Distributed Bandits with Heterogeneous Agents

Lin Yang*, Yu-Zhen Janice Chen*, Mohammad H. Hajiemaili*, John C.S. Lui†, Don Towsley*
*University of Massachusetts Amherst, Amherst, USA
†The Chinese University of Hong Kong, Hong Kong, China
{linyang,yuzhenchen,hajiesmaili,towsley}@cs.umass.edu; cslui@cse.cuhk.hk

IEEE INFOCOM 2022 - IEEE Conference on Computer Communications | 978-1-6654-5822-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/INFOCOM48880.2022.9796901

***Abstract*—This paper tackles a multi-agent bandit setting where M agents cooperate together to solve the same instance of a K-armed stochastic bandit problem. The agents are *heterogeneous*: each agent has limited access to a local subset of arms and the agents are asynchronous with different gaps between decision-making rounds. The goal for each agent is to find its optimal local arm, and agents can cooperate by sharing their observations with others. While cooperation between agents improves the performance of learning, it comes with an additional complexity of communication between agents. For this heterogeneous multi-agent setting, we propose two learning algorithms, `CO-UCB` and `CO-AAE`. We prove that both algorithms achieve order-optimal regret, which is $O\left(\sum_{i:\tilde{\Delta}_i>0} \log T/\tilde{\Delta}_i\right)$, where $\tilde{\Delta}_i$ is the minimum suboptimality gap between the reward mean of arm i and any local optimal arm. In addition, a careful selection of the valuable information for cooperation, `CO-AAE` achieves a low communication complexity of $O(\log T)$. Last, numerical experiments verify the efficiency of both algorithms.**

***Index Terms*—Multi-armed bandits, multi-agent system, heterogeneous agents, regret, communication complexity**

I. Introduction

Multi-armed bandits (MABs) [1], [2] fall into a well-established framework for learning under uncertainty that has been extensively studied since the 1950s after the seminal work of [3]. MABs have a broad range of applications including online shortest path routing, online advertisement, channel allocation, and recommender systems [2], [4]–[6]. In the basic MAB problem, a learner repeatedly pulls an arm in each round, and observes the reward/loss associated with the selected arm, but not those associated with others. The goal of the learner is to minimize regret, which compares the rewards/loss received by the learner to those accumulated by the best arm in hindsight.

Distributed MABs, which are extensions of basic MABs, have been extensively studied recently in different settings [7]–[15]. Distributed bandits are well motivated by a broad range application scenarios such as (1) large-scale learning systems [16], in domains such as online advertising and recommendation systems; (2) cooperative search by multiple robots [17], [18]; (3) applications in wireless cognitive radio [7], [19]–[21]; and distributed learning in geographically distributed communication systems, such as a set of IoT devices learning about the underlying environments [22]–[26]. Most prior work on multi-agent MABs assume that agents are *homogeneous*: all agents have full access to the set of all arms, and hence they solve the same instance of a MAB problem, with the aim to minimize the aggregate regret of the agents either in a *competition* setting [7], [9], [14], [19]–[21], [27]–[29], i.e., where multiple agents receive degraded or no rewards when they pull the same arm, or in a *collaboration/cooperation* setting [10], [13]–[15], [30]–[32], where agents pulling the same arm observe independent rewards, and agents can communicate their observations to each other in order to improve their learning performance.

A. Distributed Bandits with Heterogeneous Agents

In this paper, we study a heterogeneous version of the cooperative multi-agent MAB problem in which agents only have partial access to the set of arms. More formally, we study a multi-agent system 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* arms for agent j. Agents also appears exhibit different learning capabilities that lead to different *action rates*; agent $j \in \mathcal{A}$ can pull an arm every $1/\theta_j$ rounds, $0 < \theta_j \leq 1$. Here θ_j is the action rate of agent j. The goal of each agent is to learn the best local arm within its local set, and agents can share information on overlapping arms in their local sets to accelerate the learning process. In this model, we assume agents are fully connected and can truthfully broadcast their observed rewards to each other. We call this setup *Action-constrained Cooperative Multi-agent MAB* (`AC-CMA2B`) and formally define it in Section II.

B. Motivating Application

Cooperative multi-agent bandits have been well-motivated in the literature, and in the following, we motivate the heterogeneous-agent setting. Online advertisement is a classic application that is tackled using the bandit framework. In the online advertisement, the goal is to select an ad (arm) for a product or a search query, and the reward is the revenue obtained from ads. In the context of `AC-CMA2B`, consider a scenario that for a set of related products, a separate agent runs a bandit algorithm to select a high-reward ad for each product in the set. However, the set of available ads might have partial overlaps among multiple related products, i.e., different agents might have some overlapping arms. Hence, by leveraging the `AC-CMA2B` model, agents running different bandit algorithms can cooperate by sharing their observations to improve their performance. In this setting, different action rate among the agents also makes sense, since different products may have different popularity; hence, the agents pull arms (ads) at different rates. One may imagine similar cooperative scenarios

for recommendation systems in social networks [15] where multiple learning agents in different social networks, e.g., Facebook, Instagram, cooperate to recommend posts from overlapping sets of actions. Even more broadly, the multi-agent version of classical bandit applications is a natural extension [33]. For example, in online shortest path routing problem [34], [35], as another classic example of bandit applications, multi-agent setting could capture the case in which the underlying network is large and each agent is responsible for routing within a sub-graph in the network. Last, it is also plausible that the to have asynchronous learning among different agents in the sense that each agent has its own action rate for decision making.

C. Contributions

The paper explores the benefits of cooperation among agents in improving the learning performance as compared to independent decision making by agents. On the other hand, cooperation between agents comes with additional communication complexity. Hence, we aim to design cooperative algorithms with sublinear regret and low communication complexity.

This is challenging since these two goals can be in conflict. Intuitively, with more information exchange, the agents can benefit from empirical observations made by others, resulting in smaller regret. However, this comes at the expense of additional communication complexity due to information exchange among agents. In this paper, we tackle AC-CMA2B by developing two cooperative bandit algorithms and analyze their regret and communication complexities. The contribution is summarized as follows.

First, to characterize the regret of our algorithms, we introduce $\tilde{\Delta}_i$ as a customized notion of the suboptimality gap, which is unique to AC-CMA2B. Specifically, the parameter $\tilde{\Delta}_i$, $i \in \mathcal{K}$ (see Equation (2) for the formal definition), measures the minimum gap between the mean reward of arm i and local optimal arms of agents that include i in their local sets. Intuitively, $\{\tilde{\Delta}_i\}_{i\in\mathcal{K}}$ determine the "difficulty" of the bandit problem in a distributed and heterogeneous setting and appear in the regret bounds.

Second, we present two learning algorithms, CO-UCB and CO-AAE, which extend the Upper Confidence Bound algorithm and the Active Arm Elimination algorithm [36] to the cooperative setting, respectively. We use the notion of local suboptimality gap $\tilde{\Delta}_i$ and characterize the regrets of CO-UCB and CO-AAE and show that both algorithms achieve a regret of $O\left(\sum_{i:\tilde{\Delta}_i>0} \log T/\tilde{\Delta}_i\right)$. By establishing a regret lower bound for AC-CMA2B, we show that the above regret is optimal. To the best of our knowledge, this is the first optimality result for distributed bandits in a heterogeneous setting. Even though both algorithms are order-optimal, the regret of CO-UCB is smaller than CO-AAE by a constant factor (see Theorems 2 and 4). This is also validated by our simulations in Section VI with real data traces.

Last, we investigate the communication complexity of both algorithms, which measures the communication overhead incurred by the agents for cooperation to accelerate the learning process. In our work, communication complexity is defined to be the total number of messages, i.e., arm indices and observed rewards, exchanged by the agents. Our analysis shows that CO-UCB generally needs to send as much as $O(M\Theta T)$ amount of messages, where Θ is the aggregate action rate of all agents. Apparently, the communication complexity of CO-UCB is higher than that of CO-AAE, which is $O\left(\sum_{i:\tilde{\Delta}_i>0} \log T/\tilde{\Delta}_i^2\right)$.

We note that the authors in [33] also tackle a cooperative bandit problem with multiple heterogeneous agents with partial access to a subset of arms and different action rates. However, in [33], the goal of each agent is to find the global optimal arm, while in this work, the goal of each agent is to find its local optimal arm. This difference leads to substantially different challenges in the algorithm design and analysis. More specifically, in [33], a foundational challenge is to find an effective cooperative strategy to resolve a dilemma between pulling local vs. external arms. This is not the case in AC-CMA2B since the goal is to find the best local action. In addition, in [33], the communication complexity of algorithms is not analyzed. Our paper, instead, focuses on designing cooperative strategies with low communication complexities.

II. Model and Preliminaries

A. System Model

We consider a cooperative multi-agent MAB (CMA2B) setting, where there is a set $\mathcal{A} = \{1, \ldots, M\}$ of independent agents, each of which has partial access to a global set $\mathcal{K} = \{1, \ldots, K\}$ of arms. Let $\mathcal{K}_j \subseteq \mathcal{K}, K_j = |\mathcal{K}_j|$, be the set of arms available to agent $j \in \mathcal{A}$. Associated with arms are mutually independent sequences of i.i.d. rewards, taken to be Bernoulli with means $0 \leq \mu(i) \leq 1$, $i \in \mathcal{K}$. We assume that the local sets of some agents overlap so that cooperation among agents makes sense.

In addition to differences in their access to arms, agents also differ in their decision making capabilities. Specifically, considering decision rounds $\{1, \ldots, T\}$, agent j can pull an arm every $\omega_j \in \mathbb{N}^+$ rounds, i.e., decision rounds for agent j are $t = \omega_j, 2\omega_j, \ldots, N_j\omega_j$, 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 agent j can pull arms.

We assume that all agents can communicate with all other agents. Hence every time an agent pulls an arm, it can broadcast the arm index and the reward received to any other agent. However, there is a deterministic communication delay, d_{j_1,j_2} , between any two agents, j_1 and j_2, measured in units of decision rounds. In addition, we use d_j to denote the maximum delay from other agents to agent j.

B. Performance Metrics

At each decision round, agent j can pull an arm from $\mathcal{K}_j$. The goal of each agent is to learn the best local arm. The regret of agent j is defined as

$$R_T^j := \mu(i_j^*)N_j - \sum\nolimits_{t\in\{k\omega_j : k=0,1,\ldots,N_j\}} x_t(I_t^j), \qquad (1)$$

where i_j^* is the local optimal arm in $\mathcal{K}_j$, $I_t^j \in \mathcal{K}_j$ is the action taken by agent j at round t, and $x_t(I_t^j)$ is the realized reward.

Without loss of generality, we assume that the local sets of at least two agents overlap, i.e., $\exists j, j' \in \mathcal{A} : \mathcal{K}_j \cap \mathcal{K}_{j'} \neq \emptyset$ and the overall goal is to minimize aggregate regret of all agents, i.e., $R_T = \sum_{j\in\mathcal{A}} R_T^j$.

In addition, we assume that it is costly to send observations to other agents. To measure the communication overhead of an algorithm in AC-CMA2B, we simply assume that each message contains enough bits to transmit the index of an arm or an observation on the reward, and similar to [14], [37], the communication complexity, denoted as C_T, is defined to be the total number of messages sent by all agents in $[1, T]$.

C. Additional Notations and Terminologies

To facilitate our algorithm design and analysis, we introduce the following notations. By $\mathcal{A}_i$, we denote set of agents that can access arm i, i.e., $\mathcal{A}_i := \{j \in \mathcal{A} : i \in \mathcal{K}_j\}$. By $\mathcal{A}_i^*$, we denote the set of agents whose optimal local arm is i, i.e., $\mathcal{A}_i^* := \{j \in \mathcal{A}_i : \mu_i \geq \mu_{i'},\ \text{for } i' \in \mathcal{K}_j\}$. Note that $\mathcal{A}_i^*$ may be empty. Moreover, let $\mathcal{A}_{-i}^* = \mathcal{A}_i \setminus \mathcal{A}_i^*$ be the set of agents including i as a suboptimal arm. Finally, let M_i, M_{i^*}, and M_{-i} be the sizes of $\mathcal{A}_i$, $\mathcal{A}_{i^*}$, and $\mathcal{A}_{-i}$ respectively.

By $\Delta(i, i')$, we denote the difference in the mean rewards of arms i and i', i.e., $\Delta(i, i') := \mu(i) - \mu(i')$. Specifically, $\Delta(i^*, i)$ written as Δ_i, is the suboptimality gap in the basic bandit problem. In additional to this standard definition, we introduce the following CMA2B-specific version of the suboptimality gap, denoted by $\tilde{\Delta}_i$

$$\tilde{\Delta}_i := \begin{cases} \min_{j\in\mathcal{A}_{-i}} \Delta(i_j^*, i), & \mathcal{A}_{-i} \neq \emptyset; \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Last, we define Θ and Θ_i, $i \in \mathcal{K}$, as follows.

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

We note that both $\tilde{\Delta}_i$ and Θ_i play key roles in characterizing the regret bounds of an algorithm in AC-CMA2B. Specifically, $\tilde{\Delta}_i$ measures the minimum gap of the reward mean between the local optimal arm of agents in $\mathcal{A}_{-i}$ and arm i, and Θ_i measures the aggregate action rate of agents in $\mathcal{A}_i$, and roughly the larger the Θ_i, the higher the rate at which arm i can be pulled by the set of agents that belongs to, i.e., $\mathcal{A}_i$.

III. Algorithms

In AC-CMA2B, each agent has to identify the local optimal arm and the learning process can be accelerated by communicating with the other agents with common arms. The traditional challenge for MAB comes from the exploration-exploitation dilemma. In AC-CMA2B, the agents have to resolve this by designing learning algorithms with low regret and low communication complexity. The heterogeneity in action rates and limited access to the decision set exacerbates the design and analysis of cooperative learning algorithms for AC-CMA2B. In this section, we present two algorithms: CO-UCB and CO-AAE. CO-UCB generalizes the classic Upper Confidence Bound algorithm and achieves good regret but incurs high communication complexity. CO-AAE borrows the idea of the arm elimination strategy but incorporates a novel communication strategy tailored to reduce communication complexity. In Section IV, we derive a regret lower bound for AC-CMA2B, analyze regrets and communication complexities for both algorithms, show the order optimality of the regrets for both algorithms, and show that CO-AAE achieves low communication complexity.

A. Confidence Interval

Both CO-UCB and CO-AAE use confidence of the mean rewards to make decisions. We introduce the notion of confidence interval in the following. In AC-CMA2B, each agent computes empirical mean rewards of the arms. For arm $i \in \mathcal{K}$ with n observations, the mean reward is denoted as $\hat{\mu}(i, n)$[1], which is the average of the n observations on arm i. With these observations, we can compute a confidence interval for the true mean reward. Specifically, the width of the confidence interval for arm i and agent j at time t is defined as

$$\texttt{CI}(i, j, t) := \sqrt{\frac{\alpha \log \delta_t^{-1}}{2\hat{n}_t^j(i)}}, \tag{3}$$

where $\hat{n}_t^j(i)$ is the total number of observations (including both local observations and those received from other agents) of arm i available to agent j by time t (observations made in time slots from 1 to $t-1$). Here $\delta_t > 0$ and $\alpha > 2$ are parameters of the confidence interval. We build the following confidence interval for arm $i \in \mathcal{K}$:

$$\mu(i) \in \left[\hat{\mu}(i, \hat{n}_t^j(i)) - \texttt{CI}(i, j, t), \hat{\mu}(i, \hat{n}_t^j(i)) + \texttt{CI}(i, j, t)\right],$$

where $\mu(i)$ satisfies the upper (or lower) bound with probability at least $1 - \delta_t^\alpha$ ($0 < \delta_t \leq 1$ is a specified parameter at time slot t). One can refer to [1] for a detailed analysis of the above confidence interval.

B. CO-UCB: Cooperative Upper Confidence Bound Algorithm

In this subsection, we present CO-UCB, a cooperative bandit algorithm for the AC-CMA2B model. According to CO-UCB, each agent selects the arm with the largest upper confidence bound. For agent j, there is

$$I_t^j = \arg\max_{i\in\mathcal{K}} \hat{\mu}\left(i, \hat{n}_t^j(i)\right) + \texttt{CI}(i, j, t).$$

With each observation received from the selected arm or other agents, CO-UCB updates the mean reward estimate and the upper confidence bound. In the meantime, observations received from local arms are broadcast to other agents that contain the corresponding arm in their local sets. Details of CO-UCB are summarized in Algorithm 1. In CO-UCB, agents broadcast all observations with others. This leads to high communication complexity of the algorithm. In what follows, we present CO-AAE, an algorithm that improves the communication complexity of cooperation.

[1]In the algorithm pseudocode, we drop t and j from the notations $\hat{n}_t^j(i)$ and $\hat{\mu}(i, \hat{n}_t^j(i))$ for brevity, and simplify them as $\hat{n}(i)$ and $\hat{\mu}(i)$, respectively. The precise notation, however, is used in analysis.

Algorithm 1 The CO−UCB Algorithm for Agent j

Initialization: $\hat{n}(i) = 0$, $\hat{\mu}(i)$, $i \in \mathcal{K}_j$; $\alpha > 2$, δ_t.
for each ecision round $t = l/\theta_j$ ($l \in \{1, \ldots, N_j\}$) **do**
    Pull arm I_t^j with the highest upper confidence bound
    Increase $\hat{n}(I_t^j)$ by 1
    Update the empirical mean value of $\hat{\mu}(I_t^j)$
    Broadcast $x_t(I_t^j)$ to other agents which contains arm I_t^j
end for
for each newly received $x_t(i)$, $i \in \mathcal{K}_j$ from the past decision round **do**
    Execute Lines (4)-(5)
end for

Algorithm 2 The CO−AAE Algorithm for Agent j

Initialization: $\hat{n}(i) = 0$, $\hat{\mu}(i)$, $i \in \mathcal{K}_j$; $\alpha > 2$, δ_t.
for each received $x_\tau(i)$, $\tau < t$, $i \in \mathcal{K}_j$ for past rounds **do**
    Execute Lines (7)-(11)
end for
for each decision round $t = l/\theta_j$ ($l \in \{1, \ldots, N_j\}$) **do**
    Pull arm I_t^j from the candidate set as constructed in Equation (4) with the least observations
    Increase $\hat{n}(I_t^j)$ by 1 and update the empirical mean value, $\hat{\mu}(I_t^j)$
    Reconstruct the candidate set based on the updated values of $\hat{n}(I_t)$ and $\hat{\mu}(I_t^j)$ by using Equation (4)
    if one arm is eliminated **then**
        Broadcast the indices of eliminated arms to other agents
    end if
    if the candidate set contains more than 1 arms **then**
        Broadcast $x_t(I_t^j)$ to other agents whose candidate set contains arm I_t^j and has more than one arms
    end if
end for

C. CO−AAE: Cooperative Active Arm Elimination Algorithm

CO−AAE is independently executed by each agent and is summarized as Algorithm 2. By maintaining the confidence intervals of local arms, CO−AAE maintains a *candidate set* to track the arms likely to be the optimal local arm. The candidate set is initially the entire local set, and when the confidence interval of an arm comes to lie below that of another arm, the arm is removed from the candidate set. During execution, CO−AAE selects the arm with the fewest observations from the candidate set. The candidate set allows CO−AAE to avoid sending messages regarding low-reward arms resulting in a lower communication complexity than CO−UCB. Details are introduced below.

a) Selection Policy for Local Arms: We first present details on constructing the candidate set for agent j. We formally define the candidate set $\mathcal{C}_{j,t}$ in Eq. (4). The candidate set of j originally contains all arms in $\mathcal{K}_j$. Then, CO−AAE eliminates those arms whose confidence intervals lie below those of other arms without further consideration, and keeps the rest in a dynamic candidate set of arms. The agent updates the candidate set after pulling an arm and each time it receives an observation from another agent. Note that communication delays and action rates vary across agents. Hence, the recorded number of observations and empirical mean rewards vary among agents. To balance the number of observations among different local arms, the agent at each time slot pulls the arm within its local candidate set having the least number of observations.

b) Communication Policy: In order to reduce communication complexity, it is also crucial for CO−AAE to decide how to share information among different agents. During the execution of CO−AAE, each agent updates its candidate set with its received observations. When an arm is eliminated from an agent's candidate set, the agent broadcasts the index of the eliminated arm, such that all agents can track the candidate sets in others. In the following, we will introduce our communication policy tailored to the CO−AAE algorithm. The communication policy of CO−AAE generally follows the following two rules.

1) An agent only broadcasts observations to agents whose candidate sets contain more than one arm and contains the arms which the observations are sampled on.
2) When there is only one arm in the candidate set, the agent also broadcasts all observations to other agents.

By the first rule, the communication policy avoids transmitting redundant observations to the agents that have finished the learning task, i.e., those with only one arm in their candidate sets. The second rule prevents "fast" agents that quickly eliminated suboptimal arms from sending too many observations to the "slow" agents containing the arms whose means are close to the local optimal arm. Otherwise, sending too many observations on those arms to "slow" agents may incur $O(T)$ communication complexity in the extreme case.

Last, we note that the above communication policy can be easily implemented in practical systems and works efficiently in a fully distributed environment even when agents don't know parameters of other agents, such as action rates. Previous communication policies in distributed bandits, such as those in [13], [14] etc., require a centralized coordinator. It is also worth noting that the above communication policy cannot be applied to CO−UCB, since each agent fails to send out explicit signals on suboptimal arms.

IV. Theoretical Results

In this section, we present theoretical results for CO−UCB and CO−AAE. For AC−CMA2B, the theoretical challenge is to account for the constraint that agents can only pull arms from predetermined (and possibly overlapping sets) of arms in the regret bounds. This challenge can be tackled by incorporating the agent-specific suboptimality gaps introduced in Eq. (2) into the regret analysis. We provide upper and lower bounds for the regrets in the AC−CMA2B setting, all of which depend on the agent-specific suboptimality gaps $\tilde{\Delta}_i$. Proofs are given in Section V and in our technical report [38].

A. An Overview of Our Results

Throughout this section, by policy, we mean the way that each agent determines which arm to select in each decision round. Let $\mathsf{KL}(u, v)$ denote the Kullback-Leibler divergence between a Bernoulli distribution with parameters of u and v, i.e., $\mathsf{KL}(u, v) = u\log(u/v) + (1-u)\log((1-u)/(1-v))$.

***Theorem** 1:* (Regret Lower Bound for AC−CMA2B) Assume $\theta_j = O(1)$ for $j \in \mathcal{A}$ and a policy that satisfies

$$\mathcal{C}_{j,t} := \left\{ i \in \mathcal{K}_j : \hat{\mu}(i, \hat{n}_t^j(i)) + \texttt{cint}(i,j,t) \geq \hat{\mu}(i', \hat{n}_t^j(i)) - \texttt{cint}(i,j,t), \text{ for any } i' \in \mathcal{K}_j \right\}. \quad (4)$$

$\mathbb{E}[n_T(i)] = o(T^a)$ for any set of Bernoulli reward distributions, any arm i with $\tilde{\Delta}_i > 0$, and any $a > 0$. Then, for any set of Bernoulli reward distributions, the expected regret of any algorithm satisfies

$$\liminf_{T\to\infty} \frac{\mathbb{E}[R_T]}{\log T} \geq \sum_{i:\tilde{\Delta}_i>0} \frac{\tilde{\Delta}_i}{\mathsf{KL}(\mu_i, \mu_i + \tilde{\Delta}_i)},$$

The proof leverages techniques similar to those for establishing the classical result for the basic stochastic bandits [39] and is given in [38].

To simplify the presentation of the regret bounds for both algorithms, we introduce the following notations.

$$q_1 := 2 \sum_{j\in\mathcal{A}} \sum_{l=1}^{N_j} \sum_{i\in\mathcal{K}_j} \frac{l\Theta_i}{\theta_j} \delta_{l/\theta_j}^{\alpha},$$

$$f_i(\delta) := \sum_{j\in\mathcal{A}_i} \min\left\{ d_j\theta_j, \frac{2\alpha\log\delta^{-1}}{\Delta^2(i_j^*, i)} \right\},$$

where $\alpha > 2$, $\delta_{l/\theta_j} > 0$ are parameters specified by the algorithms, and $\delta := \max_l \delta_{l/\theta_j}$. Further, d_j is the maximum delay from other agents to agent j. The following theorem characterizes the regret of CO-UCB with regard to specified parameters α and δ_{l/θ_j}.

Theorem 2: (Expected regret of CO-UCB) When $\alpha > 2$ and $\delta_{l/\theta_j} > 0$ for any l and j, the expected regret of the CO-UCB algorithm satisfies

$$\mathbb{E}[R_T] \leq \sum_{i:\tilde{\Delta}_i>0} \left(\frac{6\alpha\log\delta^{-1}}{\tilde{\Delta}_i} + q_1 + f_i(\delta) \right).$$

We now proceed to further analyze the regret of the CO-UCB. By setting $\delta_t = 1/t$, we have

$$\begin{aligned} & 2\sum_{j\in\mathcal{A}} \sum_{l=1}^{N_j} \sum_{i\in\mathcal{K}_j} \frac{l\Theta_i}{\theta_j} \delta_{l/\theta_j}^{\alpha} \\ = & 2\sum_{j\in\mathcal{A}} \sum_{l=1}^{N_j} \sum_{i\in\mathcal{K}_j} \Theta_i \frac{1}{(l/\theta_j)^{\alpha-1}} \\ \leq & 2\sum_{j\in\mathcal{A}} \sum_{l=1}^{N_j} \Theta \frac{1}{(l/\theta_j)^{\alpha-1}} \leq \frac{2}{\alpha-2} \sum_{j\in\mathcal{A}} \Theta\theta_j^{\alpha-1}. \end{aligned}$$

We further define

$$q_2 := \frac{2}{\alpha-2} \sum_{j\in\mathcal{A}} \Theta\theta_j^{\alpha-1}.$$

Applying the above results and definitions to Theorem 2 yields the following corollary, which builds up a $O(\log T)$ regret upper bound for the CO-UCB algorithm.

Corollary 1: With $\delta_t = 1/t$ and $\alpha > 2$, the CO-UCB algorithm attains the following expected regret

$$\mathbb{E}[R_T] \leq \sum_{i:\tilde{\Delta}_i>0} \left(\frac{6\alpha\log T}{\tilde{\Delta}_i} + f_i\left(\frac{1}{T}\right) + 1 \right) + q_2.$$

We also analyze the communication complexity of CO-UCB. For simplicity, we assume that one message is needed to send an observation from an agent to another one. The total number observations made by all agents is ΘT. Then, broadcasting an observation on arm i to all other agents results in at most M communications messages. Hence, the total communication complexity of CO-UCB is $O(M\Theta T)$, which is formally summarized in the following theorem.

Theorem 3: (Communication complexity of CO-UCB) The communication complexity of CO-UCB is $O(M\Theta T)$.

Now, we proceed to present the regret and communication complexity of CO-AAE. Similar to the definition of f, we define

$$g_i(\delta) := \sum_{j\in\mathcal{A}_i} \min\left\{ d_j\theta_j, \frac{8\alpha\log\delta^{-1}}{\Delta^2(i_j^*, i)} \right\}.$$

The following theorem and corollary establishes an upper bound on the expected regret of CO-AAE.

Theorem 4: (Expected regret for CO-AAE) With $\alpha > 2$ and $\delta_{l/\theta_j} > 0$ for any l and j, the expected regret of the CO-AAE algorithm satisfies

$$\mathbb{E}[R_T] \leq \sum_{i:\tilde{\Delta}_i>0} \left(\frac{24\alpha\log\delta^{-1}}{\tilde{\Delta}_i} + q_1 + g_i(\delta) + 1 \right).$$

Corollary 2: When $\alpha > 2$ and $\delta_t = 1/t$, CO-AAE attains the following expected regret

$$\mathbb{E}[R_T] \leq \sum_{i:\tilde{\Delta}_i>0} \left(\frac{24\alpha\log T}{\tilde{\Delta}_i} + g_i\left(\frac{1}{T}\right) + 1 \right) + q_2.$$

We have the following theorem providing an upper bound for the communication complexity of CO-AAE.

Theorem 5: (Communication complexity of CO-AAE) Let $\delta_t = 1/t$ and $\alpha > 2$. The communication complexity of CO-AAE satisfies

$$C_T \leq \sum_{i\in\mathcal{K}} \left(\frac{8\alpha\log T}{\tilde{\Delta}_i^2} + \sum_{j\in\mathcal{A}_{-i}^*} d_j\theta_j + q_2 + 1 \right) (M + M_i).$$

B. Discussions

In the following, we discuss the significance of our results.

a) Regret Optimality of CO−UCB and CO−AAE: The first observation regarding Corollaries 1 and 2 is that the terms $f_i(1/T)$ and $g_i(1/T)$ of the regrets depend liearly on the delay when it is not too large. Generally, $f_i(1/T)$ and $g_i(1/T)$ relate to the number of outstanding observations that have not yet arrived. Considering the fact that $\theta_j \leq 1$, and $\mathrm{KL}(\mu_i, \mu_i + \tilde{\Delta}_i)$ satisfies

$$2\tilde{\Delta}_i^2 \leq \mathrm{KL}(\mu_i, \mu_i + \tilde{\Delta}_i) \leq \frac{\tilde{\Delta}_i^2}{(\mu(i) + \tilde{\Delta}_i)(1 - \mu(i) - \tilde{\Delta}_i)}, \quad (5)$$

one observes that both regrets match the regret lower bound in Theorem 1 when delays are bounded by a constant.

b) Comparison with Independent Policies without Cooperation: Without cooperation, we can derive a lower bound for the regret of each agent j by Theorem 2.2 in [1], that is

$$\liminf_{T\to\infty} \frac{\mathbb{E}\left[R_T^j\right]}{\log T} \geq \sum_{i \in \mathcal{K}_j : \Delta(i_j^*, i) > 0} \frac{\Delta(i_j^*, i)}{\mathrm{KL}(\mu_i, \mu_i + \Delta(i_j^*, i))}.$$

Combined with Eq. (5), the best regret that any non-cooperative algorithm can achieve for the integrated system is no better than

$$\sum_{j \in \mathcal{A}} \sum_{i : i \in \mathcal{K}_j, \Delta(i_j^*, i) > 0} \frac{\log T}{\Delta(i_j^*, i)} = \sum_{i \in \mathcal{K}} \sum_{j \in \mathcal{A}_i / \mathcal{A}_i^*} \frac{\log T}{\Delta(i_j^*, i)}.$$

Note that, with bounded delays, the regret upper bounds of both CO−UCB and CO−AAE is $O\left(\sum_{i:\tilde{\Delta}_i > 0} \log T / \tilde{\Delta}_i\right)$. By the definition of $\tilde{\Delta}_i$ in Eq. (2), we have

$$O\left(\sum_{i \in \mathcal{K}} \sum_{j \in \mathcal{A}_i / \mathcal{A}_i^*} \frac{\log T}{\Delta(i_j^*, i)}\right) \geq O\left(\sum_{i : \tilde{\Delta}_i > 0} \frac{\log T}{\tilde{\Delta}_i}\right).$$

To conclude, a non-cooperative strategy will have a much larger regret than CO−UCB and CO−AAE, especially when the number of agents is large. In Section VI, we also numerically compare our algorithms to the above independent algorithms as baseline algorithms and the numerical results match our above observation.

c) Comparison in a Special Case with Full Access to the Arms: Theorems 2 and 4 show that the regret upper bounds depend on the new suboptimality parameter $\tilde{\Delta}_i$, which measures the minimum gap between arm i and local optimal arms. Intuitively, the closer the expected reward of the local optimal arm to that of the global optimal arm, the smaller the regret will be. In the special case where each agent can access the global arm set and delays are bounded, we have $\tilde{\Delta}_i = \Delta_i$ and thus the expected regret of either CO−UCB or CO−AAE becomes $O\left(\sum_{i \in \mathcal{K}} (\alpha \log T)/\Delta_i\right)$. In the basic bandit model, a learning algorithm suffers a regret lower bound that also depends on Δ_i, i.e.,

$$\liminf_{T\to\infty} \frac{\mathbb{E}\left[R_T\right]}{\log T} \geq \sum_{i : \Delta_i > 0} \frac{\Delta_i}{\mathrm{KL}(\mu_i, \mu_i + \Delta_i)}.$$

Thus, by assuming a constant delay and Θ, the regret matches the lower bound in the special case where agents possess full access to the arms.

d) Performance with Large Delays: We are also interested in the performance of the algorithms when large delays exist in the system. We take CO−AAE as an example. In the extreme case where the maximum delay is arbitrarily large, the regret bound given in Corollary 1 becomes

$$\begin{aligned}\mathbb{E}\left[R_T\right] &\leq \sum_{i : \tilde{\Delta}_i > 0} \left(\frac{16\alpha \log T}{\tilde{\Delta}_i} + \sum_{j' \in \mathcal{A}_i} \frac{8\alpha \log T}{\Delta(i_{j'}^*, i)}\right) + q_2 + K \\ &= \sum_{i : \tilde{\Delta}_i > 0} \frac{16\alpha \log T}{\tilde{\Delta}_i} + \sum_{j \in \mathcal{A}} \sum_{i \in \mathcal{K}_j} \frac{8\alpha \log T}{\Delta(i_j^*, i)} + q_2 + K,\end{aligned}$$

where the second term dominates, and the above regret matches that of non-cooperative learning algorithms.

e) Communication Complexity: The regrets of CO−UCB and CO−AAE both drop the heavy dependency on the number of agents, but they incur much different communication overheads. By our results, CO−AAE achieves much lower communication complexity than CO−UCB, which is $O\left(\sum_{i \in \mathcal{K}} (M\alpha \log T)/\tilde{\Delta}_i^2\right)$. We leave it as an open problem to design the algorithm which simultaneously attains the lowest communication complexity.

V. Proofs

In this section, we provide full proofs for Theorem 5, but only proof sketches for theorems 2 and 4. For detailed proofs of other theorems, we referred to our technical report in [38].

A. Proof Skeletons for Regret Upper Bounds in Theorems 2 and 4

We first provide a proof sketch for the regret of CO−UCB as stated in Theorem 2. In our analysis, 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 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. More specifically, when agent j makes a Type-I decision at time t, the following equation holds for any i in $\mathcal{K}_j$.

$$\mu(i) \in \left[\hat{\mu}\left(i, \hat{n}_t^j(i)\right) - \mathtt{CI}(i, j, t), \hat{\mu}\left(i, \hat{n}_t^j(i)\right) + \mathtt{CI}(i, j, t)\right].$$

The following Lemma provides the probability that a Type-I decision happens at a particular decision round.

Lemma 1: At any time slot t when an agent makes its l-the decision, it makes a Type-I decision with a probability at least $1 - 2\sum_{i \in \mathcal{K}_j} \frac{l\Theta_i}{\theta_j} \delta_{l/\theta_j}^{\alpha}$.

In addition, we use the following lemma to upper bound the number of pulls of suboptimal arms by agent j when Type-I decision happens.

Lemma 2: If at any time $t \leq T$ agent $j \in \mathcal{A}_{-i}^*$ makes a Type-I decision and pulls arm i, i.e., $I_t^j = i$, we have

$$\hat{n}_t^j(i) \leq \frac{2\alpha \log \delta^{-1}}{\Delta^2(i_j^*, i)}.$$

Applying the above upper bound on the pulls of sub-optimal arms, we can derive the regret of pulling suboptimal arms when making Type-I decisions. Summing up the regret of making Type-II decisions and regret of pulling suboptimal arms when making Type-I decisions gives the final result stated in Theorem 2.

The proof of Theorem 4 also leverages the notions of Type-I/Type-II decisions. Specifically, with Type-I decisions, an agent is able to keep the local optimal arm in its candidate set and eventually converges its decisions to the local optimal arm. Similarly, we have the following lemma.

Lemma 3: If at any time $t \leq T$ agent $j \in \mathcal{A}^*_{-i}$ by CO-AAE makes a Type-I decision and pulls arm i, i.e., $I_t^j = i$, we have

$$\hat{n}_t^j(i) \leq \frac{8\alpha \log \delta^{-1}}{\Delta^2(i_j^*, i)} + 1.$$

We skip the rest of the proof for Theorem 4, since it follows similar steps to that of Theorem 2.

B. A proof of Theorem 5

Generally, the proof contains two steps. The first one is to upper bound the number of messages on suboptimal arms, and the second one is to upper bound that on the local optimal arms.

(1) We assume at time slot t, an agent in $\mathcal{A}^*_{-i}$ makes a Type-I decision to select arm i. From Lemma 3, we can upper bound the total number of selection times by agents in $\mathcal{A}_i$ for suboptimal arm i up to t by

$$\frac{8\alpha \log \delta^{-1}}{\Delta^2(i^*_{j_{M_{-i}}}, i)} + \sum_{j \in \mathcal{A}^*_{-i}} d_j \theta_j + 1,$$

where the second term corresponds to an upper bound for the number of outstanding observations on arm i.

Combined with the fact that the expected number of Type-II decisions for all agents is upper bounded by q, we can upper bound the expected number of observations on arm i by agents in $\mathcal{A}^*_{-i}$ as follows.

$$\begin{aligned}&\frac{8\alpha \log \delta^{-1}}{\Delta^2(i^*_{j_{M_{-i}}}, i)} + \sum_{j \in \mathcal{A}^*_{-i}} d_j \theta_j + \frac{2}{\alpha - 2} \sum_{j \in \mathcal{A}} \Theta \theta_j^{\alpha - 1} + 1 \\ =&\frac{8\alpha \log \delta^{-1}}{\Delta^2(i^*_{j_{M_{-i}}}, i)} + \sum_{j \in \mathcal{A}^*_{-i}} d_j \theta_j + q_2 + 1.\end{aligned}$$

Accordingly, the expected number of messages sent by $\mathcal{A}^*_{-i}$ to broadcast those observations on arm i is upper bounded by

$$\left(\frac{8\alpha \log \delta^{-1}}{\Delta^2(i^*_{j_{M_{-i}}}, i)} + \sum_{j \in \mathcal{A}^*_{-i}} d_j \theta_j + q_2 + 1 \right) M_i.$$

Then we can further upper bound the total number of messages sent by agents for broadcasting the observations of their suboptimal arms by

$$\sum_{i \in \mathcal{K}} \left(\frac{8\alpha \log \delta^{-1}}{\Delta^2(i^*_{j_{M_{-i}}}, i)} + \sum_{j \in \mathcal{A}^*_{-i}} d_j \theta_j + q_2 + 1 \right) M_i.$$

(2) By the rules of the CO-AAE algorithm, we have that an agent broadcasts its observations only when its candidate set has more than one arms. That is, the number of broadcast observations on the optimal arm is not larger than

$$\sum_{i \in \mathcal{K}} \left(\frac{8\alpha \log \delta^{-1}}{\Delta^2(i^*_{j_{M_{-i}}}, i)} + \sum_{j \in \mathcal{A}^*_{-i}} d_j \theta_j + q_2 + 1 \right).$$

Hence, the number of messages on the local optimal arms is upper bounded by

$$\sum_{i \in \mathcal{K}} \left(\frac{8\alpha \log \delta^{-1}}{\Delta^2(i^*_{j_{M_{-i}}}, i)} + \sum_{j \in \mathcal{A}^*_{-i}} d_j \theta_j + q_2 + 1 \right) M.$$

Combining the above two cases yields an upper bound on the expected number of messages sent by the agents, which is

$$\begin{aligned}&\sum_{i \in \mathcal{K}} \left(\frac{8\alpha \log \delta^{-1}}{\Delta^2(i^*_{j_{M_{-i}}}, i)} + \sum_{j \in \mathcal{A}^*_{-i}} d_j \theta_j + q_2 + 1 \right) (M + M_i) \\ =&\sum_{i \in \mathcal{K}} \left(\frac{8\alpha \log \delta^{-1}}{\tilde{\Delta}_i^2} + \sum_{j \in \mathcal{A}^*_{-i}} d_j \theta_j + q_2 + 1 \right) (M + M_i) \\ =&\sum_{i \in \mathcal{K}} \left(\frac{8\alpha \log T}{\tilde{\Delta}_i^2} + \sum_{j \in \mathcal{A}^*_{-i}} d_j \theta_j + q_2 + 1 \right) (M + M_i).\end{aligned}$$

This completes the proof.

VI. Numerical Experiments

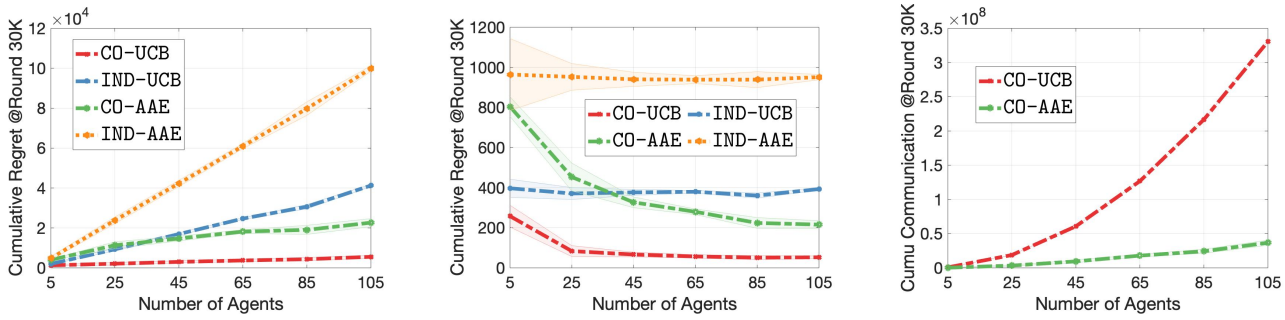
In this section, we illustrate the performance of our proposed algorithms for the AC-CMA2B settings through numerical experiments. For AC-CMA2B, our goal is to evaluate the performance of CO-UCB and CO-AAE, including regret and communication complexity, and compare them to that of non-cooperative algorithms where each agent uses only its local observations to find the best arm. Then, we investigate the impact of communication delay on the performance of proposed algorithms in AC-CMA2B.

A. Overview of Setup

We assume there are $K = 100$ arms with Bernoulli rewards with average rewards uniformly randomly taken from *AdClicks* [40]. In 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 10 independent trials and standard deviations are plotted as shaded areas. The allocation of arms to agents and number of agents differ in each experiment as explained in the corresponding sections.

B. Experimental Results

a) Experiment 1: In the first experiment, we fix the total number of arms to $K = 20$, and fix the number of arms per agent to $|\mathcal{K}_j| = 6, j \in \mathcal{A}$. We further vary the number of agents from $M = 5$ (light overlap) to $M = 105$ (heavy overlap), with step size of 20.

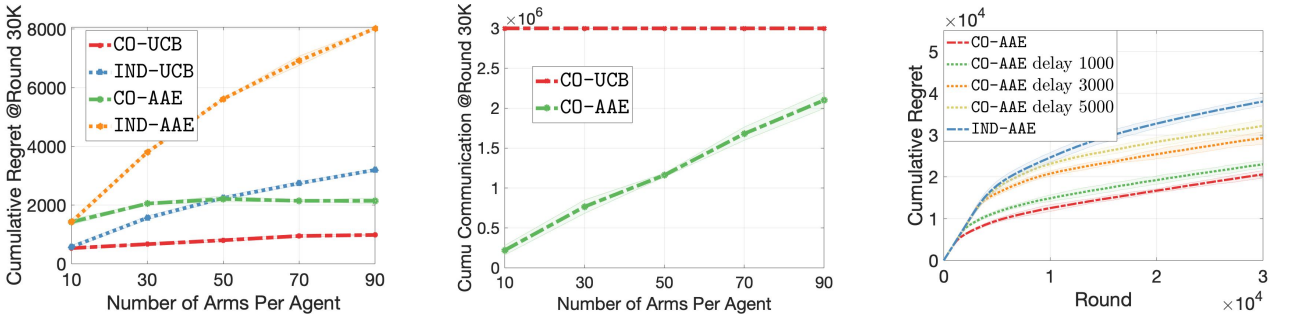


(a) Cumulative regrets with different number of agents.

(b) Average per-agent regret with different numbers of agents.

(c) Communication overhead.

Fig. 1. Simulation results for AC-CMA2B with different number of agents in the system.



(a) Per-agent regret by CO-UCB and IND-UCB.

(b) Communication overhead.

(c) Performance of CO-AAE with different values for communication delay.

Fig. 2. Simulation results for AC-CMA2B with different number of arms in each agent.

The results are shown in Figure 1. We see the observation of better performance of CO-UCB and CO-AAE as compared to IND-UCB and IND-AAE. Figure 1(a) shows a rapid increase in the cumulative regret of non-cooperative algorithms, while that of the cooperative algorithms remains the same despite the increase in the number of agents when the number of agents is larger than 65. Figure 1(b) depicts almost no change in the average per-agent regret of IND-UCB and IND-AAE, and a significant decrease for that of CO-UCB and CO-AAE, that is due to greater overlap in the local arm sets.

Figure 1(c) shows an increase of communication overheads for both CO-UCB and CO-AAE. Specifically, the CO-AAE algorithm incurs much lower communication overhead than CO-UCB in all experiments, validating our results in theorems 3 and 5. Another important observation is that, with more agents, the communication overheads for both algorithms increase. That is because, when there are more agents, there will be more possibility for agents to cooperate, with more observations exchanged on overlapping arms.

b) Experiment 2: In the second experiment, we set $K = 100$ arms, and $M = 10$ agents, and vary the number of arms in each agent j from $|\mathcal{K}_j| = 10$ with no overlap, to $|\mathcal{K}_j| = \{30, 50, 70, 90\}$ with increasing degree of overlap. The cumulative regret at 30000 rounds for five cases are reported in Figure 2(a). Figure 2(a) shows that cooperative algorithms significantly outperform non-cooperative algorithms in general cases. One can see the gap between the performance of cooperative and non-cooperative algorithms increases as the overlap increases. This observation depicts that CO-UCB and CO-AAE benefits from cooperation.

Figure 2(b) depicts the communication overheads of CO-AAE and CO-UCB, as we vary the number of arms in each agent from light overlap to heavy overlap of the local arm sets. We observe that the communication overhead of CO-AAE is much lower than CO-UCB in all experiments. In addition, the communication overhead of CO-AAE increases as the number of arms in each agent gets larger. That is because, when there are more arms in each agent, it will take more time for the agents to eliminate suboptimal arms, resulting in an increase in the communication overhead by CO-AAE.

c) Experiment 3: Last, we investigate the performance of the cooperative algorithm with different delays and take CO-AAE as an example. Toward this, we consider three additional scenarios with average delays of 1000, 3000 and 5000 slots. At each time slot, the exact delay is taken uniformly randomly in a given region. In Figure 2(c), we report the evolution of cumulative regret of CO-AAE. The results show that the regret of CO-AAE for AC-CMA2B increases and approaches the regret of IND-AAE as the delay increases.

VII. Conclusion and Future Directions

In this paper, we study the cooperative stochastic bandit problem with heterogeneous agents, with two algo-

rithms, CO−UCB and CO−AAE proposed. Both algorithms attain the optimal regret, which is independent of the number of agents. However, CO−AAE outperforms CO−UCB in communication complexity: CO−AAE needs to send $O\left(\sum_{i\in\mathcal{K}}(M\alpha\log T)/(\tilde{\Delta}_i^2)\right)$ amount of messages, while the communication complexity of CO−UCB is $O(M\Theta T)$.

This paper also motivates several open questions. A promising and practically relevant work is to design the algorithm which simultaneously attains the lowest communication complexity and the optimal regret independent of the number of agents.

ACKNOWLEDGMENT

This research is supported by NSF CAREER 2045641, CPS 2136199, CNS 2106299, CNS 2102963, CNS 1908298, GRF 14200321, the Army Research Laboratory under Cooperative Agreement W911NF17-2-0196 (IoBT CRA), and the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement W911NF-16-3-0001.

REFERENCES

[1] S. Bubeck, N. Cesa-Bianchi *et al.*, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends® in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

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

[3] H. Robbins, "Some aspects of the sequential design of experiments," *Bulletin of the American Mathematical Society*, vol. 58, no. 5, pp. 527–535, 1952.

[4] 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 *Proceedings of the 2016 ACM SIGCOMM Conference*, 2016, pp. 286–299.

[5] 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*. Citeseer, 2007, pp. 817–824.

[6] 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*, 2016, pp. 207–220.

[7] K. Liu and Q. Zhao, "Distributed learning in multi-armed bandit with multiple players," *IEEE Transactions on Signal Processing*, vol. 58, no. 11, pp. 5667–5681, 2010.

[8] 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*, 2013, pp. 19–27.

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

[10] R. K. Kolla, K. Jagannathan, and A. Gopalan, "Collaborative learning of stochastic bandits over a social network," *IEEE/ACM Transactions on Networking*, vol. 26, no. 4, pp. 1782–1795, 2018.

[11] D. Kalathil, N. Nayyar, and R. Jain, "Decentralized learning for multiplayer multiarmed bandits," *IEEE Transactions on Information Theory*, vol. 60, no. 4, pp. 2331–2345, 2014.

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

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

[14] 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*, 2020, pp. 4120–4129.

[15] 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*, vol. 3, no. 3, pp. 1–35, 2019.

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

[17] S. Li, R. Kong, and Y. Guo, "Cooperative distributed source seeking by multiple robots: Algorithms and experiments," *IEEE/ASME Transactions on mechatronics*, vol. 19, no. 6, pp. 1810–1820, 2014.

[18] 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*, vol. 48, no. 10, pp. 1715–1724, 2017.

[19] 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*, 2020, pp. 961–987.

[20] K. Liu and Q. Zhao, "Decentralized multi-armed bandit with multiple distributed players," in *2010 Information Theory and Applications Workshop (ITA)*. IEEE, 2010, pp. 1–10.

[21] L. Besson and E. Kaufmann, "Multi-player bandits revisited," in *Algorithmic Learning Theory*. PMLR, 2018, pp. 56–92.

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

[23] 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*, vol. 37, no. 10, pp. 2350–2363, 2019.

[24] O. Avner and S. Mannor, "Multi-user lax communications: a multi-armed bandit approach," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.

[25] 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*. Springer, 2017, pp. 173–185.

[26] 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.

[27] 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*, vol. 29, no. 4, pp. 731–745, 2011.

[28] E. Boursier and V. Perchet, "Sic-mmab: synchronisation involves communication in multiplayer multi-armed bandits," in *Advances in Neural Information Processing Systems*, 2019, pp. 12 071–12 080.

[29] 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.

[30] P. Landgren, V. Srivastava, and N. E. Leonard, "Social imitation in cooperative multiarmed bandits: partition-based algorithms with strictly local information," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 5239–5244.

[31] ——, "Distributed cooperative decision-making in multiarmed bandits: Frequentist and bayesian algorithms," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 167–172.

[32] U. Madhushani, A. Dubey, N. Leonard, and A. Pentland, "One more step towards reality: Cooperative bandits with imperfect communication," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[33] L. Yang, Y.-Z. J. Chen, S. Pasteris, M. Hajiesmaili, J. Lui, D. Towsley *et al.*, "Cooperative stochastic bandits with asynchronous agents and constrained feedback," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[34] Z. Zou, A. Proutiere, and M. Johansson, "Online shortest path routing: The value of information," in *2014 American Control Conference*. IEEE, 2014, pp. 2142–2147.

[35] M. S. Talebi, Z. Zou, R. Combes, A. Proutiere, and M. Johansson, "Stochastic online shortest path routing: The value of feedback," *IEEE Transactions on Automatic Control*, vol. 63, no. 4, pp. 915–930, 2017.

[36] 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*, vol. 7, no. Jun, pp. 1079–1105, 2006.

[37] Y. Wan, W.-W. Tu, and L. Zhang, "Projection-free distributed online convex optimization with $o(\sqrt{T})$ communication complexity," in *International Conference on Machine Learning*. PMLR, 2020, pp. 9818–9828.
[38] L. Yang, Y.-Z. J. Chen, M. Hajiesmaili, J. Lui, and D. Towsley, "Distributed bandits with heterogeneous agents (technical report)," *arXiv*, 2022.
[39] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in applied mathematics*, vol. 6, no. 1, pp. 4–22, 1985.
[40] "Kaggle avito context ad clicks 2015," https://www.kaggle.com/c/avito-context-ad-clicks.