A Distributed Algorithm for Sequential Decision Making in Multi-Armed Bandit with Homogeneous Rewards*

Jingxuan Zhu Romeil Sandhu Ji Liu

***Abstract*— This paper studies a distributed multi-armed bandit problem over a network of N agents, each of which can communicate only with its neighbors, where neighbor relationships are described by a connected graph $\mathbb{G}$. Each agent makes a sequence of decisions on selecting an arm from M candidates, yet it only has access to local samples of the reward for each action, which is a random variable. A distributed upper confidence bound (UCB) algorithm is proposed for the agents to cooperatively learn the best decision. It is shown that when all the agents share a homogeneous distribution of each arm reward, the algorithm achieves guaranteed logarithmic regret for all N agents at the order of $O((1+2\rho_2)^2 \log T/N)$ when T is large, where ρ_2 denotes the second largest among the absolute values of all the eigenvalues of the Metropolis matrix of $\mathbb{G}$. A sufficient condition under which the proposed distributed algorithm learns faster than the centralized (single-agent) counterpart is provided. Simulations suggest that the algorithm also works for the case when the agents have heterogeneous observations of each arm reward.**

I. Introduction

Multi-armed bandit is a decision-making problem that is common in both engineering and natural systems [1]. In a classical multi-armed bandit problem, the decision maker chooses one arm at each time from a given set of arms, and gets a reward generated according to a random variable. As different arms have different expected rewards, the goal of the decision maker is to minimize its regret. The seminal work [2] derived lower and upper bounds on asymptotic regret on this bandit selection problem. A classical and elegant algorithm named UCB1 was proposed in [3] with finite-time analysis for i.i.d. bandits. The algorithm simply maintains an index for each arm that balances exploration and exploitation and the agent selects an arm at time t according to it. It has been proved that UCB1 achieves an $O(\log T)$ regret. It is nearly impossible to survey the entire bandit literature here.

Recently, multi-agent multi-armed bandit problems have attracted increasing attention and been studied in various settings [4]–[16]. For example, the work of [5]–[10] considered a "collision" setting where agents "collide" when they simultaneously pull the same arm in wireless cognitive network system, and the work of [11] focused on a privacy-preserving problem for data sharing. Another line of research studies consensus-based distributed multi-armed bandit problems. The work of [12] considered a setting where agents only communicate to recommend arms rather than exchange reward information. The work of [13] focused on communication cost and proposed communication-efficient protocols that achieve near optimal regret.

We are motivated by the work of [14], [15] which considered a distributed setting, where a network of agents are allowed to share information over a neighbor graph and cooperatively seek an optimal arm, and proposed two algorithms called coop-UCB and coop-UCB2. While coop-UCB requires global awareness on the total number of agents in the network and the spectral gap of the underlying neighbor graph, coop-UCB2 only requires the total number of agents, but with a significantly larger regret. This paper proposes a new distributed algorithm for the same setting, which is built on the classical UCB1 algorithm [3] and Metropolis algorithm [17]. We derive an $O(\log T)$ upper bound of regret for the proposed algorithm and show how connectivity of the network affects the final regret bound of each agent. More importantly, as shown by the simulations in Section V-B, our algorithm performs well even under a heterogeneous observation setting, whereas the existing distributed algorithms cannot be applied, which implies more promising applications of our algorithm.

Although in the homogeneous reward setting, each agent can independently learn the optimal option, distributed multi-armed bandit allows cooperative decision making and thus can collect more information at each time instant. We show that under certain connectivity condition, the proposed distributed algorithm learns faster than the centralized (single-agent) counterpart. Moreover, compared with the centralized model in which a center has access to all information, cooperative learning in a distributed manner is more fault-tolerant and privacy-preserving.

Our algorithm has various practical applications, such as recommendation systems, with arms representing different recommendation choices and rewards representing users' satisfactory level of the selected recommendation. By observing each user's satisfactory level of different recommendations along with the feedback of their friends (neighbors), the recommendation system can learn to select the best recommendation for the user. The algorithm can also be applied to clinical trials, where arms represents different treatments. The information sharing between hospitals can improve the efficiency of data collecting as more samples can be observed, while only allowing direct communication of "nearby" hospitals, instead of gathering the information

*Proofs of most assertions in this paper are omitted due to space limitations and will be given in an expanded version of the paper.

J. Zhu is with the Department of Applied Mathematics and Statistics at Stony Brook University (jingxuan.zhu@stonybrook.edu). R. Sandhu is with the Departments of Bioinformatics and Computer Science at Stony Brook University (romeil.sandhu@stonybrook.edu). J. Liu is with the Department of Electrical and Computer Engineering at Stony Brook University (ji.liu@stonybrook.edu).

altogether, can reduce the information gathering difficulty as well as the risk of information leaking.

II. Problem Formulation

Consider a network consisting of N agents (or players). For ease of presentation, we label the agents from 1 through N. The agents are not aware of such a global labeling, but can differentiate between their neighbors. The set of agents is denoted by $[N] = \{1, 2, \ldots, N\}$. All agents face a common set of M arms, denoted by $[M] = \{1, 2, \ldots, M\}$. At each discrete time $t \in \{1, 2, \ldots, T\}$, each agent i makes a decision on which arm to select from the M options, and the selected arm is denoted by $a_i(t)$. When agent i selects arm k, it collects a reward $X_{i,k}(t)$, we assume that $\{X_{i,k}(t)\}_{t=1}^{T}$ is a random process. We consider a homogeneous setting in which all $X_{i,k}(t)$, $i \in [N]$, share the same expectation μ_k for each arm k. Without loss of generality, we assume that $X_{i,k}(t)$ have bounded support $[0, 1]$ and that $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_M$, so arm 1 has largest reward mean μ_1.

Each agent can communicate only with its neighbors. Neighbor relations among the N agents are described by a simple, undirected, connected graph $\mathbb{G} = (\mathcal{V}, \mathcal{E})$, with vertices corresponding to agents and edges corresponding to neighbor relations. Thus, agent j is a neighbor of agent i whenever (i, j) is an edge in $\mathbb{G}$.

The distributed multi-armed bandit problem is to devise a distributed algorithm for each agent which will enable each agent i to minimize its regret, defined as

$$R_i(T) = T\mu_1 - \sum_{t=1}^{T} \mathbf{E}[X_{a_i(t)}],$$

with the goal of achieving at least $R_i(T) = o(T)$ (i.e., $R_i(T)/T \to 0$ as $T \to \infty$) for all $i \in [N]$.

Since all the agents share the same expectation μ_k for each arm k in such a homogeneous setting, each agent can learn the best decision independently, without communicating with its neighbors, by applying the classical single-agent UCB1 algorithm [3], which achieves an upper bound for the regret at $O(\log T)$. We will present a cooperative multi-agent algorithm in the sequel and show that under certain connectivity condition, the cooperative algorithm can learn faster than UCB1.

III. Algorithm

To describe our algorithm, we begin with some notation.

Let $n_{i,k}(t)$ be the number of times agent i pulls arm k by time t. Let $\mathbb{1}$ be the indicator function that returns 1 if the statement is true and 0 otherwise. Define

$$\tilde{X}_{i,k}(t) = \frac{\sum_{\tau=1}^{t} \mathbb{1}(a_i(\tau) = k) X_{i,k}(\tau)}{n_{i,k}(t)}, \qquad (1)$$

which represents the average reward that agent i received from arm k till time t. Define

$$\Delta_k = \mu_1 - \mu_k, \quad \forall k,$$

as the gap of mean reward between arm 1 and arm k.

Each agent i has control over two variables, $\vartheta_{i,k}(t)$ and $\tilde{n}_{i,k}(t)$, which are updated as follows:

$$\vartheta_{i,k}(t+1) = \sum_{j \in [N]} w_{ij} \vartheta_{j,k}(t) + \tilde{X}_{i,k}(t+1) - \tilde{X}_{i,k}(t), \qquad (2)$$

$$\tilde{n}_{i,k}(t+1) = \max\{n_{i,k}(t), \tilde{n}_{j,k}(t), j \in \mathcal{N}_i\}, \qquad (3)$$

where $\mathcal{N}_i$ denotes the set of neighbors of agent i, and w_{ij} are the Metropolis weights used in the Metropolis algorithm [17] and defined as

$$w_{ij} = 0, \quad j \notin \mathcal{N}_i,$$
$$w_{ij} = \frac{1}{1 + \max\{|\mathcal{N}_i|, |\mathcal{N}_j|\}}, \quad j \in \mathcal{N}_i,$$
$$w_{ii} = 1 - \sum_{j \in \mathcal{N}_i} \frac{1}{1 + \max\{|\mathcal{N}_i|, |\mathcal{N}_j|\}}.$$

Here $|\mathcal{N}_i|$ equals the number of neighbors of agent i, or equivalently, the degree of vertex i in $\mathbb{G}$. It is worth emphasizing that $\vartheta_{i,k}(t)$ and $\tilde{n}_{i,k}(t)$ are updated in a distributed manner as they only use information from their neighbors. We will show later that $\tilde{n}_{i,k}(t)$ and $\vartheta_{i,k}(t)$ are agent i's estimates of $\max_{j \in [N]} n_{j,k}(t)$, which stands for the maximal number of pulls on arm k till time t, and reward mean μ_k, respectively.

Let $\vartheta_k(t)$ and $\tilde{X}_k(t)$ be the column stacks of all $\vartheta_{i,k}(t)$ and $\tilde{X}_{i,k}(t)$, respectively. Then, the N equations in (2) can be combined as

$$\vartheta_k(t+1) = W\vartheta_k(t) + \tilde{X}_k(t+1) - \tilde{X}_k(t),$$

where W is the $N \times N$ Metropolis matrix of the neighbor graph $\mathbb{G}$, which is a symmetric doubly stochastic matrix and whose ijth entry equals w_{ij}.

A detailed description of our algorithm, named Dist-UCB, is given as follows.

Initialization: At time $t = 0$, each agent i samples each arm k exactly once, sets $n_{i,k}(0) = 1$, $v_{i,k}(0) = \tilde{X}_{i,k}(0) = X_{i,k}(0)$, and $\tilde{n}_{i,k}(0) = 1$.

At each $t \in \{0, 1, \ldots, T\}$, each agent i performs the steps enumerated below in the order indicated.

1) **Transmission:** Agent i transmits $\tilde{n}_{i,k}(t)$ and $\vartheta_{i,k}(t)$ to each of its neighbours $j \in \mathcal{N}_i$; at the same time, agent i receives $\tilde{n}_{j,k}(t)$ and $\vartheta_{j,k}(t)$ from each of its neighbors $j \in \mathcal{N}_i$.
2) **Decision Making:** Agent i computes $\tilde{n}_{i,k}(t+1)$ according to (3).
   a) If there is no arm k such that $n_{i,k}(t) \leq \tilde{n}_{i,k}(t+1) - N$, agent i computes the index
   $$Q_{i,k}(t+1) = \vartheta_{i,k}(t) + C_{i,k}(t),$$
   where $C_{i,k}(t)$ is the corresponding upper confidence bound, and then pulls the arm that maximizes $Q_{i,k}(t+1)$.
   b) If there exists at least one arm k such that $n_{i,k}(t) \leq \tilde{n}_{i,k}(t+1) - N$, then agent i randomly pulls one such arm.

3) **Updating:** Agent i computes $\tilde{X}_{i.k}(t+1)$ and $\vartheta_{i,k}(t+1)$ according to (1) and (2), respectively, and updates all $n_{i,k}(t+1)$, $k \in [M]$, as follows:

$$n_{i,a_i(t+1)}(t+1) = n_{i,a_i(t+1)}(t) + 1,$$
$$n_{i,k}(t+1) = n_{i,k}(t), \quad k \neq a_i(t+1).$$

For a concise presentation of the algorithm, we refer to the pseudocode in Algorithm 1.

Algorithm 1: `Dist-UCB`

Input: $\Delta_k, \mathbb{G}, T, C_{i,k}(t)$
Output: $R_i(T)$

Initialization Each agent samples each arm exactly once. Initialize $v_{i,k}(0) = \tilde{X}_{i,k}(0) = X_{i,k}(0), \tilde{n}_{i,k}(0) = n_{i,k}(0) = 1$
for $t = 0, \ldots, T$ **do**
  $\mathcal{A}_i = \varnothing$
  Agent i sends $\tilde{n}_{i,k}(t)$ and $\vartheta_{i,k}(t)$ to $j \in \mathcal{N}_i$
  Agent i receives $\tilde{n}_{j,k}(t), \vartheta_{j,k}(t)$ from $j \in \mathcal{N}_i$
  $\tilde{n}_{i,k}(t+1) = \max\{(n_{i,k}(t)), \tilde{n}_{j,k}(t), j \in \mathcal{N}_i\}$
  if $n_{i,k}(t) \leq \tilde{n}_{i,k}(t+1) - N$ **then**
    Agent i puts k into a set $\mathcal{A}_i$
  end
  if $\mathcal{A}_i = \emptyset$ **then**
    for $k = 1, \ldots, M$ **do**
      $Q_{i,k}(t+1) = \vartheta_{i,k}(t) + C_{i,k}(t)$
    end
    $a_i(t+1) = \arg\max_k Q_{i,k}(t+1)$
  else
    $a_i(t+1)$ is randomly chosen from $\mathcal{A}_i$
  end
  $n_{i,k}(t+1) = n_{i,k}(t), \ \forall k \in [M]$
  $n_{i,a_i(t+1)}(t+1) = n_{i,a_i(t+1)}(t) + 1$
  $\vartheta_{i,k}(t+1) = \sum_{j=1}^{N} w_{ij}\vartheta_{j,k}(t) + \tilde{X}_{i,k}(t+1) - \tilde{X}_{i,k}(t)$
end
Return $R_i(T) = \sum_{\Delta_k > 0} \Delta_k n_{i,k}(T)$

According to Dist-UCB, when $N = 1$, there is no need to transmit information and the decision making step in our algorithm will be simplified to agent pulling the maximum index $Q_{i,k}(t+1) = \vartheta_{i,k}(t) + C_{i,k}(t)$. Since this $\vartheta_{i,k}(t) = \tilde{X}_{i,k}(t)$ according to (2) and $C_{i,k}(t) = \sqrt{\frac{2\log t}{n_{i,k}(t)}}$ as ρ_2 is defined as 0 when $N = 1$. It is not hard to see that our algorithm is exactly the same as UCB1 [3] when $N = 1$, thus so is the regret bound.

The main result of this paper is as follows whose proof is given in the next section.

Theorem 1: For the Dist-UCB algorithm with bounded rewards over $[0, 1]$ and

$$C_{i,k}(t) = \sqrt{\frac{2(1+2\rho_2)^2 \log t}{N n_{i,k}(t)}},$$

the regret of each agent i until time T satisfies

$$R_i(T) \leq \sum_{k>1} \left(\max\left\{ L, (3M+1)N, \frac{8(1+2\rho_2)^2}{N\Delta_k^2} \log T \right\} + \frac{2\pi^2}{3} \right) \Delta_k, \tag{4}$$

where ρ_2 is the second largest among the absolute values of all the eigenvalues of the Metropolis matrix W, and L is the smallest value such that when $t \geq L$, there holds

$$3\rho_2^{\frac{t}{N}} \leq \frac{\rho_2}{2Nt}.$$

It can be seen from the definition of L that it tends to be large when ρ_2 is close to 1, tends to be small when ρ_2 is close to 0, and would collapse to 0 when $\rho_2 = 0$.

To the best of our knowledge, there is no existing work that shows a direct relation between the regret bound and graph connectivity. From (4), it is easy to see the smaller ρ_2 is, the tighter is our regret bound. Since a smaller ρ_2 generally indicates a more connected graph, Theorem 1 implies that the more connected the neighbor graph $\mathbb{G}$ is, the tighter is each agent's regret bound, which is consistent with the intuition.

Consider the situation when T is sufficiently large, the asymptotic regret bound of agent i would be

$$R_i(t) \leq \sum_{k>1} \left(\frac{8(1+2\rho_2)^2}{N\Delta_k} + o(T) \right) \log T.$$

Comparing it with the regret bound when each agent independently applies single-agent UCB1, which is $\sum_{k>1} \left(\frac{8}{\Delta_k} + o(T) \right) \log T$, it is not hard to see that when $\rho_2 < (\sqrt{N} - 1)/2$, our regret bound is smaller. Since ρ_2 is always smaller than 1 as $\mathbb{G}$ is connected, we are able to conclude that when $N > 9$, our asymptotic regret bound is always better than that of UCB1, which implies that our distributed algorithm learns faster than the single-agent UCB1 for large-scale networks.

While Theorem 1 shows the relationship between graph connectivity and the regret bound, it requires each agent i to know ρ_2, which is global information. It is possible to relax the usage of ρ_2 and get a relatively looser bound for the regret, as shown in the following theorem.

Theorem 2: For the Dist-UCB algorithm with bounded rewards over $[0, 1]$ and

$$C_{i,k}(t) = \sqrt{\frac{18 \log t}{N n_{i,k}(t)}},$$

the regret of each agent i until time T satisfies

$$R_i(T) \leq \sum_{k>1} \left(\max \left\{ L, (3M+1)N, \frac{72}{N\Delta_k^2} \log T \right\} + \frac{2\pi^2}{3} \right) \Delta_k.$$

A. Sub-Gaussian Distribution

So far, we have assumed reward distributions to be bounded over $[0, 1]$, for the purpose of being consistent with the setting in the single-agent UCB1. It is worth emphasizing that our Dist-UCB algorithm can be easily extended to sub-Gaussian distributions, as the key step in the proof of Theorem 1, the tail bound inequality (5), holds for all sub-Gaussian random variables. Toward this end, let the optimal variance proxy of reward distribution $X_{i,k}(t)$ be no larger than σ^2. Then, we have the following result.

Theorem 3: For the Dist-UCB algorithm with sub-Gaussian random rewards, whose optimal variance proxies are no larger than σ^2, and

$$C_{i,k}(t) = \sigma \sqrt{\frac{2(1+2\rho_2)^2 \log t}{N n_{i,k}(t)}},$$

the regret of agent i until time T satisfies

$$R_i(T) \leq \sum_{k>1} \left(\max \left\{ L, (3M+1)N, \frac{8\sigma^2(1+2\rho_2)^2}{N\Delta_k^2} \log T \right\} + \frac{2\pi^2}{3} \right) \Delta_k.$$

B. A Heterogeneous Observation Setting

In more general situations, when agent i selects arm k and collects a reward $X_k(t)$, the agent may not be able to observe the exact reward; instead, it observes a biased "noisy" copy, $X_{i,k}(t)$, which is also a random variable. In this situation, to estimate the exact reward, agents need to cooperate with each other. We assume that $X_k(t)$ and $X_{i,k}(t)$ are two i.i.d. random processes. Define μ_k and $\mu_{i,k}$ as the expectation of $X_k(t)$ and $X_{i,k}(t)$, respectively, and assume they satisfy

$$\mu_k = \frac{1}{N} \sum_{i=1}^{N} \mu_{i,k},$$

which has a social meaning that the actual reward can be obtained by averaging among agents to cancel out the local bias. We also assume, without loss of generality, that $X_k(t), X_{i,k}(t)$ have bounded support and that $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_M$. Each agent i's goal is also to minimize the "actual" regret which is defined as

$$R_i(T) = T\mu_1 - \sum_{t=1}^{T} \mathbf{E}\left(X_{a_i(t)}\right)$$

for all $i \in [N]$.

While our analysis only shows the regret bound of homogeneous setting, we will show later in Section V-B that our algorithm also performs well in the heterogeneous observation setting.

IV. Analysis

To prove Theorem 1, we need the following concept.

A. Sub-Gaussian Random Variables

A random variable X with $\mu = \mathbb{E}[X]$ is called σ^2 sub-Gaussian if there is a positive σ such that

$$\mathbf{E}(e^{\lambda(X-\mu)}) \leq e^{\frac{\sigma^2\lambda^2}{2}}, \ \forall \lambda \in \mathbb{R},$$

where such σ^2 is called a variance proxy, and the smallest variance proxy is called the optimal variance proxy. Sub-Gaussian random variables have the following properties.

Property 1: (Inequality) A sub-Gaussian random variable X satisfies

$$\mathbb{P}(X - \mu \geq a) \leq e^{-\frac{a^2}{2\sigma^2}},$$
$$\mathbb{P}(\mu - X \geq a) \leq e^{-\frac{a^2}{2\sigma^2}}.$$

Property 2: (Sufficient condition) If X is a random variable with finite mean μ and $a \leq X \leq b$ almost surely, then X is $\frac{(b-a)^2}{4}$ sub-Gaussian.

Property 3: (Additivity) If X_1 is σ_1^2 sub-Gaussian and for $2 \leq i \leq n$, $(X_i|X_1, \ldots, X_{i-1})$ is σ_i^2 sub-Gaussian with σ_i being free of $X_1, \ldots, X_{i-1}$, then $X_1 + \cdots + X_i$ is sub-Gaussian with $\sigma_1^2 + \cdots + \sigma_i^2$ being one of its variance proxy.

B. Proof of Theorem 1

Denote $d_{i,j}$ as the distance between agents i and j, $\forall\ i, j \in [N]$, with $d_{i,i}$ being naturally defined as 0. It is clear that $d_{i,j} < N$ for a connected graph.

Lemma 1: For any $i \in [N]$ and $k \in [M]$,

$$\tilde{n}_{i,k}(t+1) = \max_{j \in [N]} \left\{ n_{j,k}(t - d_{i,j}) \right\}.$$

Lemma 2: For all $i \in [N]$ and $k \in [M]$, there holds $n_{i,k}(t) > \tilde{n}_{i,k}(t+1) - 3MN$.

Lemma 3: When $n_{i,k}(t) \geq (3M+1)N$, for all $i \in [N], k \in [M]$, we have $\max_{j \in [N]} n_{j,k}(t) \leq 2n_{i,k}(t)$.

Lemma 4: W^t converges to $\frac{1}{N}\mathbf{1}\mathbf{1}^\top$ as t goes to infinity, and $\left|[W^t]_{ij} - \frac{1}{N}\right| < \rho_2^t$ for all $i, j \in [N]$, where $\mathbf{1}$ denotes the vector whose entries are all equal to 1.

Proposition 1: When $n_{i,k}(t) \geq \max\{L, (3M+1)N\}$, the optimal variance proxy of $\vartheta_{i,k}(t)$ is no larger than $\frac{(1+2\rho_2)^2}{2Nn_{i,k}(t)}$.

Now we are in a position to prove Theorem 1.

Proof of Theorem 1: From Section IV-A and Proposition 1, when $n_{i,k}(t) \geq \max\{L, (3M+1)N\}$,

$$\begin{aligned} &\mathbb{P}\left(\vartheta_{i,k}(t) - \mu_k \geq \sqrt{\frac{2(1+2\rho_2)^2}{N n_{i,k}(t)}} \right) \\ &\leq \exp\left(-\frac{2(1+2\rho_2)^2}{2N n_{i,k}(t) \sigma_{i,k}^2(t)} \right) \\ &\leq \frac{1}{t^2}. \end{aligned} \tag{5}$$

Similarly,

$$\mathbb{P}\left(\mu_k - \vartheta_{i,k}(t) \geq \sqrt{\frac{2(1+2\rho_2)^2}{N n_{i,k}(t)}}\right) \leq \frac{1}{t^2}.$$

Let us go back to the algorithm, and let

$$C_{i,k}(t) = \sqrt{\frac{2(1+2\rho_2)^2 \log t}{N n_{i,k}(t)}}.$$

The UCB algorithm requires, if at time t, agent i chooses arm k instead of the optimal arm 1, there are only four possible cases:

1. $k \in \mathcal{A}_i$
2. $\vartheta_{i,k}(t) - \mu_k \geq C_{i,k}(t)$
3. $\mu_1 - \vartheta_{i,1}(t) \geq C_{i,1}(t)$
4. $\mu_1 - \mu_k < 2C_{i,k}(t)$

It is easy to verify that when

$$n_{i,k}(t) \geq \frac{8(1+2\rho_2)^2}{N\Delta_k^2} \log t,$$

case 4 does not hold. We define t' as the time such that

$$n_{i,k}(t') = \max\left\{L, (3M+1)N, \frac{8(1+2\rho_2)^2}{N\Delta_k^2} \log T\right\},$$

then

$$\begin{aligned}
&\sum_{t>t'} \mathbb{P}(\vartheta_{i,k}(t) - \mu_k \geq C_{i,k}(t)) + \mathbb{P}(\mu_1 - \vartheta_{i,1}(t) \geq C_{i,1}(t)) \\
\leq &\sum_{t>t'} \frac{2}{t^2} = \frac{\pi^2}{3},
\end{aligned}$$

which means after t', the average number of pulls of agent i on arm k due to case 2 and case 3 is no larger than $\frac{\pi^2}{3}$, consequently, the average number of pulls due to case 4 is also no larger than $\frac{\pi^2}{3}$. Thus, we have

$$\begin{aligned}
\mathbf{E}(n_{i,k}(T)) &\leq \mathbf{E}(n_{i,k}(T)|T > t') \\
&= n_{i,k}(t') + \frac{\pi^2}{3} \cdot 2 \\
&= \max\left\{L, (3M+1)N, \frac{8(1+2\rho_2)^2}{N\Delta_k^2} \log T\right\} + \frac{2\pi^2}{3}
\end{aligned}$$

for all $i \in [N], k \in [M]$.

Now we can get an upper bound of agent i's regret by following its definition:

$$\begin{aligned}
R_i(T) &= T\mu_1 - \sum_{t=1}^{T} \mathbf{E}(X_{a_i(t)}) \\
&= \sum_{k>1} \mathbf{E}(n_{i,k}(T)) \cdot \Delta_k \\
&\leq \sum_{k>1} \left(\max\left\{L, (3M+1)N, \frac{8(1+2\rho_2)^2}{N\Delta_k^2} \log T\right\} \right. \\
&\qquad \left. + \frac{2\pi^2}{3}\right) \Delta_k,
\end{aligned}$$

which completes the proof.

V. Simulations

In this section, we elucidate the above analysis with numerical experiments. First we compare our result with the classical UCB1 [3] using our original (homogeneous) setting, then change the reward distribution to a heterogeneous observation setting mentioned in Section III-B and test the performance of our algorithm under it.

A. Homogeneous Reward Setting

In the simulation below, we consider a multi-armed bandit problem with 20 arms and 40 agents, the reward distribution $X_{i,k}(t)$ is bounded over $[0, 1]$. We average the results of 50 Monte-Carlo runs to compare the average regret of agents using our algorithm to that of each agent independently applying single-agent UCB1 [3]. ρ_2 of the generated agents graph is 0.4345, and the total time T is chosen to be 10000.

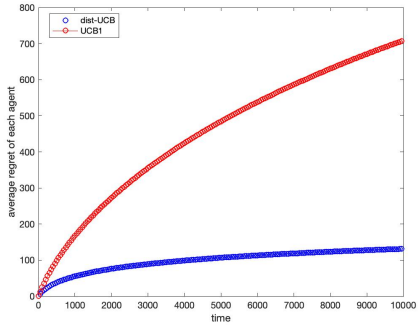

Fig. 1: simulation results comparing average regret for agents using Dist-UCB and UCB1 when $\rho_2 = 0.4345$

According to Theorem 1, when ρ_2 is very close to 1, L is possible to be the dominant term in finite time structure and might be very large, thus resulting a possible larger regret compared with that when using UCB1 within finite time analysis, our simulation still shows good result though. Below is the simulation result under the same setting above, except $\rho_2 = 0.9640$.

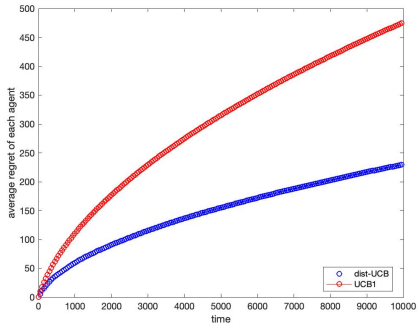

Fig. 2: simulation results comparing average regret for agents using Dist-UCB and UCB1 when $\rho_2 = 0.9640$

B. Heterogeneous Reward Setting

The result in this section is based on the heterogeneous observation setting introduced in Section III-B. In the simulation below, we consider a multi-armed bandit problem with 20 arms and 40 agents, the reward distribution $X_{i,k}(t)$ is bounded over $[0, 5]$ with different expectations among agents. We average the results of 50 Monte-Carlo runs to compare the average actual regret of agents using our algorithm and coop-UCB2 [15]. ρ_2 of our generated agents graph is 0.4813, and the total time T is chosen to be 100000.

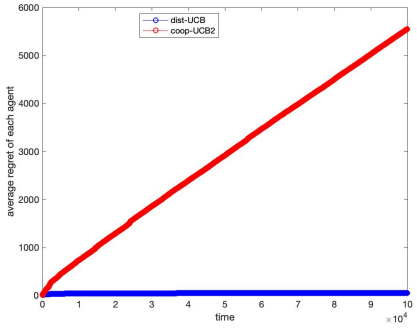

Fig. 3: simulation results comparing average regret for agents using Dist-UCB and coop-UCB2 [15] under heterogeneous observation setting

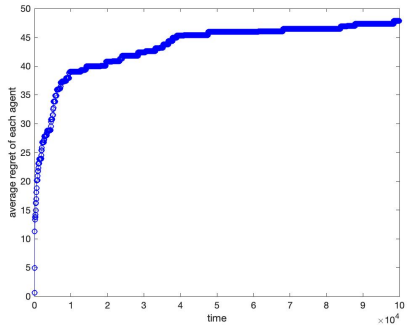

Fig. 4: simulation results of average regret for agents using Dist-UCB under heterogeneous observation setting

It is clearly shown in the simulation that coop-UCB2 is not able to handle the problem while Dist-UCB still performs quite well under heterogeneous observation setting. And to our knowledge, Dist-UCB is the first algorithm that can handle heterogeneous observation.

VI. Conclusion

In this paper, we have designed a novel algorithm to estimate the mean reward in the distributed multi-armed bandit problem. We have proved a logarithmic bound for expected cumulative regret of each agent, which indicates how graph connectivity is related with this regret. We have showed in experiments that our algorithm can also be applied to the setting where agents have heterogeneous observations of rewards. Our algorithm requires global information, the number of agents N. In future work, we aim to get around this global information and do more explorations to the heterogeneous setting, including its practical significance and detailed analysis.

References

[1] Djallel Bouneffouf and Irina Rish. A survey on practical applications of multi-armed and contextual bandits. *arXiv preprint*, 2019. arXiv:1904.10040.

[2] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[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] Mithun Chakraborty, Kai Yee Phoebe Chua, Sanmay Das, and Brendan Juba. Coordinated versus decentralized exploration in multi-agent multi-armed bandits. In *IJCAI*, pages 164–170, 2017.

[5] N. Nayyar, D. Kalathil, and R. Jain. On regret-optimal learning in decentralized multi-player multi-armed bandits. *IEEE Transactions on Control of Network Systems*, 2016.

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

[7] K. Liu and Q. Zhao. Distributed learning in cognitive radio networks: Multi-armed bandit with distributed multiple players. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3010 – 3013, 2010.

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

[9] Ilai Bistritz and Amir Leshem. Distributed multi-player bandits - a game of thrones approach. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7222–7232. Curran Associates, Inc., 2018.

[10] L. Lai, H. Jiang, and H. V. Poor. Medium access in cognitive radio networks: A competitive multi-armed bandit framework. In *42nd Asilomar Conference on Signals, Systems and Computers*, pages 98–102, 2008.

[11] Aristide CY Tossou and Christos Dimitrakakis. Differentially private, multi-agent multi-armed bandits. In *European Workshop on Reinforcement Learning (EWRL)*, 2015.

[12] Abishek Sankararaman, Ayalvadi Ganesh, and Sanjay 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.

[13] Yuanhao Wang, Jiachen Hu, Xiaoyu Chen, and Liwei Wang. Distributed bandit learning: Near-optimal regret with efficient communication. In *International Conference on Learning Representations*, 2020.

[14] P. Landgren, V. Srivastava, and N. E. Leonard. On distributed cooperative decision-making in multiarmed bandits. In *Proceedings of the European Control Conference*, 2016.

[15] P. Landgren, V. Srivastava, and N. E. Leonard. Distributed cooperative decision-making in multiarmed bandits: Frequentist and bayesian algorithms. In *Proceedings of the 55th IEEE Conference on Decision and Control*, pages 167–172, 2016.

[16] David Martínez-Rubio, Varun Kanade, and Patrick Rebeschini. Decentralized cooperative stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 4531–4542, 2019.

[17] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *International Symposium on Information Processing in Sensor Networks*, pages 63–70, 2005.