Distributed Multi-Armed Bandits

Jingxuan Zhu, *Student Member, IEEE*, Ji Liu, *Member, IEEE*

***Abstract*— This paper studies a distributed multi-armed bandit problem with heterogeneous observations of rewards. The problem is cooperatively solved by N agents assuming each agent faces a common set of M arms yet observes only local biased rewards of the arms. The goal of each agent is to minimize the cumulative expected regret with respect to the true rewards of the arms, where the mean of each arm's true reward equals the average of the means of all agents' observed biased rewards. Each agent recursively updates its decision by utilizing the information from its neighbors. Neighbor relationships are described by a time-dependent directed graph $\mathbb{G}(t)$ whose vertices correspond to agents and whose arcs depict neighbor relationships. A fully distributed bandit algorithm is proposed which couples the classical distributed averaging algorithm and the celebrated upper confidence bound (UCB) bandit algorithm. It is shown that for any uniformly strongly connected sequence of $\mathbb{G}(t)$, the algorithm achieves guaranteed regret for each agent at the order of $O(\log T)$.**

I. Introduction

Multi-armed bandit (MAB) is a fundamental reinforcement learning problem which exemplifies the exploration-exploitation trade-off as a sequential decision-making process and has a wide range of applications in natural and engineered systems including cognitive radio networks, healthcare and online recommendation systems [1]. In a classical MAB problem setting with a single decision maker (or player), the decision maker chooses one arm at each discrete time from a given finite set of arms (or choices), and collects a reward generated according to a random variable with unknown distribution. Different arms may have different unknown reward means. The target of the decision maker is to minimize its cumulative expected regret, i.e., the difference between the decision maker's accumulated expected reward and the maximum which could have obtained had all the reward information been known. Both lower and upper bounds on the asymptotic regret of this MAB problem have been derived in the seminal work [2]. Classic and elegant UCB algorithms have been proposed in [3] which achieve an $O(\log T)$ regret. Considering that the MAB problem and its variants have been studied for many years, it is impossible to survey the entire bandit literature here. For an introductory survey covering the works in this area, see a recent book [4].

Our social networks, communication infrastructure, power grids, data centers, and decision-making needs have become increasingly massive and complex in recent years. In large-scale multi-agent networks such as sensor networks and multi-robot systems, the need for distributed information processing and computing arises naturally because the sensors or robots with on-board processors are physically separated from each other. Meanwhile, in the current emerging big data era, various types of communication constraints exist, and privacy concerns become pervasive, which restricts information flow over social networks and cyber-physical-human systems, and thus precludes conventional centralized information processing and computing, including parallel computing that typically relies on the existence of an information center. With these in mind, efforts to extend conventional single-player bandit settings to multi-player frameworks have attracted increasing attention over the past years, and various multi-agent bandit problems have been proposed and developed with potential applications in different networks. Notable examples include [5]–[17], just to name a few.

Among the existing multi-agent settings, a cooperative setting complements the considerable development in the areas of distributed control and optimization, by incorporating consensus processes [18], [19] among all agents. Such a setting was first proposed in [16] which formulates a distributed MAB problem with homogeneous observations of rewards, i.e., all the agents share the same distribution of each arm's reward. The problem has recently attracted increasing attention and a few different consensus-based distributed algorithms have been proposed and developed [16], [17], [20]–[22]. It is worth pointing out that in such a homogeneous reward setting, each agent can independently learn an optimal arm using any classic single-agent UCB algorithm, without any information exchange or coordination with its neighbors. Thus, cooperation among the agents in the homogeneous reward setting is not necessary, though it may accelerate the agents' bandit learning processes. It is also worth emphasizing that all the existing distributed algorithms, tailored for the distributed MAB problem with homogeneous observations of rewards, focus on fixed neighbor graphs and require each agent be aware of certain network-wise global information, such as spectral properties of the neighbor graph or the total number of agents in the network.

Motivated by a federated learning [23] scenario where different agents hold heterogeneous datasets for the same task, a recent paper [24] formulates a distributed MAB problem with heterogeneous observations of rewards. This heterogeneity may arise due to sampling biases, local observation and data collection errors or noise. Because of the heterogeneity, learning only with local data will lead each agent to locally optimal, not globally optimal, actions. One way to

The work of J. Zhu and J. Liu was supported by the National Science Foundation under Grant No. 2230101. J. Zhu is with the Department of Applied Mathematics and Statistics at Stony Brook University (e-mail: jingxuan.zhu@stonybrook.edu). J. Liu is with the Department of Electrical and Computer Engineering at Stony Brook University (e-mail: ji.liu@stonybrook.edu).

achieve a globally optimal performance is to let all the agents cooperatively smooth out the local biases or noise, which necessitates the communication and coordination among the agents in the network. A motivating example refers to global health emergencies like the ongoing COVID-19 pandemic, in which collaborative research among different countries/regions is critical. Since there are inevitable differences in medical staff training, treatment protocols, healthcare equipment, etc. in different countries and regions, their observed effectiveness of the treatments often contains regional biases. Thus, sharing local treatment evaluations and collaborating with other countries and regions can help smooth out the local biases in treatment and thus obtain a more accurate global model for diagnosis and healthcare. Detailed discussion of this hospital treatment selection problem, including how it can be effectively formulated as a heterogeneous distributed multi-armed bandit problem, can be found in [24, Section 1].

In fact, very few papers have studied multi-agent bandit problems with heterogeneous rewards/settings. The papers [13], [25], [26] consider a system consisting of a central coordinating authority and multiple computational entities. Specifically, [13] studies a federated multi-armed bandit problem in which each arm generates independent observations at different clients with heterogeneous means, [25] considers a federated linear contextual model where the feature vectors at different clients for the same arm are heterogeneous, and [26] explores the benefits of heterogeneity in arm observability among different clients. Another recent paper [9] studies a decentralized setting in which no communication is allowed among the agents, while collision feedback is available when more than one agent selects the same arm. The work [24] is the first and only one which studies a multi-agent bandit problem with heterogeneous rewards in a cooperative distributed setting. The paper proposed a distributed algorithm based on the standard gossiping scheme for communication, and a differentially private variant is also developed. However, the distributed algorithm in [24] works only on time-invariant graphs and requires each agent be aware of global information, the network size N. The use of N in the algorithm makes it not resilient when new agents may join the network.

Distributed algorithms naturally rely on the connectivity of the possibly time-varying graphs representing communication relationships among the agents. As robots move in a multi-robot system, for example, they may leave or enter other robots' communication and sensing radii, causing dynamic, time-varying relationships. Another possible cause of time-varying graphs is unstable communication due to the realistic assumption of some non-zero communication failure probability between any two agents at any given time. Thus, there is ample motivation to develop theoretical guarantees for distributed bandit algorithms with time-varying communication graphs. It is worth emphasizing that in distributed control and optimization problems, the case of time-varying graphs is usually more challenging than the time-invariant one.

The goal of this paper is to propose a fully distributed algorithm for the distributed multi-armed bandit problem with heterogeneous observations of rewards, which does not require any global information, and to study more general time-varying graphs. We show that the proposed distributed UCB algorithm achieves a logarithmic asymptotic regret for each of the agents, provided that the underlying time-varying neighbor graphs are "uniformly strongly connected".

II. Problem Formulation

We are interested in the following distributed multi-armed bandit problem with heterogeneous observations of rewards.

Consider a network consisting of $N > 1$ agents (or players). For presentation purposes, we label the agents from 1 through N, and denote the set of agents by $[N] \triangleq \{1, 2, \dots, N\}$. The agents are not aware of such a global labeling, but can differentiate between their "neighbors". Neighbor relationships among the N agents are described by a possibly time-varying directed graph $\mathbb{G}(t) = (\mathcal{V}, \mathcal{E}(t))$, called neighbor graph, with vertex set $\mathcal{V}$ corresponding to agents and edge set $\mathcal{E}(t)$ depicting neighbor relationships at time $t \in \{0, 1, 2, \dots\}$. Specifically, agent j is a neighbor of agent i at time t whenever (j, i) is a directed edge (or an arc) in $\mathbb{G}(t)$, representing that agent i can receive information from agent j at time t. For simplicity, we assume that each agent i is always a neighbor of itself, and thus $\mathbb{G}(t)$ has a self-arc at each vertex for any time t. The neighbor graph will be assumed to be time-invariant and strongly connected or time-varying and "uniformly strongly connected".

All N agents face a common set of $M > 1$ arms (or choices), denoted by $[M] \triangleq \{1, 2, \dots, M\}$. At each time $t \in \{0, 1, \dots, T\}$, 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 an arm $k \in [M]$, it can collect a reward. The genuine reward is generated according to an unknown random variable $X_k(t)$. However, the agent cannot observe this true reward; instead, it can only observe a local biased (or noisy) copy of the reward, which is generated according to another unknown random variable $X_{i,k}(t)$. The unobservability of $X_k(t)$ can be due to local observational bias or measurement noise. We assume that $\{X_k(t)\}_{t=0}^T$ and $\{X_{i,k}(t)\}_{t=0}^T$, $i \in [N]$, are independent and identically distributed (i.i.d.) random processes. For simplicity of analysis and without loss of generality, we also assume that all X_k and $X_{i,k}$ have bounded support $[0, 1]$. The relationship between $X_k(t)$ and $X_{i,k}(t)$ is as follows. Let μ_k and $\mu_{i,k}$ be the mean of $X_k(t)$ and $X_{i,k}(t)$, respectively. For each $k \in [M]$, the mean of arm k's true reward equals the average of the means of all agents' observed rewards, i.e.,

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

Without loss of generality, assume that $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_M$, and thus arm 1 is always an optimal option.

The distributed multi-armed bandit problem is for each agent i to minimize the following cumulative expected regret:

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

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

It is worth emphasizing that since each agent i can only observe $X_{i,k}$, and $\mu_{i,1}$ is not necessarily the largest among $\mu_{i,k}$, $k \in [M]$, no agent can guarantee to solve the problem without receiving information from its neighbor(s), as any classic or state-of-the-art single-agent bandit algorithms, such as UCB1 and UCB2 [3], are designed to minimize a local cummulative expected regret $T \max_k \mu_{i,k} - \sum_{t=1}^{T} \mathbf{E}[X_{i,a_i(t)}(t)]$ for each agent i. This simple but important observation motivates the necessity of coordination among all N agents. With this observation in mind, we call the problem under consideration *heterogeneous* distributed multi-armed bandit because different agents have differing observed reward means for each arm. In the contrast, in a *homogeneous* setting where the agents observe the same reward mean for each arm, in which case they may even have different arm reward distributions, each agent can learn the bandit problem correctly and independently without using any other agent's information.

The problem just described was first proposed in [24], which solves the problem for a fixed graph in a gossip setting and provides privacy preservation guarantees in the presence of sophisticated adversaries, but requires each agent be aware of a global information, the network size N. In the next section, we will propose a fully distributed algorithm without requiring any global information at any agent. Although privacy preservation is not main consideration of this paper, we assume agents do not want to directly share their rewards possibly due to more immediate, lower-level privacy concerns, which occur in scenarios like multi-player games and social networks. Compared with [24], the most important contribution of this paper is to propose a fully distributed bandit algorithm without using any global information such as N, which works for both time-invariant, strongly connected graphs and time-varying, uniformly strongly connected graph sequences. To get around the limitation of using the network size N is technically challenging. Although the overall analysis flow looks similar to [24] at the glance, the current paper provides a much more refined algorithm design and analysis.

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 $\mathbb{1}(\cdot)$ be the indicator function that returns 1 if the statement is true and 0 otherwise. Define

$$\bar{x}_{i,k}(t) = \frac{1}{n_{i,k}(t)} \sum_{\tau=0}^{t} \mathbb{1}(a_i(\tau) = k) X_{i,k}(\tau), \tag{2}$$

which represents the average reward that agent i received from arm k till time t. Let $C(t, n_{i,k}(t))$ be the upper confidence bound function for agent i and arm k, which is a function of both $n_{i,k}(t)$ and time t. It is designed to be a decreasing function of $n_{i,k}(t)$ and will be specified in the assertions.

Each agent i has control over two sets of variables, denoted $z_{i,k}(t)$ and $m_{i,k}(t)$, which represent agent i's estimate of the network-wide true reward mean of arm k and the maximum sampling times of arm k until time t among all the N agents, respectively, and are updated as follows:

$$z_{i,k}(t+1) = \sum_{j \in \mathcal{N}_i(t)} w_{ij}(t) z_{j,k}(t) + \bar{x}_{i,k}(t+1) - \bar{x}_{i,k}(t), \tag{3}$$

$$m_{i,k}(t+1) = \max\left\{n_{i,k}(t+1),\ m_{j,k}(t),\ j \in \mathcal{N}_i(t)\right\}, \tag{4}$$

where $\mathcal{N}_i(t)$ denotes the set of neighbors of agent i including itself, and $w_{ij}(t)$, $j \in \mathcal{N}_i(t)$, are positive weights. Let $W(t)$ be the $N \times N$ matrix whose ijth entry equals $w_{ij}(t)$ if agent j is a neighbor of agent i at time t or zero otherwise. Since each agent i is always a neighbor of itself, the diagonal entries of each $W(t)$ are all positive. We assume that $W(t)$ is a "doubly stochastic matrix" for any time t where by a doubly stochastic matrix is meant a nonnegative square matrix whose row and column sums all equal one, and $w_{ij}(t)$, $j \in \mathcal{N}_i(t)$, are uniformly bounded below by some positive number.[1] In the case when $\mathbb{G}(t)$ is undirected (i.e., agent i is a neighbor of agent j whenever agent j is a neighbor of agent i), such a doubly stochastic $W(t)$ can be constructed in a distributed manner via the Metropolis algorithm [27] in which

$$w_{ij}(t) = \frac{1}{\max\{|\mathcal{N}_i(t)|, |\mathcal{N}_j(t)|\}}, \quad j \in \mathcal{N}_i(t), \quad j \neq i,$$

$$w_{ii}(t) = 1 - \sum_{j \in \mathcal{N}_i(t)} \frac{1}{\max\{|\mathcal{N}_i(t)|, |\mathcal{N}_j(t)|\}},$$

where $|\mathcal{N}_i(t)|$ denotes the number of neighbors of agent i at time t, or equivalently, the degree of vertex i when $\mathbb{G}(t)$ is undirected. It is easy to see that the Metropolis algorithm requires bi-directional communication between any pair of neighbors. For more general directed graphs in which uni-directional communication may occur, it has been shown that when $\mathbb{G}(t)$ is strongly connected, a doubly stochastic matrix $W(t)$, whose zero and nonzero pattern is consistent with $\mathbb{G}(t)$, always exists [28], and can be computed via a distributed algorithm in finite time [29]. It is worth emphasizing that $z_{i,k}(t)$ and $m_{i,k}(t)$ are updated in a distributed manner as they only use information from agent i's neighbors. The purpose of $z_{i,k}(t)$ and $m_{i,k}(t)$ is for agent i to locally estimate the network-wise true reward mean of arm k, μ_k, and the maximal number of pulls on arm k at one agent till time t over the entire network, $\max_{j\in[N]} n_{j,k}(t)$, respectively.

A detailed description of our algorithm is given as follows.

Initialization: At time $t = 0$, each agent i samples each arm k exactly once, sets $n_{i,k}(0) = 1$, $z_{i,k}(0) = \bar{x}_{i,k}(0) = X_{i,k}(0)$, $m_{i,k}(0) = 1$, and $C(0, n_{i,k}(0)) = 0$.

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) **Decision Making:**
   a) If there is no arm k such that $n_{i,k}(t) \leq m_{i,k}(t) - M$, agent i computes the index

$$Q_{i,k}(t) = z_{i,k}(t) + C(t, n_{i,k}(t)),$$

[1] The uniform lower bound is a widely-adopted technical assumption in consensus literature for guaranteeing exponentially fast consensus.

and then pulls the arm $a_i(t+1)$ that maximizes $Q_{i,k}(t)$, with ties broken arbitrarily, and receives reward $X_{i,a_i(t+1)}(t+1)$.

b) If there exists at least one arm k such that $n_{i,k}(t) \le m_{i,k}(t) - M$, agent i randomly pulls one such arm.

2) **Transmission:** Agent i broadcasts $m_{i,k}(t)$ and $z_{i,k}(t)$, $k \in [M]$, to all its current out-neighbors; at the same time, agent i receives $m_{j,k}(t)$ and $z_{j,k}(t)$, $k \in [M]$, from each current neighbor $j \in \mathcal{N}_i(t)$.

3) **Updating:**

$$\begin{aligned}
n_{i,k}(t+1) &= \begin{cases} n_{i,k}(t)+1 & \text{if } k = a_i(t+1), \\ n_{i,k}(t) & \text{if } k \neq a_i(t+1), \end{cases} \\
\bar{x}_{i,k}(t+1) &= \frac{1}{n_{i,k}(t+1)} \sum_{\tau=0}^{t+1} \mathbb{1}(a_i(\tau)=k) X_{i,k}(\tau), \\
z_{i,k}(t+1) &= \sum_{j \in \mathcal{N}_i(t)} w_{ij}(t) z_{j,k}(t) \\
&\quad + \bar{x}_{i,k}(t+1) - \bar{x}_{i,k}(t), \\
m_{i,k}(t+1) &= \max\left\{ n_{i,k}(t+1),\ m_{j,k}(t),\ j \in \mathcal{N}_i(t) \right\}.
\end{aligned}$$

Before proceeding, let us elaborate the Decision Making step. As one bottleneck of a (distributed) MAB process, insufficient exploration on some arms would cause agents to keep pulling a sub-optimal arm; this is why we have case (a). The index computed in case (a) is the summation of the local estimate on reward mean of arm k and the upper confidence bound $C(t, n_{i,k}(t))$. Since $C(t, n_{i,k}(t))$ is designed to be decreasing with respect to $n_{i,k}(t)$, pulling the arm with largest $Q_{i,k}(t)$ can be understood as the agent makes a trade-off between picking the arms with currently largest reward mean (exploitation) and the arms so far least explored (exploration). In this sense, the algorithm ensures that each arm to be sufficiently explored. In the single-agent MAB problem [3], such an exploitation/exploration trade-off is the main issue in algorithm design. However, for the heterogeneous multi-agent case under study, there is an additional critical issue: the exploration process of each agent may not be on the same page. Specifically, if an agent insufficiently explores an arm, its possibly "poor" estimation of the arm would become a drag on the estimation quality of all other agents via information fusion over the network, because the accurate estimate of the true reward mean relies on all the agents' locally observed reward means. In other words, even if an agent has made sufficient exploration on one arm, it may still be "misled" by another agent if the latter has not; this is a critical feature and challenge in the heterogeneous setting. With this in mind, case (b) is thus designed to tackle the challenge by restricting a quantitative relation between agent i's local sampling number $n_{i,k}(t)$ and the estimated global maximum sampling times $m_{i,k}(t)$. The relation guarantees that no agent in the network would fall behind in the exploration process for each arm.

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

Algorithm 1: Distributed UCB

```
Input: $\mathbb{G}(t), T, C(t, n_{i,k}(t))$
Initialization Each agent samples each arm exactly
  once. Set $z_{i,k}(0) = \bar{x}_{i,k}(0) = X_{i,k}(0)$,
  $m_{i,k}(0) = n_{i,k}(0) = 1$, and $C(0, n_{i,k}(0)) = 0$.
for $t = 0, \dots, T$ do
    $\mathcal{A}_i = \emptyset$
    if $n_{i,k}(t) \le m_{i,k}(t) - M$ then
        Agent $i$ puts index $k$ into set $\mathcal{A}_i$
    end
    if $\mathcal{A}_i = \emptyset$ then
        for $k = 1, \dots, M$ do
            $Q_{i,k}(t) = z_{i,k}(t) + C(t, n_{i,k}(t))$
        end
        $a_i(t+1) = \arg\max_k Q_{i,k}(t)$
    else
        $a_i(t+1)$ is randomly chosen from $\mathcal{A}_i$
    end
    Agent $i$ sends $m_{i,k}(t)$ and $z_{i,k}(t)$, $\forall k \in [M]$, to
      those agents $j$ for which $i \in \mathcal{N}_j(t)$
    Agent $i$ receives $m_{j,k}(t), z_{j,k}(t)$, $\forall k \in [M]$, from
      each $j \in \mathcal{N}_i(t)$
    $n_{i,k}(t+1) = n_{i,k}(t)$, $\forall k \in [M]$
    $n_{i,a_i(t+1)}(t+1) = n_{i,a_i(t+1)}(t) + 1$
    $z_{i,k}(t+1) =$
      $\sum_{j \in \mathcal{N}_i(t)} w_{ij}(t) z_{j,k}(t) + \bar{x}_{i,k}(t+1) - \bar{x}_{i,k}(t)$
      $m_{i,k}(t+1) = \max\{n_{i,k}(t+1), m_{j,k}(t), j \in \mathcal{N}_i(t)\}$
end
```

We first consider the case when the neighbor graph is time-invariant, i.e., $\mathbb{G}(t) = \mathbb{G}$ for all time t, and $\mathbb{G}$ is strongly connected. In this case, $\mathcal{N}_i(t) = \mathcal{N}_i$ and $W(t) = W$ for all $i \in [N]$ and time t, and W is an irreducible doubly stochastic matrix whose diagonal entries are all positive. Let $\mathbf{1}$ denote the N-dimensional vector whose entries all equal 1. It is well known that W^t converges to $\frac{1}{N}\mathbf{1}\mathbf{1}^\top$ exponentially fast as $t \to \infty$, i.e., there exist constants $c > 0$ and $\rho_2 \in [0,1)$ such that

$$\left\| W^t - \frac{1}{N}\mathbf{1}\mathbf{1}^\top \right\|_2 \le c\rho_2^t. \tag{5}$$

Here ρ_2 equals the second largest magnitude among all the eigenvalues of W. In the special case when W is symmetric, $c = 1$; see Lemma 8. Define $\Delta_k = \mu_1 - \mu_k$, $k \in [M]$, as the gap of mean reward between arm 1 and arm k. Then, the regret $R_i(T)$ defined in (1) can be rewritten as

$$R_i(T) = \sum_{k:\Delta_k > 0} \Delta_k \mathbf{E}(n_{i,k}(T)). \tag{6}$$

The performance of the proposed algorithm is characterized in the following theorem.

Theorem 1: Suppose that $\mathbb{G}$ is strongly connected and all N agents adhere to Algorithm 1. Then, with

$$C(t, n_{i,k}(t)) = (1+\beta_i)\sqrt{\frac{3\log t}{|\mathcal{N}_i| n_{i,k}(t)} + \frac{1}{2t}},$$

where β_i is an arbitrary positive constant, the regret of each agent i until time T satisfies

$$R_i(T) \le \sum_{k:\Delta_k>0} \left(\max\left\{ \frac{12(1+\beta_i)^4 \log T}{|\mathcal{N}_i|\Delta_k^2}, K_1 \right\} + K_2 \right) \Delta_k,$$

where K_1 and K_2 are constants defined in Remark 4.

The theorem implies that for sufficiently large T, the $\log T$ term dominates the regret bound, and thus $R_i(T) = O(\frac{\log T}{|\mathcal{N}_i|})$, which is of the same order as the classic single-agent (non-cooperative) UCB algorithms, e.g., UCB1 and UCB2 in [3], and shows that cooperation with neighbors improves the regret.

Remark 1: Compared with the classic single-agent UCB [3], the upper confidence bound in our algorithm has an additional $O(1/t)$ term, that is because it is shown in the proof of Theorem 1 that to ensure optimal rate convergence, the upper confidence bound should be designed as a summation of a $O(\log T)$ term and an extra term upper bounded by $c\sqrt{N}\rho_2^t$, which is originated from the heterogeneous observation setting. Since ρ_2 is a global information that no agent is aware of, it cannot be used in the design of the upper confidence bound. Thus, we use a $O(1/t)$ term to bound the term. Indeed, any function that converges to zero at a decaying rate slower than exponential can be used here, while this term will influence the value of constant K_1 and thus the finite-time regret bound. Study of this influence is not the focus on this paper. □

Remark 2: Consider the homogeneous case in which $\mu_{i,k} = \mu_k$ for all $i \in [N]$ and $k \in [M]$. From Theorem 1,

$$\lim_{T\to\infty} R_i(T) \le \sum_{k:\Delta_k>0} \left(\frac{12(1+\beta_i)^4}{|\mathcal{N}_i|\Delta_k} + o(1) \right) \log T.$$

It is easy to see that the asymptotic regret bound is increasing in terms of β_i, and when it is chosen to be sufficiently small, the asymptotic bound of the regret $R_i(T)$ is close to

$$\sum_{k:\Delta_k>0} \left(\frac{12}{|\mathcal{N}_i|\Delta_k} + o(1) \right) \log T.$$

Note that when each agent independently applies the classic (single-agent) UCB1 [3], the asymptotic regret bound is

$$\sum_{k:\Delta_k>0} \left(\frac{8}{\Delta_k} + o(1) \right) \log T.$$

Since $|\mathcal{N}_i| \ge 2$ for all $i \in [N]$ in a strongly connected $\mathbb{G}$ with self-arcs, we can conclude that for the homogeneous setting, with sufficiently small β_i, the asymptotic regret bound of each agent in our distributed algorithm is always better than the single-agent counterpart. For the more general heterogeneous setting, if each agent independently applies the classic UCB1 without communicating with neighbors, its regret can be as bad as linear in T, as illustrated in Fig. 2 in Section V. □

Remark 3: From Theorem 1, if $\mathbb{G}$ is a complete graph in which $|\mathcal{N}_i| = N$ for all i, the asymptotic regret bound for each agent becomes $O(\frac{\log T}{N})$, which is the largest "collaborative gain" in regret improvement it could possibly be for an N-agent network. Such a largest collaborative gain can also be achieved if each agent is assumed to be aware of the network size N. In this case, with

$$C(t, n_{i,k}(t)) = (1+\beta_i)\sqrt{\frac{3\log t}{N n_{i,k}(t)}} + \frac{1}{2t}$$

and the same arguments as in the proof of Theorem 1, it can be shown that $R_i(T) \le O(\frac{\log T}{N})$. □

Now we consider more general time-varying neighbor graphs. To this end, we need the following concepts.

Let $\mathbb{G}_p$ and $\mathbb{G}_q$ be two directed graphs with the same vertex set $\mathcal{V}$. By the composition of $\mathbb{G}_p$ with $\mathbb{G}_q$, written $\mathbb{G}_q \circ \mathbb{G}_p$, is meant the directed graph with vertex set $\mathcal{V}$ and arc set defined in such a way so that (i, j) is an arc of the composition whenever there is a vertex $k \in \mathcal{V}$ such that (i, k) is an arc of $\mathbb{G}_p$ and (k, j) is an arc of $\mathbb{G}_q$. It is clear that composition is an associative binary operation, and thus the definition extends unambiguously to any finite sequence of directed graphs with the same vertex set. A simple but important fact is the union of the arc sets of a sequence of directed graphs with self-arcs at all vertices, as considered in this paper, must be contained in the arc set of their composition, but not vice versa.

To proceed, a finite sequence of directed graphs $\mathbb{G}_1, \mathbb{G}_2, \dots, \mathbb{G}_q$ with the same vertex set is called jointly strongly connected if their composition $\mathbb{G}_q \circ \mathbb{G}_2 \circ \cdots \circ \mathbb{G}_1$ is strongly connected. We say that an infinite sequence of directed graphs $\mathbb{G}_1, \mathbb{G}_2, \dots$ with the same vertex set is uniformly strongly connected if there exists a positive integer l such that for each integer $k \ge 0$, the sequence of l graphs $\mathbb{G}_{k+1}, \mathbb{G}_{k+2}, \dots, \mathbb{G}_{k+l}$ is jointly strongly connected, i.e., the composed graph $\mathbb{G}_{k+l} \circ \cdots \circ \mathbb{G}_{k+2} \circ \mathbb{G}_{k+1}$ is strongly connected. If such an integer exists, we sometimes say that the sequence is uniformly strongly connected by sub-sequences of length l. Thus, if $\mathbb{G}(1), \mathbb{G}(2), \dots$ is a sequence of neighbor graphs which is uniformly strongly connected by sub-sequences of length l, then over any l consecutive iterations, each proper subset of the N agents can receive information from the rest.

It is easy to prove that the above definition is equivalent to the two popular joint connectivity definitions in consensus literature, namely "repeatedly jointly strongly connected" [30] and "B-connected" [31].

In the case when $\mathbb{G}(t)$ is time-varying and uniformly strongly connected, the corresponding sequence of doubly stochastic matrices $W(t)$, whose nonzero entries have a uniform positive lower bound, has the property that its backward product $W(t-1)\cdots W(1)W(0)$ converges to $\frac{1}{N}\mathbf{1}\mathbf{1}^\top$ exponentially fast as $t \to \infty$, i.e., there exist constants $c' > 0$ and $\rho \in [0, 1)$ such that

$$\left\| W(t-1)\cdots W(1)W(0) - \frac{1}{N}\mathbf{1}\mathbf{1}^\top \right\|_2 \le c'\rho^t, \qquad (7)$$

whose proof can be found in, for example, [31], [32].

The performance of Algorithm 1 over a time-varying neighbor graph is characterized in the following theorem.

Theorem 2: Suppose that $\mathbb{G}(0), \mathbb{G}(1), \mathbb{G}(2), \dots$ are uniformly strongly connected by sub-sequences of length l and all N agents adhere to Algorithm 1. Then, with

$$C(t, n_{i,k}(t)) = (1+\beta_i)\sqrt{\frac{3\log t}{\delta_i(t) n_{i,k}(t)}} + \frac{1}{2t},$$

where β_i is an arbitrary positive constant and $\delta_i(t) = \max_{0\le t'\le t} |\mathcal{N}_i(t')|$, the regret of each agent i satisfies

$$R_i(T) \le \sum_{k:\Delta_k>0} \left(\max\left\{ \frac{12(1+\beta_i)^4 \log T}{\delta_i(\tilde{K}_1)\Delta_k^2}, \tilde{K}_1 \right\} + \tilde{K}_2 \right)\Delta_k,$$

where $\tilde{K}_1$ and $\tilde{K}_2$ are constants defined in Remark 4.

Remark 4: We provide here the constants appearing in the regret bounds in Theorems 1 and 2. For Theorem 1,

$$K_1 = \max\{F_1(\beta_i),\ 2F_2(\beta_i)\},$$
$$K_2 = M^2 + 2MN + N + N(\pi^2/3 + K_1 - 1),$$

where for any $\epsilon > 0$, $F_1(\epsilon)$ is defined in (29) and $F_2(\epsilon) = \max\{f(\epsilon),\ 2(M^2+2MN+N)\}$, with $f(\epsilon)$ being the infimum of x such that when $n \ge x$, it always holds that

$$\rho_2^n + \sum_{h=2}^{n} \frac{\rho_2^{n-h}}{(h-1)h} \le \frac{\epsilon}{cNn}.$$

It will be shown in the proof of Lemma 9 that $F_2(\epsilon)$ is well-defined. For Theorem 2,

$$\tilde{K}_1 = \max\{\tilde{F}_1(\beta_i),\ 2\tilde{F}_2(\beta_i)\},$$
$$\tilde{K}_2 = M^2 + (M+1)(N-1)l + N(\pi^2/3 + \tilde{K}_1 - 1),$$

where for any $\epsilon > 0$,

$$\tilde{F}_1(\epsilon) = \max\left\{ \frac{1+\epsilon}{\epsilon\Delta},\ \inf_x \left\{ c'\sqrt{N}\rho^t < \frac{1}{2t}, \forall t \ge x \right\} \right\},$$
$$\tilde{F}_2(\epsilon) = \max\left\{ \tilde{f}(\epsilon),\ 2(M^2 + (M+1)(N-1)l) \right\},$$

with $\Delta = \min_{k:\Delta_k>0} \Delta_k$ and $\tilde{f}(\epsilon)$ being the infimum of x such that when $n \ge x$, it always holds that

$$\rho^n + \sum_{h=2}^{n} \frac{\rho^{n-h}}{(h-1)h} \le \frac{\epsilon}{c'Nn}.$$

Using the similar arguments to those in the proofs of Lemma 9 and Theorem 1, both $\tilde{F}_1(\epsilon)$ and $\tilde{F}_2(\epsilon)$ are well defined. □

From Theorem 2, it is easy to see that when T is sufficiently large, the $\log T$ term is the dominant in the regret bound, and thus $R_i(T) = O(\frac{\log T}{\delta_i(\tilde{K}_1)})$, which is of the same order of the fixed graph case; see Theorem 1 and Fig. 7.

Note that $\delta_i(t)$ in Theorem 2 makes the algorithm involve an additional updating step for each agent at each iteration, which tracks the largest number of neighbors in history, and that $\delta_i(\tilde{K}_1)$ appearing in the regret bound is a positive constant bounded above by N. A simpler version of $C(t, n_{i,k}(t))$ without involving $\delta_i(t)$ is given in the following corollary, which yields a slightly worse regret bound and is an immediate consequence of Theorem 2 using the same analysis.

Corollary 1: Suppose that $\mathbb{G}(0), \mathbb{G}(1), \mathbb{G}(2), \dots$ are uniformly strongly connected by sub-sequences of length l and all N agents adhere to Algorithm 1. Then, with

$$C(t, n_{i,k}(t)) = (1+\beta_i)\sqrt{\frac{3\log t}{n_{i,k}(t)}} + \frac{1}{2t},$$

where β_i is an arbitrary positive constant, the regret of each agent i until time T satisfies

$$R_i(T) \le \sum_{k:\Delta_k>0} \left(\max\left\{ \frac{12(1+\beta_i)^4 \log T}{\Delta_k^2}, \tilde{K}_1 \right\} + \tilde{K}_2 \right)\Delta_k,$$

where $\tilde{K}_1$ and $\tilde{K}_2$ are constants defined in Remark 4.

IV. Analysis

This section provides the analysis of Algorithm 1 and proofs of the two theorems stated in the previous section.

Let $z_k(t)$ and $\bar{x}_k(t)$ be the column stacks of all $z_{i,k}(t)$ and $\bar{x}_{i,k}(t)$, respectively. Then, the N equations in (3) can be combined as

$$z_k(t+1) = W(t) z_k(t) + \bar{x}_k(t+1) - \bar{x}_k(t), \tag{8}$$

where each $W(t)$ is an irreducible doubly stochastic matrix.

We will need the following concept. A random variable X with $\mathbf{E}[X] = \mu$ is called σ^2 sub-Gaussian if there exists a positive constant σ such that

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

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 three properties, namely Lemmas 1 to 3. The proofs for the first two lemmas can be found in Section 5.3 in [33], and Lemma 3 is a direct consequence of Hoeffding's Lemma [34].

Lemma 1: For any $a \ge 0$ and σ^2 sub-Gaussian random variable X with $\mathbf{E}[X] = \mu$,

$$\mathbf{P}(X - \mu \ge a) \le e^{-\frac{a^2}{2\sigma^2}}, \quad \mathbf{P}(\mu - X \ge a) \le e^{-\frac{a^2}{2\sigma^2}}.$$

Lemma 2: Let $X_1, \dots, X_n$ be n independent random variables such that X_i is σ_i^2 sub-Gaussian random variable for all $i \in [N]$, then $X_1 + \cdots + X_n$ is $(\sigma_1^2 + \cdots + \sigma_n^2)$ sub-Gaussian.

Lemma 3: If a random variable X has a finite mean and $a \le X \le b$ almost surely, then X is $\frac{1}{4}(b-a)^2$ sub-Gaussian.

More can be said. Let $\{X_1, \dots, X_n\}$ be a finite set of independent random variables such that for all $i \in [n]$, X_i satisfies: 1) $X_i \in [a_i, b_i]$, 2) $\mathbf{E}(X_i) = \mu_i$, and 3) X_i is σ_i^2 sub-Gaussian, then from the above three lemmas, for any $\eta \ge 0$,

$$\begin{aligned}
&\mathbf{P}\left(\sum_{i=1}^n X_i - \sum_{i=1}^n \mu_i \ge \eta \right) \le \exp\left(\frac{-2\eta^2}{\sum_{i=1}^n (b_i - a_i)^2} \right), \\
&\mathbf{P}\left(\sum_{i=1}^n \mu_i - \sum_{i=1}^n X_i \ge \eta \right) \le \exp\left(\frac{-2\eta^2}{\sum_{i=1}^n (b_i - a_i)^2} \right).
\end{aligned} \tag{9}$$

A. Time-invariant Neighbor Graph

We begin with the proof of Theorem 1. To proceed, for any $i, j \in [N]$, we use $d_{i,j}$ to denote the "distance" from vertex i to vertex j in the time-invariant neighbor graph $\mathbb{G}$. For a strongly connected graph, the distance from vertex i to vertex j is defined as the number of directed edges in a shortest directed path from vertex i to vertex j in the graph. It is natural to define $d_{i,i} = 0$ for any vertex i, and easy to see that $d_{i,j} \le N$.

For the purpose of analysis, we define $n_{i,k}(t) = m_{i,k}(t) = 0$ for all $i \in [N]$ and $k \in [M]$ when $t < 0$.

Lemma 4: For any $i \in [N]$ and $k \in [M]$,

$$m_{i,k}(t) = \max_{j\in[N]} \left\{n_{j,k}(t - d_{j,i})\right\}. \tag{10}$$

Proof: We will prove the lemma by induction on t. For the basis step, suppose that $t = 0$. In this case, $m_{i,k}(1) = \max\{n_{i,k}(0),\ m_{j,k}(0),\ j \in \mathcal{N}_i\} = 1$. Note that $\max_{j\in[N]}\{n_{j,k}(t - d_{j,i})\} = n_{i,k}(0) = 1$. Thus, (10) holds when $t = 0$.

For the inductive step, assume (10) holds at time t, and now consider time $t + 1$. Note that

$$\begin{aligned} m_{i,k}(t+1) &= \max\{n_{i,k}(t+1),\ m_{j,k}(t),\ j \in \mathcal{N}_i\} \\ &= \max\{n_{i,k}(t+1),\ n_{h,k}(t - d_{j,h}), \\ &\qquad h \in [N],\ j \in \mathcal{N}_i\}. \end{aligned}$$

It is easy to see that $d_{h,i} \le d_{j,i} + d_{h,j} = 1 + d_{h,j}$. Since $n_{i,k}(t)$ is a non-decreasing function of t by its definition,

$$\begin{aligned} m_{i,k}(t+1) &\le \max_{h\in[N]}\{n_{i,k}(t+1),\ n_{h,k}(t - d_{h,i} + 1)\} \\ &= \max_{j\in[N]}\{n_{j,k}(t - d_{j,i} + 1)\}. \end{aligned} \tag{11}$$

Fix any vertex $j \in [N]$ and let $p = (j, v_{d_{j,i}}, \dots, v_2, i)$ be a shortest path from j to i in $\mathbb{G}$. From (4),

$$\begin{aligned} m_{i,k}(t+1) &\ge m_{v_2,k}(t) \ge \cdots \ge m_{v_{d_{j,i}},k}(t - d_{j,i} + 2) \\ &\ge m_{j,k}(t - d_{j,i} + 1) \ge n_{j,k}(t - d_{j,i} + 1). \end{aligned} \tag{12}$$

Since j is arbitrarily chosen from $[N]$, $m_{i,k}(t+1) \ge \max_{j\in[N]}\{n_{j,k}(t - d_{j,i} + 1)\}$. Combining with (11),

$$m_{i,k}(t+1) = \max_{j\in[N]}\{n_{j,k}(t - d_{j,i} + 1)\}.$$

So (10) also holds at $t + 1$, which completes the induction. ■

Lemma 5: For any $i \in [N]$ and $k \in [M]$,

$$n_{i,k}(t) > m_{i,k}(t) - M(M + 2N).$$

Proof: We will prove the lemma by contradiction. Suppose that, to the contrary, there exist i and k_1 such that $n_{i,k_1}(t) \le m_{i,k_1}(t) - M(M + 2N)$. Let t' denote the first time at which the equality holds, i.e.,

$$n_{i,k_1}(t') = m_{i,k_1}(t') - M(M + 2N).$$

Here t' must exist. To see this, first note that at initial time $t = 0$, $n_{i,k}(0) > m_{i,k}(0) - M(M + 2N)$. Since both $n_{i,k}(t)$ and $m_{i,k}(t)$ either do not change or increase by 1 at each time instance, if there exists some t for which $n_{i,k_1}(t) < m_{i,k_1}(t) - M(M+2N)$, there must exist a t' between 0 and t such that $n_{i,k_1}(t') = m_{i,k_1}(t') - M(M+2N)$. From Lemma 4, there exists a $j \in [N]$ such that

$$m_{i,k_1}(t') = n_{j,k_1}(t' - d_{j,i}). \tag{13}$$

Then,

$$n_{j,k_1}(t' - d_{j,i}) - n_{i,k_1}(t') = M(M + 2N). \tag{14}$$

Also from Lemma 4, $m_{i,k_1}(t) \ge n_{j,k_1}(t - d_{j,i})$ always holds. Thus, for $t < t'$,

$$\begin{aligned} n_{j,k_1}(t - d_{j,i}) - n_{i,k_1}(t) &\le m_{i,k_1}(t) - n_{i,k_1}(t) \\ &< M(M + 2N). \end{aligned} \tag{15}$$

Since $n_{i,k}(t)$ is non-decreasing for any fixed $i \in [N]$ and $k \in [M]$, (14) and (15) imply that $n_{j,k_1}(t' - d_{j,i}) > n_{j,k_1}(t' - d_{j,i} - 1)$. This further implies that at time $t' - d_{j,i}$, agent j pulls arm k_1. Since each agent must pull an arm at each time, $\sum_{k=1}^{M} n_{i,k}(t) = t + M,\ \forall i \in [N]$. Then,

$$\begin{aligned} &\sum_{k\in[M]\setminus k_1} n_{i,k}(t') - \sum_{k\in[M]\setminus k_1} n_{j,k}(t' - d_{j,i}) \\ &> M(M + 2N) + d_{j,i}. \end{aligned}$$

Applying the Pigeonhole principle, $\exists\ k_2 \in [M]$ such that

$$n_{i,k_2}(t') - n_{j,k_2}(t' - d_{j,i}) \ge \frac{M(M + 2N)}{M - 1} > M + 2N.$$

According to the definition of $n_{i,k}(t)$, it holds that $n_{i,k}(t + 1) \le n_{i,k}(t) + 1$, which implies that $n_{i,k}(t + \tau) \le n_{i,k}(t) + \tau$ for any positive integer τ. Then,

$$\begin{aligned} n_{i,k_2}(t') &= n_{i,k_2}(t' - d_{j,i} - d_{i,j} + d_{j,i} + d_{i,j}) \\ &\le n_{i,k_2}(t' - d_{j,i} - d_{i,j}) + d_{j,i} + d_{i,j}. \end{aligned}$$

Thus,

$$\begin{aligned} &n_{i,k_2}(t' - d_{j,i} - d_{i,j}) - n_{j,k_2}(t' - d_{j,i}) \\ &> n_{i,k_2}(t') - n_{j,k_2}(t' - d_{j,i}) - d_{j,i} - d_{i,j} \\ &> M + 2N - d_{j,i} - d_{i,j} > M. \end{aligned}$$

Using (12), $m_{j,k_2}(t' - d_{j,i}) \ge n_{i,k_2}(t' - d_{j,i} - d_{i,j})$. Thus,

$$m_{j,k_2}(t' - d_{j,i}) - n_{j,k_2}(t' - d_{j,i}) > M.$$

From the description of Algorithm 1, $k_2 \in \mathcal{A}_j$. Since $\mathcal{A}_j$ is not empty, agent j randomly picks an arm in $\mathcal{A}_j$ at time $t' - d_{j,i}$ according to the Decision Making step. Meanwhile, from the preceding analysis, agent j in fact pulls arm k_1 at $t' - d_{j,i}$, which implies that $k_1 \in \mathcal{A}_j$, and thus

$$m_{j,k_1}(t' - d_{j,i}) - n_{j,k_1}(t' - d_{j,i}) \ge M > 0. \tag{16}$$

Note that from (12),

$$m_{i,k_1}(t') \ge m_{j,k_1}(t' - d_{j,i}). \tag{17}$$

Combining (13)–(17) together,

$$\begin{aligned} n_{j,k_1}(t' - d_{j,i}) = m_{i,k_1}(t') &\ge m_{j,k_1}(t' - d_{j,i}) \\ &> n_{j,k_1}(t' - d_{j,i}), \end{aligned}$$

which is a contradiction. Therefore, the lemma is true. ■

Lemma 6: For any $i \in [N]$ and $k \in [M]$, if $n_{i,k}(t) \ge 2(M^2 + 2MN + N)$, then for any $h \in [N]$,

$$\frac{1}{2} n_{h,k}(t) \le n_{i,k}(t) \le \frac{3}{2} n_{h,k}(t).$$

Proof: From (12), $m_{i,k}(t) \ge n_{h,k}(t - d_{h,i}),\ \forall h \in [N]$. Note that for any $i \in [N]$, $k \in [M]$, and $t \ge 0$, it holds that $n_{i,k}(t + 1) \le n_{i,k}(t) + 1$. Thus, for all $i, h \in [N]$, there holds

$$m_{i,k}(t) \ge n_{h,k}(t) - d_{h,i} \ge n_{h,k}(t) - N.$$

Combining this with Lemma 5, for any $i,h\in[N]$,

$$\begin{aligned} n_{i,k}(t) &\geq n_{h,k}(t) - (M^2+2MN+N),\\ n_{i,k}(t) &\leq n_{h,k}(t) + (M^2+2MN+N). \end{aligned} \tag{18}$$

Then, when $n_{i,k}(t)\geq 2(M^2+2MN+N)$, for any $h\in[N]$,

$$n_{h,k}(t)+\frac{1}{2}n_{i,k}(t)\geq n_{i,k}(t)\geq n_{h,k}(t)-\frac{1}{2}n_{i,k}(t).$$

Simplifying this inequality immediately implies the lemma. ■

Lemma 7: If W is an irreducible doubly stochastic matrix with positive diagonal entries, then there exists a positive constant c such that

$$\left\| W^t - \frac{1}{N}\mathbf{1}\mathbf{1}^\top \right\|_2 \leq c\rho_2^t, \tag{19}$$

$$\left| [W^t]_{ij} - \frac{1}{N} \right| \leq c\rho_2^t \tag{20}$$

for all $i,j\in[N]$, where ρ_2 is the second largest magnitude among all the eigenvalues of W.

The lemma is well-known and its proof can be found in [35]; see the proof of Theorem 1 in [35]. More can be said for symmetric matrices.

Lemma 8: If W is a symmetric, irreducible, (doubly) stochastic matrix with positive diagonal entries, then

$$\left\| W^t - \frac{1}{N}\mathbf{1}\mathbf{1}^\top \right\|_2 \leq \rho_2^t, \quad \left| [W^t]_{ij} - \frac{1}{N} \right| \leq \rho_2^t$$

for all $i,j\in[N]$, where ρ_2 is the second largest magnitude among all the eigenvalues of W.[2]

The proof of Lemma 8 is standard and thus omitted due to space limitations.

Lemma 9: For any $i\in[N]$, $k\in[M]$, and time t, $z_{i,k}(t)$ is a linear combination of $X_{j,k}(\tau)$, $j\in[N]$, $\tau\in\{0,1,\dots,t\}$. For any $\epsilon>0$, the absolute value of the coefficient of each $X_{j,k}(t)$ is bounded above by $\frac{(1+\epsilon)}{Nn_{j,k}(t)}$ when $n_{j,k}(t)\geq F_2(\epsilon)$, where $F_2(\epsilon)$ is defined in Remark 4.

Proof: From (2) and (3), $z_{i,k}(t)$ is a linear combination of $X_{j,k}(\tau)$ with all $j\in[N]$ and $\tau\in\{1,2,\dots,t\}$. Let $c^{(\tau)}_{i,k,j}(t)$ denote the coefficient of $X_{j,k}(\tau)$. Note that from (8),

$$\begin{aligned} z_k(t) &= Wz_k(t-1)+\bar{x}_k(t)-\bar{x}_k(t-1)\\ &= W^t z_k(0)+\sum_{\tau=0}^{t-1}W^\tau(\bar{x}_k(t-\tau)-\bar{x}_k(t-\tau-1))\\ &= \sum_{\tau=0}^{t-1}(W^{t-\tau}-W^{t-\tau-1})\bar{x}_k(\tau)+\bar{x}_k(t). \end{aligned}$$

Thus,

$$\begin{aligned} z_{i,k}(t) = \sum_j \Bigg\{ & \sum_{\tau=0}^{t-1}[W^{t-\tau}-W^{t-\tau-1}]_{ij}\bar{x}_{j,k}(\tau)\\ & + [W^0]_{ij}\bar{x}_{j,k}(t) \Bigg\}. \end{aligned}$$

Denote $\tau_{i,1},\tau_{i,2},\dots,\tau_{i,n_{i,k}(t)}$ as the ascending sequence of all time instances before time t at which agent i pulls arm k. From the initialization step of the algorithm, it is clear that $\tau_{i,1}=0$. According to update (2), for $\tau\in[\tau_{i,m},\tau_{i,m+1})$, there holds $\bar{x}_{i,k}(\tau)=\bar{x}_{i,k}(\tau_{i,m})$, $\forall i\in[N]$. Then,

$$\begin{aligned} z_{i,k}(t) = \sum_j \Bigg\{ & \sum_{h=1}^{n_{j,k}(t)-1} \left[W^{t-\tau_{j,h}} - W^{t-\tau_{j,h+1}}\right]_{ij}\bar{x}_{j,k}(\tau_{j,h})\\ & + \left[W^{t-\tau_{j,n_{j,k}(t)}}\right]_{ij}\bar{x}_{j,k}(\tau_{j,n_{j,k}(t)}) \Bigg\}, \end{aligned} \tag{21}$$

where we use $[\cdot]_{ij}$ to denote the ijth entry of a matrix. It is not hard to see from above equation that $c^{(\tau)}_{i,k,j}(t)=0$ when $\tau\neq\tau_{j,1},\tau_{j,2},\dots,\tau_{j,n_{j,k}(t)}$. Specifically, for all $i\in[N]$ and $k\in[M]$, it holds that

$$\begin{aligned} c^{(0)}_{i,k,j}(t) = \Bigg[& \sum_{h=1}^{n_{j,k}(t)-1} \frac{W^{t-\tau_{j,h}} - W^{t-\tau_{j,h+1}}}{h}\\ & + \frac{W^{t-\tau_{j,n_{j,k}(t)}}}{n_{j,k}(t)} \Bigg]_{ij}, \end{aligned} \tag{22}$$

which can also be written as

$$c^{(0)}_{i,k,j}(t) = \left[W^t - \sum_{h=2}^{n_{j,k}(t)} \frac{W^{t-\tau_{j,h}}}{(h-1)h} \right]_{ij}.$$

From (20), $c^{(0)}_{i,k,j}(t)$ satisfies

$$\begin{aligned} |c^{(0)}_{i,k,j}(t)| &\leq \frac{1}{N}\left(1-\sum_{h=2}^{n_{j,k}(t)}\frac{1}{(h-1)h}\right)+c\rho_2^t\\ &\quad + \sum_{h=2}^{n_{j,k}(t)}\frac{c\rho_2^{t-\tau_{j,h}}}{(h-1)h}\\ &= \frac{1}{Nn_{j,k}(t)}+c\rho_2^t+\sum_{h=2}^{n_{j,k}(t)}\frac{c\rho_2^{t-\tau_{j,h}}}{(h-1)h}. \end{aligned}$$

Since $0<\rho_2<1$, the smaller $t-\tau_{j,h}$ is, the larger the right side of the inequality would be, so $\rho_2^{t-\tau_{j,h}}\leq\rho_2^{n_{j,k}(t)-h}$. Thus, for $l\in(2,n_{j,k}(t))$,

$$\begin{aligned} \sum_{h=2}^{n_{j,k}(t)}\frac{\rho_2^{t-\tau_{j,h}}}{(h-1)h} &\leq \sum_{h=2}^{n_{j,k}(t)}\frac{\rho_2^{n_{j,k}(t)-h}}{(h-1)h} \\ &= \left(\sum_{h=2}^{l}+\sum_{h=l+1}^{n_{j,k}(t)}\right)\frac{\rho_2^{n_{j,k}(t)-h}}{(h-1)h}\\ &= \rho_2^{n_{j,k}(t)-1}+\sum_{h=2}^{l}\frac{\rho_2^{n_{j,k}(t)-h}(1-\rho_2)}{h}-\frac{\rho_2^{n_{j,k}(t)-l}}{l}\\ &\quad + \sum_{h=l+1}^{n_{j,k}(t)}\frac{\rho_2^{n_{j,k}(t)-h}}{(h-1)h}\\ &\leq \rho_2^{n_{j,k}(t)-1}+\sum_{h=2}^{l-1}\rho_2^{n_{j,k}(t)-h}(1-\rho_2)+\sum_{h=l+1}^{n_{j,k}(t)}\frac{1}{(h-1)h}\\ &= \rho_2^{n_{j,k}(t)-l+1}+\frac{1}{l}-\frac{1}{n_{j,k}(t)}. \end{aligned} \tag{23}$$

[2] Since W is symmetric, all its eigenvalues are real, and ρ_2 equals its second largest singular value.

Let $s = n_{j,k}(t) - l$. Then,

$$\sum_{h=2}^{n_{j,k}(t)} \frac{\rho_2^{n_{j,k}(t)-h}}{(h-1)h} \le \rho_2^{s+1} + \frac{s}{n_{j,k}(t)(n_{j,k}(t)-s)}.$$

Setting $l = n_{j,k}(t) - \lceil 2\log_{1/\rho_2} n_{j,k}(t) \rceil$, it follows that $s = \lceil 2\log_{1/\rho_2} n_{j,k}(t) \rceil$ and

$$\begin{aligned}&\lim_{n_{j,k}(t)\to\infty} \left[\left(\sum_{h=2}^{n_{j,k}(t)} \frac{\rho_2^{n_{j,k}(t)-h}}{(h-1)h}\right)\cdot n_{j,k}(t)\right]\\ &\le \lim_{n_{j,k}(t)\to\infty} \left(\frac{1}{n_{j,k}(t)} + \frac{s}{(n_{j,k}(t)-s)}\right) = 0,\end{aligned}$$

which implies that

$$\sum_{h=2}^{n_{j,k}(t)} \frac{\rho_2^{n_{j,k}(t)-h}}{(h-1)h} = o\Big(\frac{1}{n_{j,k}(t)}\Big).$$

Similarly, $\rho_2^{n_{j,k}(t)} = o(\frac{1}{n_{j,k}(t)})$. Thus, $F_2(\epsilon)$ in Remark 4 is well-defined, and when $n_{j,k}(t) \ge F_2(\epsilon)$,

$$\rho_2^{n_{j,k}(t)} + \sum_{h=2}^{n_{j,k}(t)} \frac{\rho_2^{n_{j,k}(t)-h}}{(h-1)h} \le \frac{\epsilon}{cNn_{j,k}(t)}.$$

Then, from (23), when $n_{j,k}(t) \ge F_2(\epsilon)$,

$$\begin{aligned}\big|c_{i,k,j}^{(0)}(t)\big| &\le \frac{1}{Nn_{j,k}(t)} + c\bigg(\rho_2^t + \sum_{h=2}^{n_{j,k}(t)} \frac{\rho_2^{t-\tau_{j,h}}}{(h-1)h}\bigg)\\ &\le \frac{1}{Nn_{j,k}(t)} + c\bigg(\rho_2^{n_{j,k}(t)} + \sum_{h=2}^{n_{j,k}(t)} \frac{\rho_2^{n_{j,k}(t)-h}}{(h-1)h}\bigg)\\ &\le \frac{1+\epsilon}{Nn_{j,k}(t)}.\end{aligned}$$

It is not hard to see from the definition that $c_{i,k,j}^{(\tau_{j,h})}$ is the last $n_{j,k}(t) - h + 1$ term of $c_{i,k,j}^{(0)}(t)$ in (22) for all $h > 1$. Thus, following the same arguments as above, we can conclude that

$$\big|c_{i,k,j}^{(\tau_{j,h})}(t)\big| \le \frac{1+\epsilon}{Nn_{j,k}(t)}$$

holds under the same condition of $n_{j,k}(t)$ as above, which completes the proof. ■

Now we are in a position to prove Theorem 1.

Proof of Theorem 1: Let

$$L = \max\bigg\{\frac{12(1+\beta_i)^4 \log T}{|\mathcal{N}_i|\Delta_k^2}, K_1\bigg\}.$$

From the Decision Making step of the algorithm,

$$\begin{aligned}n_{i,k}(T) &= 1 + \sum_{t=1}^{T} \mathbb{1}(a_i(t) = k)\\ &\le L + \sum_{t=1}^{T} \mathbb{1}(a_i(t) = k,\ n_{i,k}(t-1) \ge L)\\ &\le L + \sum_{t=1}^{T} \mathbb{1}\Big(z_{i,k}(t) + C(t, n_{i,k}(t)) \ge z_{i,1}(t) +\\ &\qquad\qquad C(t, n_{i,1}(t)),\ n_{i,k}(t-1) \ge L\Big)\\ &\quad + \sum_{t=1}^{T} \mathbb{1}(a_i(t) = k,\ k \in \mathcal{A}_i(t),\ n_{i,k}(t-1) \ge L).\end{aligned}$$

Thus,

$$\begin{aligned}&\mathbf{E}(n_{i,k}(T))\\ &\le L + \sum_{t=1}^{T} \mathbf{P}\Big(z_{i,k}(t) + C(t, n_{i,k}(t)) \ge z_{i,1}(t) +\\ &\qquad\qquad C(t, n_{i,1}(t)),\ n_{i,k}(t-1) \ge L\Big)\\ &\quad + \mathbf{E}\Big(\sum_{t=1}^{T} \mathbb{1}(a_i(t) = k,\ k \in \mathcal{A}_i(t),\ n_{i,k}(t-1) \ge L)\Big), \quad (24)\end{aligned}$$

in which the second and third terms stand for the number of pulls on arm k made by agent i in case a) and case b) of the Decision Making step after agent i have pulled L times of arm k, respectively. We thus divide the remaining analysis into two parts to estimate these two terms separately.

Part A: For the second term on the right hand side of (24),

$$\begin{aligned}&\sum_{t=1}^{T} \mathbf{P}\Big(z_{i,k}(t) + C(t, n_{i,k}(t)) \ge z_{i,1}(t) + C(t, n_{i,1}(t)),\\ &\qquad n_{i,k}(t-1) \ge L\Big)\\ &\le \sum_{t=1}^{T} \sum_{N_{i1}=1}^{t-1} \sum_{N_{ik}=L}^{t-1} \mathbf{P}\Big(z_{i,k}(t) + C(t, n_{i,k}(t)) \ge z_{i,1}(t)\\ &\qquad + C(t, n_{i,1}(t)), n_{i,k}(t) = N_{ik}, n_{i,1}(t) = N_{i1}\Big)\\ &\le \sum_{t=1}^{T} \sum_{N_{i1}=K_1}^{t-1} \sum_{N_{ik}=L}^{t-1} \mathbf{P}\Big(z_{i,k}(t) + C(t, n_{i,k}(t)) \ge z_{i,1}(t)\\ &\qquad + C(t, n_{i,1}(t)), n_{i,k}(t) = N_{ik}, n_{i,1}(t) = N_{i1}\Big)\\ &\quad + \sum_{t=1}^{T} \sum_{N_{i1}=1}^{K_1-1} \mathbf{P}\big(n_{i,1}(t) = N_{i1}\big). \qquad (25)\end{aligned}$$

For the second summation of (25), it holds that

$$\begin{aligned}&\sum_{t=1}^{T} \sum_{N_{i1}=1}^{K_1-1} \mathbf{P}(n_{i,1}(t) = N_{i1})\\ &= \sum_{N_{i1}=1}^{K_1-1} \sum_{t=1}^{T} \mathbf{P}(n_{i,1}(t) = N_{i1})\\ &\le \sum_{N_{i1}=1}^{K_1-1} 1 = K_1 - 1. \qquad (26)\end{aligned}$$

For each term in the first summation of (25), it can be divided into three cases as follows:

$$\begin{aligned}&\mathbf{P}\Big(z_{i,k}(t)+C(t,n_{i,k}(t)\geq z_{i,1}(t)+C(t,n_{i,1}(t)),\\&\qquad n_{i,k}(t)=N_{ik},n_{i,1}(t)=N_{i1}\Big)\\&\leq\mathbf{P}\Big(z_{i,k}(t)-\mu_k\geq C(t,n_{i,k}(t)),\ n_{i,k}(t)=N_{ik}\Big)\\&+\mathbf{P}\Big(\mu_1-z_{i,1}(t)\geq C(t,n_{i,1}(t)),\ n_{i,1}(t)=N_{i1}\Big)\\&+\mathbf{P}\Big(\mu_1-\mu_k<2C(t,n_{i,k}(t)),\ n_{i,k}(t)=N_{ik}\Big),\end{aligned}\tag{27}$$

where $N_{ik}\geq L\geq K_1$ and $N_{i1}\geq K_1$. As the analyses for estimating the first two terms on the right hand side of (27) are the same, we will only work on the first one in the sequel.

Note that $z_{i,k}(t)=\sum_{j\in[N]}\sum_{\tau=1}^{t}c_{i,k,j}^{(\tau)}(t)X_{j,k}(\tau)$. We thus decompose $z_{i,k}(t)-\mu_k$ as

$$\begin{aligned}z_{i,k}(t)-\mu_k=&\sum_{j\in[N]}\sum_{\tau=1}^{t}c_{i,k,j}^{(\tau)}(t)\big(X_{j,k}(\tau)-\mu_{jk}\big)\\&+\Big(\sum_{j\in[N]}\sum_{\tau=1}^{t}c_{i,k,j}^{(\tau)}(t)\mu_{jk}-\mu_k\Big).\end{aligned}\tag{28}$$

We first bound the second term in (28). Note that from (21),

$$\begin{aligned}&\sum_{j\in[N]}\sum_{\tau=1}^{t}c_{i,k,j}^{(\tau)}(t)X_{j,k}(\tau)\\&=\sum_{j\in[N]}\Big\{\sum_{\tau=0}^{t-1}[W^{t-\tau}-W^{t-\tau-1}]_{ij}\bar{x}_{j,k}(\tau)+[W^0]_{ij}\bar{x}_{j,k}(t)\Big\},\end{aligned}$$

which, with the definition of $\bar{x}_{i,k}(t)$, implies that

$$\begin{aligned}\sum_{\tau=1}^{t}c_{i,k,j}^{(\tau)}(t)X_{j,k}(\tau)=\Big\{&\sum_{\tau=0}^{t-1}[W^{t-\tau}-W^{t-\tau-1}]_{ij}\bar{x}_{j,k}(\tau)\\&+[W^0]_{ij}\bar{x}_{j,k}(t)\Big\}.\end{aligned}$$

It is clear that the sum of the coefficients of $X_{j,k}(\tau)$, $j\in[N]$ in (2) equals 1 for any fixed $i\in[N]$ and $k\in[M]$. Thus,

$$\sum_{\tau=1}^{t}c_{i,k,j}^{(\tau)}(t)=\sum_{\tau=0}^{t-1}[W^{t-\tau}-W^{t-\tau-1}]_{ij}+[W^0]_{ij}=[W^t]_{ij},$$

which implies that

$$\sum_{j\in[N]}\sum_{\tau=1}^{t}c_{i,k,j}^{(\tau)}(t)\mu_{jk}=W^t\begin{bmatrix}\mu_1&\mu_2&\dots&\mu_M\end{bmatrix}^\top.$$

Then, from (19),

$$\begin{aligned}&\Big|\sum_{j\in[N]}\sum_{\tau=1}^{t}c_{i,k,j}^{(\tau)}(t)(\mu_{jk}-\mu_k)\Big|\\&=\Big|\Big(W^t-\frac{1}{N}\mathbf{1}\mathbf{1}^\top\Big)\begin{bmatrix}\mu_1&\mu_2&\dots&\mu_M\end{bmatrix}^\top\Big|\\&\leq\Big\|W^t-\frac{1}{N}\mathbf{1}\mathbf{1}^\top\Big\|_2\cdot\|\begin{bmatrix}\mu_1&\mu_2&\dots&\mu_M\end{bmatrix}\|_2\leq c\sqrt{N}\rho_2^t.\end{aligned}$$

To proceed, let

$$F_1(\beta_i)=\max\Big\{\frac{1+\beta_i}{\beta_i\Delta},\ \inf_x\Big\{c\sqrt{N}\rho_2^t<\frac{1}{2t},\forall t\geq x\Big\}\Big\},\tag{29}$$

where $\Delta=\min_{k:\Delta_k>0}\Delta_k$. Since $t\rho_2^t$ converges to 0 when t goes to infinity, $F_1(\beta_i)$ is well-defined, and when $t\geq F_1(\beta_i)$, it holds that

$$\Big|\sum_{j\in[N]}\sum_{\tau=1}^{t}c_{i,k,j}^{(\tau)}(t)(\mu_{jk}-\mu_k)\Big|\leq c\sqrt{N}\rho_2^t\leq\frac{1}{2t}.\tag{30}$$

To simplify notation, let

$$h(n_{i,k}(t))=C(t,n_{i,k}(t))-\frac{1}{2t}$$

for each time $t>0$. From (21) and (30),

$$\begin{aligned}&\mathbf{P}\Big(z_{i,k}(t)-\mu_k\geq C(t,n_{i,k}(t)),\ n_{i,k}(t)=N_{ik}\Big)\\&\leq\mathbf{P}\Big(\sum_{j\in[N]}\sum_{\tau=1}^{t}c_{i,k,j}^{(\tau)}(t)\big(X_{j,k}(\tau)-\mu_{jk}\big)\geq h(n_{i,k}(t),\\&\qquad n_{i,k}(t)=N_{ik}\Big)\\&\leq\mathbf{P}\Big(\sum_{j\in[N]}\sum_{h=1}^{N_{jk}}c_{i,k,j}^{(\tau_{j,h})}(t)\big(X_{j,k}(\tau_{j,h})-\mu_{jk}\big)\geq h(N_{ik}),\\&\qquad n_{j,k}(t)=N_{jk}\text{ for all }j\in[N]\Big),\end{aligned}\tag{31}$$

where $\tau_{j,h}$ is defined in the proof of Lemma 9. For any fixed $i\in[N]$ and $k\in[M]$, define the following set $\mathcal{B}$ and event $\mathcal{C}$:

$$\mathcal{B}=\Big\{(N_{1k},\dots,N_{Nk})\ :\ N_{jk}\in\big[\frac{N_{ik}}{2},\frac{3N_{ik}}{2}\big],\ \forall\ j\in[N]\Big\},$$

$$\mathcal{C}=\Big\{\sum_{j\in[N]}\sum_{h=1}^{N_{jk}}c_{i,k,j}^{(\tau_{j,h})}(t)\big(X_{j,k}(\tau_{j,h})-\mu_{jk}\big)\geq h(N_{ik})\Big\}.$$

From Lemma 6, when $N_{ik}\geq L\geq K_1$, $\mathcal{B}$ is always nonempty. Then, expanding the right hand side of (31),

$$\begin{aligned}&\mathbf{P}\Big(z_{i,k}(t)-\mu_k\geq C(t,n_{i,k}(t)),\ n_{i,k}(t)=N_{ik}\Big)\\&\leq\max_{\mathcal{B}}\mathbf{P}(\mathcal{C})\mathbf{P}\Big(n_{j,k}(t)=N_{jk}\text{ for all }j\in[N]\mid\mathcal{C}\Big)\\&\leq\max_{\mathcal{B}}\mathbf{P}(\mathcal{C}).\end{aligned}$$

Note that for any fixed time t, all $c_{i,k,j}^{(\tau_{j,h})}(t)$ are constants and all $X_{j,k}(\tau_{j,h})$ are i.i.d. samples drawn from arm k for all $j\in[N]$ and $h\in\{1,2\dots,N_{jk}\}$. Then, applying (9),

$$\begin{aligned}&\mathbf{P}\Big(z_{i,k}(t)-\mu_k\geq C(t,n_{i,k}(t)),\ n_{i,k}(t)=N_{ik}\Big)\\&\leq\max_{\mathcal{B}}\mathbf{P}(C)\leq\max_{\mathcal{B}}\ \exp\Bigg(-\frac{2h^2(n_{i,k}(t))}{\sum_{j\in[N]}\sum_{h=1}^{N_{jk}}c_{i,k,j}^{(\tau_{j,h})}(t)}\Bigg)\\&\leq\max_{\mathcal{B}}\ \exp\Bigg(-\frac{\frac{6(1+\beta_i)^2\log t}{NN_{ik}}}{\sum_{j\in[N]}\sum_{n=1}^{N_{jk}}\frac{(1+\beta_i)^2}{N^2N_{jk}^2}}\Bigg)\\&\leq\max_{\mathcal{B}}\ \exp\Bigg(-\frac{6N}{\sum_{j\in[N]}\frac{N_{ik}}{N_{jk}}}\log t\Bigg)\\&\leq\exp\Bigg(-\frac{6N}{3N/2}\log t\Bigg)\leq t^{-4}.\end{aligned}$$

Similarly, for the second term on the right hand side of (27), it holds that for $N_{i1} \geq K_1$,

$$\mathbf{P}\Big(\mu_1 - z_{i,1}(t) \geq C(t, n_{i,1}(t)),\ n_{i,1}(t) = N_{i1}\Big) \leq t^{-4}.$$

As for the last term on the right hand side of (27), it is easy to verify that when $n_{i,k}(t) \geq \frac{12(1+\beta_i)^4 \log T}{|\mathcal{N}_i|\Delta_k^2}$, it always holds that $\mu_1 - \mu_k > 2C(t, n_{i,k}(t))$. Substituting the preceding results to (27), it follows that when $N_{ik}, N_{i1} \geq K_1$,

$$\begin{aligned}\mathbf{P}\Big(& z_{i,k}(t) + C(t, n_{i,k}(t)) \geq z_{i,1}(t) + C(t, n_{i,1}(t)),\\ & n_{i,k}(t) = N_{ik},\ n_{i,1}(t) = N_{i1}\Big) \leq 2t^{-4}.\end{aligned}$$

Combining this with (25) and (26),

$$\begin{aligned}&\sum_{t=1}^{T} \mathbf{P}\Big(z_{i,k}(t) + C(t, n_{i,k}(t)) \geq z_{i,1}(t) + C(t, n_{i,1}(t)),\\ &\qquad\qquad n_{i,k}(t-1) \geq L\Big)\\ &\leq \sum_{t=1}^{T} \sum_{N_{i1}=K_1}^{t-1} \sum_{N_{ik}=L}^{t-1} 2t^{-4} + K_1 - 1 \leq \frac{\pi^2}{3} + K_1 - 1. \quad (32)\end{aligned}$$

Part B: Now what is left is to estimate the last term on the right hand side of (24), which stands for the expected number of pulls on arm k by agent i in case b) in the Decision Making step after time instance t_1 defined as $t_1 \triangleq \arg\min_t\{n_{i,k}(t-1) = L\}$, and can be intuitively understood as the extra pulls an agent makes in order to "catch up" with the largest sample count among all the agents on arm k. In this sense, it is easy to see that any decision made according to case **b** would not affect the global maximal sample count, while any pull made according to case **a** would at most increase the global maximal sample count by 1. With this is mind, let $i_1 = \arg\max_i n_{i,k}(t_1)$ be the agent who makes the most pulls on arm k at t_1, and $g_{i,k}$ be the number of pulls made by i on arm k in case **a** after t_1, then for all $i \in [N]$, it holds that

$$n_{i,k}(T) \leq \max_j n_{j,k}(T) \leq n_{i_1,k}(t_1) + \sum_{j\in[N]} g_{j,k}.$$

Thus,

$$\begin{aligned}&\mathbf{E}\Big(\sum_{t=1}^{T} \mathbb{1}(a_i(t) = k,\ k \in \mathcal{A}_i(t),\ n_{i,k}(t-1) \geq L)\Big)\\ &= \mathbf{E}\Big(n_{i,k}(T) - n_{i,k}(t_1) - g_{i,k}\Big)\\ &\leq n_{i_1,k}(t_1) - n_{i,k}(t_1) + \mathbf{E}\Big(\sum_{j\in[N]} g_{j,k} - g_{i,k}\Big).\end{aligned}$$

From (32), $\mathbf{E}(g_{j,k}) = \frac{\pi^2}{3} + K_1 - 1$ for all $j \in [N]$, and from (18), $n_{i_1,k}(t_1) - n_{i,k}(t_1) \leq M^2 + 2MN + N$, which together imply that

$$\begin{aligned}&\mathbf{E}\Big(\sum_{t=1}^{T} \mathbb{1}(a_i(t) = k,\ k \in \mathcal{A}_i(t),\ n_{i,k}(t-1) \geq L)\Big)\\ &\leq M^2 + 2MN + N + (N-1)\Big(\frac{\pi^2}{3} + K_1 - 1\Big) = K_2.\end{aligned}$$

Combining this with (24) and (32), we have

$$\mathbf{E}\big(n_{i,k}(T)\big) \leq L + K_2.$$

Thus, from (6),

$$R_i(T) = \sum_{k:\Delta_k>0} \mathbf{E}(n_{i,k}(T))\Delta_k \leq \sum_{k:\Delta_k>0} (L + K_2)\Delta_k,$$

which completes the proof. ■

B. Time-varying Neighbor Graph

Now we analyze the time-varying neighbor graph case for proving Theorem 2.

We need the following concept for a time-varying graph sequence. Define a route over a given sequence of directed graphs with the same vertex set, $\mathbb{G}_1, \mathbb{G}_2, \ldots, \mathbb{G}_p$, as a sequence of vertices $i_0, i_1, \ldots, i_p$ such that for each $k \in \{1, 2, \ldots, q\}$, (i_{k-1}, i_k) is an arc in $\mathbb{G}_k$. It is easy to see that a route over a sequence of directed graphs which are all the same graph $\mathbb{G}$, is a walk in $\mathbb{G}$. The definition of composition of directed graphs with self-arcs at all vertices implies that if $i = i_0, i_1, \ldots, i_p = j$ is a route over a sequence $\mathbb{G}_1, \mathbb{G}_2, \ldots, \mathbb{G}_p$, then (i, j) must be an arc in the composed graph $\mathbb{G}_q \circ \mathbb{G}_{q-1} \circ \cdots \circ \mathbb{G}_1$, and vice versa.

In this subsection, we use $d_{i,j}(t)$ to denote the shortest period of time, starting from time instant t, with which there is a route from vertex i to vertex j over the time-varying neighbor graph sequence $\{\mathbb{G}(t)\}$; that is,

$$d_{i,j}(t) = \arg\min_{t'} \big\{(i, j) \text{ is an arc in } \mathbb{G}(t + t') \circ \cdots \circ \mathbb{G}(t)\big\}.$$

Since $\mathbb{G}(0), \mathbb{G}(1), \mathbb{G}(2), \ldots$ are uniformly strongly connected by sub-sequences of length l, by definition and the fact that any sequence of $N-1$ or more strongly connected graphs with self-arcs at all N vertices is a complete graph [30, Proposition 4], the composition of any $(N-1)l$ consecutive graphs in $\mathbb{G}(0), \mathbb{G}(1), \mathbb{G}(2), \ldots$ is a complete graph. Thus, for any $t \geq 0$, there is an arc from vertex i to vertex j for any $i, j \in [N]$ in the composed graph $\mathbb{G}(t+(N-1)l-1) \circ \cdots \circ \mathbb{G}(t+1) \circ \mathbb{G}(t)$. Combining this and the fact that for any two directed graphs with self-arcs at all vertices, $\mathbb{G}_p$ and $\mathbb{G}_q$, the arc set of $\mathbb{G}_p$ is a subset of the arc set of $\mathbb{G}_q \circ \mathbb{G}_p$, we have the following uniform bound for $d_{i,j}(t)$:

$$d_{i,j}(t) \leq (N-1)l.$$

It is worth noting that $d_{i,j}(t)$ is not necessarily increasing with respect to t. For example, if (i, j) is not an arc in $\mathbb{G}(1)$ but is one in $\mathbb{G}(2)$, then $d_{i,j}(1) > d_{i,j}(2) = 1$.

We use the following notation to describe the latest information j receives from i till time t :

$$D_{i,j}(t) = \arg\max_{t'}\{t - (N-1)l < t' \leq t,\ t' + d_{i,j}(t') \leq t\}. \quad (33)$$

It can be derived directly from the definition that (i, j) is an arc of $\mathbb{G}(t) \circ \cdots \circ \mathbb{G}(D_{i,j}(t))$.

With the preceding definitions, variables $n_{i,k}(t)$ and $m_{i,k}(t)$ in Algorithm 1 over time-varying neighbor graphs have similar properties to those in Lemmas 4–6 derived for time-invariant neighbor graphs.

Lemma 10: For any $i \in [N]$ and $k \in [M]$,

$$m_{i,k}(t) = \max_{j \in [N]} \left\{ n_{j,k}(D_{j,i}(t)) \right\}. \tag{34}$$

Proof: We will prove the lemma by induction on t. For the basis step, suppose that $t = 0$. In this case, $m_{i,k}(0) = \max\{n_{i,k}(0),\ m_{j,k}(0),\ j \in \mathcal{N}_i(0)\} = 1$. Note that $D_{j,i}(t) \leq t$, so $\max_{j \in [N]} \{n_{j,k}(D_{j,i}(0))\} = \max_{j \in [N]} \{n_{i,k}(0)\} = 1$. Thus, (34) holds when $t = 0$.

For the inductive step, assume (34) holds at time t, and now consider time $t+1$. Note that

$$\begin{aligned} m_{i,k}(t+1) &= \max\{n_{i,k}(t+1),\ m_{j,k}(t),\ j \in \mathcal{N}_i(t)\} \\ &= \max\{n_{i,k}(t+1),\ n_{h,k}(D_{h,j}(t)), \\ &\qquad h \in [N],\ j \in \mathcal{N}_i(t)\}. \end{aligned}$$

Since (h, j) is an arc of graph $\mathbb{G}(t) \circ \cdots \circ \mathbb{G}(D_{h,j}(t))$ and $j \in \mathcal{N}_i(t{+}1)$ (i.e., (j, i) is an arc of graph $\mathbb{G}(t{+}1)$), it follows that (h, i) must be an arc of graph $\mathbb{G}(t{+}1) \circ \cdots \circ \mathbb{G}(D_{h,j}(t))$. It implies that $D_{h,j}(t) \in \{t' : t' + d_{h,i}(t') \leq t+1\}$. Thus,

$$D_{h,i}(t+1) = \arg\max_{t'} \{t' + d_{h,i}(t') \leq t+1\} \geq D_{h,j}(t).$$

Since $n_{i,k}(t)$ is a non-decreasing function of t,

$$\begin{aligned} m_{i,k}(t+1) &= \max\big\{n_{i,k}(t+1),\ n_{h,k}(D_{h,j}(t)), \\ &\qquad h \in [N],\ j \in \mathcal{N}_i(t)\big\} \\ &\leq \max_{h \in [N]} \big\{n_{i,k}(t+1),\ n_{h,k}(D_{h,i}(t+1))\big\} \\ &= \max_{j \in [N]} \big\{n_{j,k}(D_{j,i}(t+1))\big\}. \end{aligned} \tag{35}$$

Fix any $j \in [N]$ and let $p = (j, v_{d_{j,i}(D_{j,i}(t))}, \ldots, v_2, i)$ be a route from vertex j to vertex i in $\mathbb{G}(t{+}1) \circ \cdots \circ \mathbb{G}(D_{j,i}(t{+}1))$. Then, from (4),

$$\begin{aligned} m_{i,k}(t+1) &\geq m_{v_2,k}(t) \geq \cdots \\ &\geq m_{v_{d_{j,i}(D_{j,i}(t))},k}(t - d_{j,i}(D_{j,i}(t+1)) + 1) \\ &\geq m_{j,k}(t - d_{j,i}(D_{j,i}(t+1))) \\ &\geq m_{j,k}(D_{j,i}(t+1)) \geq n_{j,k}(D_{j,i}(t+1)). \end{aligned} \tag{36}$$

Since j is arbitrarily chosen from $[N]$, it follows that

$$m_{i,k}(t+1) \geq \max_{j \in [N]} \big\{n_{j,k}(D_{j,i}(t))\big\}.$$

Combining with (35),

$$m_{i,k}(t+1) = \max_{j \in [N]} \big\{n_{j,k}(D_{j,i}(t+1))\big\}.$$

Thus, (34) holds at $t+1$, which completes the induction. ■

Lemma 11: For any $i \in [N]$ and $k \in [M]$,

$$n_{i,k}(t) > m_{i,k}(t) - M(M + (N-1)l).$$

Proof: We will prove the lemma by contradiction. Suppose that, to the contrary, there exist i and k_1 such that $n_{i,k_1}(t) \leq m_{i,k_1}(t) - M(M + (N-1)l)$. Let t' denote the first time at which the inequality holds. Then,

$$n_{i,k_1}(t') = m_{i,k_1}(t') - M(M + (N-1)l) - C,$$

where $0 \leq C \leq (N-1)l$. Here t' must exist. To see this, note that at initial time $t = 0$, $n_{i,k}(0) > m_{i,k}(1) - M(M + (N-1)l)$. In addition, $n_{i,k}(t)$ either does not change or increases at most 1 at each time by its definition, and similarly, $m_{i,k}(t)$ either does not change or increases at most $(N-1)l$ at each time. Thus, if there exists some time instance t at which $n_{i,k_1}(t) < m_{i,k_1}(t) - M(M + (N-1)l)$, there must exist another time t' between 0 and t such that $n_{i,k_1}(t') = m_{i,k_1}(t') - M(M + (N-1)l) - C$. From Lemma 10, there exists a $j \in [N]$ such that

$$m_{i,k_1}(t') = n_{j,k_1}(D_{j,i}(t')). \tag{37}$$

Then, $n_{j,k_1}(D_{j,i}(t')) - n_{i,k_1}(t') = M(M + (N-1)l) + C$. Also from Lemma 10, $m_{i,k_1}(t) \geq n_{j,k_1}(D_{j,i}(t))$ always holds. Thus, for $t < t'$,

$$\begin{aligned} & n_{j,k_1}(D_{j,i}(t)) - n_{i,k_1}(t') \leq n_{j,k_1}(D_{j,i}(t)) - n_{i,k_1}(t) \\ &\leq m_{i,k_1}(t) - n_{i,k_1}(t) < M(M + (N-1)l). \end{aligned}$$

This implies that there exists a time instance $t \in (D_{j,i}(t'-1), D_{j,i}(t')]$ at which agent j pulls arm k_1. Let $t'' \leq D_{j,i}(t')$ be the latest time instance, but no later than $D_{j,i}(t')$, at which agent j pulls arm k_1. Then, from (37),

$$n_{j,k_1}(t'') = n_{j,k_1}(D_{j,i}(t'+1)) = m_{i,k_1}(t'+1), \tag{38}$$

and thus $n_{j,k_1}(t'') - n_{i,k_1}(t') \geq M(M + (N-1)l)$. Since $t'' \leq D_{j,i}(t') \leq t'$,

$$n_{j,k_1}(t'') - n_{i,k_1}(t'') \geq M(M + (N-1)l).$$

Since each agent must pull an arm at each time, $\sum_k n_{i,k}(t) = t + M,\ \forall i \in [N]$. Then,

$$\begin{aligned} \sum_{k \in [M] \backslash k_1} n_{i,k}(t'') - \sum_{k \in [M] \backslash k_1} n_{j,k}(t'') &= n_{j,k_1}(t'') - n_{i,k_1}(t'') \\ &\geq M(M + (N-1)l). \end{aligned}$$

Applying the Pigeonhole principle, $\exists\ k_2 \in [M]$ such that

$$\begin{aligned} n_{i,k_2}(t'') - n_{j,k_2}(t'') &\geq \frac{M(M + (N-1)l)}{M-1} \\ &> M + (N-1)l. \end{aligned}$$

Since $n_{i,k}(t{+}1) \leq n_{i,k}(t){+}1$ and, from (33), $t'' \leq D_{j,i}(t'') + (N-1)l$,

$$\begin{aligned} & n_{i,k_2}(D_{i,j}(t'')) - n_{j,k_2}(t'') \\ &\geq n_{i,k_2}(t'' - (N-1)l) - n_{j,k_2}(t'') \\ &\geq n_{i,k_2}(t'') - n_{j,k_2}(t'') - (N-1)l \\ &> M + (N-1)l - (N-1)l = M. \end{aligned}$$

Using (36), $m_{j,k_2}(t'') \geq n_{i,k_2}(D_{i,j}(t''))$. Thus,

$$m_{j,k_2}(t'') - n_{j,k_2}(t'') > M.$$

From the above analysis, agent j must pull arm k_1 at time t''. According to the Decision Making step of the algorithm,

$$m_{j,k_1}(t'') - n_{j,k_1}(t'') \geq M > 0. \tag{39}$$

Note that from (36),

$$m_{i,k_1}(t') \geq m_{j,k_1}(D_{j,i}(t')). \tag{40}$$

Combining (38)–(40) together,

$$\begin{aligned} n_{j,k_1}(t'') &= m_{i,k_1}(t') \geq m_{j,k_1}(D_{j,i}(t')) \geq m_{j,k_1}(t'') \\ &> n_{j,k_1}(t''), \end{aligned}$$

which is a contradiction. Therefore, the lemma is true. ■

Lemma 12: For any $i \in [N]$ and $k \in [M]$, if $n_{i,k}(t) \geq 2(M^2 + (M+1)(N-1)l)$, then for any $j \in [N]$, there holds

$$\frac{1}{2} n_{i,k}(t) \leq n_{j,k}(t) \leq \frac{3}{2} n_{i,k}(t).$$

Proof: From (36), $m_{i,k}(t) \geq n_{j,k}(D_{j,i}(t))$ for all $i, j \in [N]$. With $n_{i,k}(t+1) \leq n_{i,k}(t) + 1$ and (33), for any $i, j \in [N]$,

$$\begin{aligned} m_{i,k}(t) &\geq n_{j,k}(D_{j,i}(t)) \geq n_{j,k}(t - (N-1)l) \\ &\geq n_{j,k}(t) - (N-1)l. \end{aligned}$$

Combining this inequality and Lemma 11, for any $i, j \in [N]$,

$$n_{i,k}(t) \geq n_{j,k}(t) - (M^2 + (M+1)(N-1)l).$$

Since the above inequality holds for any $i, j \in [N]$, exchanging indices i and j in the inequality yields

$$n_{j,k}(t) \geq n_{i,k}(t) - (M^2 + (M+1)(N-1)l).$$

Then, when $n_{i,k}(t) \geq 2(M^2 + (M+1)(N-1)l)$,

$$n_{i,k}(t) - \frac{1}{2} n_{i,k}(t) \leq n_{j,k}(t) \leq n_{i,k}(t) + \frac{1}{2} n_{i,k}(t)$$

for any $j \in [N]$, which completes the proof. ■

We are now in a position to prove Theorem 2.

Proof of Theorem 2: Following the same procedure of the proof of Lemma 9 and replacing W^t with $W(t-1)\cdots W(0)$, we can derive the estimation of the coefficient of each $X_{j,k}(\tau)$, $j \in [N]$, $\tau \in \{0, \ldots, t\}$ in $z_{i,k}(t)$ for any $i \in [N]$, $k \in [M]$, and $t > 0$ in terms of c' and ρ given in (7). With these estimations at hand, Theorem 2 can be proved using the same arguments as those in the proof of Theorem 1. ■

V. Simulations

In this section, we provide a set of simulations to validate the theorems and discuss the algorithm performance.

Note that the parameter β_i can be taken arbitrarily close to 0 in both Theorem 1 and Theorem 2. The following simulations are all performed with $\beta_i = 0.01$ except for Section V-A in which we compare the performance of each agent i with different β_i values.

A. Comparison with Different β_i Values

For the first experiment, we consider the distributed multi-armed bandit problem with 20 arms and 3 agents whose neighbor graph is a complete graph. Thus, $\mathcal{N}_i = N = 3$. Each agent i chooses β_i as 0.01, 0.1, or 1 in the design of $C(t, n_{i,k}(t))$. The reward distribution $X_{i,k}(t)$ is set to be bounded and within $[0, 1]$. The expectations $\mu_{i,k}$ of $X_{i,k}(t)$, $i \in [N]$ are set to be different. We average the results of 50 Monte-Carlo runs to compare the regret of each agent; see Fig 1. The total time T is chosen to be 10000.

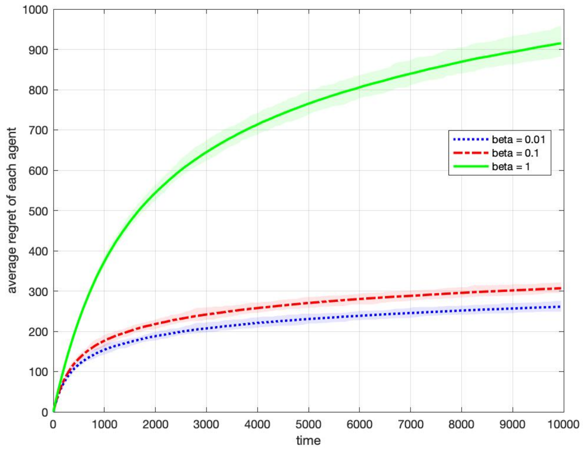

Fig. 1. Performance comparison with different $\boldsymbol{\beta_i}$ values

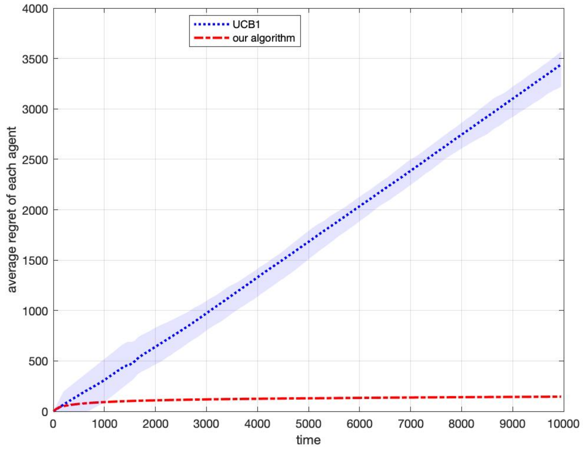

Fig. 2. Simulation results comparing the average regret for agents using Algorithm 1 and UCB1 [3] under the heterogeneous setting

It can be seen from Fig. 1 that when other factors (e.g., the number of neighbors) are fixed, the lower β_i is, the better is the algorithm performance, which is consistent with our theoretical results in Theorem 1 and Theorem 2. Because of this, all β_i are set to be 0.01 in the following simulations.

B. Comparison with the Non-cooperative Case

For this experiment, we consider the same distributed multi-armed bandit problem setting as the previous one, namely there are 20 arms and 3 agents whose neighbor graph is a complete graph, the reward distribution $X_{i,k}(t)$ is bounded within $[0, 1]$, and the expectations $\mu_{i,k}$ of $X_{i,k}(t)$, $i \in [N]$ are all different. The agents aim to find the arm with highest average expectation μ_k so as to minimize the cumulative regret. We average the results of 50 Monte-Carlo runs to compare the averaged regret of all 3 agents using Algorithm 1 with that of the non-cooperative case in which each agent independently applies UCB1 [3], a classic single-agent MAB algorithm; see Fig. 2 and Fig. 3. The total time T is chosen to be 10000.

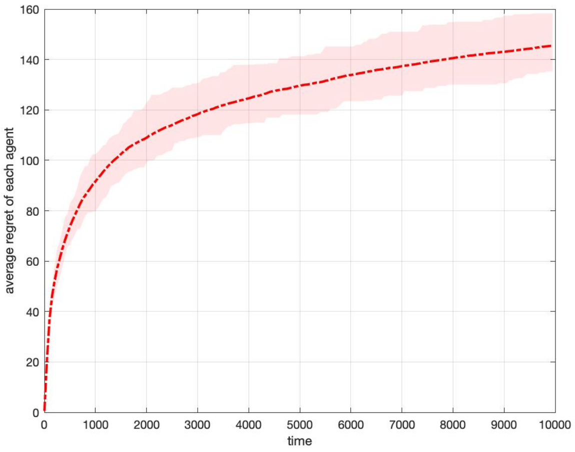

Fig. 3. Simulation of the average regret for agents using Algorithm 1 under the heterogeneous setting

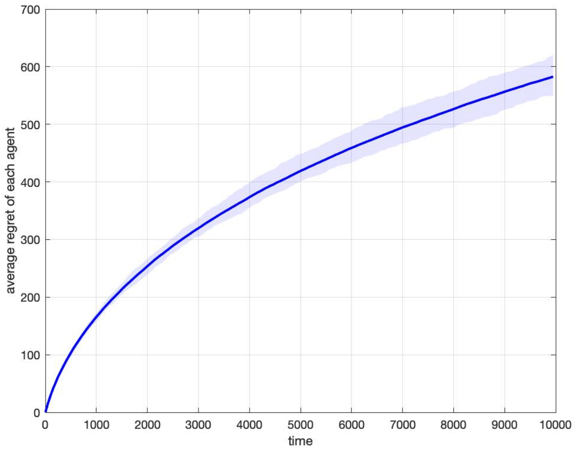

Fig. 5. Simulation of the average regret over the time-varying graph

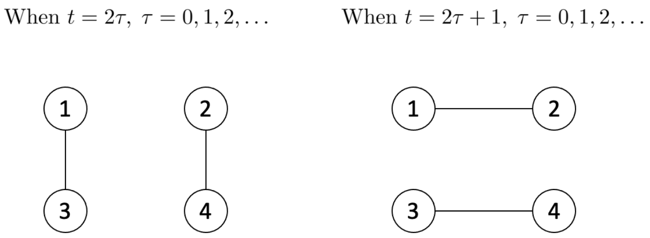
When $t = 2\tau,\ \tau = 0, 1, 2, \ldots$
When $t = 2\tau + 1,\ \tau = 0, 1, 2, \ldots$
1
2
3
4
1
2
3
4

Fig. 4. Neighbor relationships for a four-agent network

The simulation shows that the proposed Algorithm 1 works correctly for the heterogeneous setting, with the time-regret curve being logarithmic, whereas UCB1 [3] does not function well in such a setting as the time-regret curve appears to be linear, indicating that each agent cannot find the optimal arm itself without communication with its neighbors under the heterogeneous setting.

C. Time-varying Neighbor Graph

In the experiment below, we consider the distributed multi-armed bandit problem with 20 arms and over a special time-varying neighbor graph sequence. Specifically, there are 4 agents whose time-dependent neighbor relationships are described by Fig. 4. It can be seen from the figure that $\mathbb{G}(t)$ is disconnected at any time while $\mathbb{G}(2\tau+1) \circ \mathbb{G}(2\tau)$ is always connected for any nonnegative integer τ. Thus, the neighbor graph sequence is uniformly strongly connected by sub-sequences of length 2. Again, the reward distribution $X_{i,k}(t)$ is bounded within $[0, 1]$, whose expectation is different for differing agent index i, and the total time T is chosen to be 10000. See Fig. 5 for the averaged results of 50 Monte-Carlo runs of Algorithm 1.

D. Comparison between Fixed and Time-varying Graphs

We compare the performance of a multi-agent network over a fixed graph with that over a time-varying graph. Consider a six-agent network whose possible neighbor graphs are given in Fig. 6 including three graphs (a)–(c). For the fixed graph setting, the graph is a cycle graph given in (a). For the time-varying graph setting, we consider the following three cases:

1) Neighbor graph $\mathbb{G}(t)$ is (b) whenever time $t = 2\tau$, $\tau \in \{0, 1, 2, \ldots\}$, and (c) otherwise;
2) Neighbor graph $\mathbb{G}(t)$ is (a) whenever time $t = 10\tau$, $\tau \in \{0, 1, 2, \ldots\}$, and an empty graph (i.e., a graph with no edges) otherwise;
3) Neighbor graph $\mathbb{G}(t)$ is (a) whenever time $t = 1000\tau$, $\tau \in \{0, 1, 2, \ldots\}$, and an empty graph otherwise.

The regret curves are shown in Fig. 7 and Fig. 8. It is easy to see that $|\mathcal{N}_i| = \delta_i(t) = 3$ for all $i \in [N]$ and $t \geq 0$, and thus the upper confidence bound function design for the six-agent fixed graph setting and the three time-varying graph cases are identical. For both settings, we set 20 arms and $T = 10000$. The regret curves are obtained by averaging expected cumulative regret over all six agents and 50 Monte-Carlo runs.

It can be observed in Fig. 7 that the curves appear to be parallel, indicating that the corresponding coefficients of the $\log T$ term are identical and that the only difference is the constant term for the four cases (i.e., one fixed graph case and three time-varying graph cases), which is consistent with the theoretical results because $\mathcal{N}_i = \delta_i(t) = 3$. Recall the consensus convergence rates ρ_2 and ρ for fixed and time-varying graphs, respectively given in (5) and (7). It is not hard to compute that in terms of consensus convergence rate,

$$\text{Case } 3 \gg \text{Case } 2 > \text{fixed graph} > \text{Case } 1,$$

implying that Case 3 has the slowest consensus fusion and Case 1 is the fastest. From Remark 4, the constant term of the regret is determined by the consensus convergence rate, ρ_2 or ρ, and uniformly strong connectedness sub-sequence length l. The regret is an increasing function of both the consensus convergence rate and sub-sequence length l. Such a pattern is well-reflected in Fig. 7 and Fig. 8: the convergence constant for time-varying graph Case 1 is a bit better than that for the fixed graph setting; meanwhile, the length l for Case 1 ($l = 2$) is slightly worse than that in the fixed graph setting ($l = 1$),

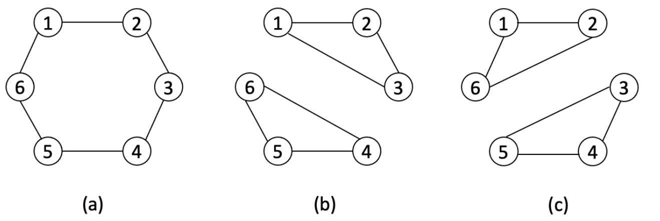

Fig. 6. Neighbor relationships for a six-agent network

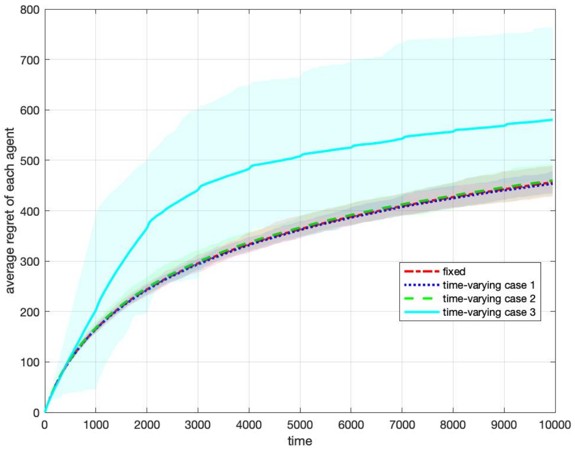

Fig. 7. Simulation comparison of the average regret between the fixed and time-varying graphs

and thus the performance difference between the two cases is almost negligible. For the time-varying graph Case 2, both the consensus convergence rate and sub-sequence length are worse than the fixed graph setting, and thus its performance is always worse than the fixed graph setting. For the time-varying graph Case 3, both the consensus convergence rate and sub-sequence length are much worse than the fixed graph setting, and thus the performance gap is relatively large. Another remarkable point is that although theoretically both the consensus convergence rate and sub-sequence length can largely affect the constant term of the regret bound (e.g., there is an $(M+1)(N-1)l$ term in the expression of $\tilde{K}_2$; see Remark 4), such a theoretical effect (e.g., a slight increase in l leads to a large increase in the regret) does not appear in the simulations. Specifically, even with $l = 10$, the performance of time-varying graph Case 2 is only slightly worse than the fixed graph setting (with $l = 1$); see Fig. 8. Only in time-varying graph Case 3 whose consensus convergence rate is sufficiently close to 1 and sub-sequence length l is sufficiently large, can we observe an obvious performance difference compared with the fixed graph setting. This observation may be due to the fact that we utilized an unknown “worst” case when deriving the regret upper bounds and indicate that the constant terms in our derived regret bounds have room to improve.

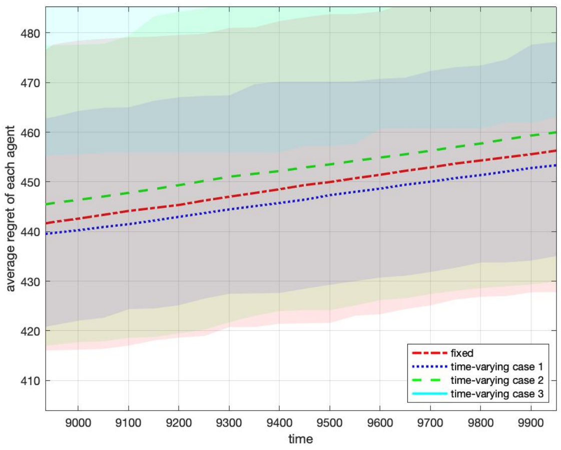

Fig. 8. A detailed-up version for Fig. 7

VI. Conclusion

In this paper, we have studied a distributed multi-armed bandit problem with heterogeneous observations of rewards over a multi-agent network. A fully distributed bandit algorithm has been proposed which is shown to achieve guaranteed regret for each agent at the order of $O(\log T)$ provided the possibly time-varying neighbor graph sequence is uniformly strongly connected. The algorithm incorporates the conventional distributed averaging algorithm with doubly stochastic matrices, which implicitly requires the underlying neighbor graph to be undirected in implementation. An immediate future direction is to relax the double stochasticity assumption for more general directed graphs allowing uni-directional communication among the agents. We will appeal to the idea of push-sum [36], which has been recently shown to be successful for the special homogeneous case [37].

For the heterogeneous setting considered here, there are many other possible formulations for each arm’s true reward mean, and the average of all agents’ observed mean rewards is not always the best choice. The special average model can be treated as a promising starting point of this line of research. Our algorithm and results can be generalized to the cases where the global reward mean is defined as any convex combination of all local reward means using the push sum idea [36] as long as each agent knows its own convex combination weight. We do not include this generalization as it will involve a different analysis approach, and leave it as a future direction. A more interesting and important future direction is to study more general and complex heterogeneous reward settings. In some network applications, the straight average, or even convex combination, of all local biased observations may not be the optimal performance index. We plan to borrow the concept of “contextual bandit” [38] to model more realistic local-global interaction in network bandit settings and “statistical heterogeneity” [39] to quantify the level of heterogeneity among the agents.

Acknowledgment

The authors are indebted to an anonymous reviewer for pointing out an error in an earlier version of the proof of Theorem 1 and wish to thank all the anonymous reviewers for their helpful comments.

References

[1] 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.

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

[5] 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.

[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] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] R. Olfati-Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[20] 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.

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

[22] 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.

[23] Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2):1–19, 2019.

[24] 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):2, 2021.

[25] R. Huang, W. Wu, J. Yang, and C. Shen. Federated linear contextual bandits. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, pages 27057–27068, 2021.

[26] A. Mitra, G.J. Pappas, and H. Hassani. Exploiting heterogeneity in robust federated best-arm identification. *arXiv preprint*, 2021. arXiv:2109.05700 [cs.LG].

[27] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of the 4th International Conference on Information Processing in Sensor Networks*, pages 63–70, 2005.

[28] B. Gharesifard and J. Cortés. Distributed strategies for generating weight-balanced and doubly stochastic digraphs. *European Journal of Control*, 18(6):539–557, 2012.

[29] A.I. Rikos and C.N. Hadjicostis. Distributed balancing with constrained integer weights. *IEEE Transactions on Automatic Control*, 64(6):2553–2558, 2018.

[30] M. Cao, A.S. Morse, and B.D.O. Anderson. Reaching a consensus in a dynamically changing environment: A graphical approach. *SIAM Journal on Control and Optimization*, 47(2):575–600, 2008.

[31] A. Nedić, A. Olshevsky, A. Ozdaglar, and J.N. Tsitsiklis. On distributed averaging algorithms and quantization effects. *IEEE Transactions on automatic control*, 54(11):2506–2517, 2009.

[32] J. Liu, S. Mou, A.S. Morse, B.D.O. Anderson, and C. Yu. Deterministic gossiping. *Proceedings of the IEEE*, 99(9):1505–1524, 2011.

[33] T. Lattimore and C. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2018.

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

[35] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems and Control Letters*, 53(1):65–78, 2004.

[36] 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.

[37] 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.

[38] A. Dubey and A. Pentland. Kernel methods for cooperative multi-agent contextual bandits. In *Proceedings of the 37th International Conference on Machine Learning*, pages 2740–2750, 2020.

[39] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar. Federated multi-task learning. In *Proceedings of the 31st Conference on Neural Information Processing Systems*, pages 4424–4434, 2017.

Jingxuan Zhu received the B.S. degree in mathematics and statistics from Wuhan University, Wuhan, Hubei Province, China, in 2018. She is currently working towards the Ph.D. degree in applied mathematics at Stony Brook University, Stony Brook, NY, USA. Her research interests are in the areas of multi-agent systems and distributed reinforcement learning.

Ji Liu received the B.S. degree in information engineering from Shanghai Jiao Tong University, Shanghai, China, in 2006, and the Ph.D. degree in electrical engineering from Yale University, New Haven, CT, USA, in 2013. He is an Assistant Professor in the Department of Electrical and Computer Engineering at Stony Brook University, Stony Brook, NY, USA. His current research interests include distributed control and optimization, distributed reinforcement learning, and resiliency of distributed algorithms.