Decentralized Randomly Distributed Multi-agent Multi-armed Bandit with Heterogeneous Rewards

Mengfan Xu
Department of Industrial Engineering and Management Sciences
Northwestern University
Evanston, IL 60208, U.S.A.
MengfanXu2023@u.northwestern.edu

Diego Klabjan
Department of Industrial Engineering and Management Sciences
Northwestern University
Evanston, IL 60208, U.S.A.
d-klabjan@northwestern.edu

Abstract

We study a decentralized multi-agent multi-armed bandit problem in which multiple clients are connected by time dependent random graphs provided by an environment. The reward distributions of each arm vary across clients and rewards are generated independently over time by an environment based on distributions that include both sub-exponential and sub-gaussian distributions. Each client pulls an arm and communicates with neighbors based on the graph provided by the environment. The goal is to minimize the overall regret of the entire system through collaborations. To this end, we introduce a novel algorithmic framework, which first provides robust simulation methods for generating random graphs using rapidly mixing Markov chains or the random graph model, and then combines an averaging-based consensus approach with a newly proposed weighting technique and the upper confidence bound to deliver a UCB-type solution. Our algorithms account for the randomness in the graphs, removing the conventional doubly stochasticity assumption, and only require the knowledge of the number of clients at initialization. We derive optimal instance-dependent regret upper bounds of order $\log T$ in both sub-gaussian and sub-exponential environments, and a nearly optimal mean-gap independent regret upper bound of order $\sqrt{T}\log T$ up to a $\log T$ factor. Importantly, our regret bounds hold with high probability and capture graph randomness, whereas prior works consider expected regret under assumptions and require more stringent reward distributions.

1 Introduction

Multi-armed Bandit (MAB) [Auer et al., 2002a,b] is an online sequential decision-making process that balances exploration and exploitation while given partial information. In this process, a single player (agent, client) aims to maximize a cumulative reward or, equivalently, minimize the cumulative loss, known as regret, by pulling an arm and observing the reward of that arm at each time step. The two variants of MAB are adversarial and stochastic MAB, depending on whether rewards are chosen arbitrarily or follow a time-invariant distribution, respectively. Recently, motivated by the development of federated learning [McMahan et al., 1273–1282, 2017], multi-agent stochastic multi-armed bandit has been drawing increasing attention (commonly referred to as multi-agent MAB). In this variant, multiple clients collaboratively work with multiple stochastic MABs to maximize the overall performance of the entire system. Likewise, regret is an important performance

Preprint. Under review.

measure, which is the difference between the cumulative reward of always pulling the global optimal arm by all clients and the actual cumulative reward gained by the clients at the end of the game, where global optimality is defined with respect to the average expected reward values of arms across clients. Thereafter, the question for each client to answer is essentially how to guarantee an optimal regret with limited observations of arms and insufficient information of other clients. Assuming the existence of a central server, also known as the controller, [Bistritz and Leshem, 2018, Zhu et al., 3–4, 2021, Huang et al., 2021, Mitra et al., 2021, Réda et al., 2022, Yan et al., 2022], allow a controller-client framework where the controller integrates and distributes the inputs from and to clients, adequately addressing the challenge posed by the lack of information of other clients. However, this centralization implicitly requires all clients to communicate with one another through the central server and may fail to include common networks with graph structures where clients perform only pair-wise communications within the neighborhoods on the given graphs. Non-complete graphs capture the reality of failed communication links. Removing the centralization assumption leads to a decentralized multi-agent MAB problem, which is a challenging but attracting direction as it connects the bandit problem and graph theory, and precludes traditional centralized processing.

In the field of decentralized multi-agent MAB, it is commonly assumed that the mean reward value of an arm for different clients is the same, or equivalently, homogeneous. This assumption is encountered in [Landgren et al., 2016a,b, 2021, Zhu et al., 2020, Martínez-Rubio et al., 2019, Agarwal et al., 2022]. However, this assumption may not always hold in practical scenarios. In recent years, there has been an increasing emphasis on heterogeneous reward settings, where clients can retain different mean values for the rewards of the same arm. The transition to heterogeneous reward settings presents additional technical challenges. Clients are unable to infer the global optimal arms without sequential communications regarding the rewards of the same arm at other clients. Such communications, however, are limited by the partially observed rewards, as other clients may not pull the same arm, and constrained by the underlying graph structure. We study the heterogeneous setting with time varying graphs.

Traditionally, rewards are assumed to be sub-Gaussian distributed. However, there has been a recent focus on MAB with heavy-tailed reward distributions. This presents a non-trivial challenge as it is harder to concentrate reward observations in sublinear time compared to the light-tailed counterpart [Tao et al., 1546–1574, 2022]. In the work of [Jia et al., 2021], sub-exponential rewards are considered and analyzed in the single-agent MAB setting with newly proposed upper confidence bounds. Meanwhile, for multi-agent MAB, heavy-tailed distributions are examined in a homogeneous setting in [Dubey and Pentland, 2730–2739, 2020]. However, the heterogeneous setting studied herein has not yet been formulated or analyzed, posing more challenges compared to the homogeneous setting, as discussed earlier.

Besides rewards, the underlying graph assumptions are essential to the decentralized multi-agent MAB problem, as increased communication among clients leads to better identification of global optimal arms and smaller regret. There are two types of graphs from a time perspective: time-invariant graphs, which remain constant over time, and time-varying graphs, which depend on time steps and are more challenging but more general. Assumptions on time-invariant graphs include complete graphs [Wang et al., 1531–1539, 2021] where all clients can communicate, regular graphs [Jiang and Cheng, 1–33, 2023] where each client has the same number of neighbors, and connected graphs under the doubly stochasticity assumption [Zhu et al., 2020, 2021, 3–4, 2021]. Independently from our work, recent work [Zhu and Liu, 2023] has focused on time-varying B-connected graphs, where the composition of every l consecutive graphs is a strongly connected graph. However, their doubly stochasticity assumption, where all elements of edge probability also called weight matrices are uniformly bounded by a positive constant, can be violated in several cases. Additionally, their graphs may be strongly correlated to meet the connectivity condition when $l > 1$, which may not always hold in practice. No research has been conducted on time-varying graphs with only connectivity constraints or without constraints on connectivity. Additionally, current time-varying graphs do not provide insight into how the graphs change over time. As the graphs are generated by the environment, similar to the generation of rewards, it remained unexplored considering random graphs in an i.i.d manner, such as random edge failures or random realizations as pointed out for future research in [Martínez-Rubio et al., 2019]. We also address this situation.

Traditionally, random graphs have often been formulated using the Erdős–Rényi (E-R) model, which has been widely adopted in various research domains. The model, described by $G(M, c)$, consists of M vertices with each pair of vertices being connected with probability c. Notably, the E-R model is 1) not necessarily connected and 2) stochastic that allows for random edge failures, and has found applications in mean-field game [Delarue, 2017] and majority vote settings [Lima et al., 2008]. Though it has only been used in numerical experiments for the decentralized multi-agent MAB setting with homogeneous rewards [Dubey and Pentland, 2730–2739, 2020], the theoretical study of this model in this context remained unexplored until this work, let alone with heterogeneous rewards and time-varying graphs. Alternatively, one can consider all connected graphs (there are exponentially many of them), and the environment can randomly sample a connected graph and produce i.i.d. samples of such random connected graphs. This approach mimics the behavior of stochastic rewards and allows the environment to exhaust the sample space of connected graphs independently, without the doubly stochasticity assumption, which, however, has not yet been studied and it is also addressed herein.

For the multi-agent MAB framework, methods in MAB are a natural extension. [Zhu et al., 3–4, 2021] adapt the UCB algorithm to the multi-agent setting. This algorithm uses weighted averages to achieve consensus among clients and is shown to have a regret of order $\log T$ for time-invariant graphs. A follow-up study in [Zhu and Liu, 2023] re-analyzes this algorithm for time-varying B-connected graphs under the aforementioned assumptions under the doubly stochasticity assumption by adding an additional term compared to UCB. An effective UCB-based method for random graphs without doubly stochasticity assumption and for sub-exponential distributed rewards remained unexplored.

This paper presents a novel contribution to the decentralized multi-agent MAB problem by studying both heterogeneous rewards and time-varying random graphs, where the distributions of rewards and graphs are independent of time. To the best of our knowledge, this is the first work to consider this problem and to investigate it with heavy-tailed reward distributions. Specifically, the paper investigates 1) heterogeneous sub-exponential and sub-gaussian distributed rewards and 2) random graphs including the possibly disconnected E-R model and random connected graphs, and applies them to the decentralized multi-agent MAB framework. This work bridges the gap between large-deviation theories for sub-exponential distributions and multi-agent MAB with heterogeneous rewards, and the gap between random graphs and decentralized multi-agent MAB.

To this end, we propose a brand new algorithmic framework consisting of three main components: graph generation, DrFed-UCB: burn-in period, and DrFed-UCB: learning period. For the learning period, we modify the algorithm by [Zhu et al., 3–4, 2021] by introducing new UCB quantities that are consistent with the conventional UCB algorithm and generalize to sub-exponential settings. We also introduce a newly proposed stopping time and a new weight matrix without the doubly stochasticity assumption to leverage more information in random graphs. A burn-in period is crucial in estimating the graph distribution and initializing the weight matrix. We embed and analyze techniques from random graphs since the number of connected graphs is exponentially large in the number of vertices, and directly sampling such a graph is an NP-hard problem. In particular, we use the Metropolis-Hastings method with rapidly mixing Markov chains, as proposed in [Gray et al., 2019], to approximately generate random connected graphs in polynomial time. We additionally demonstrate its theoretical convergence rate, making it feasible to consider random connected graphs in the era of large-scale inference.

We present comprehensive analyses of the regret of the proposed algorithm, using the same regret definition as in existing literature. Firstly, we show that algorithm DrFed-UCB achieves optimal instance-dependent regret upper bounds of order $\log T$ with high probability, in both sub-gaussian and sub-exponential settings, consistent with prior works. We add that although both [Zhu et al., 2020] and our analyses use the UCB framework, the important algorithmic steps are different and thus also the analyses. Secondly, we demonstrate that with high probability, the regret is universally upper bounded by $O(\sqrt{T}\log T)$ in sub-exponential settings, including sub-gaussian settings. This upper bound matches the upper and lower bounds in single-agent settings up to a $\log T$ factor, establishing its tightness.

The paper is organized as follows. We first introduce the notations used throughout the paper, present the problem formulation, and propose algorithms for solving the problem. Following that, we provide theoretical results on the regret of the proposed algorithm in various settings.

2 Problem Formulation and Methodologies

2.1 Problem Formulation

Throughout, we consider a decentralized system with M clients that are labeled as nodes $1, 2, \ldots, M$ on a time-varying network, which is described by an undirected graph G_t for $1 \leq t \leq T$ where parameter T denotes the time horizon of the problem. Formally, at time step t, $G_t = (V, E_t)$ where $V = \{1, 2, \ldots, M\}$ and E_t denotes the edge set consisting of pair-wise nodes and representing the neighborhood information in G_t. The neighbor set $\mathcal{N}_m(t)$ include all neighbors of client m based on G_t. Equivalently, the graph G_t can be represented by the adjacency matrix $(X^t_{i,j})_{1 \leq i,j \leq M}$ where the element $X^t_{i,j} = 1$ if there is an edge between clients i and j and $X^t_{i,j} = 0$ otherwise. We let $X_{i,i} = 1$ for any $1 \leq i \leq M$. With this notation at hand, we define the empirical graph (adjacency matrix) P_t as $P_t = \frac{(\sum_{s=1}^{t} X^s_{i,j})_{1 \leq i,j \leq M}}{t}$. It is worth emphasizing that 1) the matrix P_t is not necessarily doubly stochastic, 2) the matrix captures more information about G_t than the prior works based on $|\mathcal{N}_m(t)|$, and 3) each client m only knows the m-th row of P_t without knowledge of G_t, i.e. $\{P_t(m, j)\}_j$ are known to client m, while $\{P_t(k, j)\}_j$ for $k \neq m$ are always unknown. Let us denote the set of all possible connected graphs on M nodes as $\mathcal{G}_{\mathcal{M}}$.

We next consider the bandit problems faced by the clients. In the MAB setting, the environment generates rewards. Likewise, we again use the term, the environment, to represent the source of graphs G_t and rewards $r^m_i(t)$ in the decentralized multi-agent MAB setting. Formally, there are K arms faced by each client. At each time step t, for each client $1 \leq m \leq M$, let the reward of arm $1 \leq i \leq K$ be $r^m_i(t)$, which is i.i.d. distributed across time with the mean value μ^m_i, and is drawn independently across the clients. Here we consider a heterogeneous setting where μ^m_i is not necessarily the same as μ^j_i for $m \neq j$. At each time step t, client m pulls an arm a^t_m, only observes the reward of that arm $r^m_{a^t_m}(t)$ from the environment, and exchanges information with neighbors in G_t given by the environment. In other words, two clients communicate only when there is an edge between them.

By taking the average over clients as in the existing literature, we define the global reward of arm i at each time step t as $r_i(t) = \frac{1}{M} \sum_{m=1}^{M} r^m_i(t)$ and the subsequent expected value of the global reward as $\mu_i = \frac{1}{M} \sum_{m=1}^{M} \mu^m_i$. We define the global optimal arm as $i^* = \arg\max_i \mu_i$ and arm $i \neq i^*$ is called global sub-optimal. Let $\Delta_i = \mu_{i^*} - \mu_i$ be the sub-optimality gap. This enables us to quantify the regret of the action sequence (policy) $\{a^t_m\}^{1 \leq t \leq T}_{1 \leq m \leq M}$ as follows. Ideally, clients would like to pull arm i^* if knowledge of $\{\mu_i\}_i$ were available. Given the partially observed rewards due to bandits (dimension i) and limited accesses to information from other clients (dimension m), the regret is defined as

$$R_T = T\mu_{i^*} - \frac{1}{M} \sum_{t=1}^{T} \sum_{m=1}^{M} \mu^m_{a^t_m}$$

which measures the difference of the cumulative expected reward between the global optimal arm and the action sequence. The main objective of this paper is to develop theoretically robust solutions to minimize R_T for clients operating on time-varying random graphs that are vulnerable to random communication failures, which only require knowledge of M.

2.2 Algorithms

In this section, we introduce a new algorithmic framework that incorporates two graph generation algorithms, one for the E-R model and the other for uniformly distributed connected graphs. More importantly, the framework includes a UCB-variant algorithm that runs a learning period after a burn-in period, which is commonly referred to as a warm-up phase in statistical procedures.

2.2.1 Graph Generation

We investigate two types of graph dynamics as follows, for which we propose simulation methods that enable us to generate and analyze the resulting random graphs.

E-R random graphs At each time step t, the adjacency matrix of graph G_t is generated by the environment by element-wise sampling $X^t_{i,j}$ according to a Bernoulli distribution. Specifically, $X^t_{i,j}$ follows a Bernoulli distribution with parameter c.

Uniformly distributed connected graphs At each time step t, the random graph G_t is generated by the environment by uniformly sampling a graph from the sample space of all connected graphs $\mathcal{G}_M$, which yields the adjacency matrix $(X^t_{i,j})_{1\leq i\neq j\leq M}$ corresponding to G_t. Generating uniformly distributed connected graphs is presented in Algorithm 1. It is computationally infeasible to exhaust the sample space $\mathcal{G}_M$ since the number of connected graphs is exponentially large. To this end, we import the Metropolis-Hastings method in [Gray et al., 2019] and leverage rapidly mixing Markov chains. Remarkably, by adapting the algorithm into our setting which yields Algorithm 1, we construct a Markov chain that converges to the target distribution after a finite number of burn-in steps. This essentially traverses graphs in $\mathcal{G}_M$ through step-wise transitions, with a complexity of $O(M^2)$ from previous states. More precisely, at time step s with connected graph G_s, we randomly choose a pair of nodes and check whether it exists in the edge set. If this is the case, we remove the edge from G_s and check whether the remaining graph is connected, and only accept the graph as G_{s+1} in the connected case. If the edge does not exist in the edge set, we add it to G_s and get G_{s+1}. In this setting, let $c = c(M)$ be the number of times an edge is present among all connected graphs divided by the total number of connected graphs. It is known that

$$c = 2\frac{\log M}{M-1}, \tag{1}$$

[Trevisan]. The distribution of G_s eventually converges to the uniform distribution in $\mathcal{G}_M$. The formal statements are in Appendix.

Algorithm 1: Generate a uniformly distributed connected graph

```
Initialization: Let $\tau_1$ be given; Generate a random graph $G^{init}$ by selecting each edge with
  probability $\frac{1}{2}$;
Connectivity: make $G^{init}$ connected by adding the least many edges to get $G_0$ ;
for $t = 0, 1, 2, \ldots, \tau_1$ do
    Randomly sample an edge pair $e = (i, j)$;
    Denote the edge set of $G_s$ as $E_s$;
    if $e \in E_s$ then
        Remove $e$ from $E_s$ to get $G'_s = (V, E_s \backslash \{e\})$;
        if $G'_s$ is connected then
            $G_{s+1} = G'_s$;
        else
            reject $G'_s$ and set $G_{s+1} = G_s$;
        end
    else
        $G_{s+1} = (V, E_s \cup \{e\})$;
    end
end
```

2.2.2 Main Algorithm

In the following, we present the proposed algorithm, DrFed-UCB, which comprises of a burn-in period and a learning period described in Algorithm 2 and Algorithm 3, respectively.

We start by introducing the variables used in the algorithm with respect to client m. We use $\bar{\mu}^m_i(t), n_{m,i}(t)$ to denote reward estimators and counters based on client m's own pulls of arm i, respectively, and use $\tilde{\mu}^m_i, N_{m,i}(t)$ to denote reward estimators and counters based on the network-wide pulls of arm i, respectively. By network-wide, we refer to the clients in $\mathcal{N}_m(t)$. Denote the stopping time for the filtration $\{G_s\}^t_{s=1}$ as $h^t_{m,j} = max_{s\leq t}\{(m,j) \in E_s\}$; it represents the most recent communication between clients m and j. The weights for the network-wide and local estimators are $P'_t(m,j)$ and $d_{m,t}$ defined later, respectively.

There are two stages in Algorithm 2 where $t \leq L$ as follows. For the first τ_1 steps, the environment generates graphs based on one of the aforementioned graph generation algorithms to arrive at the steady state, while client m pulls arms randomly and updates local estimators $\{\bar{\mu}^m_i, n^m_i\}_i$. Afterwards, the environment generates the graph G_t that follows the distribution of interest, while client m updates $\{\bar{\mu}^m_i, n^m_i\}_i$ and row m of P_t by exchanging information with its neighbors in the graph G_t. Note that client m does not have any global information about G_t. At the end of the burn-in period, client m computes the network-wide estimator $\tilde{\mu}_i^m$ by taking the weighted average of local estimators of other clients (including itself), where the weights are given by the m-th row of weight matrix P' and d, which depend on P and knowledge of M and satisfy $\sum_{j=1}^M P_t'(m,j) + d_{m,t}M = 1$.

Subsequently, we describe Algorithm 3 where $t \geq L + 1$. There are four phases in one iteration enumerated below in the order indicated.

UCB Given the estimators $\tilde{\mu}_i^m(t), n_{m,i}(t), N_{m,i}(t), \bar{\mu}_i^m(t)$, client m either randomly samples an arm or pulls the arm that maximizes the upper confidence bound using $\tilde{\mu}_i^m(t), n_{m,i}(t)$, depending on whether $n_{m,i}(t) \leq N_{m,i}(t) - K$ holds for some arm i. This additional condition ensures consensus among clients regarding which arm to pull. The upper confidence bound $F(m,i,t)$ is specified as $F(m,i,t) = \sqrt{\frac{C_1 \ln t}{n_{m,i}(t)}}$ and $F(m,i,t) = \sqrt{\frac{C_1 \ln T}{n_{m,i}(t)}} + \frac{C_2 \ln T}{n_{m,i}(t)}$ in settings with sub-gaussian and sub-exponential rewards, respectively. Constants C_1 and C_2 are determined in the analyses and they depend on σ which is an upper bound of standard deviations of the reward values (it is formally defined later).

Environment and client interaction After client m pulls arm a_m^t, the environment sends the reward $r_{m,a_m^t}^t$ and the neighbor set $\mathcal{N}_m(t)$ in G_t to client m. Client m does not know the whole G_t and obtains only the neighbor set $\mathcal{N}_m(t)$.

Transmission Client m sends the maintained local and network-wide estimators $\{\bar{\mu}_i^m(t), \tilde{\mu}_i^m(t), n_{m,i}(t), N_{m,i}(t)\}_i$ to all clients in $\mathcal{N}_m(t)$ and receives the ones from them.

Update estimators At the end of an iteration, client m first updates the m-th row of matrix P_t. Subsequently, client m updates the quantities$\{\bar{\mu}_i^m(t), \tilde{\mu}_i^m(t), n_{m,i}(t), N_{m,i}(t)\}_{1 \leq i \leq K}$ adhereing to:

$$
\begin{aligned}
& t_{m,j} = max_{s \geq \tau_1}\{(m,j) \in E_s)\} \text{ and 0 if such an } s \text{ does not exist} \\
& n_{m,i}(t+1) = n_{m,i}(t) + \mathbb{1}_{a_m^t = i}, N_{m,i}(t+1) = \max\{n_{m,i}(t+1), \hat{\tilde{N}}_{i,j}^m(t), j \in \mathcal{N}_m(t)\} \\
& \bar{\mu}_i^m(t+1) = \frac{\bar{\mu}_i^m(t) \cdot n_{m,i}(t) + r_{m,i}(t) \cdot \mathbb{1}_{a_m^t = i}}{n_{m,i}(t+1)} \\
& P_t'(m,j) = \frac{M-1}{M^2} \text{ if } P_t(m,j) > 0 \text{ and 0 otherwise} \qquad (2) \\
& \tilde{\mu}_i^m(t+1) = \sum_{j=1}^M P_t'(m,j)\hat{\tilde{\mu}}_{i,j}^m(t_{m,j}) + d_{m,t} \sum_{j \in N_m(t)} \hat{\bar{\mu}}_{i,j}^m(t) + d_{m,t} \sum_{j \notin N_m(t)} \hat{\bar{\mu}}_{i,j}^m(t_{m,j}) \\
& \qquad \text{with } d_{m,t} = \frac{1 - \sum_{j=1}^M P_t'(m,j)}{M}
\end{aligned}
$$

Similar to [Zhu et al., 3–4, 2021, Zhu and Liu, 2023], the algorithm balances between exploration and exploitation by the upper confidence bound and a criterion on $n_{m,i}(t)$ and $N_{m,i}(t)$ that ensures that all clients explore arms at similar rates and thereby "staying on the same page." After interacting with the environment, clients move to the transmission stage, where they share information with the neighbors on G_t, as a preparation for the update stage.

Different from the upper confidence bound approach in [Zhu and Liu, 2023], which has an extra term of $\frac{1}{t}$ in the UCB criterion, our proposal is aligned with the conventional UCB algorithm. Meanwhile, our update rule differs from that in [Zhu et al., 3–4, 2021, Zhu and Liu, 2023] in three key aspects: (1) maintaining a stopping time $t_{m,j}$ that tracks the most recent communication to client j, and (2) updating $\tilde{\mu}_i^m$ based on both $\tilde{\mu}_i^j$ and $\bar{\mu}_i^j$ for $j \in \mathcal{N}_m(t)$, and (3) using a weight matrix based on P_t' and P_t computed from the trajectory $\{G_s\}_{s \leq t}$ in the previous steps. The first point ensures that the latest information from other clients is leveraged, in case there is no communication at the current time step. The second point ensures proper integration of both network-wide and local information, smoothing out biases from local estimators and reducing variances through averaging. The third point distills the information carried by the time-varying graphs and determines the weights of the available local and network-wide estimators, removing the need for the doubly stochasticity assumption. The algorithm assumes that the clients know M and σ^2.

We note that $t_{m,j}$ is the stopping time by definition and that $\bar{\mu}_i^m$ is an unbiased estimator for μ_i^m with a decaying variance proxy. Meanwhile, the matrices P_t' and P_t are not doubly stochastic and

Algorithm 2: DrFed-UCB: Burn-in period

```
Initialization: The length of the burn-in period is $L$ and we are also given $\tau_1 < L$; In the time
  step $t = 0$, the estimates are initialized as $\bar{\mu}_i^m(0) = 0$, $n_{m,i}(0) = 0$, $\hat{\bar{\mu}}_{i,j}^m(0) = 0$, and
  $P_0(m, j) =$ for any arm $i$ and clients $m, j$;
for $1 \leq t \leq \tau_1$ do
    The environment generates a sample graph $G_t = (V, E_t)$ based on either E-R or
      Algorithm 1;
end
for $1 \leq t \leq \tau_1$ do
    for each client $m$ do
        Sample arm $a_t^m = (t \mod K)$;
        Receive reward $r_{a_t^m}^m(t)$ and update $n_{m,i}(t) = n_{m,i}(t-1) + \mathbb{1}_{a_m^t = i}$;
        Update the local estimate for any arm $i$: $\bar{\mu}_i^m(t) = \frac{n_{m,i}(t-1)\bar{\mu}_i^m(t-1) + r_{a_t^m}^m(t) \cdot 1_{a_t^m = i}}{n_{m,i}(t-1) + 1_{a_t^m = i}}$;
    end
end
for $\tau_1 < t \leq L$ do
    The environment generates a sample graph $G_t = (V, E_t)$ based on either E-R or
      Algorithm 1;
    for each client $m$ do
        Sample arm $a_t^m = (t \mod K)$;
        Receive rewards $r_{a_t^m}^m(t)$ and update $n_{m,i}(t) = n_{m,i}(t-1) + \mathbb{1}_{a_m^t = i}$;
        Update the local estimates for any arm $i$: $\bar{\mu}_i^m(t) = \frac{n_{m,i}(t-1)\bar{\mu}_i^m(t-1) + r_{a_t^m}^m(t) \cdot 1_{a_t^m = i}}{n_{m,i}(t-1) + 1_{a_t^m = i}}$;
        Update the maintained matrix $P_t(m, j) = \frac{(t-1)P_{t-1}(m,j) + X_{m,j}^t}{t}$ for each $j \in V$;
        Send $\{\bar{\mu}_i^m(t)\}_{i=1}^{i=K}$ to all clients in $\mathcal{N}_m(t)$;
        Receive $\{\bar{\mu}_i^j(t)\}_{i=1}^{i=K}$ from all clients $j \in \mathcal{N}_m(t)$ and store them as $\hat{\bar{\mu}}_{i,j}^m(t)$.
    end
end
for each client $m$ and arm $i$ do
    For client $1 \leq j \leq M$, set $h^L(m, j) = \max_{s \geq \tau_1}\{(m, j) \in E_s\}$ or 0 if such $s$ does not exist
    $\tilde{\mu}_i^m(L+1) = \sum_{j=1}^M P'_{m,j}(L)\hat{\bar{\mu}}_{i,j}^m(h_{m,j}^L)$ where $P'_{m,j}(L) = \begin{cases} \frac{1}{M} & \text{if } P_L(m,j) > 0 \\ 0 & \text{otherwise} \end{cases}$;
end
```

keep track of the history of the random graphs. By introducing $t_{m,j}$ and P_t, we can show that the global estimator $\tilde{\mu}_i^m(t)$ behaves similarly to a sub-gaussian/sub-exponential random variable with an expectation of μ_i and a time-decaying variance proxy proportional to $\frac{1}{\min_j n_{j,i}(t)}$. This ensures that the concentration inequality holds for $\tilde{\mu}_i^m(t)$ with respect to μ_i and that client m tends to identify the global optimal arms with high probability, which plays an important role in minimizing regret. The formal statements are presented in the next section.

3 Regret Analyses

In this section, we show the theoretical guarantees of the proposed algorithm, assuming mild conditions on the environment. Specifically, we consider various settings with different model assumptions. We prove that the regret of Algorithm 3 has different instance-dependent upper bounds of order $\log T$ for settings with sub-gaussian and sub-exponential distributed rewards, and a mean-gap independent upper bound of order $\sqrt{T}\log T$ across settings. Many researchers call such a bound instance independent but we believe such a terminology is misleading and thus we prefer to call it man-gap independent, given that it still has dependency on parameters pertaining to the problem instance. The results are consistent with the regret bounds in prior works.

Algorithm 3: DrFed-UCB: Learning period

```
Initialization: For each client $m$ and arm $i \in \{1, 2, \dots, K\}$, we have $\tilde{\mu}_i^m(L+1)$,
  $N_{m,i}(L+1) = n_{m,i}(L)$; all other values at $L+1$ are initialized as 0;
for $t = L+1, L+2, \dots, T$ do
    for each client m do                                                          // UCB
        if there is no arm $i$ such that $n_{m,i}(t) \le N_{m,i}(t) - K$ then
            $a_m^t = \arg\max_i \tilde{\mu}_{m,i}(t) + F(m,i,t)$
        else
            Randomly sample an arm $a_m^t$.
        end
        Pull arm $a_m^t$ and receive reward $r_{m,a_m^t}(t)$;
    end
    The environment generates a sample graph $G_t = (V, E_t)$
      based on E-R or Algorithm 1;                                                // Env
    Each client $m$ sends $\mu_i^m(t), N_{j,i}(t), \bar{\mu}_i^m(t), \tilde{\mu}_i^m(t)$ to each client in $\mathcal{N}_m(t)$;
    Each client $m$ receives $\mu_i^j(t), N_{j,i}(t), \bar{\mu}_i^j(t), \tilde{\mu}_i^j(t)$ from all clients $j \in \mathcal{N}_m(t)$ and stores
      them as $\hat{\mu}_{i,j}^m(t), \hat{N}_{i,j}^m(t), \hat{\bar{\mu}}_{i,j}^m(t), \hat{\tilde{\mu}}_{i,j}^m(t)$;            // Transmission
    for each client m do
        for $i = 1, \dots, K$ do
            Update $P_t$ for $1 \le j \le M$ by $P_t(m,j) = \frac{(t-1)P_{t-1}(m,j) + X_{m,j}^t}{t}$;
            Update $P'_t$ for $1 \le j \le M$ by $P'_t(m,j) = \begin{cases} 1 & \text{if } P_t(m,j) > 0 \\ 0 & \text{if } P_t(m,j) = 0 \end{cases}$;
            Update $n_{m,i}(t), N_{m,i}(t)$ and $\tilde{\mu}_i^m(t)$ based on equations (2);
        end
    end
end
```

3.1 Model Assumptions

By definition, the environment is determined by how the graphs (E-R or uniform) and rewards are generated.

For reward we consider two cases.

Sub-g At time step t, the reward of arm i at client m has bounded support $[0, 1]$, and is drawn from a sub-gaussian distribution with mean $0 \le \mu_i^m \le 1$ and variance proxy $0 \le (\sigma_i^m)^2 \le \sigma^2$.

Sub-e At time step t, the reward of arm i at client m has bounded support $[0, 1]$, and follows a sub-exponential distribution with mean $0 \le \mu_i^m \le 1$ and parameters $0 \le (\sigma_i^m)^2 \le \sigma^2, 0 \le \alpha_i^m \le \alpha$.

With these considerations, we investigate four different environments (settings) based on the two graph assumptions and the two reward assumptions: Setting 1.1 corresponds to E-R and Sub-g, Setting 1.2 to Uniform and Sub-g, Setting 2.1 to E-R and Sub-e, and Setting 2.2 to Uniform and Sub-e.

For each setting, we derive upper bounds on the regret in the next section.

3.2 Regret Analyses

In this section, we establish the regret bounds formally when clients adhere to Algorithm 3 in various settings. We denote Setting 1.1, Setting 1.2 with $M < 11$, and Setting 1.2 with $M \ge 11$ as s_1, s_2 and s_3, respectively. Likewise, we denote Setting 2.1, Setting 2.2 with $M < 11$, and Setting 2.2 with $M \ge 11$ as S_1, S_2 and S_3, respectively. See Table 1 for a tabular view of the various settings.

Note that the randomness of R_T arises from both the reward and graph observations. Considering S_1, S_2, S_3 differ in the reward assumptions compared to s_1, s_2, s_3, we define an event A that preserves the properties of the variables with respect to the random graphs. Given the length of the burn-in period L_i for $i \in \{s_1, s_2, s_3\}$ and the fact that $L_{s_i} = L_{S_i}$ since it only relies on the graph assumptions, we use L to denote $\max_i L_{s_i}$. Parameters $0 < \delta, \epsilon < 1$ are any constants, and

Table 1: Settings

	E-R	uniform	M	reward
s_1	✓		any	sub-g
s_2		✓	$[1, 10]$	sub-g
s_3		✓	$[11, \infty)$	sub-g
S_1	✓		any	sub-e
S_2		✓	$[1, 10]$	sub-e
S_3		✓	$[11, \infty)$	sub-e

the parameter $c = c(M)$ represents the mean value of the Bernoulli distribution in s_1, S_1 and the probability of an edge in s_2, S_2, s_3, and S_3 among all connected graphs (see (1)). We define events $A_1 = \{\forall t \geq L, ||P_t - cE||_\infty \leq \delta\}, A_2 = \{\exists t_0, \forall t \geq L, \forall j, \forall m, t + 1 - \min_j t_{m,j} \leq t_0 \leq c_0 \min_l n_{l,i}(t+1)\}$, and $A_3 = \{\forall t \geq L, G_t \text{ is connected}\}$. Here E is the matrix with all values of 1. Constant $c_0 = c_0(K, \min_{i \neq i^*} \Delta_i, M, \epsilon, \delta)$ is defined later. Since $c = c(M)$ this implies that G_t depends on M. We define $A = A_{\epsilon,\delta} = A_1 \cap A_2 \cap A_3$, which yields $A \in \Sigma$ with Σ being the sub-σ-algebra formed by $\{\Omega, \emptyset, A, A^c\}$. This implies $E[\cdot|A]$ and $P[\cdot|A]$ are well-defined, since A only relies on the graphs and removes the differences among s_1, s_2, s_3 (S_1, S_2, S_3), enabling universal regret upper bounds.

Next, we demonstrate that event A holds with high probability.

Theorem 1. *For event $A_{\epsilon,\delta}$ and any $1 > \epsilon, \delta > 0$, we have $P(A_{\epsilon,\delta}) \geq 1 - 7\epsilon$.*

Proof Sketch. The complete proof is deferred to Appendix; we discuss the main logic here. The proof relies on bounding the probabilities of A_1, A_2, A_3 separately. For A_1, its upper bound holds by the analysis of the mixing time of the Markov chain underlying G_t and on the matrix-form Hoeffding inequality. We obtain an upper bound on $P(A_2)$ by analyzing the stopping time $t_{m,j}$ and the counter $n_{m,i}(t)$. For the last term $P(A_3)$, we show that the minimum degree of G_t has a high probability lower bound that is sufficient for claiming the connectivity of G_t. To put all together, we use the Bonferroni's inequality and reach the lower bound of $P(A_{\epsilon,\delta})$. □

Subsequently, we have the following general upper bound on the regret R_T of Algorithm 3 in the high probability sense, which holds on A in any of the settings s_1, s_2, s_3 with sub-gaussian rewards.

Theorem 2. *Let f be a function specific to a setting and detailed later. For every $0 < \epsilon < 1$ and $0 < \delta < f(\epsilon, M, T)$, in setting s_1 with $c \geq \frac{1}{2} + \frac{1}{2}\sqrt{1 - (\frac{\epsilon}{MT})^{\frac{2}{M-1}}}$, s_2 and s_3, with the time horizon T satisfying $T \geq L$, the regret of Algorithm 3 with $F(m, i, t) = \sqrt{\frac{C_1 \ln t}{n_{m,i}(t)}}$ satisfies that*

$$E[R_T|A_{\epsilon,\delta}] \leq L + \sum_{i \neq i^*} (\max\{[\frac{4C_1 \log T}{\Delta_i^2}], 2(K^2 + MK)\} + \frac{2\pi^2}{3P(A_{\epsilon,\delta})} + K^2 + (2M-1)K)$$

where the length of the burn-in period is explicitly

$$L = \max\Bigg\{ \underbrace{\frac{\ln \frac{2T}{\epsilon}}{2\delta^2}, \frac{4K \log_2 T}{c_0}}_{L_{s_1}}, \underbrace{\frac{\ln \frac{\delta}{10}}{\ln p^*} + 25\frac{1+\lambda}{1-\lambda}\frac{\ln \frac{2T}{\epsilon}}{2\delta^2}, \frac{4K \log_2 T}{c_0}}_{L_{s_2}},$$

$$\underbrace{\frac{\ln \frac{\delta}{10}}{\ln p^*} + 25\frac{1+\lambda}{1-\lambda}\frac{\ln \frac{2T}{\epsilon}}{2\delta^2}, \frac{\frac{K \ln(\frac{MT}{\epsilon})}{\ln(\frac{1}{1-\frac{2\log M}{M-1}})}}{c_0}}_{L_{s_3}} \Bigg\}$$

with λ being the spectral gap of the Markov chain in s_2, s_3 that satisfies $1 - \lambda \geq \frac{1}{2\frac{\ln 2}{\ln 2p^*}\ln 4+1}$, $p^* = p^*(M) < 1$ and $c_0 = c_0(K, \min_{i \neq i^*} \Delta_i, M, \epsilon, \delta)$, and the instance-dependent constant $C_1 = 8\sigma^2 \max\{12\frac{M(M+2)}{M^4}\}$.

Proof Sketch. The proof is carried out in Appendix; here we describe the main ideas as follows. We note that the regret is proportional to the total number of pulling global sub-optimal arms by the end of round T. We fix client m for illustration without loss of generality. We tackle all the possible cases when clients pull such a sub-optimal arm - (i) the condition $n_{m,i}(t) \leq N_{m,i}(t) - K$ is met, (ii) the upper confidence bounds of global sub-optimal arms deviate from the true means, (iii) the upper confidence bounds of global optimal arms deviate from the true means, and (iv) the mean values of global sub-optimal arms are greater than the mean values of global optimal arms. The technical novelty of our proof is in that 1) we deduce that the total number of events (ii) and (iii) occurring can be bounded by some constants using newly derived conditional concentration inequalities that hold by our upper bounds on the conditional moment generating functions and by the unbiasedness of the network-wide estimators and 2) we control (i) by analyzing the scenarios where the criteria are met, which do not occur frequently.

There are several key challenges in the analysis. The concentration inequalities are for the neighbor-wide estimators $\tilde{\mu}_i^m$, which necessitates deducing the properties of $\tilde{\mu}_i^m(t)$ for any time step t, client m and arm i. To this end, we show that $\tilde{\mu}_i^m(t)$ are unbiased estimators of μ_i conditional on A that relies on the execution of the algorithm during the burn-in period (Algorithm 2), and more importantly, we prove that $\tilde{\mu}_i^m(t)$ have variance proxies proportional to the global variable $\frac{1}{\min_j n_{j,i}(t)}$ by bounding the conditional moment generating function conditional on A and analyzing $t_{m,j}$ and the weight matrix P'. Meanwhile, the condition $n_{m,i}(t) \leq N_{m,i}(t) - K$ is based on the difference between $n_{m,i}(t)$ and $N_{m,i}(t)$. In view of that, we consider whether the clients in the neighborhood set lead to an update in $N_{m,i}(t)$ and whether client m updates $n_{m,i}(t)$ simultaneously. All the analyses are made possible by the newly proposed update rule that aligns with the new settings. □

Remark (The condition on the time horizon). *Although the above regret bound holds for any $T > L$, the same bound applies to $T \leq L$ as follows. Assuming $T \leq L$, we obtain $E[R_T|A_{\epsilon,\delta}] \leq T \leq L$ where the first inequality is by noting that the rewards are within the range of* $[0, 1]$.

Remark (The upper bound on the expected regret). *Theorem 2 states a high probability regret bound, while the expected regret is often considered in the existing literature. As a corollary of Theorem 2, we establish the upper bound on $E[R_T]$ if $\epsilon = \frac{\log T}{MT}$ as follows. Note that*

$$E[R_T] = E[R_T|A_{\epsilon,\delta}]P(A_{\epsilon,\delta}) + E[R_T|A^c_{\epsilon,\delta}]P(A^c_{\epsilon,\delta}) \leq P(A_{\epsilon,\delta}) \cdot E[R_T|A_{\epsilon,\delta}] + T \cdot (1 - P(A_{\epsilon,\delta}))$$

$$\leq (1 - 7\epsilon)(L + \sum_{i \neq i^*}(\max\{[\frac{4C_1 \log T}{\Delta_i^2}], 2(K^2 + MK)\} + \frac{2\pi^2}{3P(A_{\epsilon,\delta})} + K^2 + (2M - 1)K)) + 7\epsilon T$$

$$\leq l_1 + l_2 \log T + \sum_{i \neq i^*}(\max\{[\frac{4C_1 \log T}{\Delta_i^2}], 2(K^2 + MK)\} + \frac{2\pi^2}{3(1 - 7\epsilon)} + K^2 + (2M - 1)K) + 7\frac{\log T}{M}$$

where the first inequality uses $E[R_T|A_{\epsilon,\delta}] \leq T$ and the second inequality follows by Theorem 1. Here l_1 and l_2 are constants depending on $K, M, \delta, \min_{i \neq i^*} \Delta_i$, and λ.

Remark (Specification of the parameters). *Note that the choice of f depends on the problem settings. Specifically, in setting s_1, we set $f(\epsilon, M, T) = \frac{1}{2} + \frac{1}{4}\sqrt{1 - (\frac{\epsilon}{MT})^{\frac{2}{M-1}}}$. By the definition of c, we have $f(\epsilon, M, T) < c$. In setting s_2 with $M < 11$, we specify $f(\epsilon, M, T) = \frac{1}{2}$ which meets $f < c$ due to (1). Lastly, in setting s_3 with $M \geq 11$, we choose $f(\epsilon, M, T) = \frac{1}{2}\frac{2\log M}{M-1}$ and again we have $f < c$ due to (1).*

Remark (Comparison with previous work). *A comparison to the regret bounds in the existing literature considering sub-gaussian rewards is as follows. Our regret bounds are consistent with the prior works where the expected regret bounds are of order* $\log T$. *Note that the regret bounds in [Zhu and Liu, 2023] cannot be used here since the update rule and the settings are different. Their update rule and analyses cannot carry over to our settings, which explains why we invent these modifications and proofs. On the one hand, the time-varying graphs they consider* *do not include the E-R model, and we can find counter-examples where their doubly stochastic weight matrices W_t result in the divergence of $W_1 \cdot W_2 \dots W_T$. This makes the key proof step invalid in our framework. On the other hand, their time-varying graphs include the connected graphs when $l = 1$, but they also make an additional assumption of doubly stochastic weight matrices, which is not applicable to regular graphs. Furthermore, they study an expected regret upper bound, while we prove a high probability regret bound that captures the dynamics in the random graphs. The graph assumptions in other works, however, are stronger, such as [Zhu et al., 3–4, 2021] consider time-invariant graphs and [Wang et al., 1531–1539, 2021] assume graphs are complete [Perchet et al., 660–681, 2016]. In contrast to some work that focuses on homogeneous rewards in decentralized multi-agent MAB, we derive regret bounds of the same order $\log T$ in a heterogeneous setting. If we take a closer look at the coefficients in terms of K, M, λ, Δ_i, our regret bound is determined by $O(\max(K, \frac{1+\lambda}{1-\lambda}, \frac{1}{M^2\Delta_i})\log T)$. The work of [Zhu and Liu, 2023] arrives at $O(\max\{\frac{\log T}{\Delta_i}, K_1, K_2\})$ where K_1, K_2 are related to T without explicit formulas. Our regret is smaller when $K\Delta_i \leq 1$ and $\frac{1+\lambda}{1-\lambda}\Delta_i \leq 1$, which can always hold by rescaling Δ_i, i.e. for many cases we get substantial improvement.*

To proceed, we show a high probability upper bound on the regret $E[R_T|A_{\epsilon,\delta}]$ of Algorithm 3 for settings S_1, S_2, S_3 with sub-exponential rewards.

Theorem 3. *Let f be a function specific to a setting and defined in the above remark. For every $0 < \epsilon < 1$ and $0 < \delta < f(\epsilon, M, T)$, in settings S_1 with $c \geq \frac{1}{2} + \frac{1}{2}\sqrt{1 - (\frac{\epsilon}{MT})^{\frac{2}{M-1}}}$,$S_2, S_3$ with the time horizon T satisfying $T \geq L$, the regret of Algorithm 3 with $F(m,i,t) = \sqrt{\frac{C_1 \ln T}{n_{m,i}(t)} + \frac{C_2 \ln T}{n_{m,i}(t)}}$ satisfies*

$$E[R_T|A_{\epsilon,\delta}] \leq L + \sum_{i \neq i^*}(\Delta_i + 1) \cdot (\max([\frac{16C_1 \log T}{\Delta_i^2}], [\frac{4C_2 \log T}{\Delta_i}], 2(K^2 + MK))$$

$$+ \frac{4}{P(A_{\epsilon,\delta})T^3} + K^2 + (2M-1)K)$$

where L, C_1 are specified as in Theorem 2 and $\frac{C_2}{C_1} \geq \frac{3}{2}$.

Proof Sketch. The proof is detailed in Appendix. The proof logic is similar to that of Theorem 2. However, the main differences lie in the upper confidence bounds, which require proving new concentration inequalities and moment generating functions for the network-wide estimators. □

In addition to the instance-dependent regret bounds of order $O(\frac{\log T}{\Delta_i})$ that depend on the sub-optimality gap Δ_i which may be arbitrarily small and thereby leading to large regret, we also establish a universal, mean-gap independent regret bound that applies to settings with sub-exponential and sub-gaussian rewards.

Theorem 4. *Assume the same conditions as in Theorems 2 and 3. The regret of Algorithm 3 satisfies that*

$$E[R_T|A_{\epsilon,\delta}] \leq L_1 + \frac{4}{P(A_{\epsilon,\delta})T^3} + (\sqrt{\max(C_1, C_2)\ln T} + 1)\frac{4M}{P(A_{\epsilon,\delta})T^3} +$$

$$K(C_2(\ln T)^2 + C_2 \ln T + \sqrt{C_1 \ln T}\sqrt{T(\ln T + 1)}) = O(\sqrt{T}\ln T).$$

where $L_1 = \max(L, K(2(K^2 + MK)))$, L, C_1 is specified as in Theorem 2, and $\frac{C_2}{C_1} \geq \frac{3}{2}$. The involved constants depend on σ^2 but not on Δ_i.

Proof Sketch. A formal proof is deferred to Appendix; the general idea is as follows. We directly decompose $E[R_T|A_{\epsilon,\delta}]$ as the sum of - (i) the differences between the confidence bounds and the true mean values, (ii) the differences between the upper and lower confidence bounds, and (iii) the differences in the upper confidence bounds between the global optimal and sub-optimal arms. The first term relies on the concentration inequalities for the network-wide estimators. The second term is proportional to the cumulative sum of $\sqrt{\frac{C_1 \ln T}{n_{m,i}(t)}} + \frac{C_2 \ln T}{n_{m,i}(t)}$ which has an upper bound $O(\sqrt{T}\log T)$ by the Cauchy-Schwarz inequality. The last term is based on the number of time steps when the clients do not follow UCB, which is relevant to the criterion $n_{m,i}(t) \leq N_{m,i}(t) - K$. □

Remark. *Based on the expression of L_1, we obtain that L_1 is independent of the sub-optimality gap Δ_i. Meanwhile, we have $C_1 = 8\sigma^2 \cdot 12\frac{M(M+2)}{M^4}$ and $C_2 = \frac{3}{2}C_1 = 12\sigma^2 \cdot 12\frac{M(M+2)}{M^4}$. This implies that the established regret bound in Theorem 4 does not rely on Δ_i but does depend on σ^2. To this end, we use the terminology, mean-gap independent bounds, to only represent bounds having no dependency on Δ_i, rather than instance independent that seems to be an overclaim in this case.*

Remark. *The discussion regarding the conditions on T, the expected regret $E[R_T]$, and the parameter specifications follow the same logic as those in Theorem 2. We omit the details here.*

Remark (Comparison with previous work). *For decentralized multi-agent MAB with homogeneous heavy-tailed rewards and time-invariant graphs, [Dubey and Pentland, 2730–2739, 2020] provide an instance-dependent regret bound of order $\log T$. In contrast, our regret bound has the same order for heterogeneous settings with random graphs, as shown in Theorem 3. Additionally, we provide a mean-gap independent regret bound as in Theorem 4. In the single-agent MAB setting, [Jia et al., 2021] consider sub-exponential rewards and derive a mean-gap independent regret upper bound of order $\sqrt{T \log T}$. Our regret bound of $\sqrt{T} \log T$ is consistent with theirs, up to a logarithmic factor. Furthermore, our result is consistent with the regret lower bound as proposed in [Slivkins, 2019], up to a $\log T$ factor, indicating the tightness of our regret bound.*

References

M. Agarwal, V. Aggarwal, and K. Azizzadenesheli. Multi-agent multi-armed bandits with limited communication. *The Journal of Machine Learning Research*, 23(1):9529–9552, 2022.

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

P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002b.

I. Bistritz and A. Leshem. Distributed multi-player bandits-a game of thrones approach. *Advances in Neural Information Processing Systems*, 31, 2018.

F. Delarue. Mean field games: A toy model on an Erdös-Renyi graph. *ESAIM: Proceedings and Surveys*, 60:1–26, 2017.

A. Dubey and A. Pentland. Cooperative multi-agent bandits with heavy tails. In *International Conference on Machine Learning*, 2730–2739, 2020.

C. Gray, L. Mitchell, and M. Roughan. Generating connected random graphs. *Journal of Complex Networks*, 7(6):896–912, 2019.

R. Huang, W. Wu, J. Yang, and C. Shen. Federated linear contextual bandits. *Advances in Neural Information Processing Systems*, 34:27057–27068, 2021.

H. Jia, C. Shi, and S. Shen. Multi-armed bandit with sub-exponential rewards. *Operations Research Letters*, 49(5):728–733, 2021.

F. Jiang and H. Cheng. Multi-agent bandit with agent-dependent expected rewards. *Swarm Intelligence*, 1–33, 2023.

P. Landgren, V. Srivastava, and N. E. Leonard. On distributed cooperative decision-making in multiarmed bandits. In *2016 European Control Conference*. 243–248. IEEE, 2016a.

P. Landgren, V. Srivastava, and N. E. Leonard. Distributed cooperative decision-making in multiarmed bandits: Frequentist and Bayesian algorithms. In *2016 IEEE 55th Conference on Decision and Control*. 167–172. IEEE, 2016b.

P. Landgren, V. Srivastava, and N. E. Leonard. Distributed cooperative decision making in multiagent multi-armed bandits. *Automatica*, 125:109445, 2021.

F. W. Lima, A. O. Sousa, and M. Sumuor. Majority-vote on directed Erdős–Rényi random graphs. *Physica A: Statistical Mechanics and its Applications*, 387(14):3503–3510, 2008.

D. Martínez-Rubio, V. Kanade, and P. Rebeschini. Decentralized cooperative stochastic bandits. *Advances in Neural Information Processing Systems*, 32, 2019.

B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, 1273–1282, 2017.

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

V. Perchet, P. Rigollet, S. Chassang, and E. Snowberg. Batched bandit problems. *The Annals of Statistics*, 660–681, 2016.

C. Réda, S. Vakili, and E. Kaufmann. Near-optimal collaborative learning in bandits. In *2022-36th Conference on Neural Information Processing System*, 2022.

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

Y. Tao, Y. Wu, P. Zhao, and D. Wang. Optimal rates of (locally) differentially private heavy-tailed multi-armed bandits. In *International Conference on Artificial Intelligence and Statistics*, 1546–1574, 2022.

L. Trevisan. CS 174 randomized algorithms. URL `http://theory.stanford.edu/~trevisan/cs174/`.

Z. Wang, C. Zhang, M. K. Singh, L. Riek, and K. Chaudhuri. Multitask bandit learning through heterogeneous feedback aggregation. In *International Conference on Artificial Intelligence and Statistics*, 1531–1539, 2021.

Z. Yan, Q. Xiao, T. Chen, and A. Tajer. Federated multi-armed bandit via uncoordinated exploration. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. 5248–5252. IEEE, 2022.

J. Zhu and J. Liu. Distributed multi-armed bandits. *IEEE Transactions on Automatic Control*, 2023.

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

J. Zhu, E. Mulle, C. S. Smith, and J. Liu. Decentralized multi-armed bandit can outperform classic upper confidence bound. *arXiv preprint arXiv:2111.10933*, 2021.

Z. Zhu, J. Zhu, J. Liu, and Y. Liu. Federated bandit: A gossiping approach. In *Abstract Proceedings of the 2021 ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*, 3–4, 2021.