Proceedings of Machine Learning Research 304, 2025 ACML 2025

Multi-play Multi-armed Bandits with Shareable Arm Capacities Revisited: Settling Scarce Capacity

Hanyang Li LIHANYANG@MAIL.USTC.EDU.CN
Hong Xie XIEHONG2018@FOXMAIL.COM
Defu Lian LIANDEFU@USTC.EDU.CN
School of Computer Science and Technology, University of Science and Technology of China

Editors: Hung-yi Lee and Tongliang Liu

Abstract

This paper revisits multi-play multi-armed bandit with shareable arm capacities problem, which is tailored for resource allocation problems arising from LLM inference serving, edge intelligence, etc. We investigate the capacity-scarce setting, a common dilemma in resource allocation problems. Existing works yield sub-optimal solutions under this setting, as they rely heavily on the assumption of abundant capacities. This paper present a rather complete solution to this setting and it makes three key contributions. We establish a lower bound for the sample complexity of learning the arm capacities and propose an algorithm that exactly matches this lower bound. We derive both instance-independent and instance-dependent regret lower bounds for learning the optimal play assignment. We introduce an efficient exploration algorithm named `PC-CapUL` for the capacity-scarce setting and `PC-CapUL` matches the regret lower bounds up to an acceptable constant. `PC-CapUL` features a novel index for coordinating the exploration of multiple plays. Experiments show significant improvement over existing methods.

Keywords: Multi-play multi-armed bandit, scarce shareable arm capacity, regret bounds

1. Introduction

Multi-play multi-armed bandit (MP-MAB) is a natural and popular variant of the vanilla multi-armed bandits framework (Anantharam et al. (1987a)). MP-MAB has various applications including online advertising (Lagrée et al. (2016); Komiyama et al. (2017); Yuan et al. (2023)), power systems (Lesage-Landry and Taylor (2017)), mobile edge computing (Chen and Xie (2022); Wang et al. (2022a); Xu et al. (2023)), etc. The canonical MP-MAB model involves $K \in \mathbb{N}_+$ arms, where in each round, the learner assigns $N \in \mathbb{N}_+$ plays across the arms. Each arm can be pulled by at most one play. When an arm is pulled, it generates a reward modeled as a sample from a random variable with an unknown mean but a known tail property, such as a standard sub-Gaussian tail. Research on MP-MAB remains highly active, as evidenced by various recent generalizations (Chen and Xie (2022); Moulos (2020); Xu et al. (2023); Wang et al. (2022a); Yuan et al. (2023)).

One notable generalization of MP-MAB is MP-MAB with shareable arm capacity(MP-MAB-SAC)(Xu et al. (2023); Wang et al. (2022a,c)), which models each arm with a finite capacity while allowing multiple plays to be assigned to the same arm. When the number of plays exceeds the arm's capacity, the utility generated by the arm is shared among these plays, and the shared utility typically is the only observable value. This generalization captures the resource-sharing nature of resource allocation problems arising from LLM inference serving, edge intelligence, etc Wang et al. (2022a). However, existing works differ in the details of their models. Xu et al. (2023) considered the outcomes of individual plays in a competitive environment. Wang et al. (2022c) examined a more cooperative environment in a decentralized setting. Wang et al. (2022a) focused on a centralized and capacity-abundant scenario. Despite these contributions, the theoretical guarantees in existing MP-MAB-SAC works lack completeness, and their methods incur significant errors under the capacity-scarce setting. To address this gap, we focus on the capacity-scarce setting, aiming to establish comprehensive theoretical results and design an efficient algorithm.

To illustrate, our model considers a finite number of $K \in \mathbb{N}_+$ arms and a finite number of $N \in \mathbb{N}_+$ plays. Each arm k is characterized by a tuple (m_k, μ_k, σ), where $m_k \in \mathbb{N}_+$ models the capacity limit and $\mu_k \in \mathbb{R}_+$ models the unit-capacity reward mean. Both m_k and μ_k are unknown to the learner, with the arm capacity m_k being deterministic. The reward function of assigning $a_k \in \mathbb{N}_+$ plays to arm k is modeled as: $R_k(a_k) = \min\{a_k, m_k\}\mu_k + \epsilon_k$, where ϵ_k is a zero mean σ-sub-Gaussian random noise. Note that our reward model is rooted in the reward structure of conventional linear bandits with one dimensional features (Lattimore and Szepesvári (2020)). The capacity-scarce setting, defined by $N \geq M$ (where $M := \sum_{k=1}^{K} m_k$), is more suitable for scenarios involving intense competition under limited resources, which are frequently encountered in the real world. For example, the computing resources of certain advanced servers are bound to be in high demand, resulting in resource scarcity. Additionally, we take into account the movement cost of plays to further enhance the practicality of our model. Assigning a play to an arm incurs a constant movement cost $c \in \mathbb{R}_+$, which is assumed to satisfy $c < \min_k \mu_k$. This movement cost introduces a constraint for the exploration process.

Table 1: Summary of Main Theoretical Results

	Lower Bound	**Upper Bound**
Sample complexity	$\Omega\left(\frac{\sigma^2}{\mu_k^2}\log\delta^{-1}\right)$ (Thm 2)	$O\left(\frac{\sigma^2}{\mu_k^2}\log\delta^{-1}\right)$ (Thm 5)
Instance-dependent regret	$O\left(\sum_k \frac{c\sigma^2}{\mu_k^2}\log T\right)$ (Thm 10)	$O\left(\sum_k \left(\frac{N}{m_k}c + K(\mu_k - c)\right)\frac{\sigma^2}{\mu_k^2}\log T\right)$ (Thm 14)
Instance-independent regret	$O\left(\sigma\sqrt{TK}\right)$ (Thm 7)	$O\left(K\sigma\sqrt{NT\log T}\right)$ (Thm 16)

1.1. Main Results and Contributions

In our MP-MAB-SAC problem setting, the theoretical conclusions are summarized in Table 1. Our contributions can be categorized into the following three aspects:
Sample complexity. We establish a lower bound $\Omega(\frac{\sigma^2}{\mu_k^2}\log\delta^{-1})$ for the sample complexity of learning the arm capacity, and propose an active inference algorithm named `ActInfCap`

that achieves this lower bound exactly. The key finding is that the difficulty of learning arm capacity is determined by the per-capacity reward mean. We introduce new uniform confidence intervals for arm capacity estimation and a novel approach that actively probes an arm using its capacity's UCB or LCB for data-efficient learning. In the data-gathering process,the UCB and LCB are adopted alternately. These findings provide new insights into arm capacity estimation and provide foundational building blocks for designing data-efficient exploration algorithms.

Regret lower bounds. We prove an instance-independent regret lower bound $\Omega(\sigma\sqrt{TK})$ and an instance-dependent regret lower bound $\Omega(\sum_{k=1}^{K}\frac{c\sigma^2}{\mu_k^2}\log T)$. Notably, these regret lower bounds are independent of the arm capacity m_k. At first glance, this may appear counterintuitive; however, it aligns with our sample complexity lower bound, which also demonstrates that sample complexity is independent of arm capacity. Additionally, the dependence on the reward mean μ_k is consistent with dependence observed in the sample complexity. This finding highlights that the difficulty of learning the optimal action is primarily governed by the number of arms K and the per-unit capacity reward mean μ_k. Increasing the number of arms or decreasing the reward mean would make the learning process more challenging.

Data efficient exploration. We propose an adaptive algorithm named `PC-CapUL`, which leverages prioritized coordination of arm capacity upper/lower confidence bounds (UCB/LCB) to efficiently balance the exploration-exploitation trade-off. We prove both instance-dependent and instance-independent upper bounds for `PC-CapUL`, which match the corresponding lower bounds up to certain acceptable model-dependent factors. Numerical experiments validate the data efficiency of `PC-CapUL` in the capacity-scarce setting. The main idea of `PC-CapUL` consists of four key aspects: (1) *Preventing excessive UEs.* At each time slot, ensure that the arms played by united explorations(UE) in the previous time slot are played by individual explorations(IE) in the current time slot. Here, UE/IE refers to explorations where the number of plays assigned to an arm equals its capacity UCB/LCB. (2) *Balancing UEs and IEs.* Excessive IEs on a single arm is also not advisable. (3) *Favorable arms win UE first.* At each time slot, when multiple arms compete for UEs, we resolve this competition using a carefully selected criterion defined as $OrcInd_{k,t}$, which is determined based on insights from the sample regret. (4) *Stop learning when converges.* At each time slot t, once the upper and lower bounds of an arm's capacity converge, no further exploration is performed on that arm.

2. Related Work

Methodology perspective. To the best of our knowledge, MP-MAB was first studied by Anantharam et al. (1987a), where an asymptotic regret lower bound was established and an algorithm achieving the lower bound asymptotically was proposed. The regret lower bound in the finite time is achieved by Komiyama et al. (2015) via Thompson sampling. Markovian rewards variant of MP-MAB was studied in Anantharam et al. (1987b). Some recent generalization of MP-MAB include: cascading MP-MAB where the order of plays is captured into the reward function (Lagrée et al. (2016); Komiyama et al. (2017)), MP-MAB with switching cost (Agrawal et al. (1990); Jun (2004)), MP-MAB with budget constraint (Luedtke et al. (2019); Xia et al. (2016); Zhou and Tomlin (2018)) and MP-MAB with a stochastic number of plays in each round (Lesage-Landry and Taylor (2017)), sleeping MP-MAB (Yuan et al. (2023)), MP-MAB with shareable arm capacities (Chen and Xie (2022); Wang et al. (2022a); Xu et al. (2023)).

Our work falls into the research line of MP-MAB with shareable arm capacities (Chen and Xie (2022); Wang et al. (2022a,b); Xu et al. (2023); Mo and Xie (2023)). The shareable arm capacities models can be categorized into two types: (1) stochastic arm capacity but with feedback on the realization of arm capacity (Chen and Xie (2022); Mo and Xie (2023)); (2) deterministic capacity without any realization of the arm capacity (Wang et al. (2022a,b); Xu et al. (2023)). Although the difference seems small, the two settings lead to fundamentally different research problems and techniques for addressing it. For the stochastic arm capacity line, Chen and Xie (2022) models the arm capacity as a random variable, but in each round the sample of the arm capacities of all arms are revealed to the decision, i.e., expert feedback on arm capacity. One can directly estimate the distribution of arm capacity from the capacity samples. Mo and Xie (2023) generalizes this model to the distributed setting, and uses the realization of the arm capacity as a signal for coordination. However, the deterministic arm capacity is technically different. Though the capacity is deterministic, it is unknown and the decision maker can only access samples from the reward function, while no samples on the arm capacity can be observed. Wang et al. (2022a,b); Xu et al. (2023) consider the setting in which multiple strategic agents compete for the resource. Nash equilibrium in the offline setting is established. Our work revisits this research line, motivated by the lack of specific and detailed studies about the capacity-scarce setting, in which existing MP-MAB-SAC methods yield sub-optimal regret levels.

Applications perspective. MP-MAB-SAC is a versatile model with numerous real-world applications. As illustrated in Wang et al. (2022a), MP-MAB-SAC can be applied to edge computing, cognitive ratio applications , online advertisement placement etc. To avoid repetition, we present another instance of MP-MAB-SAC application. Here we elaborate on how to map our model to LLM inference serving applications (Li et al. (2024)). In this context, each arm corresponds to a deployment instance of an LLM. The arm capacity models the number of queries that an LLM can process within a given time slot. Due to the multiplexing behavior of computing systems, the capacity is unknown and the processing is uncertain (Zhu et al. (2023)). An LLM deployed on more powerful computing facilities would be modeled with larger capacity. The reward mean μ_k can be mapped to the capability of an LLM such as large, medium, or small LLM mixed inference serving. The cost c can be interpreted as the communication cost incurred when transmitting queries to a commercial LLM server. Running an LLM service involves various expenses, including those for computing resources, IT operations, and system maintenance. Transmitting an end user's query to the server also incurs communication costs, especially when the query includes a lengthy prompt. The cost c serves as an aggregate abstraction of these various costs.

3. Model and problem Formulation

By default, for any integer $N \in \mathbb{N}_+$: $[N] := \{1, \dots, N\}$. Consider $K \in \mathbb{N}_+$ arms indexed by $[K]$ and $N \in \mathbb{N}_+$ plays to be assigned to these arms. Each arm $k \in [K]$ is characterized by a tuple (m_k, μ_k, σ), where $m_k \in [N]$ and $\mu_k \in \mathbb{R}$ and $\sigma \in \mathbb{R}$. Here, m_k models the capacity of arm k, μ_k models the per-unit reward mean of arm k, and $\sigma \in \mathbb{R}_+$ models tail property of the reward, i.e., σ-sub-Gaussian. Both m_k and μ_k are unknown to the learner, and the capacity m_k is deterministic. We consider the scarce arm capacity setting, such that $N \geq M$, where $M := \sum_{k=1}^{K} m_k$ denotes the total amount of capacities across all arms. For every play there is a constant movement cost c to an arm, which is known to the learner. The movement cost can model the transmission cost in edge intelligence applications, etc. From a learning perspective, it introduces a cost constraint to exploration. Let $a_k \in [N]$ denote the number of plays assigned to arm $k \in [K]$. The reward function associated with a_k is:

$$R_k(a_k) = \min\{a_k, m_k\}\mu_k + \epsilon_k.$$

Consider $T \in \mathbb{N}_+$ time slots. Let $a_{k,t} \in [N] \cup \{0\}$ denote the number of plays assigned to the arm k at time slot t, and the action taken at time t is represented by the vector $\mathbf{a}_t := (a_{1,t}, a_{2,t}, ..., a_{K,t})$. The action space $\mathcal{A}$ is:

$$\mathcal{A} := \left\{ (a_1, a_2, ..., a_K) \in \mathbb{N}^K \middle| \sum\nolimits_{k\in[K]} a_k \leq N \right\}.$$

Let $U_{k,t}$ denote the utility of the action $\mathbf{a}_t$ at time slot t on arm k , defined as the reward minus the movement cost:

$$U_{k,t} := R_k(a_{k,t}) - c \cdot a_{k,t}.$$

We then define the expected utility for action $\mathbf{a}_t$ as $f(\mathbf{a}_t)$:

$$f(\mathbf{a}_t) : = \mathbb{E}\left[\sum\nolimits_{k\in[K]} U_{k,t}\right] = \sum\nolimits_{k\in[K]} \left(\min\{a_{k,t}, m_k\} \cdot \mu_k - c \cdot a_{k,t}\right).$$

Let $\mathbf{a}^*$ denote the optimal action $\mathbf{a}$ that maximizes the expected utility $f(\mathbf{a})$,i,e.: $\mathbf{a}^* := \arg\max_{\mathbf{a}} f(\mathbf{a})$. It is evident that the optimal action is $\mathbf{a}^* = (m_1, m_2, ..., m_K)$. The challenge lies in distinguishing the capacities of all arms, where the order plays a crucial role in this problem. The objective is to minimize the regret over T time slots, which is defined as $\mathrm{Reg}(T)$:

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

The regret generated on the arm k is defined as $\mathrm{Reg}_k(T)$:

$$\mathrm{Reg}_k(T) := T(m_k\mu_k - cm_k) - \mathbb{E}\left[\sum\nolimits_{t=1}^{T} \left(\min\{a_{k,t}, m_k\} \cdot \mu_k - c \cdot a_{k,t}\right)\right].$$

As mentioned above, the movement cost c can model the transmission cost in edge intelligence application. It is reasonable to set $c > 0$, as the transmission cost is usually non-zero. Moreover, there is a significant distinction between the cases of $c > 0$ and $c = 0$. When $c > 0$, the optimal action is unique, and the primary objective becomes learning the capacities of all arms. However, when $c = 0$, the optimal actions are no longer unique. For any action $\mathbf{a}$ satisfying $a_k \geq m_k$ for all $k \in [K]$, no regret is generated. This shifts the goal from learning the arm capacities to finding such actions $\mathbf{a}$. This idea contrasts with that in Wang et al. (2022a), where learning the capacities of the optimal arms is essential. To maintain practicality and consistency with prior work, we do not consider the case when $c = 0$.

4. Sample Complexity of Estimating Arm Capacity

4.1. Sample Complexity Lower Bound

We focus on understanding the complexity of inferring the arm capacities, as this directly determines the optimal allocation of plays. Specifically, we consider the scenario where, for a fixed arm k, an inference algorithm π_{Inf} generates samples by assigning $a_{k,t} \in [N]$ plays to the arm."

Definition 1 **((Wang et al. (2022a)))** *An action $a_{k,t}$ is United Exploration (UE) if $a_{k,t} > m_k$. An action $a_{k,t}$ is individual exploration (IE) if $a_{k,t} \leq m_k$.*

Note that $1 \leq m_k < N$ is assumed as a prior, making both UEs and IEs feasible for π_{Inf}. We consider a space of all inference algorithm π_{Inf} that can adaptively adjust the numbers of both UEs and IEs.

Theorem 2 *For any inference algorithm π_{Inf}, there exists an instance of arm k such that:*

$$\mathbb{P}\left[\hat{m}_{k,t} \neq m_k | t \leq \frac{2\sigma^2}{\mu_k^2} \log\left(\frac{1}{4\delta}\right)\right] \geq 1 - \delta,$$

where $\hat{m}_{k,t}$ denotes the estimator produced by π_{Inf}.

Remark 3 *Theorem 2 establishes a lower bound of $\Omega(\frac{\log \delta^{-1}}{\mu_k^2})$ for the sample complexity of estimating arm capacity. This lower bound is independent of the arm's capacity but solely depends on μ_k.*

4.2. Sample Efficient Algorithm

Uniform confidence interval for arm capacity. First we formally define $\tau_{k,t}$ and $\iota_{k,t}$ as the number of IEs and UEs on arm k up to time slot t:

$$\tau_{k,t} = \sum\nolimits_{s=1}^{t} \mathbb{1}\{a_{k,s} \leq m_k\}, \iota_{k,t} = \sum\nolimits_{s=1}^{t} \mathbb{1}\{a_{k,s} > m_k\}.$$

Since the real capacity m_k is unknown during the training process, we should use the confidence interval $[m_{k,t}^l, m_{k,t}^u]$ rather than the capacity m_k itself to calculate an empirical version of $\tau_{k,t}$ and $\iota_{k,t}$. We define the empirical versions of $\tau_{k,t}$ and $\iota_{k,t}$ as $\hat{\tau}_{k,t}$ and $\hat{\iota}_{k,t}$, respectively:

$$\hat{\tau}_{k,t} = \sum\nolimits_{s=1}^{t} \mathbb{1}\{a_{k,s} \leq m_{k,s-1}^l\},$$
$$\hat{\iota}_{k,t} = \sum\nolimits_{s=1}^{t} \mathbb{1}\{a_{k,s} \geq m_{k,s-1}^u\}.$$

Another term we require is the cumulative squared sum of plays in IEs.

$$\hat{V}_{k,t} = \sum\nolimits_{s=1}^{t} a_{k,s}^2 \cdot \mathbb{1}\{a_{k,s} \leq m_{k,s-1}^l\}.$$

The estimator of μ_k up to time slot t is denoted as $\hat{\mu}_{k,t}$. Let $\upsilon_k := m_k\mu_k$ and the estimator of $m_k\mu_k$ up to time slot t is denoted as $\hat{\upsilon}_{k,t}$:

$$\hat{\mu}_{k,t} = \left(\sum\nolimits_{s=1}^{t} a_{k,s}\left(U_{k,s} + c \cdot a_{k,s}\right) \cdot \mathbb{1}\left\{a_{k,s} \leq m_{k,s-1}^{l}\right\}\right) \Big/ \hat{V}_{k,t}, \tag{1}$$

$$\hat{\upsilon}_{k,t} = \left(\sum\nolimits_{s=1}^{t} \left(U_{k,s} + c \cdot a_{k,s}\right) \cdot \mathbb{1}\left\{a_{k,s} \geq m_{k,s-1}^{u}\right\}\right) \Big/ \hat{\iota}_{k,t}. \tag{2}$$

To simplify the notation, we define the function :

$$\phi\left(x, \delta\right) := \sqrt{\left(1 + \frac{1}{x}\right)\frac{2\log\left(2\sqrt{x+1}/\delta\right)}{x}}.$$

Lemma 4 *The confidence intervals of the estimators $\hat{\mu}_{k,t}$ and $\hat{\upsilon}_{k,t}$ can be calculated as:*

$$\hat{\mu}_{k,t} \in [\mu_k - \sigma\phi(\hat{V}_{k,t}, \delta), \mu_k + \sigma\phi(\hat{V}_{k,t}, \delta)], \tag{3}$$

$$\hat{\upsilon}_{k,t} \in [\upsilon_k - \sigma\phi\left(\hat{\iota}_{k,t}, \delta\right), \upsilon_k + \sigma\phi\left(\hat{\iota}_{k,t}, \delta\right)]. \tag{4}$$

For any adaptive algorithm using the first K time slots for initialization, when $\sigma\phi(\hat{V}_{k,t}, \delta) < \hat{\mu}_{k,t}$, define event A_k :

$$A_k := \left\{\forall t \in [T], t > K, (3)(4) \text{ is correct}, m_k \in \left[\frac{\hat{\upsilon}_{k,t} - \sigma\phi\left(\hat{\iota}_{k,t}, \delta\right)}{\hat{\mu}_{k,t} + \sigma\phi(\hat{V}_{k,t}, \delta)}, \frac{\hat{\upsilon}_{k,t} + \sigma\phi\left(\hat{\iota}_{k,t}, \delta\right)}{\hat{\mu}_{k,t} - \sigma\phi(\hat{V}_{k,t}, \delta)}\right]\right\}.$$

Then for fixed k, the probability that A_k holds is at least $1 - \delta$.

This lemma ensures that our confidence intervals are accurate with high probability during the learning process. Let $A := \bigcap_{k=1}^{K} A_k$. One straightforward application of the union bound inequality shows that A holds with a probability of at least $1 - K\delta$. When event A occurs, the confidence bounds for all estimators are correct, and the capacity confidence bounds are accurate for all $k \in [K]$ and $t \in [T]$. Consequently, the capacity of any arm should not exceed the sum of the lower bounds of the capacities of the other arms. We now can define the capacity confidence lower bound $m_{k,t}^{l}$ and the upper bound $m_{k,t}^{u}$ as the end points of the capacity confidence interval of m_k, and we refine the bounds under the assumption that event A occurs as follows:

$$m_{k,t}^{l} := \max\left\{\left\lceil\frac{\hat{\upsilon}_{k,t} - \sigma\phi(\hat{\iota}_{k,t}, \delta)}{\hat{\mu}_{k,t} + \sigma\phi(\hat{V}_{k,t}, \delta)}\right\rceil, 1\right\}, \tag{5}$$

$$m_{k,t}^{u} := \min\left\{\left\lfloor\frac{\hat{\upsilon}_{k,t} + \sigma\phi\left(\hat{\iota}_{k,t}, \delta\right)}{\hat{\mu}_{k,t} - \sigma\phi(\hat{V}_{k,t}, \delta)}\right\rfloor, N - \sum_{i \neq k}^{K} m_{i,t}^{l}\right\}. \tag{6}$$

Now we compare the arm capacity confidence intervals with those in Wang et al. (2022a):

$$m_{k,t}^{l} = \max\left\{\left\lceil\frac{\hat{\upsilon}_{k,t}}{\hat{\mu}_{k,t} + \sigma\phi\left(\hat{\tau}_{k,t}, \delta\right) + \sigma\phi\left(\hat{\iota}_{k,t}, \delta\right)}\right\rceil, 1\right\},$$

$$m_{k,t}^{u} = \min\left\{\left\lfloor\frac{\hat{\upsilon}_{k,t}}{\hat{\mu}_{k,t} - \sigma\phi\left(\hat{\tau}_{k,t}, \delta\right) - \sigma\phi\left(\hat{\iota}_{k,t}, \delta\right)}\right\rfloor, N\right\}.$$

The similarity in the capacity confidence intervals stems from adopting the same approach as in Wang et al. (2022a), where UEs and IEs are treated separately for capacity estimation. In both Wang et al. (2022a) and our work, the UCB and LCB of the capacity m_k are derived solely from the empirical mean values of μ_k and υ_k. However, the key distinction lies in how the estimation error of UE, represented by the term $\sigma\phi\left(\hat{\imath}_{k,t},\delta\right)$, is handled. While Wang et al. (2022a) place this term in the denominator, we place it above the denominator. This modification results in our UCB being smaller and our LCB being larger than theirs for the same UEs and IEs. Consequently, their method requires more rounds of UEs and IEs for the confidence intervals to converge. This is substantiated through experiments, and the results are presented in the supplementary materials.

$$m^l_{k,t} = \max\left\{\left\lceil\frac{\hat{\upsilon}_{k,t}}{\hat{\mu}_{k,t} + \sigma\phi(\hat{V}_{k,t},\delta) + \sigma\phi\left(\hat{\imath}_{k,t},\delta\right)}\right\rceil, 1\right\},$$

$$m^u_{k,t} = \min\left\{\left\lfloor\frac{\hat{\upsilon}_{k,t}}{\hat{\mu}_{k,t} - \sigma\phi(\hat{V}_{k,t},\delta) - \sigma\phi\left(\hat{\imath}_{k,t},\delta\right)}\right\rfloor, N\right\}.$$

Algorithm 1 states `ActInfCap`, which estimates the arm capacity by adaptively probing the arm with varying numbers of plays to generate samples. Specifically, `ActInfCap` leverages the UCB and LCB to guide sample generation for each arm. At the core of `ActInfCap` lies the newly proposed confidence intervals of the arm capacity. In `ActInfCap`, the UEs and IEs are assigned alternately, allowing the UCB and LCB of the arm capacity to approach each other as more utilities are observed.

Algorithm 1 `ActInfCap`(k, T)

```
Initialize: $t \leftarrow 0$, $m^l_{k,0} \leftarrow 1$, $m^u_{k,0} \leftarrow N$.
Do one UE and one IE respectively.
Observe $U_{k,1}$ and $U_{k,2}$. $m^u_{k,2} \leftarrow N, m^l_{k,2} \leftarrow 1, t \leftarrow 2$.
while $t < T$ and $m^l_{k,t-1} < m^u_{k,t-1}$ do
    $t \leftarrow t+1$
    if $t$ is an odd number then
        $a_{k,t} \leftarrow m^l_{k,t-1}$ plays to arm $k$, observe $U_{k,t}$.
        Update $m^l_{k,t}, m^u_{k,t}$ via Equation (5) and (6)
    else
        $a_{k,t} \leftarrow m^u_{k,t-1}$ plays to arm $k$, observe $U_{k,t}$.
        Update $m^l_{k,t}, m^u_{k,t}$ via Equation (5) and (6)
    end if
end while
Return $m^u_{k,t}$
```

Theorem 5 *The output of Algorithm 1, i.e.,* $m^u_{k,t}$ *satisfies:*

$$\mathbb{P}\left[\hat{m}^u_{k,t} = m_k | t \geq 10240\frac{\sigma^2}{\mu_k^2}\log\left(\frac{2}{\delta}\right) + 2\right] \geq 1 - \delta.$$

Remark 6 *Theorem 5 states that Algorithm 1 achieves a sample complexity that exactly matches the lower bound. This closes the sample complexity gap. Furthermore, this theorem implies that the number of explorations required to determine the capacity m_k is unrelated to m_k itself.*

5. Regret Lower Bounds and Sample Efficient Algorithms

5.1. Regret Lower Bounds

Theorem 7 *Given K and M, for any learning algorithm π, its instance-independent minimax regret lower bound is:*

$$\mathbb{E}\left[Reg\left(T,\pi\right)\right] \geq \frac{\sigma}{64e\sqrt{2}}\sqrt{TK}.$$

Remark 8 *This theorem indicates that the regret lower bound depends on $\sqrt{K}$ for the number of arms K and $\sqrt{T}$ for the learning horizon T. Notably, there is no dependence on the arm capacity m_k, which is consistent with the sample complexity bound stated in Theorem 2 and Theorem 5. Although Theorem 7 follows the conventional paradigm (Lattimore and Szepesvári (2020)), it is technically non-trivial. The key idea lies in carefully balancing the trade-off between the per-time-slot regret and the challenge of learning the arm capacities. When the utility is small, the per-time-slot regret is also small. However, distinguishing the capacities becomes more difficult since the gaps in expected utilities are small when the capacity gaps are the same.*

Theorem 9 *For any consistent learning strategy π, the regret generated on the arm k is lower-bounded as:*

$$\liminf_{T\to\infty} \frac{\mathbb{E}\left[Reg_k\left(T,\pi\right)\right]}{\log\left(T\right)} \geq 2\frac{c\sigma^2}{\mu_k^2}.$$

Following the naming convention in the sample complexity section, we refer to the regret generated on an arm, where there are always sufficient plays for UEs and IEs, as "sample" regret. The sample regret lower bound is presented above. This lower bound is derived by analyzing the expected number of UEs where $a_{k,t} > m_k$ during the learning process. A direct corollary for bounding the total regret can then be obtained by summing the sample regret across all K arms:

Theorem 10 *Given K, $\{m_k\}_{k\in[K]}$, and $\{\mu_k\}_{k\in[K]}$, for any consistent learning strategy π, it holds*

$$\liminf_{T\to\infty} \frac{\mathbb{E}\left[Reg\left(T,\pi\right)\right]}{\log\left(T\right)} \geq 2\sum_{k=1}^{K}\frac{c\sigma^2}{\mu_k^2}.$$

Remark 11 *Theorem 9 and 10 state that the instance-dependent regret lower bound depends on μ_k^{-2} . This indicates that smaller values of μ_k make it more challenging to learn the optimal action. Additionally, there is no dependence on the arm capacity m_k. The key idea in our proof of Theorem 9 and 10 is to derive a lower bound for the expected number of suboptimal actions over the entire T time slots.*

5.2. Efficient Exploration Algorithm

Motivated by the strong performance of the algorithm 1 in sample complexity upper bound proof, we now turn our attention to its sample regret upper bound. It should be noted that with alternating UEs and IEs, $m^l_{k,t}$ and $m^u_{k,t}$ converge to m_k, as demonstrated in the expressions (5) and (6). Consequently, subsequent UEs and IEs generate less regret, given that we assign $m^l_{k,t}$ or $m^u_{k,t}$ plays to arm k for IE or UE, respectively.

Theorem 12 *The regret generated by Algorithm 1 can be upper-bounded as:*

$$\begin{aligned}
&\mathbb{E}[Reg_k\left(T\right)] \\
\leq &\left(2048c + \frac{512\pi^2\left(\mu_k - c\right)}{3}\left(\frac{1024\pi^2}{3} + \frac{2048}{m_k}\right)\cdot\frac{N}{m_k}c\right)\frac{\sigma^2}{\mu_k^2}\log\left(T\right) + 2\max\left(m_k\mu_k, Nc\right) \\
=&O\left(\left(\frac{N}{m_k}c + \left(\mu_k - c\right)\right)\frac{\sigma^2}{\mu_k^2}\log\left(T\right)\right).
\end{aligned}$$

Remark 13 *This regret bound serves as a relatively tight upper bound when N and m_k do not differ significantly, and when the μ_k and c are close in value. Under these assumptions, the upper bound can be expressed as $O\left(\frac{c\sigma^2}{\mu_k^2}\log\left(T\right)\right)$. The N in the numerator of the regret upper bound originates from the UEs during the initial time slots, when the $m^u_{k,t}$ is not yet well-learned. The number of such suboptimal UEs is positively correlated with $\log(T)$, as it is derived from the confidence interval, which is also correlated with $\log(T)$. This indicates that the sample regret lower bound presented in Theorem 9 is relatively tight. Importantly, this sample regret upper bound offers valuable insights for designing efficient algorithms under the capacity-scarce setting.*

Efficient exploration algorithm. Algorithm 2 outlines `PC-CapUL`, an abbreviation of Prioritized Coordination of Capacities' UCB and LCB.

(1) **Preventing excessive UEs**(Line 10). At each time slot, we ensure that the arms played by UEs in the previous time slot are not played by UEs again in the current time slot. Compared to IEs, UEs are play-consuming, especially during the early time slots when the capacity confidence intervals are not yet well learned. Overloading a particular arm with UEs can hinder the learning process for other arms that also require UEs to refine their capacity intervals. This occurs because there are often insufficient plays for all arms to be explored with UEs simultaneously. Furthermore, alternating exploration between UEs and IEs is, to some extent, an optimal strategy, as proven in Theorem 12. This approach ensures that the regret generated by UEs is minimized.

(2)**Balancing UEs and IEs**(Line 12). Excessive IEs on a single arm is also not advisable, as an increase in $\hat{V}_{k,t}$ alone cannot efficiently improve $m^l_{k,t}$ without a corresponding increase in $\hat{\iota}_{k,t}$. Therefore, it is reasonable to balance the number of UEs and IEs. One approach to maintain this balance is using $OrcInd_{k,t}$ as a criterion, defined as

$$OrcInd_{k,t} := \phi(\hat{V}_{k,t}, \delta) + \phi\left(\hat{\iota}_{k,t}, \delta\right).$$

The $OrcInd_{k,t}$ stands for "orchestra index". Here, $\sigma\phi\hat{V}_{k,t}, \delta)$ and $\sigma\phi\left(\hat{\iota}_{k,t}, \delta\right)$ measure the length of the confidence intervals for υ_k and μ_k respectively. The sum of these values

Algorithm 2 `PC-CapUL`

```
1: Notation: $\boldsymbol{m}^l_t := (m^l_{k,t} : k \in [K]), \boldsymbol{m}^u_t := (m^u_{k,t} : k \in [K]), \boldsymbol{U}_t := (U_{k,t} : k \in [K]), \hat{\boldsymbol{\tau}}_t := (\hat{\tau}_{k,t} : k \in [K]), \hat{\boldsymbol{V}}_t := (\hat{V}_{k,t} : k \in [K]), \hat{\boldsymbol{\iota}}_t := (\hat{\iota}_{k,t} : k \in [K]), \hat{\boldsymbol{\mu}}_t := (\hat{\mu}_{k,t} : k \in [K]), \hat{\boldsymbol{\upsilon}}_t := (\hat{\upsilon}_{k,t} : k \in [K])$.
   $\boldsymbol{Cndt} := (Cndt_k : k \in [K])$ is a binary vector indicating continue exploration (1) or not (0).
   $\boldsymbol{w} := (w_k, k \in [K])$ is a binary vector with entry 1 indicating do IE and 0 indicating do UE.
   $\odot$ denotes the Hadamard product, $\boldsymbol{e}_k$ denotes a unit vector with $k$-th entry being 1.
2: Initialization: $\boldsymbol{m}^l_0 \leftarrow \boldsymbol{1}, \boldsymbol{m}^u_0 \leftarrow (N - K + 1)\boldsymbol{1}, \hat{\boldsymbol{\tau}}_0 \leftarrow \boldsymbol{0}, \hat{\boldsymbol{V}}_0 \leftarrow \boldsymbol{0}, \hat{\boldsymbol{\iota}}_0 \leftarrow \boldsymbol{0}, \boldsymbol{Cndt} \leftarrow \boldsymbol{1}$.
3: for $1 \leq t \leq K$ do
4:    The $t$-th arm do UE and all others do IE: $\boldsymbol{w} \leftarrow \boldsymbol{1} - \boldsymbol{e}_t$.
5:    Arm assignment: $\boldsymbol{a}_t \leftarrow (\boldsymbol{1} - \boldsymbol{w}) \odot \boldsymbol{m}^u_{t-1} + \boldsymbol{w} \odot \boldsymbol{m}^l_{t-1}$.   Observe $\boldsymbol{U}_t$, $\boldsymbol{m}^l_t \leftarrow \boldsymbol{m}^l_{t-1}$.
6:    $\boldsymbol{m}^l_t \leftarrow \boldsymbol{m}^l_{t-1}, \boldsymbol{m}^u_t \leftarrow \boldsymbol{m}^u_{t-1}, \hat{\boldsymbol{\tau}}_t \leftarrow \hat{\boldsymbol{\tau}}_{t-1} + \boldsymbol{w}, \hat{\boldsymbol{V}}_t \leftarrow \hat{\boldsymbol{V}}_{t-1} + \boldsymbol{w} \odot \boldsymbol{a}_t \odot \boldsymbol{a}_t, \hat{\boldsymbol{\iota}}_t \leftarrow \hat{\boldsymbol{\iota}}_{t-1} + \boldsymbol{1} - \boldsymbol{w}$, $\hat{\boldsymbol{\mu}}_t$ with (1), $\hat{\boldsymbol{\upsilon}}_t$ with (2).
7: end for
8: while $K + 1 \leq t \leq T$ do
9:    if $\boldsymbol{Cndt} \neq \boldsymbol{0}$ then
10:      Record the arms whose actions are UEs at last time slot: $w_k \leftarrow \mathbb{I}\{a_{t-1,k} \geq m^u_{k,t}\}, \forall k$.
11:      Converged arms: $w_k \leftarrow \mathbb{I}\{Cndt_k = 0\}, \forall k.$,   Update capacity needs: $M_{needs} \leftarrow (\boldsymbol{1} - \boldsymbol{w}) \cdot \boldsymbol{m}^u_{t-1} + \boldsymbol{w} \cdot \boldsymbol{m}^l_{t-1}$.
12:      $\boldsymbol{\ell} \leftarrow$ sort arms based on $OrcInd_{k,t} = \phi(\hat{V}_{k,t}, \delta) + \phi\,(\hat{\iota}_{k,t}, \delta)$ in descending order with $Cndt_k \neq 0$.
13:      for $k = 1, \ldots, K$ do
14:         if $M_{needs} > N$ then
15:            The ranked $k$-th arm (with index $\ell_k$) do IE, and update it to the vector $\boldsymbol{w} \leftarrow \boldsymbol{w} + \boldsymbol{e}_{\ell_k}$ .
16:            Update capacity needs: $M_{needs} \leftarrow (\boldsymbol{1} - \boldsymbol{w}) \cdot \boldsymbol{m}^u_{t-1} + \boldsymbol{w} \cdot \boldsymbol{m}^l_{t-1}$.
17:         end if
18:      end for
19:      $\boldsymbol{a}_t \leftarrow (\boldsymbol{1} - \boldsymbol{w}) \odot \boldsymbol{m}^u_{t-1} + \boldsymbol{w} \odot \boldsymbol{m}^l_{t-1}$.   Observe $\boldsymbol{U}_t$.
20:      $\hat{\boldsymbol{\tau}}_t \leftarrow \hat{\boldsymbol{\tau}}_{t-1} + \boldsymbol{w}, \hat{\boldsymbol{V}}_t \leftarrow \hat{\boldsymbol{V}}_{t-1} + \boldsymbol{w} \odot \boldsymbol{a}_t \odot \boldsymbol{a}_t, \hat{\boldsymbol{\iota}}_t \leftarrow \hat{\boldsymbol{\iota}}_{t-1} + \boldsymbol{1} - \boldsymbol{w}$, Update $\hat{\boldsymbol{\mu}}_t$ with (1), $\hat{\boldsymbol{\upsilon}}_t$ with (2), $\boldsymbol{m}^l_t$ with (5), $\boldsymbol{m}^u_t$ with (6), $Cndt_k \leftarrow \mathbb{I}\{\boldsymbol{m}^l_{k,t} < \boldsymbol{m}^u_{k,t}\}, \forall k$.
21:   else
22:      Observe $\boldsymbol{U}_t$, $\boldsymbol{a}_t \leftarrow \boldsymbol{m}^l_{t-1}, \boldsymbol{m}^l_t \leftarrow \boldsymbol{m}^l_{t-1}, \boldsymbol{m}^u_t \leftarrow \boldsymbol{m}^u_{t-1}$.
23:   end if
24: end while
```

provides a comprehensive measure of how well the confidence intervals of υ_k and μ_k are learned. Assigning too many UEs or IEs to arm k will result in a relatively larger $OrcInd_{k,t}$ compared to arms with more balanced assignments.

(3) **Favorable arms win UE first**(Line 12-18). The criterion $OrcInd_{k,t}$ also measures the regret generated by IEs on arm k. A larger $OrcInd_{k,t}$ value corresponds to greater

per-round regret caused by IEs on arm k. Since the regret generated by UEs can be upper-bounded by preventing consecutive UEs on the same arm, we can focus primarily on the regret generated by IEs. A larger $OrcInd_{k,t}$ value indicates that the capacity interval for arm k is not well learned, suggesting that UEs should be assigned to this arm to reduce the IE regret in subsequent rounds.

(4) **Stop learning when converges** (Line 11, and Line 21-23). At each time slot t, once the upper and lower bounds of an arm's capacity converge, no further exploration is required for that arm. The correctness of the estimated capacity is guaranteed by Lemma 4. Furthermore, this allows more flexible explorations on other arms, as UEs are no longer required on arms that have been sufficiently learned. Consequently, this accelerates the convergence of confidence intervals for all arms.

Regret upper bounds. The following theorems state the regret upper bounds of Algorithm 2.

Theorem 14 *The instance-dependent regret upper bound for Algorithm 2 is:*

$$\begin{aligned}&\mathbb{E}[Reg\left(T\right)]\\ \leq&\sum_{k=1}^{K}\left(2048c+\frac{4096\pi^2\left(\mu_k-c\right)}{3}\cdot K+\left(\frac{1024\pi^2}{3}+\frac{2048}{m_k}\right)\cdot\frac{N}{m_k}c\right)\frac{\sigma^2}{\mu_k^2}\log\left(T\right)\\ &+\sum_{k=1}^{K}2K\max\left(\mu_k m_k,Nc\right)\\ =&O\left(\sum_{k=1}^{K}\left(\frac{N}{m_k}c+K\left(\mu_k-c\right)\right)\frac{\sigma^2}{\mu_k^2}\log\left(T\right)\right).\end{aligned}$$

Remark 15 *The N in the numerator of the regret upper bound arises from the early UEs, when the $m_{k,t}^u$ values are not well learned and all plays are assigned to ensure a valid UE on that arm. The K in the regret upper bound mainly results from compulsory IEs when there are insufficient plays for all arms to freely choose UEs or IEs. Consider the scenario where the UCBs of the capacities are well-learned, such that $\sum_{k=1}^{K} m_{k,t}^u \leq N$. In this scenario, each arm can be explored with alternating IEs and UEs. This eliminates the need for compulsory IEs and removes the dependence of the regret upper bound on K. However, the primary challenge lies in limiting the number of such IEs before $\sum_{k=1}^{K} m_{k,t}^u \leq N$. An improved bound for the regret caused by these IEs could potentially be achieved by introducing factors like the magnitude of N/M, alongside a more detailed analysis.*

Theorem 16 *The instance-independent regret upper bound for Algorithm 2 is:*

$$\begin{aligned}&\mathbb{E}\left[Reg(T)\right]\\ \leq&\sigma\sqrt{\left(2048M+\frac{4096\pi^2}{3}KM+5500NK\right)K\left(T\log\left(T\right)\right)}+\sum_{k=1}^{K}2K\max\left(\mu_k m_k,Nc\right)\\ =&O\left(K\sigma\sqrt{NT\log(T)}\right).\end{aligned}$$

Remark 17 *The arm capacity M appears in this instance-independent regret upper bound, suggesting that regret is positively related to the capacity. However, there is another dominant term NK^2, since $N \geq M$. As explained in the previous section, N arises from poor-performing UEs in the early time slots, while K reflects the competition for resources among UEs. Therefore, a trade-off should exist between these two terms in the regret bound. With more plays, it takes fewer time slots for $\sum_{k=1}^{K} m_{k,t}^{u} \leq N$. Characterizing this relationship could potentially reduce N to a sublinear factor in the regret upper bound, which we leave for future investigation. Furthermore, our experimental results provide supporting evidence that such sublinear dependence is likely to hold in practice.*

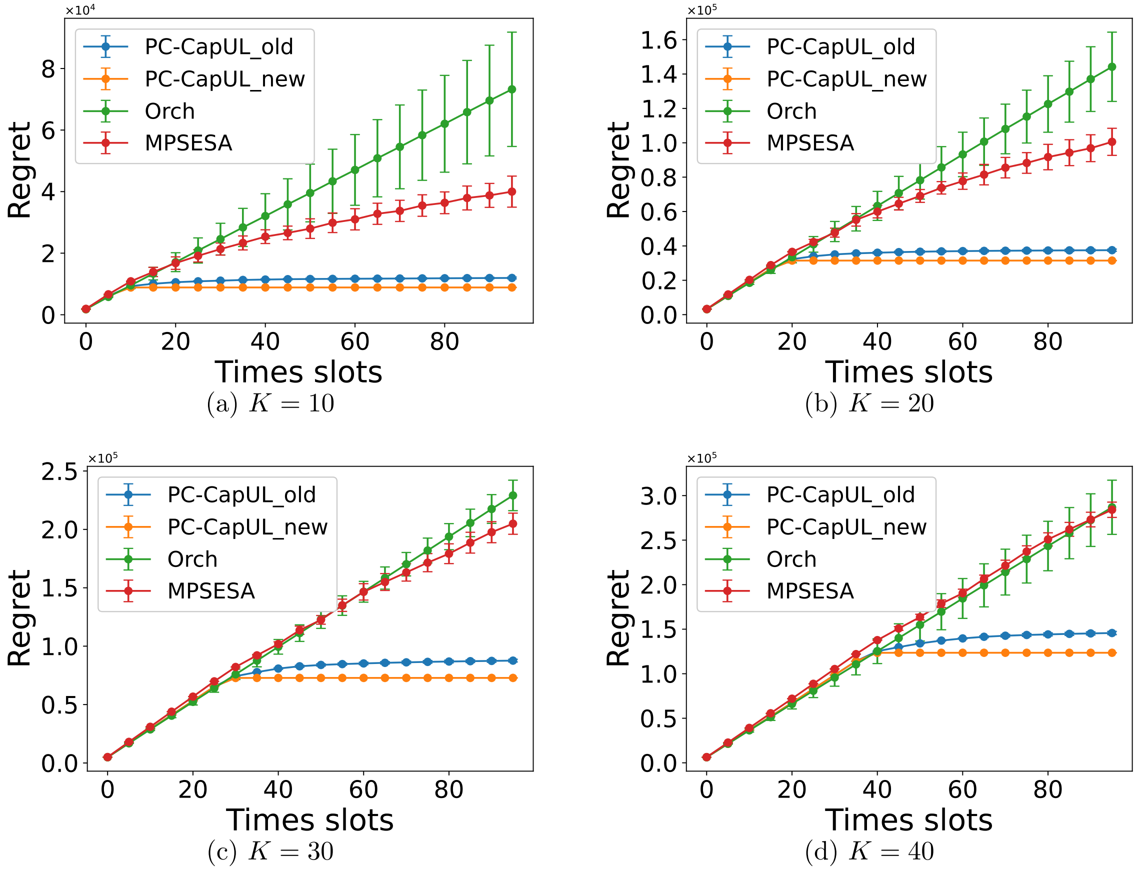
6. Experiments

6.1. Experiment Setting

This section states the experiment setting, including the number of plays, arms, comparison baselines and parameter settings, etc. The capacity of each arm setting: $m_k = 10 + [\ell \times \text{Rand}(0,1)]$, where $\ell = 5, 10, 15, 20$. Number of arms: $K = 10, 20, 30, 40$. Number of plays: $N = M, M + 0.1M, M + 0.2M, M + 0.4M$. Movement cost: $c = 0.2, 0.1, 0.01$, We consider the default parameters unless we mention to vary them explicitly $\ell = 10, K = 20, N = M + 0.1M, c = 0.1$. We conduct simulations to validate the performance of our algorithm and compare it to other algorithms adapted from MAB. We consider three baselines: MP-SE-SA, Orch proposed in Wang et al. (2022a), and a variant of our proposed algorithm `PC-CapUL-old`, which replaces the our arm capacity estimator with that of Wang et al. (2022a). Other details are shown in the the supplementary materials

6.2. Impact of Number of Arms

In Figures 1, we set K as $10, 20, 30, 40$, respectively. It is evident that as the number of arms increases, all algorithms require more exploration to identify the true capacities of each arm, as shown in both the lower and upper bound theorems. For all values of K, our algorithms outperform the two baseline algorithms, and the one with better estimators converges much faster. In our simulation of 2000 time slots, the regret of Orch in 1(a) converges to around 4.5×10^5 after 1700 time slots, which is much slower than ours. The difference in convergence speed can be attributed to two main factors. First, Orch has far fewer attempts for UEs in the same time slot due to its parsimonious and maladaptive strategy. The UEs are only allowed in even rounds in Orch. In contrast, `PC-CapUL-old` assigns UEs or IEs to arm k depending on how well the μ_k and m_k are learned. Second, our confidence intervals are more precise, and converge with fewer explorations. Additional experiments were conducted to verify this, with the results presented in the supplementary materials. It is noteworthy that across all parameter settings, the standard deviations of the regret for Orch and MP-SE-SA are significantly larger than that of `PC-CapUL`. The primary reason is that the former two algorithms assign UEs and IEs in a more random manner compared to `PC-CapUL`.



(a) $K = 10$

(b) $K = 20$

(c) $K = 30$

(d) $K = 40$

Figure 1: Impact of number of Arms. (a) $K = 10$, (b) $K = 20$, (c) $K = 30$, (d) $K = 40$.

7. Conclusion

This paper revisits the multi-play multi-armed bandit problem with shareable arm capacities in the capacity-scarce setting, a scenario absent from existing MP-MAB-SAC research. In this paper, we establish more complete theoretical results compared to those discussed in other settings in prior works. We close the sample complexity gap, and derive both instance-dependent and instance-independent lower bounds for this setting. We design an algorithm named `PC-CapUL`, in which we use prioritized coordination of arm capacities upper/lower confidence bound (UCB/LCB) to efficiently balance the exploration-exploitation trade-off. We prove both instance-dependent and instance-independent upper bounds for `PC-CapUL`, which match the lower bounds up to some acceptable model-dependent factors. Numerical experiments validate the data efficiency of `PC-CapUL`.

8. Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 62476261.

References

R Agrawal, M Hegde, D Teneketzis, et al. Multi-armed bandit problems with multiple plays and switching cost. *Stochastics and Stochastic reports*, 29(4):437–459, 1990.

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, 1987a.

Venkatachalam Anantharam, Pravin Varaiya, and Jean Walrand. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part ii: Markovian rewards. *IEEE Transactions on Automatic Control*, 32(11):977–982, 1987b.

Junpu Chen and Hong Xie. An online learning approach to sequential user-centric selection problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6231–6238, 2022.

Tackseung Jun. A survey on the bandit problem with switching costs. *de Economist*, 152 (4):513–541, 2004.

Junpei Komiyama, Junya Honda, and Hiroshi Nakagawa. Optimal regret analysis of Thompson sampling in stochastic multi-armed bandit problem with multiple plays. In *International Conference on Machine Learning*, pages 1152–1161. PMLR, 2015.

Junpei Komiyama, Junya Honda, and Akiko Takeda. Position-based multiple-play bandit problem with unknown position bias. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5005–5015, 2017.

Paul Lagrée, Claire Vernade, and Olivier Cappé. Multiple-play bandits in the position-based model. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 1605–1613, 2016.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Antoine Lesage-Landry and Joshua A Taylor. The multi-armed bandit with stochastic plays. *IEEE Transactions on Automatic Control*, 63(7):2280–2286, 2017.

Baolin Li, Yankai Jiang, Vijay Gadepally, and Devesh Tiwari. Llm inference serving: Survey of recent advances and opportunities. *arXiv preprint arXiv:2407.12391*, 2024.

Alex Luedtke, Emilie Kaufmann, and Antoine Chambaz. Asymptotically optimal algorithms for budgeted multiple play bandits. *Machine Learning*, 108:1919–1949, 2019.

Jinyu Mo and Hong Xie. A multi-player mab approach for distributed selection problems. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 243–254. Springer, 2023.

Vrettos Moulos. Finite-time analysis of round-robin kullback-leibler upper confidence bounds for optimal adaptive allocation with multiple plays and markovian rewards. *Advances in Neural Information Processing Systems*, 33:7863–7874, 2020.

Xuchuang Wang, Hong Xie, and John C. S. Lui. Multiple-play stochastic bandits with shareable finite-capacity arms. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 23181–23212. PMLR, 2022a. URL https://proceedings.mlr.press/v162/wang22af.html.

Xuchuang Wang, Hong Xie, and John C. S. Lui. Multi-player multi-armed bandits with finite shareable resources arms: Learning algorithms & applications. In Luc De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 3537–3543. ijcai.org, 2022b. doi: 10.24963/IJCAI.2022/491. URL https://doi.org/10.24963/ijcai.2022/491.

Xuchuang Wang, Hong Xie, and John C. S. Lui. Multi-player multi-armed bandits with finite shareable resources arms: Learning algorithms & applications, 2022c. URL https://arxiv.org/abs/2204.13502.

Yingce Xia, Tao Qin, Weidong Ma, Nenghai Yu, and Tie-Yan Liu. Budgeted multi-armed bandits with multiple plays. In *IJCAI*, pages 2210–2216, 2016.

Renzhe Xu, Haotian Wang, Xingxuan Zhang, Bo Li, and Peng Cui. Competing for shareable arms in multi-player multi-armed bandits. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Jianjun Yuan, Wei Lee Woon, and Ludovik Coba. Adversarial sleeping bandit problems with multiple plays: Algorithm and ranking application. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23, page 744–749, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702419. doi: 10.1145/3604915.3608824. URL https://doi.org/10.1145/3604915.3608824.

Datong Zhou and Claire Tomlin. Budget-constrained multi-armed bandits with multiple plays. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Banghua Zhu, Ying Sheng, Lianmin Zheng, Clark Barrett, Michael I Jordan, and Jiantao Jiao. On optimal caching and model multiplexing for large model inference. *arXiv preprint arXiv:2306.02003*, 2023.