.g., [Kalathil *et al.*, 2014; Bistritz and Leshem, 2018; Mehrabian *et al.*, 2020; Shi *et al.*, 2021]. Another variant of MMAB models took game theoretical results, e.g., Nash equilibrium, into consideration. For example, Boursier and Perchet [2020] considered selfish players who might deceive other players and race to occupy the best arms for their own good. Liu *et al.* [2020] and Sankararaman *et al.* [2021] studied the two-sided markets in MMAB, i.e., both arms and players have preference for choosing each other. Lastly, Magesh and Veeravalli [2021] considered a variant that when the number of players selecting an arm exceeds a threshold, all player receive zero reward, while our model has no such threshold. This is an important difference because this threshold can be utilized to communicate and coordinate. Different from above variants, our model introduces finite shareable resources arms into MMAB.

3 Model Formulation

3.1 The Decision Model

Consider $K \in \mathbb{N}_+$ arms and $M \in \mathbb{N}_+$ ($M < K$) players (the condition relaxation is discussed in Appendix B.1). The arm $k \in [K] := \{1, 2, \ldots, K\}$ is associated with (m_k, X_k), where $m_k \in \mathbb{N}_+$ ($m_k \leqslant M$) denotes its maximal shareable resources capacity, while X_k is a random variable with support in $[0, 1]$ to model the arm's "per-load" stochastic reward. Denote the reward mean as $\mu_k := \mathbb{E}[X_k]$. Without loss of generality, assume these reward means have a descending order $\mu_1 > \mu_2 > \ldots > \mu_K$. Reward means μ_k (including their order) and resource capacities m_k are all unknown to players.

Consider a finite time horizon with length $T \in \mathbb{N}_+$. In each time slot $t \in \{1, 2, \ldots, T\}$, each player $i \in \{1, 2, \ldots, M\}$ selects one arm to pull. Let $\boldsymbol{a}_t := (a_{1,t}, a_{2,t}, \ldots, a_{K,t})$ denote the assignment profile of players, where $a_{k,t}$ denotes the number of players pulling arm k at time slot t. The assignment profile satisfies $\sum_{k=1}^{K} a_{k,t} = M$, capturing that no players are idle in any given time slots. When $a_{k,t}$ players share the arm k at time slots t, the total reward they obtain from arm k is

$$R_{k,t} := \min\{a_{k,t}, m_k\} X_{k,t},$$

where the scaler factor $\min\{a_{k,t}, m_k\}$ describes the amount of resources of arm k utilized by $a_{k,t}$ players. In other words, when $a_{k,t} \leqslant m_k$, they utilize $a_{k,t}$ resources, e.g., each player can enjoy one unit of resources; while if $a_{k,t} > m_k$, all m_k resources of the arm k are shared by $a_{k,t}$ players. We focus on maximizing all players' total reward and how the reward $R_{k,t}$ distributed among $a_{k,t}$ players is not of interest in this work.

The expected reward of an assignment profile $\boldsymbol{a}_t$ is the summation of each arms' reward, which can be expressed as

$$f(\boldsymbol{a}_t) = \mathbb{E}\Big[\sum\nolimits_{k=1}^{K} R_{k,t}\Big] = \sum\nolimits_{k=1}^{K} \min\{a_{k,t}, m_k\}\mu_k.$$

Hence, *the optimal assignment profile* $\boldsymbol{a}^*$ for maximizing per time slot's reward would be exactly m_1 number of players choosing arm 1, m_2 players choosing arm 2, and so on, until all players are assigned. This profile can be expressed as

$$\boldsymbol{a}^* := \Big(m_1, m_2, \ldots, m_{L-1}, M - \sum\nolimits_{k=1}^{L-1} m_k, 0, \ldots, 0\Big),$$

where $L := \min\{l : \sum_{k=1}^{l} m_k \geqslant M\}$ denotes *the least favored arm index* among the optimal assignment profile. We call selected arms in the optimal profile $\boldsymbol{a}^*$ as *optimal arms*, and the remaining as *sub-optimal arms*. Note that $\boldsymbol{a}^*$ is unknown to players because μ_k and m_k are unknown.

3.2 Online Learning Problem

When pulling arm k, a player not only observes arm k's reward, but also some additional feedback on the degree of competition on this arm. We consider two types of feedback: **Sharing demand information (SDI):** at time slot t, players who pull arm k can also observe the number of players $a_{k,t}$ that selects the arm k; **Sharing demand awareness (SDA):** at time slot t, players who pull arm k only know whether the arm is shared by others or not, i.e., $\mathbb{1}\{a_{k,t} > 1\}$. We note that the SDI feedback is more informative than SDA because knowing $a_{k,t}$ directly implies $\mathbb{1}\{a_{k,t} > 1\}$. But SDA is easier to implement since the player only needs to procure 1-bit information per time. Note that players cannot freely communicate with each other and they can only infer the environment from feedback.

We aim to design decentralized algorithms for players to select arms and our objective is to maximize the total reward of all players. In each time slot, the algorithm prescribes an arm for each player given the player's feedback up to time slot t, which together forms an assignment profile denoted by $\boldsymbol{a}_t$. We define regret to quantify the performance of an algorithm when comparing with the optimal profile $\boldsymbol{a}^*$,

$$\mathbb{E}[\mathrm{Reg(T)}] = \sum\nolimits_{t=1}^{T} \left(f(\boldsymbol{a}^*) - f(\boldsymbol{a}_t)\right).$$

A smaller regret implies that an algorithm achieves a larger reward, or a reward which is closer to the optimal profile's.

4 Decentralized Parsimonious Exploration for the SDI Feedback Algorithm

We begin with the high-level idea of the DPE-SDI algorithm, and then present the detailed design of its several phases. Finally, we show DPE-SDI has a $O(\log(T))$ sub-linear regret.

4.1 Overview of the Design

DPE-SDI employs a leader-follower structure: one player is the leader, and the rest $M - 1$ players are followers. The leader takes the responsibility of collecting observations (both rewards and SDI feedback) itself and updates its information (e.g., empirical optimal arms) to followers. Followers do not need to communicate anything to the leader, i.e., they only receive information. DPE-SDI consists of three phases: *initialization phase*, *exploration-exploitation phase* and *communication phase*. The *initialization phase* (line 2) detects the number of players M and assigns ranks to players. The player with rank 1 becomes the leader. After the *initialization phase*, DPE-SDI runs the *exploration-exploitation phase* repeatedly and, when necessary, runs the *communication phase* in which the leader updates followers' information. The *exploration-exploitation phase* (line 4) conducts *exploitation* such that the followers exploit empirical optimal arms, and three types of explorations: (1) *parsimonious exploration*, where the leader parsimoniously explores empirical sub-optimal arms; (2) *individual exploration*, where the leader individually explores empirical optimal arms; and (3) *united exploration*, where all players pull the same arms for estimating their maximum resources capacities m_k. When it ends, the leader updates its estimated parameters (line 7 in Algorithm 1). In *communication phase*, the leader sends his updated parameters to followers (line 9) and followers receive them (line 11). After communication, the algorithm goes back to the *exploration-exploitation phase* (line 4). Algorithm 1 outlines the DPE-SDI algorithm.

Notations. We use i to denote each player's rank. Denote the empirical optimal arm set as $\mathcal{S}_t$, the empirical least favored arm index as $\mathcal{L}_t$, and the parsimonious exploration arm set as $\mathcal{E}_t$ (define later). We also denote the lower and upper confidence bounds of arms' shareable resources as $\boldsymbol{m}_t^l := (m_{1,t}^l, \ldots, m_{K,t}^l)$ and $\boldsymbol{m}_t^u := (m_{1,t}^u, \ldots, m_{K,t}^u)$. We summarize the information that leader sends to followers as $\boldsymbol{\Upsilon}_t := (\mathcal{S}_t, \mathcal{L}_t, \boldsymbol{m}_t^l, \boldsymbol{m}_t^u)$. Arms' useful statistics (e.g., empirical means) are aggregated as $\boldsymbol{\Lambda}_t$ (define later).

Algorithm 1 `DPE-SDI`

```
Initialization: $\mathcal{E}_t \leftarrow [K], \boldsymbol{\Upsilon}_t = (\mathcal{S}_t, \mathcal{L}_t, \boldsymbol{m}_t^l, \boldsymbol{m}_t^u) \leftarrow ([K], 1, (1, \dots, 1)^T, (K, \dots, K)^T)$, and $\boldsymbol{\Lambda}_t \leftarrow \emptyset$.
▷ Initialization phase
$(i, M) \leftarrow$ DPE-SDI.Init()
while $t \leqslant T$ do
▷ Exploration-exploitation phase
    $\boldsymbol{\Lambda}_t \leftarrow$ DPE-SDI.Explo($i, M, \mathcal{E}_t, \boldsymbol{\Upsilon}_t, \boldsymbol{\Lambda}_t$)
    $\boldsymbol{\Upsilon}_{\text{pre}} \leftarrow \boldsymbol{\Upsilon}_t$                ▷ Record previous info.
▷ Communication phase
▷ Leader performs update and sends info. to followers.
    if $i = 1$ then
        $(\mathcal{E}_t, \boldsymbol{\Upsilon}_t) \leftarrow$ DPE-SDI.Update($\boldsymbol{\Lambda}_t, \boldsymbol{\Upsilon}_t$)
        if $\boldsymbol{\Upsilon}_t \neq \boldsymbol{\Upsilon}_{\text{pre}}$ then
            DPE-SDI.CommSend($\boldsymbol{\Upsilon}_t, \boldsymbol{\Upsilon}_{\text{pre}}$)
▷ Followers receive leader's update info.
    else if $i \neq 1$ and leader informs then
        $\boldsymbol{\Upsilon}_t \leftarrow$ DPE-SDI.CommRece($\boldsymbol{\Upsilon}_{\text{pre}}$)
```

4.2 Initialization Phase

Our initialization phase, consisting of two steps: *rally* and *orthogonalization*, is simpler and more effective than the previous ones in MMAB, e.g., [Wang *et al.*, 2020], due to the advantage of the SDI feedback. The feedback of the rally step — all players pull arm 1 at time slot 1 — is equal to the number of players M. Knowing M, we then orthogonalize M players to M arms (i.e., each player individual chooses an arm) and the player's orthogonalized arm index is set as its rank. We provide the phase's details in Appendix B.1.

4.3 Exploration-Exploitation Phase

The exploration-exploitation phase consists of exploitation (followers), individual exploration (leader), parsimonious exploration (leader), and united exploration (all players). Exploitation and parsimonious/individual exploration are done in parallel, in which followers exploit the empirical optimal arms while the leader at times deviates to explore empirical sub-optimal ones. Denote $\hat{\boldsymbol{a}}_t^* := (\hat{a}_{1,t}^*, \dots, \hat{a}_{K,t}^*)$ as the empirical optimal assignment profile at time t. Given $\hat{\boldsymbol{a}}_t^*$, the empirical optimal arm set $\mathcal{S}_t$ is $\{k : \hat{a}_{k,t}^* > 0\}$. Note that the leader does not need to transmit the profile $\hat{\boldsymbol{a}}_t^*$ to followers, because followers can **recover $\hat{\boldsymbol{a}}_t^*$ from $\mathcal{S}_t, \mathcal{L}_t, \boldsymbol{m}_t^l$** (Algorithm 2's line 2): $\hat{a}_{k,t}^*$ is equal to $m_{k,t}^l$ when k is in $\mathcal{S}_t$ but not equal to $\mathcal{L}_t$, and the $\hat{a}_{\mathcal{L}_t,t}^*$ is equal to $M - \sum_{k \in \mathcal{S}_t, k \neq \mathcal{L}_t} m_{k,t}^l$. Next, we illustrate the phase's four components respectively.

Exploitation. Players exploit arms according to the empirical optimal profile $\hat{\boldsymbol{a}}_t^*$, and they play the empirical optimal arms in turn. To illustrate, we take $(i + t) \pmod M$ as their circulating ranks for choosing arms in time slot t. Note that several players may share the same arm, e.g., when $\hat{a}_{k,t}^* > 1$ for some k. To realize this, we apply **a rotation rule** (Algorithm 2's line 4): at time slot t, player with rank i plays arm j such that $\sum_{n=1}^{j} \hat{a}_{n,t}^* \geqslant (i + t) \pmod M > \sum_{n=1}^{j-1} \hat{a}_{n,t}^*$. That means players with a circulating rank greater than $\sum_{n=1}^{j-1} \hat{a}_{n,t}^*$ and no greater than $\sum_{n=1}^{j} \hat{a}_{n,t}^*$ choose arm j.

Individual exploration. Individual exploration (IE) happens when the explored arm's resource capacity is not exceeded by the number of players pulling the arm, i.e, $a_{k,t} \leqslant m_k$. The simplest case of IE is that an arm is played by one player, but it also includes cases that several players share an arm. The leader's actions during the exploitation and parsimonious exploration can all be regarded as IEs because $a_{k,t} \leqslant m_{k,t}^l$. Divided by $a_{k,t}$, IE's rewards can be used to estimate the reward mean $\hat{\mu}_{k,t} := S_{k,t}^{\text{IE}} / \tau_{k,t}$, where $S_{k,t}^{\text{IE}}$ is the total IE's reward feedback (divided by $a_{k,t}$) up to time slot t, and $\tau_{k,t}$ is the total times of IE for arm k up to time slot t.

Parsimonious exploration. Denote $u_{k,t}$ as the KL-UCB index of arm k at time slot t [Cappé *et al.*, 2013]: $u_{k,t} = \sup\{q \geqslant 0 : \tau_{k,t} \operatorname{kl}(\hat{\mu}_{k,t}, q) \leqslant f(t)\}$, where $f(t) = \log(t) + 4 \log\log(t)$. The parsimonious exploration arm set $\mathcal{E}_t$ consists of empirical sub-optimal arms whose KL-UCB indexes are larger than the least favored arm $\mathcal{L}_t$'s empirical mean, i.e., $\mathcal{E}_t := \{k : u_{k,t} \geqslant \hat{\mu}_{\mathcal{L}_t,t}, \hat{a}_{k,t}^* = 0\}$. When the leader is supposed to play the least favored arm $\mathcal{L}_t$ for exploitation (also IE), with a probability of 0.5, he will uniformly choose an arm from $\mathcal{E}_t$ to explore; or otherwise, the leader exploits the arm $\mathcal{L}_t$. This parsimonious exploration idea was first made in [Combes *et al.*, 2015] for learning-to-rank algorithms and was further utilized by [Wang *et al.*, 2020] in MMAB.

United exploration. United exploration (UE) refers to all M players rallying on an arm so as to acquire a sample of $m_k X_k$, i.e., the reward when the arm's resources are fully utilized. Collecting this kind of observation is crucial in estimating an arm's shareable resources capacity m_k (see Appendix B.2). Let $\mathcal{S}_t' := \{k \in \mathcal{S}_t : m_{k,t}^l \neq m_{k,t}^u\}$ denote the set of empirical optimal arms whose resource capacities have not been learnt. In one round of exploration-exploitation phase, all M players unitedly explore each arm in $\mathcal{S}_t'$ once. Denote $S_{k,t}^{\text{UE}}$ as the total reward of arm k in UE up to time t, $\iota_{k,t}$ as the total times of UE for arm k up to time t, and thus the empirical mean estimate of $\mathbb{E}[m_k X_k]$ is $\hat{\nu}_{k,t} := S_{k,t}^{\text{UE}} / \iota_{k,t}$. The output statistics $\boldsymbol{\Lambda}_t$ consists of each arm's IE's total reward $\boldsymbol{S}_t^{\text{IE}} := (S_{1,t}^{\text{IE}}, \dots, S_{K,t}^{\text{IE}})$, IE times $\boldsymbol{\tau}_t := (\tau_{1,t}, \dots, \tau_{K,t})$, and their UE counterparts: $\boldsymbol{S}_t^{\text{UE}} := (S_{1,t}^{\text{UE}}, \dots, S_{K,t}^{\text{UE}}), \boldsymbol{\iota}_t := (\iota_{1,t}, \dots, \iota_{K,t})$.

Algorithm 2 outlines the exploration-exploitation phase. After the phase, the leader updates $\boldsymbol{\Upsilon}_t$ via `DPE-SDI.Update`. Its detailed procedure is deferred to Appendix B.2.

4.4 Communication Phase

After updating $\boldsymbol{\Upsilon}_t = (\mathcal{S}_t, \mathcal{L}_t, \boldsymbol{m}_t^l, \boldsymbol{m}_t^u)$, if there are parameter changes, leader will initialize a round of communication to update followers' $\boldsymbol{\Upsilon}_t$. For simplicity, we illustrate the communication as six steps.

- Initial step: leader signals followers to communicate.
- 2nd step: notify followers arms to be removed from $\mathcal{S}_t$.
- 3rd step: notify followers arms to be added in $\mathcal{S}_t$.
- 4th step: notify followers the least favored arm $\mathcal{L}_t$.
- 5th step: update followers' resources' lower bounds $\boldsymbol{m}_t^l$.
- 6th step: update followers' resources' upper bounds $\boldsymbol{m}_t^u$.

The detailed updates of these steps (including conditions and pseudo-codes) and some potential improvements are presented in Appendix B.3.

Algorithm 2 `DPE-SDI.Explo`$(i, M, \mathcal{E}_t, \boldsymbol{\Upsilon}_t, \boldsymbol{\Lambda}_t)$

```
Output: $\boldsymbol{\Lambda}_t := (\boldsymbol{S}_t^{\text{IE}}, \boldsymbol{S}_t^{\text{UE}}, \boldsymbol{\tau}_t, \boldsymbol{\iota}_t)$
$\hat{\boldsymbol{a}}_t^* \leftarrow$ Recover$(\mathcal{S}_t, \mathcal{L}_t, \boldsymbol{m}_t^l)$
for $M$ times do
    $j_t \leftarrow$ Rotate$(i, M, \hat{\boldsymbol{a}}_t^*, t)$
    ▷ Parsimonious exploration by leader
    if $i = 1$ then
        if $\mathcal{E}_t \neq \emptyset$ and $j_t = \mathcal{L}_t$ then
            w.p. $1/2$, Play arm $l \sim \mathcal{E}_t$ uniformly
            w.p. $1/2$, Play arm $j_t$
    ▷ Individual exploration by leader
        else Play arm $j_t$
        $S_{k,t}^{\text{IE}} \leftarrow S_{k,t}^{\text{IE}} + r_{k,t}/\hat{a}_{k,t}^*, \tau_{k,t} \leftarrow \tau_{k,t} + 1$
    ▷ Exploitation by followers
    else if $i \neq 1$ then Play arm $j_t$
    $t \leftarrow t + 1$.
▷ United exploration by all players
$\mathcal{S}_t' \leftarrow \{k \in \mathcal{S}_t : m_{k,t}^l \neq m_{k,t}^u\}$.
for $k \in \mathcal{S}_t'$ do
    All players pull arm $k$.
    if $i = 1$ then $S_{k,t}^{\text{UE}} \leftarrow S_{k,t}^{\text{UE}} + r_{k,t}; \iota_{k,t} \leftarrow \iota_{k,t} + 1$
    $t \leftarrow t + 1$
```

4.5 Regret Bound of DPE-SDI

In the following theorem, we state a regret upper bound for `DPE-SDI` algorithm. Its proof is presented in Appendix D.

Theorem 1. *For any given set of parameters $K, M, \boldsymbol{\mu}, \boldsymbol{m}$ and $0 < \delta < \min_{1 \leqslant k \leqslant K-1}(\mu_k - \mu_{k+1})/2$, the regret of* `DPE-SDI` *is upper bounded as follows:*

$$\begin{aligned}\mathbb{E}[\text{Reg(T)}] \leqslant & \sum\nolimits_{k=L+1}^{K} \frac{(\mu_L - \mu_k)(\log T + 4\log(\log T))}{\text{kl}(\mu_k + \delta, \mu_L - \delta)} \\ & + \sum\nolimits_{k=1}^{M} 49 w_k m_k^2 \mu_k^{-2} \log T \qquad (1) \\ & + 588 K M m_M^2 \mu_M^{-2} \log T + 156 M^3 K^3 (4 + \delta^2)\end{aligned}$$

where $w_k := f(\boldsymbol{a}^*) + \mu_1 - (m_k + 1)\mu_k$ is the highest cost of one round of IE and UE.

Eq.(1)'s first two terms correspond to parsimonious exploration and united exploration of the exploration-exploitation phase respectively. The last two terms cover the cost of initialization phase, communication phase, exploiting sub-optimal arms, and falsely estimating resources. Note that `DPE-SDI`'s $O(\log T)$ regret is tight in term of T, as it matches MMAB's regret lower bound $\Omega(\log T)$ [Anantharam *et al.*, 1987].

5 Synchronization Involved Communication for the SDA Feedback Algorithm

Recall that in `DPE-SDI.CommSend`'s initial step, the leader signals followers to start communication via abnormally deviating the empirical optimal assignment $\hat{\boldsymbol{a}}_t^*$. That is, leader sticks to an arm such that the number of players (leader pluses followers) pulling the arm $a_{k,t}$ is greater than the expected. While under the SDI feedback followers can sense the increase of $a_{k,t}$, in the SDA's $\mathbb{1}\{a_{k,t} > 1\} = 1$ feedback followers may fail. Not being able to sense the difference makes the communication protocol devised for `DPE-SDI` not applicable to the SDA feedback. To address this issue, we propose the `SIC-SDA` algorithm with a phase-based communication protocol. Because it is phase-based, communications start and end at a pre-defined scheme and do not need initial steps.

The `SIC-SDA` algorithm consists of four phases: *initialization phase, exploration phase, communication phase,* and *exploitation phase.* The objective of *initialization phase* is to probe the total number of players M and assign ranks to players. After the *initialization phase*, `SIC-SDA` runs the *exploration phase* and *communication phase* iteratively (i.e., in a loop). In the loop, whenever an optimal arm is identified with high confidence, `SIC-SDA` will assign players to exploit the arm's resources. In other words, these players leave the loop and execute *exploitation phase* so to zoom in on the optimal arm. As the algorithm runs, players in the two-phase loop will gradually enter the *exploitation phase*. Due the limited space, we defer the `SIC-SDA` algorithm's detail and its $O(\log T)$ regret proof to Appendix C and E respectively.

6 Simulations & Applications

6.1 Synthetic Data Simulations

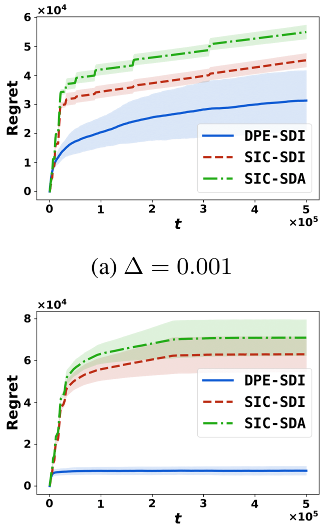
We consider an environment with 9 arms and 6 players, which is the common simulation setting in MMAB literatures [Boursier and Perchet, 2019; Wang *et al.*, 2020]. The "per-load" reward random variables follow the Bernoulli distribution. Their reward means are a random permutation of a decreasing array. The array starts from 0.9 and each successor is smaller than its predecessor by Δ, e.g., when $\Delta = 0.025$, it is $[0.90, 0.875, 0.85, 0.825, 0.80, 0.775, 0.75.0.725, 0.70]$. These 9 arms' resource capacities are $[3, 2, 4, 2, 1, 5, 2, 1, 3]$. For each experiment, we calculate average and standard variance (as shaded region) over 50 simulations. Except `DPE-SDI` and `SIC-SDA`, we also apply the `SIC-SDA` algorithm to the SDI feedback setting and call this as `SIC-SDI`.

Figure 1 compares these three algorithms' performance for different Δs. First, the `DPE-SDI` algorithm has a much smaller regret than the other two. The intuition is that `DPE-SDI` is based on KL-UCB which is theoretically better than elimination utilized by `SIC-SDA` and `SIC-SDI` [Cappé *et al.*, 2013]. Second, we compare `DPE-SDI` and `SIC-SDI` under the same SDI feedback. When Δ becomes smaller, the gap between `DPE-SDI`'s regret and `SIC-SDI`'s also becomes smaller. This observation implies that when optimal arms are difficult to identify (e.g., $\Delta = 0.001$), the efficacy of both algorithms is similar. Lastly, we note that the `SIC-SDI` algorithm outperforms `SIC-SDA`, which is consistent with the fact that the SDI feedback is more informative than SDA.

6.2 Real World Applications

We consider two scenarios: (1) an edge computing system for SDA and (2) a 5G/4G wireless networking system for SDI.

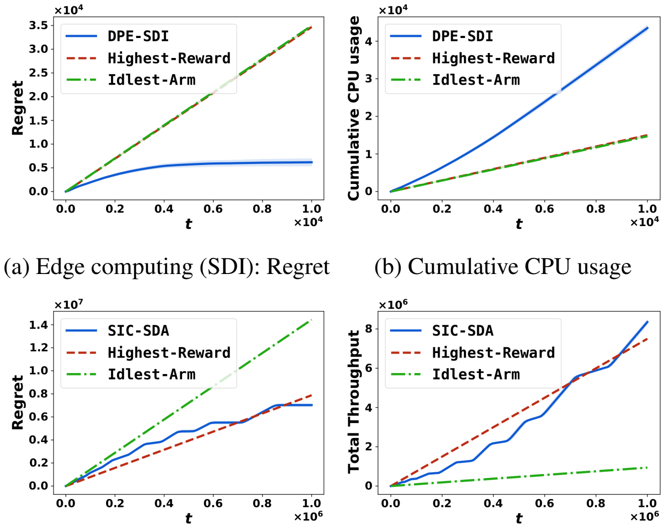
Edge computing. The edge computing system Precog [Drolia *et al.*, 2017] was designed for image processing. Its edge nodes contain mobile devices and personal desktop computers, which fits the SDA setting. According to their specifications, we consider 7 edge computing nodes (arms):



(a) $\Delta = 0.001$ (b) $\Delta = 0.012$

(c) $\Delta = 0.025$ (d) $\Delta = 0.037$

Figure 1: Synthetic data simulations



(a) Edge computing (SDI): Regret (b) Cumulative CPU usage

(c) 5G/4G network (SDA): Regret (d) Total throughput

Figure 2: Real world data simulations

CPU Speed (GHz)	1.5	2.1	1.2	2.5	2.0	1.3	2.6
# of CPU Cores	3	2	4	2	1	2	3

To scale the CPU speed as "pre-load" reward mean in $[0, 1]$, we divide them by 3. The number of CPU cores corresponds to arms' shareable resource capacities m_k. We assume there are 6 parallel tasks (players) in the system.

5G & 4G network. 5G started to serve consumers from 2019 and will coexist with 4G for a long time. When a smartphone uses wireless service, it needs to choose between 5G and 4G networks. The smartphone only has the SDI feedback as it doesn't know the exact number of users using a particular base station. Narayanan *et al.* [2020] measured the 5G performance on smartphone and compared it with 4G. We pick the 8 parallel TCP connections as our typical setting, in which the 5G's throughput (THR) is around 8 times higher than 4G's, and 4G's round-trip time (RTT) latency is around 4 times slower than 5G's. From [Narayanan *et al.*, 2020]'s experimental results, we consider an environment consisting of two 5G base stations (underlined) and eighteen 4G base stations (20 arms) and 18 smartphones (18 players):

RTT (100ms)	1.2	1.1	4.2	4.0	4.5	3.5	5.0	4.2	5.5	3.9
THR (100Mbps)	9.2	8.1	1.2	1.2	1.4	1.1	1.3	1.2	1.1	1.4
RTT (100ms)	4.8	5.5	3.7	4.7	3.2	5.1	4.4	5.3	4.9	4.1
THR (100Mbps)	1.0	1.1	1.2	1.0	1.3	1.2	1.0	1.1	1.3	1.2

We use their RTT latencies' reciprocals as arms' "per-load" reward means and their THR's integer rounding as arms' maximal resource capacities.

We apply the `DPE-SDI` and `SIC-SDA` algorithms to solve these two scenarios respectively. We also implement two heuristic policies according to each player's own observations: the `Highest-Reward` policy in which players choose the arm with highest empirical reward mean, and the `Idlest-Arm` policy in which players select the arm that is most infrequently shared with others. Figure 2a and 2b show that in the edge computing scenario, `DPE-SDI` outperforms other two heuristic policies. In Figure 2c and 2d for the 5G/4G wireless networking scenario, when time slot is small, the `Highest-Reward` policy is better than `SIC-SDA`. Because the policy can detect two powerful 5G base stations immediately and utilize them from the very beginning. But in the long run, the `SIC-SDA` algorithm will find the optimal profile and finally can outperform both heuristic policies.

7 Conclusion

We propose a novel multi-player multi-armed bandits model in which several players can share an arm, and their total reward is the arm's "per-load" reward multiplied by the minimum between the number of players selecting the arm and the arm's maximum shareable resources. Under the SDI and SDA feedback respectively, we design the `DPE-SDI` and `SIC-SDA` algorithms coordinating decentralized players to explore arms' reward means and capacities, exploit the empirical optimal allocation profile, and communicate with each other. Especially, `DPE-SDI` has a simpler initialization phase than the previous algorithms' in MMAB due to the SDI feedback. We prove that both algorithms achieve the logarithmic regrets that are tight in term of time horizon. We also compare these two algorithms' performance using synthetic and real world data and show their utilities in wireless networking and edge computing.

Acknowledgements

The work of John C.S. Lui was supported in part by SRFS2122-4S02. The work of Hong Xie was supported by Chongqing Talents: Exceptional Young Talents Project (cstc2021ycjh-bgzxm0195). Hong Xie is the corresponding author.

References

[Anandkumar *et al.*, 2011] Animashree Anandkumar, Nithin Michael, Ao Kevin Tang, and Ananthram Swami. Distributed algorithms for learning and cognitive medium access with logarithmic regret. *IEEE Journal on Selected Areas in Communications*, 29(4):731–745, 2011.

[Anantharam *et al.*, 1987] Venkatachalam Anantharam, Pravin Varaiya, and Jean Walrand. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part i: Iid rewards. *IEEE Transactions on Automatic Control*, 32(11):968–976, 1987.

[Besson and Kaufmann, 2018] Lilian Besson and Emilie Kaufmann. Multi-player bandits revisited. In *Algorithmic Learning Theory*, pages 56–92. PMLR, 2018.

[Bistritz and Leshem, 2018] Ilai Bistritz and Amir Leshem. Distributed multi-player bandits-a game of thrones approach. In *Advances in Neural Information Processing Systems*, pages 7222–7232, 2018.

[Boursier and Perchet, 2019] Etienne Boursier and Vianney Perchet. Sic-mmab: Synchronisation involves communication in multiplayer multi-armed bandits. In *Advances in Neural Information Processing Systems*, volume 32, pages 12071–12080, 2019.

[Boursier and Perchet, 2020] Etienne Boursier and Vianney Perchet. Selfish robustness and equilibria in multi-player bandits. In *Conference on Learning Theory*, pages 530–581. PMLR, 2020.

[Cappé *et al.*, 2013] Olivier Cappé, Aurélien Garivier, Odalric-Ambrym Maillard, Rémi Munos, and Gilles Stoltz. Kullback-leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, pages 1516–1541, 2013.

[Combes *et al.*, 2015] Richard Combes, Stefan Magureanu, Alexandre Proutiere, and Cyrille Laroche. Learning to rank: Regret lower bounds and efficient algorithms. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 231–244, 2015.

[Drolia *et al.*, 2017] Utsav Drolia, Katherine Guo, and Priya Narasimhan. Precog: Prefetching for image recognition applications at the edge. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, pages 1–13, 2017.

[Jouini *et al.*, 2009] Wassim Jouini, Damien Ernst, Christophe Moy, and Jacques Palicot. Multi-armed bandit based policies for cognitive radio's decision making issues. In *2009 3rd International Conference on Signals, Circuits and Systems (SCS)*, pages 1–6. IEEE, 2009.

[Kalathil *et al.*, 2014] Dileep Kalathil, Naumaan Nayyar, and Rahul Jain. Decentralized learning for multiplayer multiarmed bandits. *IEEE Transactions on Information Theory*, 60(4):2331–2345, 2014.

[Lai and Robbins, 1985] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[Landgren *et al.*, 2016] Peter Landgren, Vaibhav Srivastava, and Naomi Ehrich Leonard. Distributed cooperative decision-making in multiarmed bandits: Frequentist and bayesian algorithms. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 167–172. IEEE, 2016.

[Liu and Zhao, 2010] Keqin Liu and Qing Zhao. Decentralized multi-armed bandit with multiple distributed players. In *2010 Information Theory and Applications Workshop (ITA)*, pages 1–10. IEEE, 2010.

[Liu *et al.*, 2020] Lydia T Liu, Horia Mania, and Michael Jordan. Competing bandits in matching markets. In *International Conference on Artificial Intelligence and Statistics*, pages 1618–1628. PMLR, 2020.

[Magesh and Veeravalli, 2021] Akshayaa Magesh and Venugopal V Veeravalli. Decentralized heterogeneous multiplayer multi-armed bandits with non-zero rewards on collisions. *IEEE Transactions on Information Theory*, 2021.

[Martínez-Rubio *et al.*, 2019] David Martínez-Rubio, Varun Kanade, and Patrick Rebeschini. Decentralized cooperative stochastic bandits. *Advances in Neural Information Processing Systems*, 32:4529–4540, 2019.

[Mehrabian *et al.*, 2020] Abbas Mehrabian, Etienne Boursier, Emilie Kaufmann, and Vianney Perchet. A practical algorithm for multiplayer bandits when arm means vary among players. In *International Conference on Artificial Intelligence and Statistics*, pages 1211–1221. PMLR, 2020.

[Narayanan *et al.*, 2020] Arvind Narayanan, Eman Ramadan, Jason Carpenter, Qingxu Liu, Yu Liu, Feng Qian, and Zhi-Li Zhang. A first look at commercial 5g performance on smartphones. In *Proceedings of The Web Conference 2020*, pages 894–905, 2020.

[Rosenski *et al.*, 2016] Jonathan Rosenski, Ohad Shamir, and Liran Szlak. Multi-player bandits–a musical chairs approach. In *International Conference on Machine Learning*, pages 155–163. PMLR, 2016.

[Sankararaman *et al.*, 2021] Abishek Sankararaman, Soumya Basu, and Karthik Abinav Sankararaman. Dominate or delete: Decentralized competing bandits in serial dictatorship. In *International Conference on Artificial Intelligence and Statistics*, pages 1252–1260. PMLR, 2021.

[Shi *et al.*, 2021] Chengshuai Shi, Wei Xiong, Cong Shen, and Jing Yang. Heterogeneous multi-player multi-armed bandits: Closing the gap and generalization. *Advances in Neural Information Processing Systems*, 34, 2021.

[Wang *et al.*, 2020] Po-An Wang, Alexandre Proutiere, Kaito Ariu, Yassir Jedra, and Alessio Russo. Optimal algorithms for multiplayer multi-armed bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 4120–4129. PMLR, 2020.