Journal of Machine Learning Research 23 (2022) 1-24 Submitted 2/21; Revised 5/22; Published 7/22

# Multi-Agent Multi-Armed Bandits with Limited Communication

**Mridul Agarwal**[*] AGARW180@PURDUE.EDU

**Vaneet Aggarwal** VANEET@PURDUE.EDU

**Kamyar Azizzadenesheli** KAMYAR@PURDUE.EDU

*Purdue University*

**Editor:** Aurelien Garivier

## Abstract

We consider the problem where $N$ agents collaboratively interact with an instance of a stochastic $K$ arm bandit problem for $K \gg N$. The agents aim to simultaneously minimize the cumulative regret over all the agents for a total of $T$ time steps, the number of communication rounds, and the number of bits in each communication round. We present Limited Communication Collaboration - Upper Confidence Bound (LCC-UCB), a doubling-epoch based algorithm where each agent communicates only after the end of the epoch and shares the index of the best arm it knows. With our algorithm, LCC-UCB, each agent enjoys a regret of $\tilde{O}\left(\sqrt{(K/N+N)T}\right)$, communicates for $O(\log T)$ steps and broadcasts $O(\log K)$ bits in each communication step. We extend the work to sparse graphs with maximum degree $K_G$ and diameter $D$ to propose LCC-UCB-GRAPH which enjoys a regret bound of $\tilde{O}\left(D\sqrt{(K/N+K_G)DT}\right)$. Finally, we empirically show that the LCC-UCB and the LCC-UCB-GRAPH algorithms perform well and outperform strategies that communicate through a central node.

## 1. Introduction

We consider a setup where $N$ agents, connected over a network, interact with a multi-armed bandit (MAB) environment (Lattimore and Szepesvári, 2020). The agents aim to collaborate with other agents in the network to minimize their regret. The agents also aim to reduce the number of messages and the size of messages communicated with others. Consider a case of an e-commerce company serving its users by recommending its vast number of items through multiple servers for quick response times. It attempts to learn the user preferences using a MAB algorithm. If each of the multiple servers runs their own algorithm, they waste the large amount of data that other servers collect. Or, if they communicate after every recommendation, the communication complexity becomes high within the servers themselves.

As observed from the example above, communicating after each time-step is not favorable because of the increased communication cost. If $N$ agents communicate after every round to reduce the regret for $T$ time steps, their total regret is lower bounded by the regret of a super-agent solving the MAB problem with $NT$ time steps. This bounds the

[*] This work was supported in part by the U.S. National Science Foundation under grant CCF-2149588.

total regret as at least $\Omega(\sqrt{NKT})$ or a per agent regret of $\Omega(\sqrt{KT/N})$. Whereas, if the $N$ agents interact with the MAB problem independently, without any information exchange with other agents, the individual regret is upper bounded by $\tilde{O}(\sqrt{KT})$. We aim to find an algorithm that can obtain the regret bound of the super-agent setup, *i.e.*, $\tilde{O}(\sqrt{KT/N})$, though with limited communication between the agents.

We provide an algorithm, Limited Communication Collaboration - UCB, (LCC-UCB), to minimize the regret. LCC-UCB divides the arms among multiple agents, such that each agent interacts with the MAB instance but plays arms only from a subset of all the arms. The algorithm proceeds in epochs which double in duration, where the agents use the UCB algorithm to find the best arm in their smaller MAB problem and communicate at the end of each epoch. On receiving the messages from other agents, each agent updates its set of arms and restarts its algorithm. We prove the regret of LCC-UCB is upper bounded by $\tilde{O}\left(\sqrt{(K/N+N-1)\,T}\right)$. For $N=1$, the regret of the LCC-UCB algorithm reduces to the standard regret bounds of $\tilde{O}(\sqrt{KT})$.

We also consider a general setup where the network of agents may not be completely connected, and the agents may not be able to broadcast knowledge to all the other agents at once. Under such a case, we propose LCC-UCB-GRAPH algorithm that subdivides epochs into sub-epochs of equal length. The agents restart their UCB algorithm in each sub-phase with the new information available from their neighbors. We show that the regret bound of this modified algorithm with divided phases changes to $\tilde{O}\left(D\sqrt{(K/N+K_G)\,DT}\right)$, where $K_G$ is the maximum degree of the nodes in the graph. Also, the increased communication complexity of this algorithm is bounded by $O\left(K_G D \log T\right)$ message exchanges per node. The key novelty in both the algorithms is that the gap between the recommended arms and the optimal arm reduces with epochs.

Finally, we simulate and compare our algorithms with other communication protocols. We show that the algorithm behaves close to the communication strategy where the agents share the knowledge at each time step. For the LCC-UCB-GRAPH algorithm, we consider sparse graphs with more than 100 nodes. We observe that the LCC-UCB-GRPAH algorithm performs better (in terms of median cumulative regret per agent, over 30 independent runs) than the communication strategy where the agents share local data with all their neighbors at every time step. Further, the LCC-UCB and the LCC-UCB-GRAPH algorithms also outperform the DEMAB algorithm (Wang et al., 2020) where agents communicate for only $O(N\log(NK))$ rounds.

## 2. Related Works

Optimal action selection problem dates back to (Thompson, 1933), and since then, many algorithms have been proposed and studied to solve the MAB problem ranging from index-based policies (Gittins, 1979), Optimism in the Face of Uncertainty based UCB algorithm (Auer, 2002; Auer and Ortner, 2010; Audibert et al., 2009), to Thompson Sampling algorithm (Agrawal and Goyal, 2013). All the algorithms achieve an upper bound on regret $\tilde{O}(\sqrt{KT})$ and match the lower bound of $\Omega(\sqrt{KT})$ up to logarithmic factors. Since then, various generalizations and extensions have been proposed to solve various online learning problems using a bandit framework (Abbasi-Yadkori et al., 2011; Li et al., 2010; Lattimore et al., 2018; Yang et al., 2020). However, all these problems consider a single agent interacting with the environment.

Since the last decade, there has been a thrust in studying distributed agents solving an instance of MAB problem. Kanade et al. (2012) consider a model where $N$ agents talk to a central controller at every round. However, they considered the problem of reducing the communication cost for each agent connected in a star topology with a controller as the central node, which is unlike our setup where we allow any topology, including the central node/agent. Hillel et al. (2013) consider the problem of reducing communication cost for stochastic bandits in a setup where every agent can communicate with each other. Their work also bounds the total communication rounds by $O(\log_2 T)$ using an action elimination-based algorithm. However, their agents communicate the estimates of arm rewards for all the $K$ arms in each message, whereas we bound the number of bits required in each message by $O(\log_2 K)$. Shahrampour et al. (2017) consider a setup where multiple agents collectively select an arm at a time step and observe different rewards sampled from different distributions for each agent.

Other works consider a setup where the agents talk to only one of the other nodes in a network at any given time step (gossiping style algorithm) (Landgren et al., 2016; Martínez-Rubio et al., 2019; Wang et al., 2020). However, they allow their agents to communicate at every time-step, which is a different setup and do not optimize a regret-communication trade-off. Further, they also send estimates of arm rewards in each message. Sankararaman et al. (2019); Chawla et al. (2020) also consider gossip style algorithms. Similar to us, these works divide the time horizon into epochs of variable length. Their strategies also divide the arms among the agents, and the agents unicast the knowledge of the best arm they have using $O(\log K)$ bits in each epoch. However, because of gossip style communication protocols, an agent becomes aware of the best arm after it has already incurred $O(\frac{1}{\Delta^2})$ regret, which translates to a problem independent bound of $\tilde{O}(T^{2/3})$. We note that we use the same number of communication as these papers while achieving better regret bound of $\tilde{O}(T^{1/2})$. Further, we can convert the proposed broadcast based communication of our work to a unicast based strategy by sending a message to each neighbor at one timestep for $N$ timesteps.

Wang et al. (2019); Dubey and Pentland (2020a,b) consider the problem of distributed linear bandits. They considered a fully connected network to reduce the communication messages and reduce the average regret for $N$ agents. In contrast, we aim to find bounds on the regret of each of the $N$ agents for $K$-armed stochastic bandits.

Wang et al. (2019) propose DEMAB algorithm for a distributed bandit setup where all the nodes communicate with a central node. The setup assumes knowledge of the time horizon to cleverly obtain a bound on the number of communication messages independent of time. The DEMAB algorithm is based on action elimination that also proceeds in epochs with duration growing exponentially after an initial period of length $T/(NK)$ where every agent eliminates arms independently. In each epoch, the algorithm generates new estimates of arm rewards discarding the old samples. This results in high constants $O(\sqrt{2^{14}})$ in the regret term. The regret bounds of the proposed LCC-UCB algorithm only exceeds the regret of DEMAB for $\log_2 T > 2^{14}/144$. Additionally, the DEMAB algorithm requires a central coordinating node, which may not always be the case. Lastly, for an unknown time horizon, the number of messages increases back to $O(\log T)$, the same as ours.

The proposed algorithm, LCC-UCB, obtains $\tilde{O}(\sqrt{(N/K)T})$ for each agent with messages of size $O(\log K)$ with a total of $O(\log T)$ messages, thus achieving the regret of $\tilde{O}(\sqrt{T})$ Additionally, the proposed LCC-UCB-GRAPH algorithm works well on sparse graphs with a large number of agents with communication complexity of $O(D\log_2 T)$. A summary of recent algorithms is provided in Table 1.

| Algorithm | Regret | Bits per message | Rounds | Comments |
|---|---|---|---|---|
| DEMAB (Wang et al., 2020) | $\tilde{O}(\sqrt{\frac{KT}{N}})$ | $O(\log T)$ | $O(N\log NK)$ | known $T$, client-server message |
| GosInE (Chawla et al., 2020) | $\tilde{O}((\frac{K}{N}+2)^{1/3}T^{2/3})$ | $O(\log K)$ | $O(\log T)$ | unicast message, general graphs |
| LCC-UCB (this work) | $\tilde{O}(\sqrt{\left(\frac{K}{N}+N\right)T})$ | $O(\log K)$ | $O(\log T)$ | broadcast message, completely connected graphs |
| LCC-UCB-GRAPH (this work) | $\tilde{O}(\sqrt{\left(\frac{K}{N}+K_G\right)D^3T})$ | $O(\log K)$ | $O(D\log T)$ | broadcast message, general graphs |

Table 1: Summary of the baseline algorithms and the algorithms presented in this work.

## 3. Problem Formulation

We consider a completely connected network of $N$ agents, indexed as $n \in [N] = \{1, 2, \cdots, N\}$. We also consider $N$ independent instances of same stochastic $K$ armed bandit. Each agent $n \in [N]$ interacts with a fixed bandit instance $m \in [N]$ over $T$ time steps. For simplicity, we assume $m = n$.

Let $\{X_{i,m,t}\}_{i\in[K],m\in[N],t\in[T]}$ be a sequence of random variables defined on $(\Omega, \mathcal{F})$, where $\Omega = [0,1]^{KNT}$ and $\mathcal{F} = \mathcal{B}(\Omega)$. For each $i$, we assume that $X_{i,m,t}$ are identically distributed and are independent across all $N$ instances and $T$ time steps. Let $\mu_i$ denote the expected value of for the random variables $X_{i,m,t}$ for all $m$, $t$, and $i$. At time $t$, agent $n \in [N]$ selects arm $i_n(t)$ using a deterministic policy and observes $r_{n,t} = X_{i_n(t),n,t}$.

A super-agent policy would ideally have all $N$ agents communicating all information each time to selects arms $i_n(t)$ which is $\mathcal{F}_{t-1} = \{i_1(1), r_{1,1}, \cdots, i_N(1),\ r_{N,1}, \cdots, i_1(t-1), r_{1,t-1} \cdots, i_N(t-1), r_{N,t-1}\}$-measurable. However, the agents do not communicate at every time step and communicates only arm indices and hence arm $i_n(t)$ selected by agent $n$ is $\mathcal{F}_{n,t-1}$-measurable where $\mathcal{F}_{n,t-1}$ is the local information of arms played and their rewards available to agent $n$ and the indices of the arms communicated by other agents. $\mathcal{F}_{n,t-1}$ will be defined precisely in Section 5.

For our analysis, we assume that $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$. However, the ordering is unknown to the agents. We also define the gap between the expected reward of the best arm and the expected reward of arm $i$ as $\Delta_i := \mu_1 - \mu_i$. For our analysis we assume $0 \leq \mu_i \leq 1 \; \forall \; i \in [K]$. For our system model, we assume that $N \ll K$ as observed in many practical setups. For example, an e-commerce website will have many more products listed than the number of servers deployed.

Since our agents are completely connected, all agents can communicate with each other (we later relax this assumption in Section 6). This implies, whenever an agent broadcasts a message, all the other $N-1$ agents receive the message. Further, we assume that each agent only communicates the index of the best arm it knows. This requires $\lceil \log(K) \rceil$ bits for every message and since there are $N-1$ other agents to send the message, the total bits required by any agent is $(N-1)\lceil \log(K) \rceil$ bits in every communication round. We assume that each agent $n \in [N]$ interacts with the bandit environment equal number of times in an epoch. Further the synchronization time at the end of each epoch is assumed to be negligible.

An agent $n$ aims to minimize its cumulative regret over time $T$, $R_n(T)$, defined as:

$$R_n(T) = T\mu_1 - \mathbb{E}\left[\sum_{t=1}^{T} \sum_{i=1}^{K} \mu_i \mathbf{1}\{i_n(t) = i\}\right] \tag{1}$$

Note that minimizing regret $R_n(T)$ for all agents $n \in [N]$ also minimizes the total cumulative regret over the agents as well.

## 4. LCC-UCB Algorithm

We design our algorithm LCC-UCB on the basis of the fact that the regret of UCB algorithms (Auer, 2002; Bubeck et al., 2011; Lattimore and Szepesvári, 2020) scales as $\tilde{O}(\sqrt{KT})$. We reduce the per step regret by distributing the $K$ arms among the $N$ agents in growing in length epochs. An agent $n$ chooses to interact with a potentially smaller set of arms $\mathcal{S}_n$ where $\mathcal{S}_n = \{\big((n-1)\lceil \frac{K}{N} \rceil \mod K\big) + 1, \cdots, \big((n\lceil \frac{K}{N} \rceil - 1) \mod K\big) + 1\}$. For the first epoch, i.e., $j = 0$, each agent starts with possibly sub-optimal arms, even the worst possible arms. As the algorithm proceeds, in epoch $j \geq 1$, agents broadcast the most played arm by UCB algorithm during epoch $j$ to all the other agents. Each agent $n \in [N]$ receives $\mathcal{R}_{n,j}$, a set of arm recommendations from other $N-1$ agents. The agent now runs the UCB algorithm (Bubeck et al., 2011) over the arms in the augmented set $\mathcal{A}_{n,j} = \mathcal{S}_n \cup \mathcal{R}_{n,j}$. At the end of any epoch, the agent purges any old recommendations it has and starts again with the new recommendations received after an epoch. This ensures that the number of arms with any agent does not exceed $K' := \lceil K/N \rceil + N - 1$. This approach helps to bound the regret of any agent $n \in [N]$ by $\tilde{O}\left(\sqrt{(\lceil K/N \rceil + N - 1)\, T}\right)$.

The LCC-UCB algorithm running at an agent $n \in [N]$ is described in Algorithm 1. The algorithm at agent $n$ receives the set of initial arms $\mathcal{S}_n$, the indices of other agents, and the total horizon $T$. At each epoch $j$, agent $n$ maintains a set $\mathcal{R}_{n,j}$ of the arms received from the remaining $[N] \setminus \{n\}$ agents. For the first epoch $\mathcal{R}_{n,1} = \emptyset$ as the agent has not heard anything from the remaining agents and the augmented set is same as the initial set of arms, $\mathcal{A}_{n,0} = \mathcal{S}_n$. As the algorithm proceeds, it runs the UCB algorithm (Auer et al., 2002; Bubeck et al., 2011), described in Algorithm 2, on the arms in the augmented set $\mathcal{A}_{n,j}$ for epoch duration $K'(K'+1)2^j$. If at time $t$, remaining time is not sufficient to run a complete epoch of duration $T_j$, it just runs the UCB algorithm for the remaining horizon $T - t$.

**Algorithm 1** LCC-UCB$(n, \mathcal{S}_n, [N] \setminus \{n\}, T)$

```
$t = 0, j = 0, K' = |\mathcal{S}_\setminus + N - 1|$
$\mathcal{R}_{n,j} = \emptyset$
while $s < T$ do
    Set augmented set $\mathcal{A}_{n,j} = \mathcal{S}_n \cup \mathcal{R}_{n,j}$
    $i^* = \mathrm{UCB}(n, s, \mathcal{A}_{n,j}, \min\left(T - s, K'(K'+1)2^j\right))$
    $s = s + \min\left(T - s, K'(K'+1)2^j\right)$
    $j = j + 1$
    Send $i^*$ to other $[N] \setminus \{n\}$ agents
    Receive most played arms of $[N] \setminus \{n\}$ agents as $\mathcal{R}_{n,j}$
end while
```

**Algorithm 2** UCB$(n, \mathcal{A}, T_j)$

```
$t_j = 0$
$N_i(t_j) = 0, \hat{\mu}_i = 0 \ \forall \ i \in \mathcal{A}$
for $t_j = 1, \cdots, T_j$ do
    Obtain reward $r_{n,t}$ by playing arm $i_n(t)$, where
        $$i_n(t) = \arg\max_{i \in \mathcal{A}} \left\{ \hat{\mu}_i + \sqrt{\frac{2\log(t_j)}{N_i(t_j)}} \right\}$$
    $N_i(t_j) = N_i(t_j - 1) + \mathbf{1}_{\{i_t = i\}} \ \forall \ i \in \mathcal{A}$
    Update $\hat{\mu}_{i_t} = \frac{\hat{\mu}_i \times N_i(t_j - 1) + r_{n,t}}{N_i(t_j)}$
end for
Return $i^* = \arg\max_{i \in \mathcal{A}} N_i(T_j)$
```

## 5. Main Result

We now state the main result for bounding the regret and number of communication rounds for the proposed LCC-UCB algorithm.

**Theorem 1** *The regret of any agent $n$ following* LCC-UCB *algorithm is bounded by*

$$R_n(T) \leq O\left(\sqrt{K'T\log T}\right), \tag{2}$$

*where $K' = \lceil K/N \rceil + N - 1$.*

To prove Theorem 1, we first state the necessary lemmas required for the construction of the proof.

We start by making two important observations for our analysis. First, the communication does not happen at every time step. Second, any agent $n$ only receives the index of the most played arm from the other agents at the end of every epoch, and does not receive any additional information about the arms played and the rewards obtained by the other agents in the epoch. Thus, following the LCC-UCB algorithm, in epoch $j \geq 1$, an agent uses a policy which is $\mathcal{F}_{n,t-1} = \{i_1^*, \cdots, i_{n-1}^*, i_{n+1}^*, \cdots, i_N^*, i_n(s), r_{n,s}, \cdots, i_n(t-1), r_{n,t-1}\}$-measurable, where $s$ is the first time step of the epoch $j$ and $\{i_1^*, \cdots, i_{n-1}^*, i_{n+1}^*, \cdots, i_N^*\}$ are the most played arms by the other agents which agent $n$ receives at the end of the previous epoch at time step $s-1$.

For our setup, we assumed that the rewards samples obtained by playing an arm are independent across time. This allows to use the Hoeffding's bound stated in the following Lemma, Lemma 2.

**Lemma 2** *(Hoeffding, 1994, Hoeffding's bound) If $X_1, X_2, \cdots, X_n$ are $n$ independent random variables such that $X_i \in [0,1]$ for all $i = 1, \cdots, n$, then*

$$Pr\left(\frac{X_1 + \cdots + X_n}{n} - \mathbb{E}\left[\frac{X_1 + \cdots + X_n}{n}\right] \geq \epsilon\right) \leq \exp\left(-2n\epsilon^2\right), \text{ and} \tag{3}$$

$$Pr\left(\mathbb{E}\left[\frac{X_1 + \cdots + X_n}{n}\right] - \frac{X_1 + \cdots + X_n}{n} \geq \epsilon\right) \leq \exp\left(-2n\epsilon^2\right) \tag{4}$$

Note that, the LCC-UCB algorithm bounds regret when agent 1 recommends an arm $i^*$ which is "close" to the best arm ($i = 1$) from its augmented set $\mathcal{A}_{1,j}$ at every epoch, and then, in the following epoch, every other agent $n$ minimizes the regret with respect to the their augmented sets $\mathcal{A}_{n,j+1}$ which now contain the arm $i^*$.

Since the agent runs UCB algorithm (Algorithm 2) which returns the most played arm for each epoch, we want to analyse the properties of the most played arm. We now state and prove the lemma that the most played arms by the UCB algorithm is "good", or $\mu_{i^*} \geq \mu_1 - \tilde{\Delta}_j$, with high probability for some $\tilde{\Delta}_j$.

**Lemma 3** *For any epoch $j$, such that $T_j \geq K'(K'+1)$, instance of the UCB Algorithm 2 running at agent 1 returns an arm $i_j^*$ that satisfies $\mu_{i_j^*} \geq \mu_1 - \tilde{\Delta}_j$, with probability atleast*

$$1 - K'\left(\frac{T_j}{K'} - 1\right)^{-2}, \tag{5}$$

*for $\tilde{\Delta}_j = \sqrt{\frac{16K' \log T}{T_j}}$.*

**Proof** We first note that the augmented set at agent 1 contains the best arm 1 as arm index $\left((n-1)\lceil \frac{K}{N} \rceil \mod K\right) + 1 \in \mathcal{S}_n$ for $n = 1$. From Algorithm 2 instance that ran at epoch $j$, $N_i(T_j)$ is the number of times arm $i \in \mathcal{A}_{1,j}$ is played in epoch $j$. We now prove that the arm $i_j^* = \arg\max_{i \in \mathcal{A}_{1,j}} N_i(T_j)$ is at most $\tilde{\Delta}_j$ far from the true optimal arm 1.

For time step $t_j$ in epoch $j$, we construct an event where arm $i$ is selected and the total plays $N_i(t_j - 1)$ of arm $i$ has exceeded some number $l$ as $\mathcal{G}_{t_j}(i) = \{\{i_t = i\} \cap \{N_i(t_j - 1) \geq l_i\}\}$

for $l_i = 1 + \frac{8 \log T}{\Delta_i^2}$ and $t_j \geq K + 1$ as each arm is played atleast once. We first bound the probability of the event $\mathcal{G}_{t_j}(i)$ using the probability measure induced on the observed samples, as a result of the policy's interaction with the $K$-armed bandit instance 1. Note that the most played arms returned by agents $2 \leq n \leq N$ in epoch $j-1$ lie in the set $\mathcal{A}_{1,j}$, or $\{i^*_{2,j-1}, \cdots, i^*_{N,j-1}\} \subset A_{1,j}$. Conditioned on $\mathcal{F}_{1,t-1} = \{i^*_{2,j-1}, \cdots, i^*_{N,j-1}, i_n(s), r_{n,s}, \cdots, i_n(t-1), r_{n,t-1}\}$, we bound $Pr\left(\mathcal{G}_{t_j}(i) | \mathcal{F}_{1,t-1}\right)$ as:

$$Pr\left(\mathcal{G}_{t_j}(i)|\mathcal{F}_{1,t-1}\right) = Pr\left(\{\{i_t = i\} \cap \{N_i(t_j-1) \geq l_i\}\}|\mathcal{F}_{1,t-1}\right) \tag{6}$$

$$\leq Pr\left(\{i_t = i\}|\{N_i(t_j-1) \geq l_i\}, \mathcal{F}_{1,t-1}\right) \tag{7}$$

$$= Pr\left(\left\{\hat{\mu}_i + \sqrt{\frac{2\log t_j}{N_i(t_j)}} \geq \hat{\mu}'_i + \sqrt{\frac{2\log t_j}{N_{i'}(t_j)}}, \forall i' \in \mathcal{A}_{1,j}\right\} | \left\{N_i(t_j-1) \geq l_i\right\}, \mathcal{F}_{1,t-1}\right) \tag{8}$$

$$\leq Pr\left(\left\{\hat{\mu}_i + \sqrt{\frac{2\log t_j}{N_i(t_j)}} \geq \hat{\mu}_1 + \sqrt{\frac{2\log t_j}{N_1(t_j)}}\right\} | \left\{N_i(t_j-1) \geq l_i\right\}, \mathcal{F}_{1,t-1}\right) \tag{9}$$

$$\leq Pr\left(\left\{\hat{\mu}_i + \sqrt{\frac{2\log t_j}{N_i(t_j)}} \geq \min_{1 \leq N_1(t_j) \leq t_j} \hat{\mu}_1 + \sqrt{\frac{2\log t_j}{N_1(t_j)}}\right\} | \left\{N_i(t_j-1) \geq l_i\right\}, \mathcal{F}_{1,t-1}\right) \tag{10}$$

$$\leq \sum_{N_1(t_j)=1}^{t_j} Pr\left(\left\{\hat{\mu}_i + \sqrt{\frac{2\log t_j}{N_i(t_j)}} \geq \hat{\mu}_1 + \sqrt{\frac{2\log t_j}{N_1(t_j)}}\right\} | \left\{N_i(t_j-1) \geq l_i\right\}, \mathcal{F}_{1,t-1}\right) \tag{11}$$

$$\leq \sum_{N_1(t_j)=1}^{t_j} Pr\left(\left\{\hat{\mu}_i + \frac{\Delta_i}{2} \geq \hat{\mu}_1 + \sqrt{\frac{2\log t_j}{N_1(t_j)}}\right\} | \left\{N_i(t_j-1) \geq l_i\right\}, \mathcal{F}_{1,t-1}\right) \tag{12}$$

$$\begin{aligned}
&= \sum_{N_1(t_j)=1}^{t_j} Pr\left(\left\{\hat{\mu}_i + \frac{\Delta_i}{2} \geq \hat{\mu}_1 + \sqrt{\frac{2\log t_j}{N_1(t_j)}}\right\} \cap \left\{\hat{\mu}_1 + \sqrt{\frac{2\log t_j}{N_1(t_j)}} < \mu_1\right\} | \right.\\
&\left\{N_i(t_j-1) \geq l_i\right\}, \mathcal{F}_{1,t-1}) +\\
&\quad \sum_{N_1(t_j)=1}^{t_j} Pr\left(\left\{\hat{\mu}_i + \frac{\Delta_i}{2} \geq \hat{\mu}_1 + \sqrt{\frac{2\log t_j}{N_1(t_j)}}\right\} \cap \left\{\hat{\mu}_1 + \sqrt{\frac{2\log t_j}{N_1(t_j)}} \geq \mu_1\right\} | \right.\\
&\left\{N_i(t_j-1) \geq l_i\right\}, \mathcal{F}_{1,t-1})
\end{aligned} \tag{13}$$

$$\begin{aligned}
&\leq \sum_{N_1(t_j)=1}^{t_j} Pr\left(\left\{\hat{\mu}_1 + \sqrt{\frac{2\log t_j}{N_1(t_j)}} < \mu_1\right\} | \left\{N_i(t_j-1) \geq l_i\right\}, \mathcal{F}_{1,t-1}\right) +\\
&\quad \sum_{N_1(t_j)=1}^{t_j} Pr\left(\left\{\hat{\mu}_i + \frac{\Delta_i}{2} \geq \mu_1\right\} | \left\{N_i(t_j-1) \geq l_i\right\}, \mathcal{F}_{1,t-1}\right)
\end{aligned} \tag{14}$$

$$\begin{aligned}
&\leq \sum_{N_1(t_j)=1}^{t_j} Pr\left(\left\{\hat{\mu}_1 + \sqrt{\frac{2\log t_j}{N_1(t_j)}} < \mu_1\right\}\right)\\
&\quad + \sum_{N_1(t_j)=1}^{t_j} Pr\left(\left\{\hat{\mu}_i - \mu_i \geq \frac{\Delta_i}{2}\right\} | \left\{N_i(t_j-1) \geq l_i\right\}, \mathcal{F}_{1,t-1}\right)
\end{aligned} \tag{15}$$

$$\leq \sum_{N_1(t_j)=1}^{t_j} \left(\exp\left(-2N_1(t_j)\frac{2\log t_j}{N_1(t_j)}\right) + \exp\left(-2N_i(t_j)\frac{2\log t_j}{N_i(t_j)}\right)\right) \tag{16}$$

$$\leq \sum_{N_1(t_j)=1}^{t_j} \left(1/t_j^4 + 1/t_j^4\right) = 2/t_j^3 \tag{17}$$

Equation (8) follows from the fact that the UCB algorithm will select arm $i$ if the upper confidence bound of the arm $i$ is highest among all the arms in set $\mathcal{A}_{1,j}$. Note that the true best arm, arm 1, lies in the set $\mathcal{A}_{1,j}$ and this gives Equation (9). Equation (11) follows by taking union bound over all possible values of $N_1(t_j)$. Equation (12) follows by replacing the lower bound of $N_i(t_j)$ obtained from the conditioning. Equation (14) follows from the fact that the sets in Equation (14) contains the sets in Equation (13). The first term in Equation (15) holds because the confidence intervals for arm 1 are independent of the number of samples of arm $i$ and $\mu_1 = \mu_i + \Delta_i$. Equation (16) follows from Hoeffding's concentration bound. Using the law of total probability, we get $Pr(\mathcal{G}_{t_j}(i)) \leq 2t_j^{-3}$.

We can now use the probability of the event $\mathcal{G}_{t_j}(i)$, to bound the probability of the event that the number of plays of an arm exceeds $l_i$ by using union bound. Specifically we have:

$$Pr\left(N_i(T_j) \geq l_i\right) \leq \bigcup_{t_j=l_i}^{T_j} Pr\left(\mathcal{G}_{t_j}(i)\right) \tag{18}$$

$$\leq \sum_{t_j=l_i}^{T_j} 2t_j^{-3} \tag{19}$$

$$< \sum_{t_j=l_i}^{\infty} 2t_j^{-3} \tag{20}$$

$$\leq \int_{t_j=l_i-1}^{\infty} 2t_j^{-3} = \frac{1}{(l_i-1)^2} \tag{21}$$

Now, for an arm $i$ such that $\Delta_i > \sqrt{\frac{8K'\log T}{T_j - K'}} \geq \sqrt{\frac{16K'\log T}{T_j}} =: \tilde{\Delta}_j$, we have,

$$l_i = 1 + \frac{8\log T}{\Delta_i^2} \tag{22}$$

$$< 1 + \frac{8\log T}{\tilde{\Delta}_j^2} \tag{23}$$

$$\leq 1 + \frac{T_j - K'}{K'} = \frac{T_j}{K'}. \tag{24}$$

Hence, $N_i(T_j) \leq T_j/K' - 1$ with probability at least $1 - (T_j/K - 1)^{-2}$. Now, let $\mathcal{B}_{1,j} = \{i \in \mathcal{A}_{1,j} | \mu_i < \mu_1 - \tilde{\Delta}_j\}$ be the set of "bad" arms in the augmented set of agent 1 in epoch $j$. Then for any arm $i \in \mathcal{B}_{1,j}$, we have $N_i(T_j) < T_j/K'$ with probability at least

$1 - K'(T_j/K-1)^{-2}$. Thus,

$$\begin{aligned}
\mathbb{P}\left(N_i(T_j) \geq T_j/K' \text{ for any } i \in \mathcal{B}_{1,j}\right) &= \cup_{i\in\mathcal{B}_{1,j}} \mathbb{P}\left(N_i(T_j) \geq T_j/K'\right) && (25)\\
&\leq \sum_{i\in\mathcal{B}_{1,j}} \mathbb{P}\left(N_i(T_j) \geq T_j/K'\right) && (26)\\
&\leq \sum_{i\in\mathcal{B}_{1,j}} (T_j/K'-1)^{-2} && (27)\\
&\leq \sum_{i\in\mathcal{A}_{1,j}} (T_j/K'-1)^{-2} && (28)\\
&= K'(T_j/K-1)^{-2} && (29)
\end{aligned}$$

Thus, $\mathbb{P}\left(N_i(T_j) < T_j/K' \text{ for all } i \in \mathcal{B}_{1,j}\right) \geq 1-K'(T_j/K-1)^{-2}$ . Thus, the probability that $N_i(T_j) < T_j/K'$ for all $i \in \mathcal{B}_{1,j}$ is at least $1 - K'(T_j/K-1)^{-2}$.

After bounding the number of plays of arms $i$, such that $\mu_i \leq \mu_1 - \tilde{\Delta}_j$, with high probability, we show that the most played arm $i^*_j$ has expected reward $\mu_{i^*_j} \geq \mu_1 - \tilde{\Delta}_j$. We have:

$$\begin{aligned}
\max_{i\in\mathcal{A}_{1,j}\setminus\mathcal{B}_{1,j}} N_i(T_j) &\geq \frac{1}{|\mathcal{A}_{1,j}\setminus\mathcal{B}_{1,j}|} \sum_{i\in\mathcal{A}_{1,j}\setminus\mathcal{B}_{1,j}} N_i(T_j) && (30)\\
&= \frac{1}{|\mathcal{A}_{1,j}\setminus\mathcal{B}_{1,j}|} \left(T_j - \sum_{i\in\mathcal{B}_{1,j}} N_i(T_j)\right) && (31)\\
&> \frac{1}{|\mathcal{A}_{1,j}\setminus\mathcal{B}_{1,j}|} \left(T_j - \sum_{i\in\mathcal{B}_{1,j}} \left(\frac{T_j}{K'}\right)\right) && (32)\\
&= \frac{1}{|\mathcal{A}_{1,j}\setminus\mathcal{B}_{1,j}|} \left(K'\frac{T_j}{K'} - |\mathcal{B}_{1,j}|\left(\frac{T_j}{K'}\right)\right) && (33)\\
&= \frac{1}{|\mathcal{A}_{1,j}\setminus\mathcal{B}_{1,j}|} \left(K' - |\mathcal{B}_{1,j}|\right)\frac{T_j}{K} && (34)\\
&= \frac{1}{|\mathcal{A}_{1,j}\setminus\mathcal{B}_{1,j}|} |\mathcal{A}_{1,j}\setminus\mathcal{B}_{1,j}|\frac{T_j}{K} = \frac{T_j}{K} && (35)
\end{aligned}$$

This proves that the most played arm in $\mathcal{A}_{1,j}$, $i^*_j$, is at most $\tilde{\Delta}_j$ far from the optimal arm 1. ■

After showing that the agent 1 returns a good arm after each epoch, we now show that the regret of all the other agents is bounded in the following epoch $j+1$. Lemma 4 bounds the regret of an agent $n$ running UCB Algorithm 2 during an epoch $j$. We then sum over all the epochs to obtain the total regret of the algorithm. We focus our analysis on an agent $n$. The analysis of the remaining agents follows identically.

**Lemma 4 (UCB regret bound)** *The regret of any agent $n$ running UCB algorithm described in Algorithm 2 for an epoch $j \geq 2$ with $T_j$ time steps is upper bounded by*

$$R(T_j) \leq 6\sqrt{2K'T_j \log T} + \frac{16K'^3}{T_j} + 2K' \tag{36}$$

**Proof** We first consider the case of an agent $n \neq 1$. The agent receives recommendations from all the other $N-1$ agents including the agent 1 and hence contains the arm $i^*$ recommended by the agent 1.

To analyze the regret, we first create some events that will help in analysis. The first event denotes the case where the agent 1, after the end of epoch $j-1$, recommends arm $i^*$ such that $\mu_{i^*} \geq \mu_1 - \tilde{\Delta}_{j-1}$. We denote this event as $\tilde{\mathcal{G}}_1$. Further note that $N_i(T_j)$ is the number of times agent plays arm $i \in \mathcal{A}_{n,j}$ in epoch $j$. We note that when the event $\tilde{\mathcal{G}}_1$ occurs $\Delta_{i^*} \leq \tilde{\Delta}_{j-1}$. We assume that $i^*$ satisfies $\mu_{i^*} = \max_{i\in\mathcal{A}_{n,j}} \mu_i$. In case the assumption is not valid, we redefine $i^*$ as $i^* = \arg\max_{i\in\mathcal{A}_{n,j}} \mu_i$, and we still have $\mu_1 - \mu_{i^*} \leq \tilde{\Delta}_j$. Also, for the simplicity of notation, we define $\Delta_{i^*,i} = \mu_{i^*} - \mu_i$. Then, using the regret decomposition lemma (Lattimore and Szepesvári, 2020, Lemma 4.5), the regret of the UCB algorithm for epoch $j$ is upper bounded as:

$$R(T_j) = \sum_{i\in\mathcal{A}_{n,j}} \mathbb{E}\left[\Delta_i N_i(T_j)\right] \tag{37}$$

$$= \sum_{i\in\mathcal{A}_{n,j}} \mathbb{E}\left[(\mu_1 - \mu_i) N_i(T_j)\right] \tag{38}$$

$$= \sum_{i\in\mathcal{A}_{n,j}} \mathbb{E}\left[(\mu_1 - \mu_{i^*} + \mu_{i^*} - \mu_i) N_i(T_j)\right] \tag{39}$$

$$= \sum_{i\in\mathcal{A}_{n,j}} \mathbb{E}\left[(\Delta_{i^*} + \Delta_{i^*,i}) N_i(T_j)\right] \tag{40}$$

$$= \sum_{i\in\mathcal{A}_{n,j}} \mathbb{E}\left[\Delta_{i^*} N_i(T_j)\right] + \sum_{i\in\mathcal{A}_{n,j}} \mathbb{E}\left[(\Delta_{i^*,i}) N_i(T_j)\right] \tag{41}$$

$$= \sum_{i\in\mathcal{A}_{n,j}} \mathbb{E}\left[\Delta_{i^*} N_i(T_j) | \tilde{\mathcal{G}}_1\right] Pr(\tilde{\mathcal{G}}_1) + \sum_{i\in\mathcal{A}_{n,j}} \mathbb{E}\left[\Delta_{i^*} N_i(T_j) | \tilde{\mathcal{G}}_1^c\right] Pr(\tilde{\mathcal{G}}_1^c) + \sum_{i\in\mathcal{A}_{n,j}} \mathbb{E}\left[(\Delta_{i^*,i}) N_i(T_j)\right] \tag{42}$$

$$\leq \sum_{i\in\mathcal{A}_{n,j}} \tilde{\Delta}_{j-1} \mathbb{E}\left[N_i(T_j) | \tilde{\mathcal{G}}_1\right] Pr(\tilde{\mathcal{G}}_1) + \sum_{i\in\mathcal{A}_{n,j}} \mathbb{E}\left[N_i(T_j) | \tilde{\mathcal{G}}_1^c\right] Pr(\tilde{\mathcal{G}}_1^c) + \sum_{i\in\mathcal{A}_{n,j}} \mathbb{E}\left[(\Delta_{i^*,i}) N_i(T_j)\right] \tag{43}$$

$$\leq \tilde{\Delta}_{j-1} \mathbb{E}\left[\sum_{i\in\mathcal{A}_{n,j}} N_i(T_j) | \tilde{\mathcal{G}}_1\right] + Pr(\tilde{\mathcal{G}}_1^c) \mathbb{E}\left[\sum_{i\in\mathcal{A}_{n,j}} N_i(T_j) | \tilde{\mathcal{G}}_1^c\right] + \sum_{i\in\mathcal{A}_{n,j}} \mathbb{E}\left[(\Delta_{i^*,i}) N_i(T_j)\right] \tag{44}$$

$$\le \tilde{\Delta}_{j-1}T_j + K'\left(\frac{K'}{T_{j-1}-K'}\right)^2 T_j + \sum_{i\in\mathcal{A}_{n,j}} \mathbb{E}\left[(\Delta_{i^*,i})N_i(T_j)\right] \tag{45}$$

$$\le 4\sqrt{\frac{K'\log T}{T_{j-1}}}T_j + K'\left(\frac{2K'}{T_{j-1}}\right)^2 T_j + \sum_{i\in\mathcal{A}_{n,j}} \mathbb{E}\left[(\Delta_{i^*,i})N_i(T_j)\right] \tag{46}$$

$$\le 4\sqrt{2K'T_j\log T} + \frac{16K'^3}{T_j} + \sum_{i\in\mathcal{A}_{n,j}} \mathbb{E}\left[(\Delta_{i*,i})N_i(T_j)\right] \tag{47}$$

We now focus on the last term. We define event where the UCB algorithm plays arm $i$ after the number of plays of an arm $i$ is has crossed $l_i$, or

$$\mathcal{G}_{n,i}(t_j) = \left\{\{i_t = i\} \cap \{N_i(t_j-1)\ge l_i\}\right\}, \text{ where } l_i = \frac{1}{\Delta_{i^*,i}}, \tag{48}$$

Again, similar to Lemma 3, we use Hoeffding's concentration bound to upper bound the probability of the event $\mathcal{G}_{t_j}(i)$ by $2t_j^{-3}$. Then we can bound the last term in Equation 47 as:

$$\sum_{i\in\mathcal{A}_{n,j}} \mathbb{E}\left[(\Delta_{i*,i})N_i(T_j)\right] \le \sum_{i\in\mathcal{A}_{n,j}} \Delta_{i*,i}l_i + \sum_{i\in\mathcal{A}_{n,j}}\sum_{t_j=l_i}^{T_j} Pr\left(\mathcal{G}_{n,i}(t_j)\right) \tag{49}$$

$$\le \sum_{i\in\mathcal{A}_{n.j}} \Delta_{i^*,i}\left(1+\frac{8\log T}{\Delta^2_{i^*,i}}\right) + \sum_{i\in\mathcal{A}_{n.j}}\sum_{t_j=1}^{T_j} t_j^{-2} \tag{50}$$

$$\le \sum_{i\in\mathcal{A}_{n.j}} \frac{8\log T}{\Delta_{i^*,i}} + K' + \frac{K'\pi^2}{6} \tag{51}$$

$$\le \sqrt{8K'T_j\log T} + K' + \frac{K'\pi^2}{6} \tag{52}$$

Replacing the value in Equation 47, we get the required result for $n \neq 1$.

Further, note that for $n = 1$, the true optimal arm 1 is always present in $\mathcal{A}_{1,j}$ for all $j \geq 1$. ■

We are now ready to prove Theorem 1. We first note that for epoch $j = 0$, not agents have yet communicated, and hence the regret of any agent is trivially bounded by $T_0 = K'(K'+1)$. For the later epochs, we sum over the regret incurred in each epoch using Lemma 4. To do so, we first bound the total number of epochs. Let the total number of epochs be $J$, then noting that the total number of time steps is $T$, we get:

$$T \le \sum_{j=0}^{J-1} K'(K'+1)2^j < 2T$$

$$\implies 2^J - 1 < \frac{2T}{K'(K'+1)}$$

$$\implies J < \log_2\left(\frac{T}{K'(K'+1)}+1\right)$$

$$\implies J = \lfloor \log_2 \left( \frac{T}{K'(K'+1)} + 1 \right) \rfloor$$

After bounding the regret in each epoch $R(T_j)$ and bounding the total number of epochs, we can bound the total regret as,

$$R_n(T) = \sum_{j=0}^{J-1} R(T_j) \tag{53}$$

$$\leq 6 \sum_{j=1}^{J-1} \sqrt{2K'T_j \log T} + \sum_{j=1}^{J-1} \frac{16K'^3}{T_j} + 3K' \log_2(2T+1) + K'(K'+1) \tag{54}$$

$$\leq 6\sqrt{2K' \log T} \sum_{j=1}^{J-1} \sqrt{T_j} + 16K' + 3K' \log_2(2T+1) + K'(K'+1) \tag{55}$$

$$\leq 6\sqrt{2K' \log T} \sum_{j=1}^{J-1} \sqrt{K'(K'+1)2^j} + 16K' + 3K' \log_2(2T+1) + K'(K'+1) \tag{56}$$

$$\leq 6\sqrt{2K' \log T}\sqrt{K'(K'+1)} \sum_{j=1}^{J-1} 2^{j/2} + 16K' + 3K' \log_2(2T+1) + K'(K'+1) \tag{57}$$

$$\leq 6\sqrt{2K' \log T} \left( \sqrt{K'(K'+1)} \sum_{j=1}^{J-1} 2^{j/2} \right) + 16K' + 3K' \log_2(2T+1) + K'(K'+1) \tag{58}$$

$$\leq 6(\sqrt{2}+1)\sqrt{2K' \log T(2T)}\sqrt{K'(K'+1)2^J} + 16K' + 3K' \log_2(2T+1) + K'(K'+1) \tag{59}$$

$$\leq 36\sqrt{K'T \log T} + 16K' + 3K' \log_2(2T+1) + K'(K'+1), \tag{60}$$

where Equation (59) follows from summing over the geometric progression in Equation (58)

**Theorem 5** *For* LCC-UCB *algorithm, total number of bits exchanged by an agent is bounded by* $O\left(N \log(K) \log(T)\right)$.

**Proof** An agent sends or receives only arm index, which requires $\log_2(K)$ bits. In each epoch, the agents communicates with $N-1$ agents and sends and receives $2(N-1)\log_2(K)$ bits. Finally, there are $\log_2(T)$ epochs. This bounds the total number of bits as $O\left(K \log(K) \log(T)\right)$. ∎

We note that the algorithm proposed by Sankararaman et al. (2019) also divides the time horizon into epochs with $K$ arms divided among $N$ agents. However, they consider the first few epochs to be of fixed length where agents only explore to find the best arm within themselves. Our algorithm runs UCB from the very first epoch. Also, the length of the first epoch is $o(1)$ in LCC-UCB algorithm which limits the regret. These novel changes allow for a significantly improved regret bound as compared to the state of the art with limited communication rounds.

## 6. Extension to general network structures

So far we assumed that all the nodes are connected to each and every other node. However, this might not always be true. We now assume a general structure where a graph $G = (V, E)$ that has the different agents as vertices and the connections as edges represents the network structure. We assume that the graph representing the network is sparsely connected with a small diameter and degree, for example Erdős-Rényi graphs (Chung and Lu, 2001). We assume that the maximum degree of $G$ is $K_G$ and the diameter of $G$ is $D$. We assume that each agent knows the diameter $D(< N)$ of the graph and the maximum degree $K_G$ of the nodes. Further, we assume that each node is aware of its neighbors in order to communicate with them. We do not require the complete graph structure in order to proceed with the proposed algorithm.

For this setup, we assume that an agent or node can communicate with only its neighbors. Under this assumption, it may take multiple epochs for the knowledge of the best arm to reach an agent that may not have the best arm to begin with. Further, the number of epochs where an agent does not hear from the agent that has the best arm is bounded by the diameter $D$. Also, the maximum size of $\mathcal{A}_{n,j}$ is now upper bounded by $\lceil \frac{K}{N} \rceil + K_G$ instead of $\lceil \frac{K}{N} \rceil + N - 1$.

We first start with a direct extension of the result in Theorem 1, and by understanding the issues in the direct extension, will propose an algorithm to improve the results for general networks. The following result gives a corollary for Theorem 1 for general graphs.

**Corollary 6** *For graph $G = (V, E)$ with agents as nodes $V$,* LCC-UCB *algorithm results in a regret bound of:*

$$R_n(T) \leq \tilde{\mathcal{O}}\left(2^D K'^2 + \sqrt{2^D K' T}\right) \tag{61}$$

*where $D$ is the diameter of the graph $G$, $K' = \left(\left\lceil \frac{K}{N} \right\rceil + K_G\right)$ and $K_G$ is the maximum degree of any node in the graph $G$.*

**Proof** An agent $n \neq 1$ receives arm recommendations only from its neighboring nodes which results in reduction of $K'$ from $\lceil K/N \rceil + N$ to $\lceil K/N \rceil + K_G$. However, this also implies that the $n \neq 1$ does not obtains information about a good arm from the agent 1 directly. Note that applying Lemma 3 on UCB algorithm ran by agent $n \neq 1$ suggests that the agent recommends an arm $i_n^*$ such that $\mu_{i_n^*} \geq \mu_{i^*} - \tilde{\Delta}_j$ where $i^* = \arg\max_{i \in \mathcal{A}_{n,j}} \mu_i$. This implies that the agent (or node) located farthest from the agent 1 receives knowledge about a good arm, **(1)** only after $D$ epochs for the very first time, and **(2)** the best arm in the received $i^* = \arg\max_{i \in \mathcal{R}_{n,j}} \mu_i$ set $i^*$ satisfies $\mu_i^* \geq \mu_1 - \sum_{j=1}^{D} \tilde{\Delta}_{j-1}$.

This results in an additional constant regret during the first $D$ epochs as:

$$\sum_{j=0}^{D-1} T_j = \sum_{j=0}^{D-1} (K' + 1)K' 2^j = (K' + 1)K'(2^D - 1) \tag{62}$$

Further, the gap incurred from receiving a bad recommendation in each epoch scales as:

$$(\mu_1 - \mu_{i^*})T_j \leq \sum_{j'=1}^{D} \tilde{\Delta}_{j'-1} T_j = \sum_{j'=1}^{D} 4\sqrt{\frac{K' \log T}{T_{j'-1}}} T_j \tag{63}$$

$$= \sum_{j'=1}^{D} 4\sqrt{K' 2^{j'} T_j \log T} \tag{64}$$

$$= 4\sqrt{K' T_j \log T} \sum_{j'=1}^{D} 2^{j'/2} \tag{65}$$

$$= 4\sqrt{K' 2(2^D - 1) T_j \log T} \tag{66}$$

■

**Remark 7** *Note that for $D = 1$ and $K_G = N - 1$, or the case for a completely connected graph, the result of Theorem 1 is obtained.*

To avoid the exponential blow-up of $2^D$ in the regret, we first consider a strategy where an agent forwards the messages from one neighbor to all the other neighbor. However, this increases the message size from $O(K_G \log_2 K)$ bits to $O(N \log_2 K)$ bits. Further, additional complexity is added to reduce repeated propagation of messages. In order to avoid the potential exponential increase in regret or increase in the message size and the communication complexity, we propose a modification of the LCC-UCB algorithm as LCC-UCB-GRAPH algorithm. The proposed LCC-UCB-GRAPH algorithm is described in Algorithm 3.

**Algorithm 3** LCC-UCB-GRAPH($\mathcal{S}_n, G, T_0, T$)

```
 1: $t = 0, j = 0$
 2: $\mathcal{R}_{n,1,0} = \emptyset$
 3: for $t < T$ do
 4:     $d = 1$
 5:     for $d \leq D$ do
 6:         Set augmented set $\mathcal{A}_{n,d,j} = \mathcal{S}_n \cup \mathcal{R}_{n,d,j}$
 7:         $i^* = \text{UCB}(\mathcal{A}_{n,d,j}, \min(T - t, K'(K'+1)2^j))$
 8:         $t = t + K'(K'+1)2^j$
 9:         Send $i^*$ to neighbors
10:         Receive most played arms of neighbors as $\mathcal{R}_{n,d,j}$
11:         $d = d + 1$
12:     end for
13:     $j = j + 1$
14: end for
```

The LCC-UCB-GRAPH algorithm further divides an epoch $j$ into $D$ sub-epochs indexed as $d$. The duration of each sub-epoch in epoch $j$ is $T_j = K'(K'+1)2^j$. Now, the LCC-UCB-GRAPH algorithm restarts UCB algorithm for sub epochs (Line 6-12). Additionally, the agents now communicate after every sub-epoch, but, only with their neighbors. This gives the $K' \leq \lceil \frac{K}{N} \rceil + K_G$.

Note that results from sub-epoch $d$ of epoch $j$ are propagated throughout the graph by the time sub-epoch $d$ starts in epoch $j+1$. Hence, for $\tilde{\Delta}_j := \sqrt{\frac{16K' \log(T)}{T_j}}$, this approach allows to propagate arms with $\Delta_i \leq D\tilde{\Delta}_{j-1}$ instead of $\sum_{j'=j-D}^{j} \tilde{\Delta}_{j'}$. Based on this modification, we can bound the regret of LCC-UCB-GRAPH algorithm and the number of bits required for communication by LCC-UCB-GRAPH algorithm.

**Theorem 8** *Let $G = (V, E)$ be the graph representing the network structure of agents $n \in [N]$, and let $D$ be the diameter of the graph $G$ and let $K_G$ be the maximum degree of the vertices of the graph $G$. Then, the regret of any agent $n$ following* LCC-UCB-*GRAPH algorithm is bounded by*

$$R_n(T) \leq \tilde{O}\left(D\sqrt{DK'T}\right), \tag{67}$$

*where $K' = \lceil \frac{K}{N} \rceil + K_G$.*

**Proof** Note that at the beginning of the phase of a sub-epoch $d$ in epoch $j$, the information from the farthest node $D$ edges away is also received for epoch $j-1$ sub-epoch $d$. This is because exactly $D$ communication rounds happens between sub-epoch, epoch pair $d, j-1$ and $d, j$. Further, each intermediate $D$ nodes drifts from the optimal arm found in sub-epoch, epoch $d, j-1$ by at most $\tilde{\Delta}_{j-1}$. This suggest that instead of receiving an arm with $\Delta_i \leq \tilde{\Delta}_{j-1}$, the node actually receives an arm $i^* = \arg\max_{i \in \mathcal{A}_{n,d,j}} \mu_i$ with $\Delta_{i^*} \leq D\tilde{\Delta}_{j-1}$. Hence, extending Lemma 4 with $D$ hops, the regret $R(d,j)$ in each sub-epoch $d$ and epoch $j$ is now upper bounded as

$$R(d,j) \leq 2(2D+1)\sqrt{2K'T_j \log T} + \frac{16DK'^3}{T_j} + 2K' \tag{68}$$

In Equation (68), the extra factors of $D$ comes from the fact that now each of the agents in $D$ hops recommends an arm $i$ such that $\mu_{i_d^*} \geq \mu_{i_{d-1}^*} - \tilde{\Delta}_j$ for all $d \geq 1$ and $i_0^* = 1$, the true best arm. Note that the duration of any sub-epoch $d$ is $K'(K'+1)2^j$ and it depends only on the epoch $j$. Hence, the regret $R(d,j)$ is only a function of epoch count $j$.

The total regret of the agent $n$, which is the sum of regrets over all sub-epochs in every epoch, can now be bounded as:

$$\begin{aligned}
R_n(T) &= \sum_{j=0}^{J-1} \sum_{d=1}^{D} R(d,j) \\
&= \sum_{j=1}^{J-1} \sum_{d=1}^{D} R(d,j) + \sum_{d=1}^{D} R(d,0) \\
&= \sum_{j=1}^{J-1} \sum_{d=1}^{D} \left(2(2D+1)\sqrt{2K'T_j \log T} + \frac{16DK'^3}{T_j} + 3K'\right) + \sum_{d=1}^{D} K'(K'+1) \qquad (69) \\
&= 2(2D+1)D\sqrt{2K' \log T} \sum_{j=1}^{J-1} \sqrt{T_j} + DJ\frac{16DK'^3}{T_j} + 3DJK' + DK'(K'+1) \\
&= 12(2D+1)\sqrt{K'DT \log T} + 16D^2K' + 3K'D\log_2(2T+1) + DK'(K'+1)
\end{aligned}$$

■

The key novelty of LCC-UCB-GRAPH algorithm is to let sub-epochs $0 \leq d < D$ collect the messages from the entire graph. The equal length of each sub-epoch avoids the exponential blow-up in the regret. Further, the exponential length of each epoch $j$ still keeps the total messages in logarithmic order of $T$.

**Theorem 9** *For* LCC-UCB-GRAPH *algorithm, total number of bits exchanged by an agent is bounded by* $O\left(K_G D \log(K) \log(T)\right)$.

**Proof** An agent sends or receives only arm index, which requires $\log_2(K)$ bits. The agent communicates at the end of every sub-epoch of every epoch. In each communication, the agents talks to at most $K_G$ neighbors and sends and receives $2K_G \log_2(K)$ bits. Finally, there are $D$ sub-epochs in every $\log_2(T)$ epochs. This bounds the total number of bits as $O\left(DK_G \log(K) \log(T)\right)$. ■

Results from Theorem 8 and Theorem 9 suggest that it is possible to reduce the regret from an exponential order of the diameter $D$ of the graph $G$ at the expense of $D$ times more communication rounds. Further, since each communication involves only exchange of arm indices, the cost of communication is not high ($O(K_G \log_2 K)$ bits) for power constrained devices such as sensor networks.

### Optimizing Regret using Knowledge of Graph

We now show that the additional knowledge of the complete graph structure could be utilized to reduce the regret bound of the proposed algorithm. As a motivating example, consider $L$ cliques of size $N/L$ connected in a line. An example for $L = 3$ and $N = 15$ is shown in Figure 1. Here, we can run the LCC-UCB algorithm on individual cliques to obtain a per-agent regret of $O(\sqrt{\frac{KT}{N/L} + NT/L})$. In contrast, running LCC-UCB algorithm on the entire graph incurs a regret of $O(L\sqrt{L\left(KT/N + NT/L\right)})$. Hence, working with cliques separately helps in improving the regret by a factor of $L$ in this case.

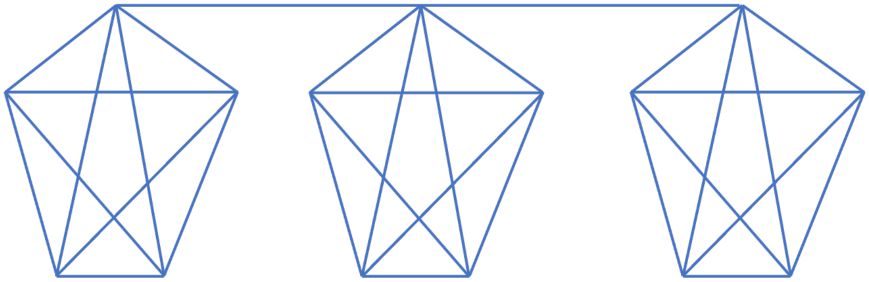


Figure 1: A topology where 3 cliques, each of size 5, are connected via bridges.

The example above can be extended to partition the graph into sub-graphs. Let the graph $G$ can be divided into $C$ subgraphs - $G_1, \cdots, G_C$ such that each node is contained in exactly one of the sub-graphs, and the links in the sub-graphs are subset of the links in the graph. Let $D_{G_c}$ be the diameter of the sub-graph $G_c$, $N_{G_c}$ be the number of nodes in the sub-graph $G_c$, and $K_{G_c}$ is the maximum degree of any node in $G_c$. Then, for a given choice of the sub-graphs, the regret can be given as

$$O\left(\max_c\left(D_{G_c}\sqrt{D_{G_c}KN_{G_c}T + D_{G_c}N_{G_c}^2K_{G_c}T}\right)\right) \tag{70}$$

Finding the best possible split of sub-graphs that optimizes the above metric can be performed for the given graph. This split can help obtain improved results in some graphs as illustrated in the example above, where we obtained an improvement of $O(L)$.

## 7. Evaluations

We consider various problem setups to evaluate our algorithms. We compare the proposed algorithms, LCC-UCB and LCC-UCB-GRAPH with a no-communication strategy and a full communication strategy. The details of the no-communication and the full communication setup are:

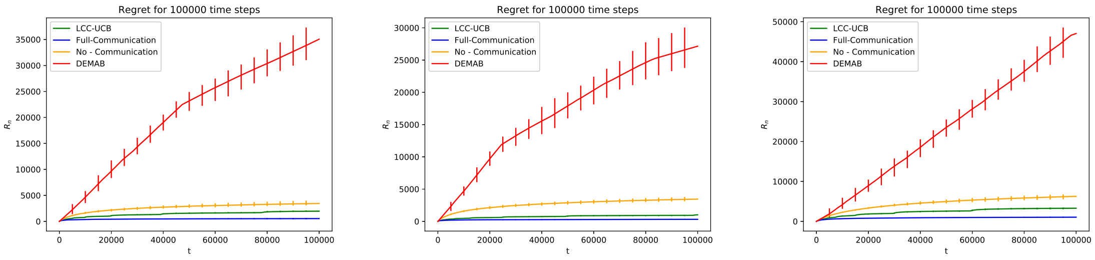
- **No-Communication Setup:** We consider the case where each agent $n \in [N]$ works in isolation. The initial set $\mathcal{S}_n$ of every agent contains all the $K$ arms. Each agent uses the standard UCB algorithm to interact with the environment and reduce its regret. Hence, each agent incurs a regret of $\tilde{O}(\sqrt{KT})$. Since the agents are not communicating, they do not reduce the regret.

- **Full-Communication Setup:** We consider the case where each agent $n \in [N]$ can talk only to its neighbors and observes the arms played and the rewards obtained by the neighbors at every time step. The initial set $\mathcal{S}_n$ of every agent again contains all the arms. Each agent uses the standard UCB algorithm to interact with the environment. For a fully connected graph, the regret of any agent $n$ scales as $\tilde{O}(\sqrt{KT/N})$. However, for a general graph, the regret of agent $n$ scales as $\tilde{O}(\sqrt{KT/N_n})$, where $N_n$ is the number of neighbors of agent $n$. The proposed LCC-UCB-GRAPH algorithm helps the agent to reduce the regret to $\tilde{O}(D\sqrt{DKT/N})$ by effectively propagating the knowledge about the best arm throughout the network. This approach requires significantly higher communication $O(T)$ as compared to the algorithms proposed in this paper $O(\log T)$.

We also compare with the DEMAB algorithm, proposed by Wang et al. (2020), which requires only $O(M\log(MK)$ communication rounds for known time horizons.

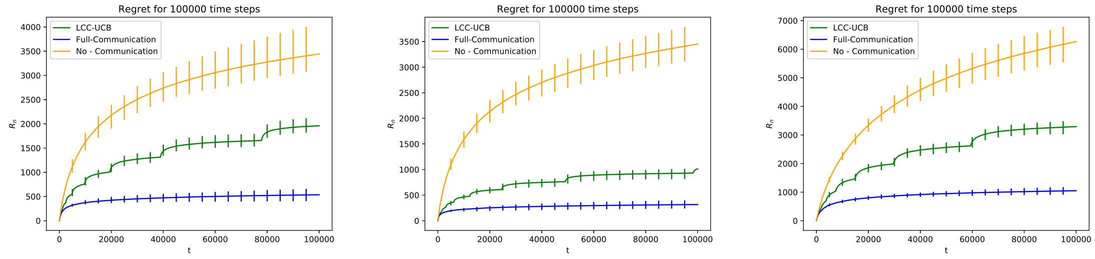
We first present the comparison results for Algorithm 1. We consider a horizon of $T = 10^5$ steps. We study the behaviour of the algorithm by varying the number of agents $N$ and the number of arms $K$. We choose three pairs $(N, K)$, which are $(10, 100)$, $(20, 100)$, $(10, 200)$. We present the result in Fig. 2 for 30 independent runs for expected rewards drawn from uniform $\mathbb{U}(0, 1)$ distribution. We plot the median of the cumulative regret incurred by a single angle at each time step and the 95% confidence intervals.

We first note that the regret of the DEMAB algorithm is even larger than the no-communication strategy. The high regret in the DEMAB algorithm is expected because the algorithm purges the observations collected after each epoch. Further, the agents do not share the knowledge of the best arm and continue to redivide the remaining arms to



(a) $(N, K) = (10, 100)$ (b) $(N, K) = (20, 100)$ (c) $(N, K) = (10, 200)$

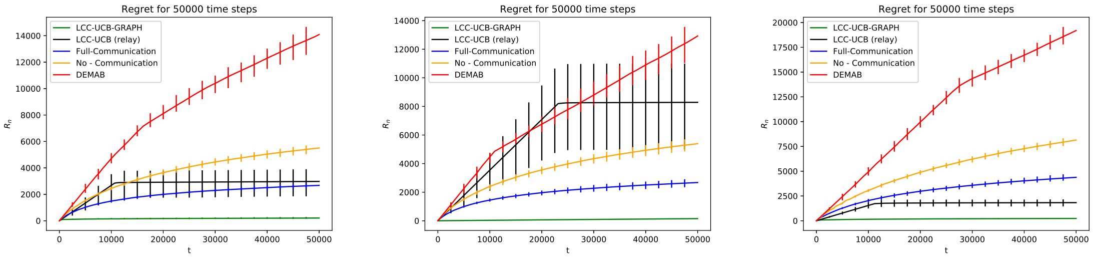
Figure 2: Per-step cumulative regret for a single agent under various communication strategies.



(a) $(N, K) = (10, 100)$ (b) $(N, K) = (20, 100)$ (c) $(N, K) = (10, 200)$

Figure 3: Per-step cumulative regret for a single agent under various communication strategies. (Excluding plots from DEMAB algorithm to the regret growth of other algorithms)

quickly eliminate the bad arms, and hence not all agents are able to exploit the best arm. This results in the high regret of the algorithm. To show the scale between the remaining communication strategies, we plot the regret curves with the DEMAB algorithm in Figure 3.



(a) $(N, K) = (100, 250)$ (b) $(N, K) = (150, 250)$ (c) $(N, K) = (100, 500)$

Figure 4: Per-step cumulative regret for a single agent, in a sparse graph, under various communication strategies.

The start of an epoch $j$ can be observed as the jumps in the cumulative regret. We observe that the initial epochs incur the largest regret despite the duration $T_j$ being small.

This is because the agents are not aware of the best arm yet and are exploring from possibly worst arms. Also, the regret grows very slowly in the later phase because most agents send the same arm index (the optimal arm) and the effective regret in the later rounds increase only as $\tilde{O}\left(\sqrt{(1+\lceil N/K\rceil)\,T_j}\right)$, instead of the upper bound of $\tilde{O}\left(\sqrt{(N-1+\lceil N/K\rceil)\,T_j}\right)$. We note that for small number of agents $N$ compared to the number of arms $K$, $(N, K) = (10, 100)$ and $(N, K) = (10, 200)$, the algorithm performs closer to the optimal case where the agents could communicate with each other as observed from Fig. 3a and Fig. 3c. This is because of the reduced overhead of re-sampling new arms obtained from all the agents.

We now evaluate the proposed LCC-UCB-GRAPH algorithm on sparse graphs. We specifically consider Erdős-Rényi graphs $G(N, p)$ where $N \geq 100$ vertices are a swarm of $N$ agents. Also, $p = 10/N \geq \ln N/N$ is the edge selection probability. This gives an expected number of total edges in the graph to be $5N$. We consider only connected graphs (If the resulting graph is not connected, we sample another graph.). Once initiated, the graph does not changes structure over the subsequent time steps. This setup is typically used in placement of IoT devices communicating with only neighbors (Avner and Mannor, 2016; Sankararaman et al., 2019).

We again consider 3 cases of $(N, K)$ which are $(100, 250)$, $(150, 250)$, and $(100, 500)$. We present the result in Fig. 4 for 30 independent runs. Along with the expected rewards of the arms, graph structure is also different for each run. We plot the median of the cumulative regret incurred by a single angle at each time step and the 95% confidence intervals.

We note that for $K = 250$, the performance is similar for $N = 100$ (Fig. 4b) and $N = 150$ (Fig. 4b). This is expected for no-communication strategy as the number of arms are same. For LCC-UCB-GRAPH algorithm, this makes sense as the degree of the graph $K_G$ is higher than the the number of arms allocated to every agent $\lceil K/N\rceil$. For full communication strategy, this happens because the expected degree of each agent is same for both graphs. Each agent can access data from only neighbors, and that remains same. On doubling $K$ from 250 to 500, we observe that the regret increases at lower rate for LCC-UCB-GRAPH than for the other two strategies. This is again attributed to the fact that $K_G$ dominates $\lceil K/N\rceil$ term in regret. We note that the performance of the DEMAB algorithm is still sub-par to the all the other three strategies. Note that the LCC-UCB-GRAPH algorithm accumulates extremely low regret because of the reduced arms per agent ($\leq 5$) and the degree of any node is also very low as we considered sparse $G(N, p)$ graphs with $p = 10/N$.

As expected, we note that the proposed strategy performs better than the no communication strategy. Further, we note that the proposed strategy even outperforms the strategy where communication happens after every time step and lags behind in initial time steps only. This is because, for the always communicating setup, an agent only shares its knowledge with its neighbors and thus is not able to fully utilize the graph with $N$ agents. For the initial time steps, the LCC-UCB-GRAPH algorithm performs pure exploration, hence incurs regret.

We also compare the performance of the LCC-UCB-GRAPH algorithm against a modified LCC-UCB algorithm which relays messages from other nodes. This modification allows every agent to receive recommendations from all the other agents after every epoch. However, the performance of the LCC-UCB-GRAPH algorithm is significantly better than the relay based LCC-UCB algorithm which justifies the sub-epoch based modification used in LCC-UCB-GRAPH. LCC-UCB algorithm wastes a significant portion of the time to generate good recommendations and hence incur a large regret. The better performance of the LCC-UCB-GRAPH algorithm is because after each epoch, an agent only receives arm updates from its neighbors, and hence, the $\sqrt{K/N + K_G}$ term in regret is very small.

## 8. Conclusion

We considered the problem of reducing communication rounds between $N$ agents and minimizing the regret of agents interacting with an instance of a Multi Armed Bandit problem with $K$ arms for time horizon $T$. We proposed two algorithm LCC-UCB for fully connected networks and LCC-UCB-GRAPH for sparse networks with maximum degree $K_G$ and diameter $D$. We analyzed the algorithms and obtain regret bound of $\tilde{O}(\sqrt{T(N + K/N)})$ and $\tilde{O}(D\sqrt{D(K/N + K_G)T})$ for LCC-UCB and LCC-UCB-GRAPH algorithms respectively. We found that the algorithms perform well empirically with the LCC-UCB-GRAPH algorithm outperforming every time communication strategy in which an agent shares knowledge only with its neighbors. Further, both the LCC-UCB and the LCC-UCB-GRAPH algorithm beat the existing state of the art results. Additionally, the low bit complexity for communication in both the algorithms makes them a suitable choice for power constrained devices. We conjecture when $N$ agents are connected with graph of diameter $D = o(N^{1/3})$, and the agents can only communicate the index of the best known arm, the lower bound of regret of any agent scales as $O(D\sqrt{D(KT/N + NT})$. However, proving the conjecture remains an open problem. As future work, the setting where the agents interact with non-identical bandit instances can be considered. Moreover, considering limited communication setups for federated reinforcement learning setups (Agarwal et al., 2021) is also an interesting direction for future work.

## References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

Mridul Agarwal, Bhargav Ganguly, and Vaneet Aggarwal. Communication efficient parallel reinforcement learning. In *Uncertainty in Artificial Intelligence*, pages 247–256. PMLR, 2021.

Shipra Agrawal and Navin Goyal. Further optimal regret bounds for thompson sampling. In *Artificial intelligence and statistics*, pages 99–107, 2013.

Jean-Yves Audibert, Sébastien Bubeck, et al. Minimax policies for adversarial and stochastic bandits. In *COLT*, volume 7, pages 1–122, 2009.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Peter Auer and Ronald Ortner. Ucb revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65, 2010.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Orly Avner and Shie Mannor. Multi-user lax communications: a multi-armed bandit approach. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016.

Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, 412(19):1832–1852, 2011.

Ronshee Chawla, Abishek Sankararaman, Ayalvadi Ganesh, and Sanjay Shakkottai. The gossiping insert-eliminate algorithm for multi-agent bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 3471–3481. PMLR, 2020.

Fan Chung and Linyuan Lu. The diameter of sparse random graphs. *Advances in Applied Mathematics*, 26(4):257–279, 2001.

Abhimanyu Dubey and Alex Pentland. Kernel methods for cooperative contextual bandits. In *International Conference on Machine Learning*, 2020a.

Abhimanyu Dubey and AlexSandy' Pentland. Differentially-private federated linear bandits. *Advances in Neural Information Processing Systems*, 33, 2020b.

John C Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2):148–164, 1979.

Eshcar Hillel, Zohar S Karnin, Tomer Koren, Ronny Lempel, and Oren Somekh. Distributed exploration in multi-armed bandits. In *Advances in Neural Information Processing Systems*, pages 854–862, 2013.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The collected works of Wassily Hoeffding*, pages 409–426. Springer, 1994.

Varun Kanade, Zhenming Liu, and Bozidar Radunovic. Distributed non-stochastic experts. In *Advances in Neural Information Processing Systems*, pages 260–268, 2012.

Peter Landgren, Vaibhav Srivastava, and Naomi Ehrich Leonard. Distributed cooperative decision-making in multiarmed bandits: Frequentist and bayesian algorithms. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 167–172. IEEE, 2016.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Tor Lattimore, Branislav Kveton, Shuai Li, and Csaba Szepesvari. Toprank: A practical algorithm for online stochastic ranking. In *Advances in Neural Information Processing Systems*, pages 3945–3954, 2018.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

David Martínez-Rubio, Varun Kanade, and Patrick Rebeschini. Decentralized cooperative stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 4529–4540, 2019.

Abishek Sankararaman, Ayalvadi Ganesh, and Sanjay Shakkottai. Social learning in multi agent multi armed bandits. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(3):1–35, 2019.

Shahin Shahrampour, Alexander Rakhlin, and Ali Jadbabaie. Multi-armed bandits in multi-agent networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2786–2790. IEEE, 2017.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Po-An Wang, Alexandre Proutiere, Kaito Ariu, Yassir Jedra, and Alessio Russo. Optimal algorithms for multiplayer multi-armed bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 4120–4129. PMLR, 2020.

Yuanhao Wang, Jiachen Hu, Xiaoyu Chen, and Liwei Wang. Distributed bandit learning: Near-optimal regret with efficient communication. In *International Conference on Learning Representations*, 2019.

Jiaqi Yang, Wei Hu, Jason D Lee, and Simon Shaolei Du. Impact of representation learning in linear bandits. In *International Conference on Learning Representations*, 2020.