Cooperative Stochastic Bandits with Asynchronous Agents and Constrained Feedback

Anonymous Author(s)
Affiliation
Address
email

Abstract

Motivated by the scenario of large-scale learning in distributed systems, this paper studies a scenario where M agents cooperate together to solve the same instance of a K-armed stochastic bandit problem. The agents have limited access to a local subset of arms and are asynchronous with different gaps between decision-making rounds. The goal is to find the global optimal arm and agents are able to pull any arm, however, they can only observe the reward when the selected arm is local. The challenge is a tradeoff for agents between pulling a local arm with observable feedback, or pulling external arms without feedback and relying on others' observations that occur at different rates. We propose `AAE-LCB`, a two-stage algorithm that prioritizes pulling local arms following an active arm elimination policy, and switches to other arms only if all local arms are dominated by some external arms. We analyze the regret of `AAE-LCB` and show it matches the regret lower bound up to a K factor.

1 Introduction

Multi-armed bandits (MABs) [11, 34] is a remarkably successful online learning framework that has been extensively studied since the 1950s [32]. It has a broad range of applications including datacenter optimization, web advertising, and recommender systems [34, 20, 26, 10]. In the basic MAB problem, a learner repeatedly chooses an action (pulls an arm) in each round, and collects (and observes) the reward associated with the selected arm, but not rewards associated with the unselected arms. The goal of the learner is to maximize the long-term reward collected, and the performance metric is regret, which is the difference between the expected reward for the best arm and that received by the learner.

The distributed/multi-agent/multiplayer MAB problem is an extension of the basic MAB that has been studied extensively recently in different settings [29, 35, 6, 23, 22, 17, 30, 36, 33]. This problem is motivated by several applications such as (1) large-scale learning systems [14] in domains such as finance or recommendation systems; (2) cooperative search by multiple robots [27, 21]; (3) the application of wireless cognitive radio [12, 29, 28, 5]; and (4) distributed learning in distributed systems (e.g., a set of IoT devices learning about an underlying environment) [31, 16, 4, 7, 37].

Most prior work on the multi-agent MAB problem assumes that agents have full access to all arms, and hence they solve the same MAB problem with the goal of minimizing the aggregate regret of the agents either in a *competition* setting [2, 9, 12, 36, 6, 8, 29, 28, 5] where agents receive degraded or no rewards when pulling the same arm, or in a *collaboration/cooperation* setting [30, 36, 25, 24, 23, 33], where agents receive independent rewards when they pull the same arm, and agents can communicate to improve their learning performance. We review the literature in the supplementary material.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

In this paper, we study a heterogeneous version of the cooperative multi-agent stochastic bandit problem with a set $\mathcal{A} = \{1, \ldots, M\}$ of agents and a set $\mathcal{K} = \{1, \ldots, K\}$ of arms. Agent $j \in \mathcal{A}$ has access to a subset $\mathcal{K}_j \subseteq \mathcal{K}$ of arms. We refer to arms in $\mathcal{K}_j$ as *local* and the remaining arms in $\mathcal{K} \setminus \mathcal{K}_j$ as *external* arms. Agents come with different learning capabilities and can pull an arm every $1/\theta_j$ rounds, with $0 < \theta_j \leq 1$ as the action rate of agent j. The goal is to collaboratively find the optimal arm in $\mathcal{K}$. An agent can pull any arm and receives a reward; however, it *observes* the reward only when it pulls a local arm. It instantaneously forwards this observation to all other agents. We call this setup *Feedback-constrained Cooperative Multi-agent MAB* (`FC-CMA2B`) and formally define it in Section 2.

The `FC-CMA2B` setting arises naturally in large-scale learning systems that are often geographically distributed for domains such as web search, finance, and recommendation systems. In such large-scale systems, it is usually impossible to run a single bandit algorithm. Instead, a distributed set of agents work to solve the problem, each with access to a subset of the action space. However, the entire learning task is integrated and the goal is to find the best global action, e.g., the most relevant search result in a web search application, or the best possible video recommendation in the YouTube recommendation system among millions of possible recommendations. In the above scenario, learning among different agents may be *asynchronous* in the sense that each agent has its own action (decision making) rate, which is captured by `FC-CMA2B`.

The algorithmic challenge in `FC-CMA2B` originates from a nontrivial tradeoff that an agent must make between pulling a local arm with an observable reward, and pulling an external arm with the expectation of a larger but not observable reward. One might simply run a cooperative `UCB` algorithm on each agent without any distinction between local and external arms and pull the arm with the highest confidence interval. Then, if the selected arm is local, the agent can broadcast its observation to others. However, this strategy suffers a poor regret. The reason is intuitive. Consider a case where a suboptimal arm is only observable by a slow agent, i.e., an agent with a very small action rate θ_j. In such a case, the upper confidence bound of the suboptimal arm will remain large due to the limited observations available only to the slow agent. Since the indexing policy of cooperative `UCB` suggests pulling the arm with the highest upper confidence bound, then, the suboptimal arm will be continuously pulled by everyone, resulting in a poor regret.

Contributions. This paper makes the following contributions.

Algorithm design using a two-stage learning strategy. To tackle the above challenge of `FC-CMA2B`, a proper bandit algorithm should be able to collect sufficient reward information, which is possible only by pulling local arms. To achieve this, our high-level idea is to distinguish between local and external arms and prioritize the local arms and only switch to external arms when they are clearly better. We implement this idea by proposing `AAE-LCB`, a two-stage learning strategy, which in the first stage selects local arms based on a carefully-designed *active arm elimination* (`AAE`) policy, and in the second stage selects external arms based on a *lower confidence bound* (`LCB`) policy.

More specifically, in the *first stage* of `AAE-LCB`, each agent pulls only from a dynamically-constructed candidate set of local arms. Then through an `AAE` process, a local arm is removed from the candidate set when its confidence interval falls below that of at least one other (local or external) arm. With this strategy, `AAE-LCB` makes a sufficient number of observations of local arms, and resolves issues encountered by the cooperative `UCB`. For an agent whose local arms do not include the optimal arm, the elimination process continues until all local arms are eliminated, and when so, the algorithm enters a *second stage*, where the agent only pulls external arms.

In the second stage, the agent totally relies on external observations that may occur at different rates. Differences in observation rates complicates decision making regarding choice of external arm such that the agent has to trade off between well observed arms, and those that are not well observed but likely to be optimal. To tackle this tradeoff, `AAE-LCB` pulls an external arm with the largest `LCB`, and the intuition is that an arm with greater `LCB` is both better observed and more likely to be optimal. Note that selecting based on largest `UCB` may lead to pulling external arms whose upper confidence bounds are large due to a lack of samples.

Regret lower and upper bounds. We analyze the regret of `AAE-LCB` and show that it achieves a regret of $O(\sum_{i:\Delta_i>0} \Theta_i/\Theta_{i^*} \times K \log T/\Delta_i)$, where Θ_i is the action rate of arm i defined as the sum of the action rates of agents containing arm i and i^* is the optimal arm. We further establish a regret lower bound for `FC-CMA2B`, that shows `AAE-LCB` is optimal up to a factor of K. For comparison, we

investigate the regret of a baseline algorithm, AAE-AAE, which pulls external arms based on another AAE process and show that it exhibits poor performance in the presence of agents with low action rates. Specifically, we show that AAE-AAE suffers a regret of $\Omega(\Theta/\Theta_{\min}\log T)$, where $\Theta = \sum_{j\in\mathcal{A}}\theta_j$ and $\Theta_{\min} = \min_{i\in\mathcal{K}}\sum_{j:i\in\mathcal{K}_j}\theta_j$ are the aggregate action rate of all agents, and the *minimum* aggregate action rate of agents containing any arm i, respectively. This result shows that regret can be arbitrarily bad when $\Theta_{\min}$, i.e., the minimum observation rate among all arms (either optimal or suboptimal), is small. Hence, instead of regret depending on the observation rate of the least observable (and possibly suboptimal) arms, the regret of AAE-LCB depends on that of the optimal arm.

Numerical results. Through brief numerical experiments, we verify our theoretical observations and show that the cooperative extension of UCB and AAE-AAE are either unable to gain sufficient observations for the global optimal arm or vulnerable to the action rate of slow agents with suboptimal arms, while AAE-LCB is robust to both issues and achieves much better performance.

2 Problem Setup

Feedback-constrained Cooperative Multi-agent MAB (FC-CMA2B). We consider a multi-agent stochastic bandit setting with a set $\mathcal{A} = \{1,\dots,M\}$ of independent agents existing over the entire time period, and a set $\mathcal{K} = \{1,\dots,K\}$ of arms. Associated with arms are mutually independent sequences of i.i.d. Bernoulli rewards with mean $0 \le \mu(i) \le 1$, for arm $i \in \mathcal{K}$. Agent $j \in \mathcal{A}$ has full access to a subset $\mathcal{K}_j \subseteq \mathcal{K}, K_j = |\mathcal{K}_j|$, of arms. We refer to $\mathcal{K}_j$ and $\mathcal{K}\setminus\mathcal{K}_j$ as the sets of local and external arms of agent j, respectively. Agents are allowed to pull and receive a reward from any arm from $\mathcal{K}$, however, they only receive observations from local arms. That is to say, if the selected arm is external, the agent does not observe the reward even though the reward is allocated. Last, we make no assumption regarding overlaps among sets $\mathcal{K}_j$.

In addition to heterogeneity in access to arms, agents are also heterogeneous in their decision making capabilities. Specifically, considering the decision rounds $\{1,\dots,T\}$, agent j is able to pull an arm every $\omega_j \in \mathbb{N}^+$ rounds, i.e., decision rounds for agent j are $t = \omega_j, 2\omega_j, \dots, N_j\omega_j$, where $N_j = \lfloor T/\omega_j \rfloor$. Parameter ω_j represents the *inter-round gap* of agent j. For simplicity of analysis, we define $\theta_j := 1/\omega_j$ as the *action rate* of agent j. Intuitively, the larger θ_j, the faster the agent j is in pulling arms.

The goal of each agent is to learn the global optimal arm. The regret for agent j in FC-CMA2B is

$$R_T^j := \mu(i^*)N_j - \sum\nolimits_{t\in\{k\omega_j : k=1,\dots,N_j\}} x_t(I_t^j),$$

where i^* is the optimal arm in $\mathcal{K}$, and $I_t^j \in \mathcal{K}$ is the pulled arm by agent j at time t. Agent j receives the reward $x_t(I_t^j)$ whose value is observable only if $I_t^j \in \mathcal{K}_j$. The ultimate goal is to minimize the aggregate regret of the agents, i.e., $R_T = \sum_{j\in\mathcal{A}} R_T^j$.

We note that when two agents select the same arm, they collect stochastically independent rewards, and there is no reward degradation such as occurs in competition settings. We also consider full and truthful cooperation among agents, i.e., after committing to one decision and receiving a reward from the pulled arm, an agent broadcasts observations to others at no cost and no delay. In the supplementary material, we further consider a more general model with communication delays among agents.

Additional notations and terminologies. To facilitate our algorithm design and analysis, we introduce the following notations. Let $\mathcal{A}_i$ denote the set of agents whose local arm sets include arm i,

$$\mathcal{A}_i := \{j \in \mathcal{A} : i \in \mathcal{K}_j\}, \quad \forall i \in \mathcal{K}.$$

By $\Delta(i', i)$, we denote the difference in mean rewards of arms i' and i, i.e., $\Delta(i', i) := \mu(i') - \mu(i)$. When $i' = i^*$, we rewrite $\Delta(i^*, i)$ as Δ_i, which is known as the suboptimality gap in the basic bandit problem. Last, we define Θ and Θ_i, $i \in \mathcal{K}$, as follows

$$\Theta := \sum_{j\in\mathcal{A}}\theta_j, \quad \text{and} \quad \Theta_i := \sum_{j\in\mathcal{A}_i}\theta_j,$$

where Θ_i is the action rate of arm i across all agents in $\mathcal{A}_i$. Intuitively, the larger Θ_i is, the higher the rate at which arm i can be pulled by agents in $\mathcal{A}_i$. The action rate Θ_i plays a key role in characterizing the regret bounds.

Algorithm 1 `AAE-LCB`: A Cooperative Bandit Algorithm for Agent j in the `FC-CMA2B` setting

1: **Initialization:** $\hat{n}^j(i) = 0$, $\hat{\mu}(i)$, $i \in \mathcal{K}$, $\mathcal{C}_j = \mathcal{K}_j$, $\alpha > 2$, δ_t.
2: **for** each decision round t **do**
3: **for** each new received $x_\tau(i)$, $i \in \mathcal{K}$, $\tau < t$, **do**
4: Execute line (8)-(10) for arm i
5: **end for**
6: **if** $\mathcal{C}_j \neq \emptyset$ **then** ▷ *Stage 1: Pulling local arms through an* `AAE` *process*
7: Pull arm I_t^j with the least no. of observations from $\mathcal{C}_j$
8: Increase counter for the selected arm by 1: $\hat{n}^j(I_t) \leftarrow \hat{n}^j(I_t) + 1$
9: Update estimate of the mean value of the selected arm $\hat{\mu}(I_t)$
10: Reconstruct candidate set based on updated values of $\hat{n}^j(I_t)$ and $\hat{\mu}(I_t^j)$ as in Equation (2)
11: Broadcast $x_t(I_t^j)$ to other agents
12: **else** ▷ *Stage 2: Pulling external arms through a* `LCB` *process*
13: Select an external arm with the largest lower confidence bound as in Equation (3)
14: **end if**
15: **end for**

3 Algorithm Design

Each agent in the `FC-CMA2B` setting has to resolve a tradeoff between two categories of arms: (i) local arms with observable rewards, and (ii) external arms whose rewards are not observable but may be better than local arms. This heterogeneity in information feedback motivates our two-stage learning algorithm, `AAE-LCB`. In the first stage, agents running `AAE-LCB` are focused on sampling local arms to collect sufficient information that is useful for all agents in finding the global optimal arm. And when with high probability, an agent is confident that its local set does not contain the optimal arm, it moves to the second stage by pulling external arms based on others' observations.

3.1 Stage 1: Selecting a local arm

In order to converge to the global optimal arm, agents must collect sufficient observations in the first stage. Toward this, our idea is to prioritize pulling local arms as much as possible and only switch to external arms when the agent finds an external arm that is *sufficiently better* than all its local arms with high confidence. To implement this idea, we extend the active arm elimination (`AAE`) algorithm [18, 3] to capture the additional issues in `FC-CMA2B`. Similar to the basic `AAE` algorithm, in our algorithms, we start off by constructing a dynamic candidate set of local arms and gradually eliminating arms from it. The elimination process is similar to the basic `AAE`, which eliminates an arm when its confidence interval falls below that of at least one other arm. However, the difference is that while the candidate set includes local arms, the criteria for eliminating an arm is based on a comparison between both local and external arms.

Specifically, we define the width of the confidence interval for arm i and agent j at time t as

$$\texttt{cint}(i, j, t) := \sqrt{\frac{\alpha \log \delta_t^{-1}}{2\hat{n}_t^j(i)}}, \tag{1}$$

where $\hat{n}_t^j(i)$ is the total number of observations (either local or received from other agents) of arm i available to agent j by time t (observations made in time slots from 1 to $t-1$). Also, $\delta_t > 0$ and $\alpha > 2$ are parameters of the confidence interval. The removal of arm i from agent j's candidate set at time t with empirical mean of $\hat{\mu}(i, \hat{n}_t^j(i))$ occurs if there is another arm i', such that the difference in empirical means of i and i' is larger than the widths of the confidence intervals, i.e., $\hat{\mu}(i', \hat{n}_t^j(i')) - \hat{\mu}(i, \hat{n}_t^j(i)) > \texttt{cint}(i', j, t) + \texttt{cint}(i, j, t)$. This means that with high probability, arm i is not optimal. Note that i' could be either a local or an external arm. More formally, agent j constructs the following dynamic candidate set $\mathcal{C}_{j,t}$ at time t during its learning process.

$$\mathcal{C}_{j,t} = \left\{ i \in \mathcal{K}_j : \hat{n}_t^j(i) = 0 \ \texttt{ or } \ \hat{\mu}(i, \hat{n}_t^j(i)) - \hat{\mu}(i', \hat{n}_t^j(i')) \leq \texttt{cint}(i, j, t) + \texttt{cint}(i', j, t), \forall i' \in \mathcal{K} \right\}. \tag{2}$$

Note that in the above construction the local candidate set of agent j includes all local arms not yet observed and all others not dominated by any other local or external arm. If the local candidate set

is not empty, an agent always prioritize selecting local arms and, following basic AAE, pulls a local arm in the candidate set with the least number of observations, and then updates the empirical mean values and broadcasts the observation to others as in Lines 8-10 of Algorithm 1.

3.2 Stage 2: Selecting an external arm

As Stage 1 of the learning process moves forward, it is possible that the local candidate set of an agent will become empty, i.e., an external arm eventually dominates all local arms. In this situation, the question becomes how to pick an external arm. The answer to this question is critical in designing cooperative algorithms with low regret.

Different action rates among different agents can lead arms to suffer different observation limitations that can invalidate the selection indices commonly used in the bandit algorithms. Specifically, when a suboptimal arm is only accessible to slow agents, estimates of the empirical mean reward of this arm will be updated more slowly than those of other arms pulled by faster agents, and thus contain more errors. In this way, those algorithms, which are unaware of confidence of estimates, may be misled into selecting a suboptimal arm with low-level confidence in the FC-CMA2B setting. For example, "slow arms", i.e., those arms that are only in the local sets of slow agents, have much looser confidence interval and a much larger upper confidence bound than others. As a result, fast agents running a cooperative version of AAE or UCB-based algorithms that pull external arms based on estimated upper confidence bounds may be misled into continuously selecting a suboptimal "slow" arm because its upper confidence bound is large due to insufficient number of observations. To address this, we propose to follow a Lower Confidence Bound (LCB) policy, which selects the external arm with the largest lower confidence bound, i.e.,

$$I_t^j = \arg\max_{i \in \mathcal{K}/\mathcal{K}_j} \hat{\mu}(i, \hat{n}_t^j(i)) - \texttt{cint}(i, j, t). \tag{3}$$

An important observation on LCB is that the lower confidence bound of an arm is large only if it is well observed and hence is likely to be the optimal arm. Indeed this is not the case if the selection policy is based on the largest upper confidence bound, since external arms with low observation might have large upper confidence bound.

3.3 Baseline Algorithms: CO-UCB and AAE-AAE

To show the advantages of AAE-LCB, we introduce two baseline algorithms. The first one is CO-UCB, a cooperative version of UCB that the agents do not distinguish between local and external arms. In other words, each agent running CO-UCB, pulls the arm with the largest upper confidence bound and, if the information is observable, broadcasts it to all other agents. Further, each agent will updates its confidence intervals not only based on its local observations, but also, based on the information received from other agents.

The second algorithm is AAE-AAE, which has a similar two-stage structure to prioritize pulling local arms as AAE-LCB, but adapts another layer of active arm elimination for pulling external arms. In the second stage, i.e., when the local set is emptied, AAE-AAE randomly pulls an external arm whose confidence interval have overlaps with others' and dynamically reconstructs the external candidate set similar to the process in the first stage, but for external arms. Because external arms with few observations has loose confidence interval, they satisfy the criteria to be in the candidate set, hence, the agents running AAE-AAE may repeatedly pull them.

In Section 5, we numerically compare the performance of both baseline algorithms with AAE-LCB and show the poor performance of CO-UCB due to the same treatment for local and external arms, and the improvement of AAE-LCB over AAE-AAE due to better treatment of external arms. In addition, in the next section, we theoretically compare the regret of AAE-LCB and AAE-AAE.

4 Regret Results

Two main regret results are highlighted in the following two theorems. We first introduce some terminology to facilitate the presentation of the results. Consider an algorithm π running on agents in FC-CMA2B. We say that π is *consistent*, if its regret satisfies $\mathbb{E}\left[R_T(\pi)\right] = O((T\Theta)^\sigma)$ as $T \to +\infty$ for any $\sigma > 0$, and for any set of Bernoulli reward distributions. Further, let $\mathrm{KL}(\mu_i, \mu_i + \Delta_i)$ refer to the the Kullback-Leibler divergence between a Bernoulli of parameter μ_i and $\mu_i + \Delta_i$.

Theorem 1 (Asymptotic Regret Lower Bound for `FC-CMA2B`**)** *For any consistent algorithm π, its expected regret satisfies*

$$\liminf_{T\to+\infty,\Theta/\Theta_{i^*}\to+\infty} \frac{\mathbb{E}\left[R_T(\pi)\right]}{\log(T\Theta)} = \Omega\Big(\frac{\Theta}{\Theta_{i^*}} \sum_{i:\Delta_i>0} \frac{\Delta_i}{\mathrm{KL}(\mu_i, \mu_i+\Delta_i)}\Big).$$

The proof for the lower bound consists of two steps. In the first one, we prove that under any given agent action rates, any algorithm suffers a regret of $\Omega(\log(T\Theta))$ for large T. This proof uses techniques commonly used to prove lower bounds for stochastic bandits. In the second step, we prove that, with large enough T and Θ/Θ_{i^*}, no algorithm can have a sublinear dependency on Θ/Θ_{i^*}. This is proved by contradiction. The details are given in the supplementary material.

Theorem 2 (Expected Regret for `AAE-LCB`**)** *The expected regret of* `AAE-LCB` *has the following upper bound*

$$\mathbb{E}\left[R_T\right] \leq \sum_{i:\Delta_i>0} \max\left\{\frac{4\alpha\log\delta^{-1}}{\Delta_i}\left(2+\frac{K\Theta_i}{\Theta_{i^*}}\right), \frac{12\Theta_i\alpha K\log\delta^{-1}}{\Delta_i\Theta_{i^*}}\right\}$$
$$+2\left(1+\frac{\Theta_i}{\Theta_{i^*}}\right)\sum_{j\in\mathcal{A}}\sum_{l=1}^{N_j}\sum_{i\in\mathcal{K}_j}\frac{l\Theta_i}{\theta_j}\delta^{\alpha}_{l/\theta_j}.$$

Corollary 1 *With $\delta_t = 1/t$ and $\alpha > 2$, we have the following regret for the* `AAE-LCB` *algorithm.*

$$\mathbb{E}\left[R_T\right] = O\left(\sum_{i\in\mathcal{K}}\frac{K\Theta_i\log T}{\Theta_{i^*}\Delta_i}\right).$$

The prove is given in Section 4.1. From Corollary 1 and Theorem 1, we observe that the proposed `AAE-LCB` algorithm matches the lower bound up to a factor K.

We also compare the regret of `AAE-LCB` to that of `AAE-AAE`. With $\delta_t = 1/T$, and $\alpha > 4$, the regret of `AAE-AAE` could be characterized as

$$\mathbb{E}\left[R_T\right] = \Omega\left(\frac{\Theta}{\Theta_{\min}}\log T\right), \tag{4}$$

where $\Theta_{\min} := \min_i \Theta_i$. The detailed derivation of the the above regret is in the supplementary materials. The regret of `AAE-AAE` in (4) strongly depends on $\Theta_{\min}$, i.e., the smallest aggregate action rate among all arms. Hence, as $\Theta_{\min}$ goes to zero, the regret of `AAE-AAE` significantly degrades. The regret of `AAE-LCB` shown in Corollary 1, however, depends only on the action rate of the optimal arm. Hence, `AAE-LCB` outperforms `AAE-AAE`, in the sense that its performance is independent of the slowest arm.

4.1 A Proof for Theorem 2

There are two contributions to the regret of `AAE-LCB`: one due to pulling suboptimal local arms, and the other one due to pulling suboptimal external arms. To prove Theorem 2, we analyze these two contributions. Note that, observations are generated whenever agents pull local arms. Thus, the first contribution to regret can be upper bounded by analyzing the relationship between the confidence interval and the number of observations obtained from each arm. By analyzing the algorithm rules of `AAE-AAE`, we upper bound the expected number of observations required from suboptimal arms to identify the optimal arm if it lies in the local set, as well as the first contribution to regret. Regarding the second contribution to regret, the main difficulty comes form the heterogeneity of action rates associated with the arms. To upper bound this contribution to regret, we first upper bound the time period needed for the agents containing the optimal arm in their local sets generate enough observations such that the lower confidence bound of the optimal arm is higher than those of any others. With this, we can prove the the final result by upper bounding the number of times that suboptimal external arms need to be selected in the above proved time period.

Now, we proceed to formally prove the regret result. First, we categorize decisions made by the agents into Type-I and Type-II decisions. Type-I corresponds to the decisions of an agent when the

actual mean values of local arms lie in the confidence intervals calculated by the agent, otherwise, Type-II decision happens, i.e., the actual mean value of some local arm is not within the calculated confidence interval. Specifically, when agent j makes a Type-I decision at time t, we have

$$\mu(i) \in \left[\hat{\mu}\left(i, \hat{n}_t^j(i)\right) - \texttt{cint}(i,j,t), \hat{\mu}\left(i, \hat{n}_t^j(i)\right) + \texttt{cint}(i,j,t)\right], \quad i \in \mathcal{K}_j.$$

With Type-I decisions, an agent can keep the local optimal arm in its candidate set and eventually converges its decisions to the local optimal arm. The following Lemma, whose proof is given in the supplementary material, provides the probability that a Type-I decision happens at a particular decision round.

Lemma 1 *Agent j running* `AAE-AAE` *makes a Type-I decision, in round $t = l/\theta_j$, with a probability of at least* $1 - 2\sum_{i\in\mathcal{K}_j} \frac{l\Theta_i}{\theta_j}\delta^{\alpha}_{l/\theta_j}$.

In the `AAE-LCB` algorithm, a suboptimal arm pulled by agent j lies either in the candidate set or is the one with the largest lower confidence bound among all arms. Thus, we split the regret analysis into two cases: *Case I: local arm selection*, where a suboptimal decision is made during the arm elimination period for arms lying in the local set; and *Case II: external arm selection*, where a suboptimal decision is made based on the lower confidence bound when selecting external arms. In the following, we analyze the regret in each case separately.

Case I: regret due to local arm selection: For a suboptimal arm i, we upper bound the total number of pulling times by agents in $\mathcal{A}_i$, which is actually equal to $\hat{n}_t^j(i)$. First, we focus on the cases that the algorithm makes a Type-I decision at time t, i.e., the mean value of any arm lies in its confidence interval calculated by agent j. Then, at time t, if agent j in $\mathcal{A}_i$ selects arm i, we have

$$2\texttt{cint}(i,j,t) + 2\texttt{cint}(i^*,j,t) \geq \Delta_i. \tag{5}$$

Otherwise, we have

$$\begin{aligned}\hat{\mu}(i, \hat{n}_t^j(i)) + \texttt{cint}(i,j,t) \leq & \mu_i + 2\texttt{cint}(i,j,t) < \mu_i + \Delta_i - \texttt{cint}(i^*,j,t)\\ = & \mu_{i^*} - 2\texttt{cint}(i^*,j,t) \leq \hat{\mu}(i^*, \hat{n}_t^j(i^*)) - \texttt{cint}(i^*,j,t),\end{aligned}$$

implying that arm i is strictly dominated by i^*, hence, i can not be selected by agent j, contradicting the assumption that i is selected by j. It follows from Equation (5) that

$$\max\left\{2\texttt{cint}(i,j,t), 2\texttt{cint}(i^*,j,t)\right\} \geq \frac{\Delta_i}{2},$$

and by the definition of $\texttt{cint}(.)$ in Equation (1), we have

$$\min\left\{\hat{n}_t^j(i), \hat{n}_t^j(i^*)\right\} \leq \frac{8\alpha\log\delta_t^{-1}}{\Delta_i^2}. \tag{6}$$

Now, we focus on Type-II decisions. Let Q denotes the number of Type-II decisions. By Lemma 1, we have

$$\mathbb{E}\left[Q\right] \leq 2\sum_{j\in\mathcal{A}}\sum_{l=1}^{N_j}\sum_{i\in\mathcal{K}_j}\frac{l\Theta_i}{\theta_j}\delta^{\alpha}_{l/\theta_j}. \tag{7}$$

By combining Equations (6) and (7), we have

$$\min\left\{\mathbb{E}\left[\hat{n}_T^j(i)\right], \mathbb{E}\left[\hat{n}_T^j(i^*)\right]\right\} \leq \frac{8\alpha\log\delta^{-1}}{\Delta_i^2} + \mathbb{E}\left[Q\right]. \tag{8}$$

Then, we have

$$\mathbb{E}\left[\hat{n}_T^j(i^*)\right] \geq \frac{T\Theta_{i^*} - \mathbb{E}\left[Q\right]}{K} \geq \frac{\Theta_{i^*}n_T(i)}{\Theta_i K} - \frac{\mathbb{E}\left[Q\right]}{K}, \tag{9}$$

where the first inequality is based on the fact that the expected number of decision rounds with the optimal arm in the candidate set is at least $T\Theta_{i^*} - \mathbb{E}\left[Q\right]$, and the second one is based on the fact that $T \geq \hat{n}_T(i)/\Theta_i$. Combining the results in Equations (8) and (9), we get

$$\mathbb{E}\left[\hat{n}_T(i)\right] \leq \max\left\{\frac{8\alpha\log\delta^{-1}}{\Delta_i^2}, \frac{8\alpha K\Theta_i\log\delta^{-1}}{\Theta_{i^*}\Delta_i^2}\right\} + \left(1 + \frac{\Theta_i}{\Theta_{i^*}}\right)\mathbb{E}\left[Q\right]. \tag{10}$$

Case II: regret due to external arm selection: Now, we aim at upper bounding the expected number of selection times for a suboptimal arm i by the agents outside set $\mathcal{A}_i$. Again, we assume that agent j makes a Type-I decision at time slot t. Consider the case that $I_t^j = i$ and i is not within $\mathcal{K}_j$. By algorithm rules, we have that arm i has the largest lower confidence bound. We claim that

$$2\sqrt{2}\texttt{cint}(i^*, j, t) \geq \Delta_i. \tag{11}$$

Otherwise, we have

$$\begin{aligned}\hat{\mu}(i, \hat{n}_t^j(i)) - \texttt{cint}(i, j, t) \leq & \mu(i) = \mu(i^*) - \Delta_i \\ < & \hat{\mu}(i^*, \hat{n}_t^j(i^*)) + \texttt{cint}(i^*, j, t) - 2\texttt{cint}(i^*, j, t) \\ = & \hat{\mu}(i^*, \hat{n}_t^j(i^*)) - \texttt{cint}(i^*, j, t),\end{aligned}$$

contradicting the rules of the algorithm. Thus, at time t, the selected arm i satisfies

$$\Delta_i \leq 2\sqrt{\frac{\alpha K \log \delta_t^{-1}}{t\Theta_{i^*} - Q}}.$$

The above equation is obtained by replacing $\hat{n}_t^j(i^*)$ in equation $2\sqrt{2}\texttt{cint}(i^*, j, t) \geq \Delta_i$ with $[t\Theta_{i^*} - Q]/K$, since $\hat{n}_t^j(i^*) \geq [t\Theta_{i^*} - Q]/K$.

For any agent j, the largest time slot when the agent makes a Type-I decision and a suboptimal arm i lies in the candidate set is $4\frac{\alpha K \log \delta^{-1}}{\Delta_i^2 \Theta_{i^*}} + \frac{Q}{\Theta_{i^*}}$. Then, the regret spent on the arm i in other agents is upper bounded by $4\frac{\alpha K \log \delta^{-1}}{\Delta_i}\frac{\Theta_i}{\Theta_{i^*}} + \frac{Q_2 \Theta_i}{\Theta_{i^*}} + Q$.

Summing up the above two pieces of regret and the expected number of Type-II decisions yields

$$\begin{aligned}\mathbb{E}\left[R_T\right] \leq & \sum_{i:\Delta_i > 0} \max\left\{\frac{4\alpha \log \delta^{-1}}{\Delta_i}\left(2 + \frac{K\Theta_i}{\Theta_{i^*}}\right), \frac{12\Theta_i \alpha K \log \delta^{-1}}{\Delta_i \Theta_{i^*}}\right\} \\ & + 2\left(1 + \frac{\Theta_i}{\Theta_{i^*}}\right) \sum_{j \in \mathcal{A}} \sum_{l=1}^{N_j} \sum_{i \in \mathcal{K}_j} \frac{l\Theta_i}{\theta_j} \delta_{l/\theta_j}^{\alpha}.\end{aligned}$$

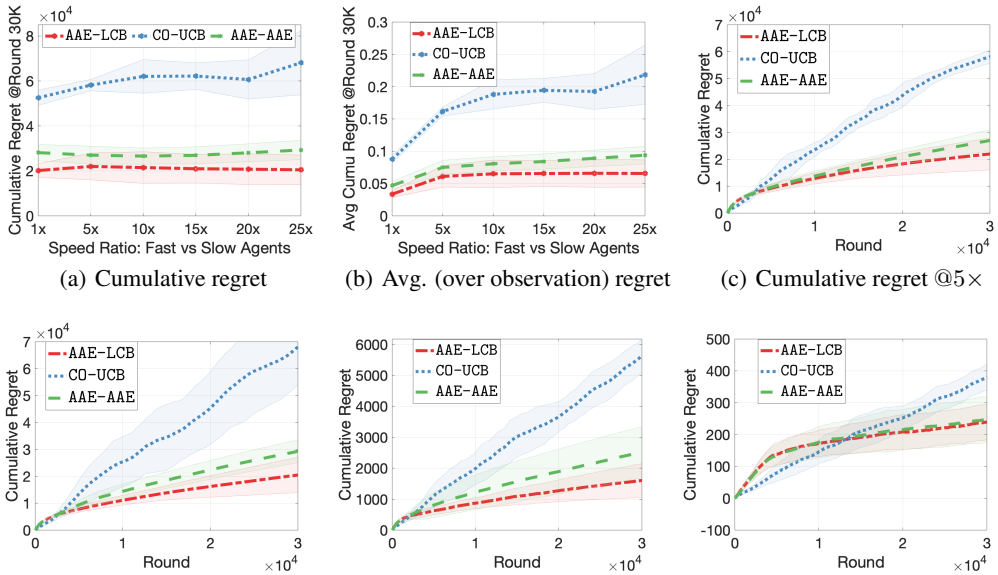
This completes the proof.

5 Numerical Experiments

Our goal in this section is to numerically investigate the performance of `AAE-LCB` and compare it to that of `AAE-AAE` and `CO-UCB` (see Section 3.3), and show that `AAE-LCB` effectively resolves the challenge slow agents present, while neither `AAE-AAE` nor `CO-UCB` do so. More specifically, by comparing `AAE-LCB` with `CO-UCB`, our goal is to verify the importance of two-stage learning in the design of `AAE-LCB`, and by comparing `AAE-LCB` and `AAE-AAE`, our goal is to verify the importance of using `LCB` as the indexing policy for external arm selection.

Experimental setup. We assume there are $K = 100$ arms with Bernoulli rewards with average rewards uniformly randomly taken from *Ad-Clicks* [1]. In our experiments, we report the cumulative regret after 30,000 rounds, which corresponds to the number of decision rounds of the fastest agent. All reported values are averaged over 20 independent trials. We have 20 agents, each with 12 arms selected from among a set of $K = 100$ arms into two categories of 10 "fast" agents each with action rate of 1, and 10 "slow" agents with varying action rates of less than 1. We compare the performances of `AAE-LCB` and `AAE-AAE` and `CO-UCB` that are introduced in Section 3.3.

Experimental results. In Figures 1(a) and 1(b), we fix the action rate of the fast agents and vary the ratio between action rate of fast and slow agents from $5\times$ to $25\times$ with steps of 5. The results show relatively sound performance of `AAE-LCB` as the speed ratio between fast and slow agents increases. However, the performance of `CO-UCB` decreases substantially as shown in Figure 1(b) for average per-agent regret, which is mainly due to treating local and external arms in the same way.



(a) Cumulative regret (b) Avg. (over observation) regret (c) Cumulative regret @5×

(d) Cumulative regret @25× (e) Per-agent regret (fast agent) (f) Per-agent regret (slow agent)

Figure 1: Regret of AAE-LCB vs. AAE-AAE vs. CO-UCB with two groups of "fast" and "slow" agents and varying action rate ratio between them. Notable observations: AAE-LCB outperforms CO-UCB significantly and outperforms AAE-AAE slightly. Comparing (c) and (d) shows that improvement of AAE-LCB over CO-UCB increases as the difference between action rates increases. Comparing (e) and (f) shows substantial regret degradation of CO-UCB in fast agent due to existence of slow agents.

This further verifies our intuitions for the need to distinguish between local and external arms as the main motivation for developing AAE-LCB. AAE-AAE and AAE-LCB exhibit similar performance showing the importance of prioritizing the local arms, however, AAE-LCB performs slightly better since it uses a better external arm selection policy than that of AAE-AAE. The evolution of cumulative regrets over time in Figures 1(c) and 1(d) and per-agent regret for fast and slow agents in Figures 1(e) and 1(f) also demonstrate that AAE-LCB outperform the alternatives.

6 Conclusion and Future Directions

In this paper, we proposed bandit algorithms with near-optimal regret for a cooperative stochastic bandit problem among multiple agents playing the same instance of the problem, each with limited access to the set of arms and with different decision-making capabilities.

A limitation of this work is that in the current result, there is a gap of factor K between the regret of AAE-LCB and the regret lower bound. We leave developing an algorithm with the optimal regret for the FC-CMA2B setting as an open problem. Additionally, a critical, yet practically relevant question is how to extend the algorithms to the case with communication delay and cost between agents. We address the communication delay in the supplementary material of the paper. Regarding communication cost, the question is how to design algorithms that can provide a tradeoff between the regret and communication costs. While this tradeoff has been studied extensively in recent works [13, 15, 19, 35], none of them consider this tradeoff in the presence of heterogeneous agents. This task is challenging since in each decision-making round, each agent should make a nontrivial decision on whether or not broadcast with other agents or a subset of agents. Another question arises by considering topological constraints for agents. In this work, we focus our analysis to a set of fully-connected agents. In practice, however, geographically distributed agents might have limited access to other agents over an underlying graph. This setting also has been studied in recent works such as [23, 33]; however, there is no work on the heterogeneous version of this setting. Last, we do not see any negative societal impacts of our work.

References

[1] Kaggle avito context ad clicks 2015. https://www.kaggle.com/c/avito-context-ad-clicks.

[2] A. Anandkumar, N. Michael, A. K. Tang, and A. Swami. Distributed algorithms for learning and cognitive medium access with logarithmic regret. *IEEE Journal on Selected Areas in Communications*, 29(4):731–745, 2011.

[3] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[4] O. Avner and S. Mannor. Multi-user lax communications: a multi-armed bandit approach. In *Proc. of IEEE INFOCOM*, pages 1–9, 2016.

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

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

[7] R. Bonnefoi, L. Besson, C. Moy, E. Kaufmann, and J. Palicot. Multi-armed bandit learning in iot networks: Learning helps even in non-stationary settings. In *International Conference on Cognitive Radio Oriented Wireless Networks*, pages 173–185. Springer, 2017.

[8] E. Boursier, E. Kaufmann, A. Mehrabian, and V. Perchet. A practical algorithm for multiplayer bandits when arm means vary among players. In *AISTATS 2020*, 2020.

[9] E. Boursier and V. Perchet. SIC-MMAB: synchronisation involves communication in multi-player multi-armed bandits. In *Advances in Neural Information Processing Systems*, pages 12071–12080, 2019.

[10] G. Bresler, D. Shah, and L. F. Voloch. Collaborative filtering with low regret. In *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*, pages 207–220, 2016.

[11] S. Bubeck, N. Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[12] S. Bubeck, Y. Li, Y. Peres, and M. Sellke. Non-stochastic multi-player multi-armed bandits: Optimal rate with collision information, sublinear without. In *Conference on Learning Theory*, pages 961–987, 2020.

[13] S. Buccapatnam, J. Tan, and L. Zhang. Information sharing in distributed stochastic bandits. In *Proc. of IEEE INFOCOM*, pages 2605–2613, 2015.

[14] N. Cesa-Bianchi, T. Cesari, and C. Monteleoni. Cooperative online learning: Keeping your neighbors updated. In *Algorithmic Learning Theory*, pages 234–250. PMLR, 2020.

[15] M. Chakraborty, K. Y. P. Chua, S. Das, and B. Juba. Coordinated versus decentralized exploration in multi-agent multi-armed bandits. In *IJCAI*, pages 164–170, 2017.

[16] S. J. Darak and M. K. Hanawal. Multi-player multi-armed bandits for stable allocation in heterogeneous ad-hoc networks. *IEEE Journal on Selected Areas in Communications*, 37(10):2350–2363, 2019.

[17] A. Dubey and A. Pentland. Cooperative multi-agent bandits with heavy tails. In *Proc. of ICML*, 2020.

[18] E. Even-Dar, S. Mannor, and Y. Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(Jun):1079–1105, 2006.

[19] E. Hillel, Z. S. Karnin, T. Koren, R. Lempel, and O. Somekh. Distributed exploration in multi-armed bandits. In *Advances in Neural Information Processing Systems*, pages 854–862, 2013.

[20] J. Jiang, R. Das, G. Ananthanarayanan, P. A. Chou, V. Padmanabhan, V. Sekar, E. Dominique, M. Goliszewski, D. Kukoleca, R. Vafin, et al. Via: Improving internet telephony call quality using predictive relay selection. In *Proc. of ACM SIGCOMM*, pages 286–299, 2016.

[21] L. Jin, S. Li, L. Xiao, R. Lu, and B. Liao. Cooperative motion generation in a distributed network of redundant robot manipulators with noises. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(10):1715–1724, 2017.

[22] D. Kalathil, N. Nayyar, and R. Jain. Decentralized learning for multiplayer multiarmed bandits. *IEEE Transactions on Information Theory*, 60(4):2331–2345, 2014.

[23] R. K. Kolla, K. Jagannathan, and A. Gopalan. Collaborative learning of stochastic bandits over a social network. *IEEE/ACM Transactions on Networking*, 26(4):1782–1795, 2018.

[24] P. Landgren, V. Srivastava, and N. E. Leonard. Distributed cooperative decision-making in multiarmed bandits: Frequentist and bayesian algorithms. In *Proc. of IEEE CDC*, pages 167–172, 2016.

[25] P. Landgren, V. Srivastava, and N. E. Leonard. Social imitation in cooperative multiarmed bandits: partition-based algorithms with strictly local information. In *Proc. of IEEE CDC*, pages 5239–5244, 2018.

[26] J. Langford and T. Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pages 817–824. Citeseer, 2007.

[27] S. Li, R. Kong, and Y. Guo. Cooperative distributed source seeking by multiple robots: Algorithms and experiments. *IEEE/ASME Transactions on mechatronics*, 19(6):1810–1820, 2014.

[28] K. Liu and Q. Zhao. Decentralized multi-armed bandit with multiple distributed players. In *Proc. of Information Theory and Applications Workshop (ITA)*, pages 1–10, 2010.

[29] K. Liu and Q. Zhao. Distributed learning in multi-armed bandit with multiple players. *IEEE Transactions on Signal Processing*, 58(11):5667–5681, 2010.

[30] D. Martínez-Rubio, V. Kanade, and P. Rebeschini. Decentralized cooperative stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 4529–4540, 2019.

[31] S. McQuade and C. Monteleoni. Global climate model tracking using geospatial neighborhoods. In *Proc. of AAAI*, 2012.

[32] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

[33] A. Sankararaman, A. Ganesh, and S. Shakkottai. Social learning in multi agent multi armed bandits. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(3):1–35, 2019.

[34] A. Slivkins. Introduction to multi-armed bandits. *arXiv preprint arXiv:1904.07272*, 2019.

[35] B. Szorenyi, R. Busa-Fekete, I. Hegedus, R. Ormándi, M. Jelasity, and B. Kégl. Gossip-based distributed stochastic bandit algorithms. In *International Conference on Machine Learning*, pages 19–27, 2013.

[36] P.-A. Wang, A. Proutiere, K. Ariu, Y. Jedra, and A. Russo. Optimal algorithms for multiplayer multi-armed bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 4120–4129, 2020.

[37] W. Xia, T. Q. Quek, K. Guo, W. Wen, H. H. Yang, and H. Zhu. Multi-armed bandit based client scheduling for federated learning. *IEEE Transactions on Wireless Communications*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]