2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Decentralized Multi-Armed Bandit Can Outperform Classic Upper Confidence Bound: A Homogeneous Case over Strongly Connected Graphs

Jingxuan Zhu Ji Liu

Abstract— **This paper studies a homogeneous decentralized multi-armed bandit problem, in which a network of multiple agents faces the same set of arms, and each agent aims to minimize its own regret. A fully decentralized upper confidence bound (UCB) algorithm is proposed for a multi-agent network whose neighbor relations are described by a directed graph. It is shown that the decentralized algorithm guarantees each agent to achieve a lower logarithmic asymptotic regret compared to the classic UCB algorithm, provided the neighbor graph is strongly connected. The improved asymptotic regret upper bound is reciprocally related to the maximal size of a local neighborhood within the network. The roles of graph connectivity, maximum local degree, and network size are analytically elucidated in the expression of regret.**

I. Introduction

Multi-armed bandit (MAB) is a basic yet fundamental reinforcement learning problem, with a wide range of practical applications in natural and engineered systems. These applications include clinical trials, adaptive routing, cognitive radio networks, and online recommendation systems [2]. The problem has various formulations. In a classical and conventional MAB problem setting, a single decision maker (or player) sequentially selects one arm from a given finite set of arms (or choices) at each discrete time. Subsequently, the decision maker receives a reward corresponding to the chosen arm, which is generated according to a random variable with an unknown distribution. In general, different arms have different distributions and reward means. The goal of the decision maker is to minimize its cumulative (expected) regret, namely the difference between the decision maker's accumulated (expected) reward and the maximum which could have been obtained had the reward information been known. For this conventional MAB problem, both lower and upper bounds on the asymptotic regret were derived in the seminal work [3]. Additionally, classic UCB algorithms, known as UCB1 and UCB2, were proposed in [4], which achieve an asymptotic $O(\log T)$ regret. Due to the extensive study of multi-armed bandit problems over decades, it is impossible to survey the entire literature here. For an introductory survey for MAB, see a recent book [5].

*The proofs of all assertions in this paper are omitted due to space limitations and can be found in the arXiv version of the paper [1].

The work of J. Liu was supported by the National Science Foundation (NSF) under grant 2230101. J. Zhu is currently with Zhejiang Lab and was previously affiliated with the Department of Applied Mathematics and Statistics at Stony Brook University. The majority of the work was completed while J. Zhu was at Stony Brook University. J. Liu is with the Department of Electrical and Computer Engineering at Stony Brook University (ji.liu@stonybrook.edu).

Over the past years, there has been increasing interest to extend conventional single-player bandit settings to multi-player frameworks.

Multi-agent MAB problems have been studied in various settings [6]–[23], to name a few. For example, [6], [7], [10], [24] preclude communications among agents but allow them to receive "collision" signals when more than one agent selects the same arm, which has applications in wireless communication and cognitive radio. A distributed setting with a central controller is studied in [14], [19] in a federated learning context. Other federated bandit settings are considered in [13], [20], [25] with additional focus on theoretical privacy preservation.

Among all the existing multi-agent settings, we are motivated by a cooperative setting which makes use of a consensus process [26] among all agents. Such a cooperative setting was first proposed in [17] with homogeneous reward distributions, that is, all agents share the same distribution of each arm's reward.

A. Problem Formulation

As mentioned in the introduction, we are interested in a decentralized multi-armed bandit problem formulated as follows. Consider a multi-agent network consisting of N agents (or players). For presentation purposes, we label the agents from 1 through N. It is worth emphasizing that the agents are not aware of such a global labeling, but each agent can differentiate between its neighbors. The neighbor relations among the N agents are described by a directed graph $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ with N vertices, where the vertex set $\mathcal{V} = [N] \triangleq \{1, 2, \ldots, N\}$ represents the N agents and the set of directed edges (or arcs) $\mathcal{E}$ depicts the neighbor relations Specifically, agent j is an in-neighbor of agent i if $(j, i) \in \mathcal{E}$, and similarly, agent k is an out-neighbor of agent i if $(i, k) \in \mathcal{E}$. Each agent can send information to its out-neighbors and receive information from its in-neighbors. Thus, the directions of edges represent the directions of information flow. For convenience, we assume that each agent is always an in- and out-neighbor of itself. We use $\mathcal{N}_i(t)$ and $\mathcal{N}_i^-(t)$ to denote the in- and out-neighbor set of agent i at time t, respectively, i.e., $\mathcal{N}_i(t) = \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$ and $\mathcal{N}_i^-(t) = \{k \in \mathcal{V} : (i, k) \in \mathcal{E}\}$. It is clear that $\mathcal{N}_i(t)$ and $\mathcal{N}_i^-(t)$ are nonempty as they both contain index i. Clearly, a directed graph $\mathbb{G}$ may allow uni-directional communication among the agents. In the case when (i, j) is an edge in $\mathbb{G}$ as long as (j, i) is an edge in the graph, $\mathbb{G}$ can be simplified to an undirected graph which only allows bi-directional communication.

All N agents face a common set of M arms (or decisions) which is denoted by $[M] \triangleq \{1, 2, \ldots, M\}$. At each discrete time $t \in \{0, 1, 2, \ldots, T\}$, each agent i makes a decision on which arm to select from the M choices, and the selected arm is denoted by $a_i(t)$. If agent i selects an arm k, it will receive a random reward $X_{i,k}(t)$. For each $i \in [N]$ and $k \in [M]$, $\{X_{i,k}(t)\}_{t=1}^{T}$ is an unknown i.i.d. random process. For each arm $k \in [M]$, all $X_{i,k}(t)$, $i \in [N]$, share the same expectation μ_k. It is worth emphasizing that this setting allows different agents to have different reward probability distributions for each arm, so long as their means are the same. Without loss of generality, we assume that all $X_{i,k}(t)$ have bounded support $[0, 1]$ and that $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_M$, which implies that arm 1 has the largest reward mean and thus is always an optimal choice.

The goal of the decentralized multi-armed bandit problem just described is to devise a decentralized algorithm for each agent in the network which will enable agent i to minimize its expected cumulative regret, defined as

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

at an order at least as good as $R_i(T) = o(T)$, i.e., $R_i(T)/T \to 0$ as $T \to \infty$, for all $i \in [N]$.

B. Related Work

The above homogeneous cooperative multi-agent MAB problem has recently attracted increasing attention and quite a few different consensus-based decentralized algorithms have been proposed and developed [17], [18], [25], [27]–[29]. Recently, cooperative multi-agent bandits have been extended to heterogeneous reward settings, wherein different agents may have distinct reward distributions and means for each arm. A heterogeneous decentralized problem was first proposed and solved in [25] using the idea of gossiping to improve communication efficiency and privacy. Its algorithm is "partially" decentralized because it relies on the network size. A fully decentralized algorithm was later designed in [30]. The work of [31] considers a heterogeneous setting but focuses on a complete graph, which implicitly allows each agent to collect all other agents' information. Another heterogeneous setting over complete graphs has been studied in [22], [23], where different agents are associated with distinct arm subsets or bandits.

This paper focuses on the homogeneous setting. Note that in the homogeneous reward distribution setting, each agent in a network actually can independently learn an optimal arm using any conventional single-agent UCB algorithm, ignoring any information received from other agents. Notwithstanding, all the existing algorithms for the decentralized multi-armed bandit problem with homogeneous reward distributions require that each agent be aware of certain network-wise global information, such as spectral properties of the underlying graph or total number of agents in the network, except for our earlier work [32]. Such a requirement leads to a counterintuitive observation: compared with the conventional single agent case, each agent in a multi-agent network can collect more arm-related information while its bandit learning becomes more restrictive or less independent.

Even though [32] shows that collaborating with neighbors can improve regret bounds compared to the classic single-agent UCB1 [4], it does not exhibit any specific dependence on the network or neighborhood. This means that the incentive for collaboration is not reflected in the learning rates. In other words, a network- and neighborhood-independent improvement fails to capture the gain an agent should experience if it has more neighbors with which to communicate. Intuitively, a lower regret should be attainable when more neighbors are available. Another limitation of the algorithm in [32] is that it only works for undirected graphs. The incentive issue may be resolved by applying the heterogeneous decentralized algorithm in [30] to the homogeneous case. However, it relies on a doubly stochastic consensus update matrix and thus implicitly requires that the underlying graph be undirected or weight-balanced[1]. It is unclear how its algorithm design can be applied to general weight-unbalanced directed graphs. The algorithm in [34], to our knowledge, is the only one in the literature crafted for general directed graphs, but it requires each agent be aware of the network size N.

C. Contribution

With the preceding discussion in mind, this paper proposes a new fully decentralized UCB algorithm for the homogeneous setting over general directed neighbor graphs, without using any network-wide information. The proposed decentralized algorithm not only outperforms its classic single-agent counterpart, UCB1, ensuring that each agent in the network achieves an improved asymptotic regret upper bound, but also guarantees that this bound is reciprocally related to the maximal size of a neighborhood within the network, provided the neighbor graph is strongly connected. This contribution incentivizes collaboration among all neighboring agents in any strongly connected network. In particular, the established asymptotic regret surpasses that of all existing fully decentralized cooperative algorithms. This includes algorithms described in [32] and [30] whose asymptotic regrets are $C \log T$, with the coefficient C being either a constant or reciprocally related to each agent's local degree.

II. Algorithm

Before introducing the algorithm, we first articulate the technical challenges of algorithm design.

A. Technical Challenges

The variables mentioned here will be explained in detail in the next subsection. The primary technical challenge of decentralized bandit algorithm design lies in accurately estimating reward means in a multi-agent network. Each agent i iteratively updates its reward mean estimate $z_{i,k}(t)$ for each arm k. This estimation process is intrinsically tied

[1]A weighted directed graph is called weight-balanced if the sum of all in-weights equals the sum of all out-weights at each of its vertices [33].

to the agent's confidence level in arm k, as reflected by the variance of $z_{i,k}(t)$. It is worth emphasizing that, contrary to intuition, having access to more information does not necessarily guarantee improved accuracy in reward mean estimates. This is because the reward information propagated over the network does not consist of raw rewards. Instead, what is transmitted is a linear composition of one-time rewards at different agents over time. Therefore, accurately evaluating the confidence level of the reward mean estimate requires a careful design of auxiliary transmitted variables and their updating rules. This issue becomes even more challenging for general weight-unbalanced directed graphs.

The existing literature fails to resolve this issue in a fully decentralized manner, except for our earlier works [32] and [30], which are tailored solely for undirected graphs or special weight-balanced directed graphs, and thus rely on a doubly stochastic consensus update matrix. To address the challenge in general directed graphs, we appeal to the idea of push-sum [35] based on a column stochastic matrix. While this idea was previously applied in directed graphs in our earlier work [34], it required each agent to know the network size. Here we introduce a new local update $l_i(t)$ at each agent i to estimate the maximal neighborhood size. This fully decentralized process not only avoids the need for the network size but also, by incorporating $l_i(t)$ into the design of the upper confidence bound function $C_{i,k}(t)$, leads to a tighter asymptotic regret compared to those in [32] and [30].

Another challenge in decentralized bandit algorithm design is ensuring exploration consistency among different agents. This is because insufficient sampling of an arm by one agent can negatively affect the reward mean estimation of its out-neighbors, and through information propagation, ultimately hinders the estimation of all agents. Here we refine the operation of case b) in the Decision Making step, which enables agents to achieve faster exploration consistency compared to counterpart operations described in [30], [32], [34], as demonstrated in Lemma 8 compared to Lemma 6 in [30], [32], [34].

B. Algorithm Design

We next introduce some important variables to help present our algorithm.

Local sample counter and sampling estimate: Let $n_{i,k}(t)$ be the number of times agent i pulls arm k up to time t. Let $m_{i,k}(t)$ be agent i's estimate of the maximal sampling times of arm k up to time t over its neighborhood, which is updated as follows:

$$m_{i,k}(t+1) = \max\Big\{n_{i,k}(t+1),\ \max_{j\in\mathcal{N}_i} m_{j,k}(t)\Big\}. \quad (1)$$

The variable $m_{i,k}(t)$ and its update (1) help agent i keep track of the maximal sampling times of arm k among all those agents in the network that lie within the same connected component.

Local sample mean and reward mean estimate: Let $\bar{x}_{i,k}(t)$ be the sample mean, representing the average reward that agent i receives from arm k up to time t, which is updated as follows:

$$\bar{x}_{i,k}(t) = \frac{1}{n_{i,k}(t)} \sum_{\tau=0}^{t} \mathbb{1}(a_i(\tau)=k) X_{i,k}(\tau),$$

where $\mathbb{1}(\cdot)$ is the indicator function that returns 1 if the statement is true and 0 otherwise. Let $z_{i,k}(t)$ be agent i's estimate of the reward mean of arm k up to time t, which is updated, along with an auxiliary variable $y_i(t)$, as follows:

$$z_{i,k}(t+1) = \sum_{j\in\mathcal{N}_i} w_{ij} z_{j,k}(t) + \bar{x}_{i,k}(t+1) - \bar{x}_{i,k}(t), \quad (2)$$

$$y_i(t+1) = \sum_{j\in\mathcal{N}_i} w_{ij} y_j(t), \quad (3)$$

where $w_{ij} = 1/|\mathcal{N}_j^-|$ for all $j \in \mathcal{N}_i$. Let W be the $n \times n$ matrix whose ijth entry equals w_{ij} if $j \in \mathcal{N}_i$ and zero otherwise. It is easy to see that W is a column stochastic matrix whose zero and nonzero pattern are consistent with the neighbor graph $\mathbb{G}$. The updates (2) and (3) make use of the idea of push-sum [35], a clever approach to distributed averaging over directed graphs. The term $\bar{x}_{i,k}(t+1) - \bar{x}_{i,k}(t)$ can be regarded as a "coarse gradient", which requires the reward at time $t+1$. We also define $\tilde{z}_{i,k}(t) = z_{i,k}(t)/y_i(t)$ which will be used in the algorithm.

Local arm index set: Each agent i keeps and updates an arm index set $\mathcal{A}_i(t) = \{k \in [M] : n_{i,k}(t) < m_{i,k}(t)\}$ at each time t, which serves as the index collection of those arms that "fall behind" in exploring.

Local estimate of maximal neighborhood size: Each agent i maintains a variable $l_i(t)$ to estimate the maximal neighborhood size within the network, which is initialized as $l_i(0) = |\mathcal{N}_i \cup \mathcal{N}_i^-|$ and updated as

$$l_i(t+1) = \max_{j\in\mathcal{N}_i} l_j(t). \quad (4)$$

It is easy to see that if $\mathbb{G}$ is strongly connected, all $l_i(t)$, $i \in [N]$ will reach the maximum value, $\max_{i\in[N]} |\mathcal{N}_i \cup \mathcal{N}_i^-|$, in a finite number of time steps at $t = d(\mathbb{G})$, where $d(\mathbb{G})$ denotes the diameter of $\mathbb{G}$. It is worth emphasizing that each agent does not need to know when this update process stops.

Local upper confidence bound function: Each agent i needs to specify a design object in its local implementation, namely its upper confidence bound function, $C(t, n_{i,k}(t))$, which will be used to quantify agent i's belief on its estimate of arm k's reward mean. Upper confidence bound functions are critical in single-agent UCB algorithm design. Coordination among the agents allows us to design the following upper confidence bound function:

$$C_{i,k}(t) = \left(1 + \sqrt{\frac{l_i(t)}{n_{i,k}(t)}}\right) \sqrt{\frac{2\log t}{l_i(t) n_{i,k}(t)}} \quad (5)$$

which, as we will see, is "better" than that in the conventional single-agent UCB1.

A detailed description of the proposed decentralized UCB algorithm is presented as follows.

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

$z_{i,k}(0) = \bar{x}_{i,k}(0) = X_{i,k}(0)$, $y_i(0) = 1$, $l_i(0) = |\mathcal{N}_i \cup \mathcal{N}_i^-|$, and $C_{i,k}(0) = 0$.

Iteration: Between clock times t and $t+1$, $t \in \{0, 1, \ldots, T\}$, each agent i performs the steps enumerated below in the order indicated.

1) **Transmitting:** Agent i transmits $y_i(t)/|\mathcal{N}_i^-|$, $l_i(t)$, $z_{i,k}(t)/|\mathcal{N}_i^-|$ and $m_{i,k}(t)$ for each arm k, to its out-neighbors; simultaneously, agent i receives these quantities from each of its in-neighbors $j \in \mathcal{N}_i$.
2) **Decision Making:** Let $\mathcal{A}_i(t) = \{k \in [M] : n_{i,k}(t) < m_{i,k}(t)\}$. Agent i pulls exactly one arm according to the following rule:
   a) If $\mathcal{A}_i(t) = \emptyset$, agent i pulls arm
   $$a_i(t+1) = \arg\max_{k \in [M]} \left(\tilde{z}_{i,k}(t) + C_{i,k}(t)\right)$$
   with ties broken arbitrarily.
   b) If $\mathcal{A}_i(t) \neq \emptyset$, agent i pulls arm
   $$a_i(t+1) = \arg\max_{k \in \mathcal{A}_i(t)} \left(m_{i,k}(t) - n_{i,k}(t)\right)$$
   with ties broken arbitrarily.
3) **Updating:** Agent i first updates $n_{i,k}(t+1)$ and $\bar{x}_{i,k}(t+1)$ according to the Decision Making step, then updates $m_{i,k}(t+1)$, $z_{i,k}(t+1)$, $y_i(t+1)$, $l_i(t+1)$ according to (1)–(4), respectively, and finally sets $\tilde{z}_{i,k}(t+1) = z_{i,k}(t+1)/y_i(t+1)$.

For a concise presentation of the algorithm, we refer to the pseudocode below.

It is straightforward to verify that in the extreme single-agent case, namely when $N = 1$, the proposed decentralized multi-agent algorithm simplifies to the classic upper confidence bound algorithm, UCB1, as proposed in [4]. Therefore, we term it **Decentralized UCB1**.

Algorithm 1: Decentralized UCB1

Input: $\mathbb{G}, T, C(t, n_{i,k}(t))$

Initialization: Each agent i samples each arm k exactly once and sets $m_{i,k}(0) = n_{i,k}(0) = 1$, $\tilde{z}_{i,k}(0) = z_{i,k}(0) = \bar{x}_{i,k}(0) = X_{i,k}(0)$, $y_i(0) = 1$, $l_i(0) = |\mathcal{N}_i \cup \mathcal{N}_i^-|$, and $C_{i,k}(0) = 0$.

for $t = 0, \ldots, T$ **do**
  $\mathcal{A}_i(t) = \emptyset$
  for $k = 1, \ldots, M$ **do**
    if $n_{i,k}(t) < m_{i,k}(t)$ **then**
      Agent i adds index k into set $\mathcal{A}_i(t)$
    end
  end
  if $\mathcal{A}_i(t) = \emptyset$ **then**
    $a_i(t+1) = \arg\max_{k \in [M]} \left(\tilde{z}_{i,k}(t) + C_{i,k}(t)\right)$ // optimal arm in belief
  else
    $a_i(t+1) = \arg\max_{k \in \mathcal{A}_i(t)} \left(m_{i,k}(t) - n_{i,k}(t)\right)$ // for exploration consistency
  end
  Agent i transmits $y_i(t)/|\mathcal{N}_i^-|$, $l_i(t)$, $z_{i,k}(t)/|\mathcal{N}_i^-|$ and $m_{i,k}(t)$ for each arm k, to its out-neighbors // information transmission
  $n_{i,a_i(t+1)}(t+1) = n_{i,a_i(t+1)}(t) + 1$
  $n_{i,k}(t+1) = n_{i,k}(t)$, $k \neq a_i(t+1)$, $k \in [M]$
  $\bar{x}_{i,k}(t+1) = \frac{1}{n_{i,k}(t+1)} \sum_{\tau=0}^{t+1} \mathbb{1}(a_i(\tau) = k) X_{i,k}(\tau)$
  Agent i updates $m_{i,k}(t+1)$, $z_{i,k}(t+1)$, $y_i(t+1)$, $l_i(t+1)$ according to (1)–(4)
  $\tilde{z}_{i,k}(t+1) = \frac{z_{i,k}(t+1)}{y_i(t+1)}$, $k \in [M]$ // information updating
end

C. Main Results

To state the main result, we need the following notation. For each arm k, let $\Delta_k = \mu_1 - \mu_k$ denote the difference in reward means between arm k and the optimal arm. Let $l_{\max} = \max_{i \in [N]} |\mathcal{N}_i \cup \mathcal{N}_i^-|$ represent the largest local neighborhood size within the network. Let ρ_2 denote the second largest magnitude among all eigenvalues of the column stochastic matrix W. It is well know that $\rho_2 < 1$ if $\mathbb{G}$ is strongly connected, which follows as a direct consequence of the Perron-Frobenius Theorem (cf. Lemma 4).

Theorem 1: Suppose that all N agents adhere to Algorithm 1. If $\mathbb{G}$ is strongly connected, then for any $\epsilon > 0$, the regret of each agent i up to time T satisfies

$$R_i(T) \leq \sum_{k:\ \Delta_k > 0} \left(\frac{8(1+\epsilon)^2 \log T}{l_{\max} \Delta_k} + \Gamma_{i,k}(\epsilon, \mathbb{G}) \right), \quad (6)$$

where $\Gamma_{i,k}(\epsilon, \mathbb{G})$ is a constant defined in Remark 1.

Note that the upper regret bound in the above theorem is of the form $(1+\epsilon)^2 C_1 \log T + C_2(\epsilon)$, where C_1 and C_2 are two algorithm-dependent constants with the latter depending on an arbitrary positive ϵ. Such a similar form of upper regret bound is standard in the multi-armed bandit literature. Notable examples include the non-asymptotic bound of the KL-UCB algorithm, $(1+\epsilon)C_1 \log T + C_2(\epsilon) + C_3 \log(\log T)$, [36, Theorem 2] and the optimal problem-dependent bound of Thompson sampling, $(1+\epsilon)C_1 \log T + O(M\epsilon^{-2})$ [37, Theorem 1.1]. From the following remark, it will be easy to see that in the special single-agent case, namely $N = 1$, $\Gamma_{i,k}(\epsilon, \mathbb{G}) = O(M\epsilon^{-2})$, which is of the same order as Thompson sampling. From Theorem 1, it is easy to see that

$$R_i(T) \leq \inf_{\epsilon > 0} \sum_{k:\ \Delta_k > 0} \left(\frac{8(1+\epsilon)^2 \log T}{l_{\max} \Delta_k} + \Gamma_{i,k}(\epsilon, \mathbb{G}) \right),$$

which is a simple theoretical improvement, but without an explicit expression.

Remark 1: We define the constant $\Gamma_{i,k}(\epsilon, \mathbb{G})$ here. First note that the column stochastic matrix W, network size N, and $l_{\max}$ are uniquely determined by the neighbor graph $\mathbb{G}$. When $\mathbb{G}$ is strongly connected, $\rho_2 < 1$ and $\lim_{t \to} W^t = v\mathbf{1}'$, where $v \in \mathbb{R}^N$ is a positive stochastic vector and $\mathbf{1}$ denotes the vector in $\mathbb{R}^N$ whose entries all equal 1; furthermore, there exists a constant $\zeta > 0$ such that $|[W^t]_{ij} - v_i| \leq \zeta \rho_2^t$

for all $i, j \in [N]$, where v_i denotes the ith entry of v (cf. Lemma 4).[2] Let $\alpha_i(\mathbb{G})$ be the smallest value such that for all $x \geq \alpha_i(\mathbb{G})$, there holds[3]

$$\frac{-2\sqrt{x}\log x}{x\log\rho_2 + 2\log x} + \frac{2}{\sqrt{x}} \leq \frac{v_i\sqrt{l_{\max}}}{\zeta}, \tag{7}$$

Such $\alpha_i(\mathbb{G})$ must exist (cf. Lemma 1) and it is straightforward to verify that $\alpha_i(\mathbb{G}) = O(1+\log^{-2}\rho_2)$. Next let $\beta(\mathbb{G})$ be the unique solution of x to the following equation:

$$\frac{(x-(3N+1))^{\frac{3}{2}}(\sqrt{x}+\sqrt{l_{\max}})}{x^{\frac{3}{2}}(\sqrt{x-(3N+1)}+\sqrt{l_{\max}})} = \frac{7}{8}. \tag{8}$$

Such $\beta(\mathbb{G})$ must exist (cf. Lemma 2) and it is straightforward to verify that $\beta(\mathbb{G}) = O(N)$. It is clear that $\beta(\mathbb{G}) \geq 3N+1$. To proceed, let $c = \lceil(\log(1-\sqrt{6/7})v_i)/(\log\rho_2)\rceil$, where $\lceil\cdot\rceil$ denotes the ceiling function, and define

$$\gamma_i(\mathbb{G}) = 2\sum_{\tau=c}^{\infty}\tau^{-\frac{7}{2}\left(1-\frac{\rho_2^\tau}{v_i}\right)^2+2}, \tag{9}$$

which is of the order $O(1)$ (cf. Lemma 3). Then, $\Gamma_{i,k}(\epsilon,\mathbb{G})$ is defined as follows:

$$\Gamma_{i,k}(\epsilon,\mathbb{G}) = \Big(\max\Big\{\kappa_i(\mathbb{G}), \frac{l_{\max}}{\epsilon^2}, \frac{\log(1-\sqrt{\frac{6}{7}})v_i}{\log\rho_2}\Big\} + 2\kappa_i(\mathbb{G}) + 2\gamma_i(\mathbb{G}) + 3N - 1\Big)\Delta_k, \tag{10}$$

where $\kappa_i(\mathbb{G}) = \max\{\alpha_i(\mathbb{G}) + 3N + 1, \beta(\mathbb{G})\}$. It is worth emphasizing that, with $\alpha_i(\mathbb{G}) = O(1+\log^{-2}\rho_2)$, $\beta(\mathbb{G}) = O(N)$, and $\gamma_i(\mathbb{G}) = O(1)$, it is easy to see that $\Gamma_{i,k}(\epsilon,\mathbb{G}) = O(N\max\{\epsilon^{-2}, \log^{-2}\rho_2\})$. □

From Remark 1, it is clear that the constant term in each agent i's upper bound of the regret, $\sum_{k:\Delta_k>0}\Gamma_{i,k}(\epsilon,\mathbb{G})$, will increase if, while assuming other variables remain unchanged, N or ρ_2 increases.

It is well known that for the classic single-agent (non-cooperative) UCB1 algorithm, the upper bound on the agent's regret, denoted as $R(T)$, is $\sum_{k:\Delta_k>0}\left(\frac{8\log T}{\Delta_k} + (1+\frac{\pi^2}{3})\Delta_k\right)$ [4, Theorem 1], which implies $\lim_{T\to\infty}\frac{R(T)}{\log T} \leq \sum_{k:\Delta_k>0}\frac{8}{\Delta_k}$. From Theorem 1, since (6) holds for any $\epsilon > 0$, $\lim_{T\to\infty}\frac{R_i(T)}{\log T} < \sum_{k:\Delta_k>0}\frac{8}{l_{\max}\Delta_k}$. Since $\mathbb{G}$ is assumed to be strongly connected, $l_{\max} \geq 3$. We have thus proved the following result:

Corollary 1: The proposed decentralized multi-agent UCB1 achieves an improved per-agent asymptotic regret upper bound compared to its single-agent counterpart, with the improvement being inversely related to the maximal size of a local neighborhood within the multi-agent network, provided that the network is strongly connected.

The corollary immediately implies that a strongly connected network of multiple agents can collectively outperform the non-cooperative case.

[2]We sometimes use $[M]_{ij}$ to denote the ijth entry of a matrix M.

[3]We treat the rare extremal case $\rho_2 = 0$ with $\lim_{\rho\to 0^+}\log^{-1}\rho_2 = 0$.

Lower Bound: For any strongly connected graph, the proposed decentralized UCB1 algorithm has an asymptotic lower bound $\Omega((\log T)/N)$ on each agent's regret. The argument is as follows: Among all strongly connected graphs with N agents, a complete graph yields the lowest network regret, as each agent can access information from all other agents. This is essentially equivalent to a single-agent case in which the agent can pull N times at each time. It is well known that the asymptotic lower bound of the classic single-agent UCB1 algorithm, in which the agent pulls exactly once every time, is $\Omega(\log T)$ [3, Theorem 1]. Then, with pulling N times at each time, the regret lower bound becomes $\Omega(\log NT)$. Since this regret equals to the network regret of the complete graph case and all N agents are homogeneous, each agent's regret has a lower bound $(1/N)\Omega(\log NT)$, which asymptotically equals $\Omega((\log T)/N)$. This is in line with (6) for the complete graph case in which $l_{\max} = N$.

For any strongly connected graph, the iterative process described in (4) for estimating the maximal neighborhood size can be replaced by a flooding process to estimate the network size. In this flooding process, each agent iteratively collects the identity numbers of its in-neighbors and transmits all collected identity numbers to its out-neighbors. Consequently, all agents will have an accurate network size value in finite time. Using the same analysis, this approach will lead to a further improved regret upper bound by replacing $l_{\max}$ with N in (6), leading to an asymptotic bound of optimal order $O((\log T)/N)$. However, implementing this flooding idea requires a global unique identity and extensive storage and communication capacities at each agent.

Remark 2: If we divide time into communication epochs of fixed constant length and allow each agent to communicate and make arm decisions only at the start of each epoch, while continuing to select the same arm until the end of the epoch, we can reduce communication while experiencing only a constant-level increase in regret. □

III. Analysis

This section provides a comprehensive analysis of the proposed Decentralized UCB1 algorithm (excluding proofs) and a sketched proof of Theorem 1.

A. Properties of Constants

We begin with the following lemmas supporting the constant definitions in Remark 1.

Lemma 1: $\alpha_i(\mathbb{G})$ exists.

Lemma 2: Equation (8) has a unique minimum solution.

Lemma 3: $\gamma_i(\mathbb{G})$ defined in (9) is of the order $O(1)$.

B. Preliminary Results

We next introduce some preliminary results which are more or less well known and will be used later.

Lemma 4: If $\mathbb{G}$ is strongly connected, then $W_\infty \triangleq \lim_{t\to} W^t = v\mathbf{1}'$, where $v \in \mathbb{R}^N$ is a positive stochastic vector, and there exists a constant $\zeta > 0$ such that $|[W^t]_{ij} - v_i| \leq \zeta\rho_2^t$ for all $i, j \in [N]$.

Lemma 5: (Hoeffding's inequality [38, Theorem 2]) Let $\{X_1, \ldots, X_n\}$ be a finite set of independent random variables such that each X_i satisfies $X_i \in [a_i, b_i]$ and $\mathbf{E}(X_i) = \mu_i$. Then, for any $\eta \geq 0$,

$$\mathbf{P}\Big(\sum_{i=1}^{n} X_i - \sum_{i=1}^{n} \mu_i \geq \eta\Big) \leq \exp\Big(\frac{-2\eta^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\Big),$$
$$\mathbf{P}\Big(\sum_{i=1}^{n} X_i - \sum_{i=1}^{n} \mu_i \leq -\eta\Big) \leq \exp\Big(\frac{-2\eta^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\Big).$$

C. Simple Update Properties

We will also need the following simple properties respectively for updates (3) and (1).

Lemma 6: If $\mathbb{G}$ is strongly connected, then there exists a constant $\eta > 0$ such that $N \geq y_i(t) \geq \eta$ for all i and t.

It is easy to prove that $N \geq y_i(t)$. The above lemma is a special case of Corollary 2 (b) in [39].

For any $i, j \in [N]$, we use $d_{i,j}$ to denote the distance from vertex i to vertex j, which is defined as the number of directed edges in a shortest directed path from vertex i to vertex j in the neighbor graph $\mathbb{G}$, provided at least one such directed path exists. It is natural to define $d_{i,i} = 0$ for all $i \in [N]$. In the case when $\mathbb{G}$ is strongly connected, all $d_{i,j}$, $i, j \in [N]$ are well defined and it is easy to see that $d_{i,j} \leq d(\mathbb{G})$ for all $i, j \in [N]$. For the purpose of analysis, define $n_{i,k}(t) = m_{i,k}(t) = 0$ for all $i \in [N]$ and $k \in [M]$ when $t < 0$.

Lemma 7: For any $i \in [N]$, $k \in [M]$, and $t \geq 0$,

$$m_{i,k}(t) = \max_{j \in [N]} \{n_{j,k}(t - d_{j,i})\}.$$

The above lemma was proved in [30, Lemma 4].

D. Key Lemmas

We then present the following two key intermediate steps for analyzing the proposed algorithm.

The first key lemma guarantees that the difference in exploration times of each arm among different agents in the network is always bounded.

Lemma 8 (Exploration Consistency): $|n_{i,k}(t) - n_{j,k}(t)| \leq 3N + 1$ for any $i, j \in [N]$, $k \in [M]$, and $t > 0$.

The second key lemma establishes a tight error bound for each agent i's reward mean estimate for each arm k.

Lemma 9 (Estimation Confidence): If $n_{i,k}(t) \geq \alpha_i(\mathbb{G}) + 3N + 1$, then for any $i \in [N]$, $k \in [M]$, and $t > 0$,

$$z_{i,k}(t) \geq v_i \sum_{j \in [N]} \left(1 - \sqrt{\frac{l_{\max}}{n_{j,k}(t)}}\right) \bar{x}_{j,k}(t),$$
$$z_{i,k}(t) \leq v_i \sum_{j \in [N]} \left(1 + \sqrt{\frac{l_{\max}}{n_{j,k}(t)}}\right) \bar{x}_{j,k}(t).$$

where v_i and $\alpha_i(\mathbb{G})$ are defined in Remark 1.

E. Sketched Proof of Main Theorem

We are now in a position to outline the proof of Theorem 1.

Sketched Proof of Theorem 1: There are two key intermediate steps toward the analysis of the Decentralized UCB1 algorithm. The first key step is Lemma 8, which guarantees that the difference in exploration times of each arm among different agents in the network is always bounded. We call it the exploration consistency step. The proof of Lemma 8 makes use of Lemma 7, which establishes a property of the update (1) by expressing each agent i's local sampling estimate $m_{i,k}(t)$ in terms of all its in-neighbors' local sample counters. The other key step is to establish a tight error bound for $z_{i,k}(t)$, each agent i's reward mean estimate for each arm k. This is achieved through Lemma 9, which quantifies both lower and upper bounds via the maximal size of a local neighborhood within the network, $l_{\max}$, and the unique dominant eigenvector v of the push-sum update matrix W. We call it the estimation confidence step. The proof of Lemma 9 leverages the exploration step in Lemma 8, as well as the convergence property of the push-sum weight matrix W given in Lemma 4.

Equipped with the two key lemmas, we are able to prove the main theorem. To enhance the clarity and refinement of our analysis, we divide the proof of Theorem 1 into two parts. Part A analyzes the number of times agents choose sub-optimal arms when they execute case a) in the Decision Making step, while Part B focuses on case b). In Part A, we systematically evaluate each agent's confidence in its reward mean estimate of each arm for all possible random processes. This evaluation involves applying the "slicing" technique to the sample count, as shown in Equation (28) in [1]. This evaluation process also makes use of Lemma 6, which is a well-known property of the update (3) in push-sum. We then utilize Hoeffding's inequality (cf. Lemma 5) in conjunction with Lemma 9 in Equation (29) in [1]. This allows us to show that the expected sampling frequency of any sub-optimal arm is "negligible" after a sufficient number of samplings, as indicated in Equation (33) in [1]. This result not only establishes an upper bound for the number of sub-optimal samplings in Part A but also leads to a low frequency of sub-optimal samplings in Part B when combined with Lemma 8. We finally obtain the regret upper bound (6) by combining the findings from both Part A and Part B. ■

IV. Conclusion

In this paper, we have proposed a novel fully decentralized UCB algorithm for a homogeneous multi-agent multi-armed bandit problem, without using any network-wide information. It has been shown that the proposed algorithm achieves an asymptotic regret upper bound which is reciprocally related to the maximal size of a local neighborhood within a multi-agent network, provided the network is strongly connected. To our knowledge, this is the best asymptotic regret bound in the existing fully decentralized bandit literature.

The proposed algorithm utilizes the push-sum idea to handle general weight-unbalanced directed graphs. Conse-

quently, it "inherits" a well-known limitation from push-sum, that is, it requires each agent to be aware of its out-neighbors. Additionally, the algorithm analysis relies on the fact that an irreducible nonnegative matrix W has a strictly positive dominant eigenvector v. Therefore, the presented analysis tool cannot be applied to more general weakly connected graphs. These two limitations are directions for future work. Tailoring the proposed algorithm to cope with Byzantine attacks [40] is another potential future direction.

ACKNOWLEDGEMENT

The authors wish to thank Samruddhi Suresh Pednekar (Department of Applied Mathematics and Statistics, Stony Brook University) for proofreading this work and thank all the anonymous reviewers for their helpful comments.

REFERENCES

[1] J. Zhu and J. Liu. Decentralized multi-armed bandit can outperform classic upper confidence bound: A homogeneous case over strongly connected graphs. *arXiv preprint*, 2024. arXiv:2111.10933v3 [cs.LG].

[2] D. Bouneffouf, I. Rish, and C. Aggarwal. Survey on applications of multi-armed and contextual bandits. In *Proceedings of the 2020 IEEE Congress on Evolutionary Computation*, 2020.

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

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

[5] A. Slivkins. Introduction to multi-armed bandits. *Foundations and Trends in Machine Learning*, 12(1-2):1–286, 2019.

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

[7] K. Liu and Q. Zhao. Distributed learning in multi-armed bandit with multiple players. *IEEE Transactions on Signal Processing*, 58(11):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 *Proceedings of the 30th International Conference on Machine Learning*, pages 19–27, 2013.

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

[10] I. Bistritz and A. Leshem. Distributed multi-player bandits – a game of thrones approach. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7222–7232, 2018.

[11] 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.

[12] Y. Wang, J. Hu, X. Chen, and L. Wang. Distributed bandit learning: Near-optimal regret with efficient communication. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

[13] A. Dubey and A. Pentland. Differentially-private federated linear bandits. In *Proceedings of the 34th Conference on Neural Information Processing Systems*, pages 6003–6014, 2020.

[14] C. Shi and C. Shen. Federated multi-armed bandits. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pages 9603–9611, 2021.

[15] U. Madhushani and N.E. Leonard. A dynamic observation strategy for multi-agent multi-armed bandit problem. In *Proceedings of the 2020 European Control Conference*, pages 1677–1682, 2020.

[16] U. Madhushani and N.E. Leonard. Heterogeneous explore-exploit strategies on multi-star networks. *IEEE Control Systems Letters*, 5(5):1603–1608, 2020.

[17] P. Landgren, V. Srivastava, and N.E. Leonard. On distributed cooperative decision-making in multiarmed bandits. In *Proceedings of the 2016 European Control Conference*, pages 243–248, 2016.

[18] D. Martínez-Rubio, V. Kanade, and P. Rebeschini. Decentralized cooperative stochastic bandits. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, pages 4531–4542, 2019.

[19] C. Shi, C. Shen, and J. Yang. Federated multi-armed bandits with personalization. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, 2021.

[20] T. Li, L. Song, and C. Fragouli. Federated recommendation system via differential privacy. In *Proceedings of the 2020 IEEE International Symposium on Information Theory*, pages 2592–2597, 2020.

[21] M. Agarwal, V. Aggarwal, and K. Azizzadenesheli. Multi-agent multi-armed bandits with limited communication. *Journal of Machine Learning Research*, 23(212):1–24, 2022.

[22] L. Yang, Y.-Z.J. Chen, M.H. Hajiemaili, J.C.S. Lui, and D. Towsley. Distributed bandits with heterogeneous agents. In *Proceedings of the 2022 IEEE International Conference on Computer Communications*, pages 200–209, 2022.

[23] R. Chawla, D. Vial, S. Shakkottai, and R. Srikant. Collaborative multi-agent heterogeneous multi-armed bandits. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 4189–4217, 2023.

[24] 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*, 5(1):597–606, 2018.

[25] Z. Zhu, J. Zhu, J. Liu, and Y. Liu. Federated bandit: A gossiping approach. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 5(1):Article 2, 2021.

[26] A. Jadbabaie, J. Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[27] 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.

[28] P. Landgrena, V. Srivastavab, and N.E. Leonarda. Distributed cooperative decision making in multi-agent multi-armed bandits. *Automatica*, 125:109445, 2021.

[29] J. Zhu, R. Sandhu, and J. Liu. A distributed algorithm for sequential decision making in multi-armed bandit with homogeneous rewards. In *Proceedings of the 59th IEEE Conference on Decision and Control*, pages 3078–3083, 2020.

[30] J. Zhu and J. Liu. Distributed multi-armed bandits. *IEEE Transactions on Automatic Control*, 68(5):3025–3040, 2023. Special Issue on Learning for Control.

[31] Z. Wang, C. Zhang, M. K. Singh, L. Riek, and K. Chaudhuri. Multitask bandit learning through heterogeneous feedback aggregation. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 1531–1539. PMLR, 13–15 Apr 2021.

[32] J. Zhu and J. Liu. Distributed multi-armed bandit over arbitrary undirected graphs. In *Proceedings of the 60th IEEE Conference on Decision and Control*, pages 6976–6981, 2021.

[33] B. Gharesifard and J. Cortés. Distributed continuous-time convex optimization on weight-balanced digraphs. *IEEE Transactions on Automatic Control*, 59(3):781–786, 2013.

[34] J. Zhu and J. Liu. A distributed algorithm for multi-armed bandit with homogeneous rewards over directed graphs. In *Proceedings of the 2021 American Control Conference*, pages 3038–3043, 2021.

[35] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proceedings of the 44th Annual Symposium on Foundations of Computer Science*, pages 482–491, 2003.

[36] A. Garivier and O. Cappé. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 359–376, 2011.

[37] S. Agrawal and N. Goyal. Near-optimal regret bounds for Thompson sampling. *Journal of the ACM*, 64(5):1–24, 2017.

[38] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[39] A. Nedić and A. Olshevsky. Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615, 2015.

[40] J. Zhu, A. Koppel, A. Velasquez, and J. Liu. Byzantine-resilient decentralized multi-armed bandits. *Transactions on Machine Learning Research*, 2024. accepted and available at `https://openreview.net/forum?id=JoYMJJdvry`.