Cooperative Multi-agent Bandits: Distributed Algorithms with Optimal Individual Regret and Communication Costs

Anonymous authors
Paper under double-blind review

Abstract

Recently, there has been extensive study of cooperative multi-agent multi-armed bandits where a set of distributed agents cooperatively play the same multi-armed bandit game. The goal is to develop bandit algorithms with the optimal group and individual regrets and low communication between agents. Prior algorithms either cannot achieve constant communication costs or fail to achieve optimal individual regrets. This paper presents a simple yet effective communication policy and integrates it into a learning algorithm for cooperative bandits. Our algorithm achieves the best of both paradigms: optimal individual regret and constant communication costs. We also provide a tight communication lower bound that matches the constant communication upper bound of our algorithms in all terms, suggesting the optimality of our algorithm design and analysis.

1 Introduction

Recently, there has been a surge of various online learning problems in distributed settings, where a set of agents perform individual learning algorithms to complete a common task and can cooperate with each other to improve the performance of the learning process. Distributed online learning is naturally motivated by a broad range of applications where computational resources are geographically distributed, and a group of machines has to communicate with each other to complete a common task cooperatively. Examples include inference engines in a software-defined network, servers in a data center, and drones in a swarm. In distributed online learning settings, agents take actions over time and receive sequential samples associated with the selected actions. While the agents can cooperate to speed up the learning process, it comes at the expense of communication overhead in sharing sequential samples with others. Hence, distributed online learning problems involve a natural trade-off between learning performance and communication overheads.

Table 1: A comparison summary of prior literature and this work.

Algorithm	**Group regret**	**Individual regret**	**Communication cost**
`DPE2` (leader-follower) (Wang et al., 2020a)	$O(\sum_k \Delta_k^{-1} \log T)$	$O(\sum_k \Delta_k^{-1} \log T)$	$O(K^2M^2\Delta_{\min}^{-2})$
`GosInE` (Chawla et al., 2020)	$O((\sum_k \Delta_k^{-1} + 2M) \log T)$	$O(((\sum_k \Delta_k^{-1}/M) + 2) \log T)$	$O(\log T)$
`ComEx` (Madhushani & Leonard, 2021)	$O(\sum_k \Delta_k^{-1} \log T)$	$O(\sum_k \Delta_k^{-1} \log T)$	$O(KM \log T)$
`Dec_UCB` (Zhu et al., 2021)	$O(\sum_k \Delta_k^{-1} \log T)$	$O((\sum_k \Delta_k^{-1}/M) \log T)$	$O(MT)$
`UCB-TCOM` (Wang et al., 2023)	$O(\sum_k \Delta_k^{-1} \log T)$	$O((\sum_k \Delta_k^{-1}/M) \log T)$	$O(KM \log \log T)$
`BatchedMAB` (Karpov & Zhang, 2024)	$O(\sum_k \Delta_k^{-1} \log T)$	$O((\sum_k \Delta_k^{-1}/M) \log T)$	$O(KM \log \Delta_{\min}^{-1})$
`DoE-bandit`	$O(\sum_k \Delta_k^{-1} \log T)$	$O((\sum_k \Delta_k^{-1}/M) \log T)$	$O(M \sum_k \log \Delta_k^{-1})$
Communication Lower Bound	$O(\sum_k \Delta_k^{-1} \log T)$	$O((\sum_k \Delta_k^{-1}/M) \log T)$	$\Omega(\max\{\sum_k \log \Delta_k^{-1}, M\})$

This paper focuses on studying Cooperative Multi-Agent Multi-Armed Bandit (`CMA2B`) problems where multiple agents tackle the same instance of a bandit problem. In the standard setting of `CMA2B`, a set of M independent agents existing over the entire time horizon pull an arm at each time from a common set of K arms. Associated with arms are mutually independent sequences of i.i.d.

$[0, 1]$-valued rewards with mean $0 \leq \mu(k) \leq 1$, for arm $k \in \mathcal{K}$. Each agent has full access to the set of arms: agents are allowed to pull and receive a reward from any arm without any reward degradation when pulling the same arm. The goal of each agent is to learn the best arm, with performance characterized by group regret and maximum individual regret according to different application scenarios. In addition to regret, another important metric is the communication overheads that the agents spend in cooperative learning.

The above `CMA2B` problem is a natural extension of the basic MAB problem (Auer et al., 2002; Bubeck, 2010) in a cooperative multi-agent setting, with extensive recent literature, to name a few (Szorenyi et al., 2013; Landgren et al., 2016; Chakraborty et al., 2017; Kolla et al., 2018; Martínez-Rubio et al., 2019; Féraud et al., 2019; Wang et al., 2020a; Bistritz & Bambos, 2020; Chawla et al., 2020; Wang et al., 2020b; Madhushani & Leonard, 2021; Zhu et al., 2021; Yang et al., 2021; Chen et al., 2023). In terms of solution design, the prior work could be categorized into two paradigms of leader-follower, where a leader agent coordinates the learning process, and fully distributed algorithms, where there is no central coordinator.

In the leader-follower paradigm Shi et al. (2021a); Mehrabian et al. (2020); Shi et al. (2021b); Shi & Shen (2021); Wang et al. (2019; 2020a); Bar-On & Mansour (2019); Chakraborty et al. (2017); Dubey et al. (2020), a leader agent coordinates the learning process among all agents. The state-of-the-art result in this paradigm is the `DPE2` algorithm proposed in (Wang et al., 2020a) and achieves the optimal group regret with a constant number of communication overheads[1], Yet, `DPE2` (and all other leader-follower-based algorithms) relies on a structure where the leader solely pays the exploration costs and incurs almost all the regret in the system. Hence, by nature, this paradigm fails to achieve a good individual regret since all the regret is imposed on the leader agent. It is worth noting that in many practical applications, agents' individual regrets are crucial for a system's overall performance. For example, in a drone swarm, the failure/misbehavior of a single drone, e.g., it crashes into other drones, can dramatically degrade the whole system's overall performance; or in network measurement, the slowest inference engine determines how fast the network parameters, e.g., traffic flows and channel bandwidths, are learned.

An alternative approach is to remove the leader as the central coordinator and design fully distributed cooperative algorithms. While there has been a success in achieving the optimal group and individual regrets for fully distributed algorithms, they still fail to achieve low communication overhead, such as those in the leader-follower-based algorithms. Early works in this space, e.g., (Buccapatnam et al., 2015; Yang et al., 2021; 2022) adopted immediate broadcasting as their communication scheme, incurring a high communication cost of $O(T)$. More recent works (Martínez-Rubio et al., 2019; Wang et al., 2019; Chawla et al., 2020), improved the communication overhead of the cooperative algorithms to $O(\log T)$ by optimizing the use of communication budget. The state-of-the-art in this line of work is the `UCB-TCOM` algorithm (Wang et al., 2023) that achieves the optimal individual regret of $O(K/M \log T)$ with communication cost of $O(KM \log \log T)$. Despite the above efforts, prior to this work, no existing algorithm, either based on leader-follower or fully distributed, achieves optimal group and individual regret with constant communication costs.

Besides the literature on distributed multi-agent bandits, there is a line of works on batched bandits (Perchet et al., 2016; Gao et al., 2019; Esfandiari et al., 2021; Jin et al., 2021; Karpov & Zhang, 2024) that relate to `CMA2B`. In batched bandits, the time horizon is separated into several batches, and the reward observations of pulling arms during each batch are only revealed at the end of the batch. This scheme is similar to the distributed bandits, where the observations of other agents after the last communication are only revealed at these agents' next communication. Therefore, the batched bandits algorithm can adapt to our multi-agent bandits setting. The current state-of-the-art batched algorithm, `BatchedMAB` (Karpov & Zhang, 2024), requires $O(K \log \Delta_{\min}^{-1})$ batches to attain the near-optimal problem-dependent regret bound. That is, transferring their algorithms to the distributed setting leads to $O(KM \log \Delta_{\min}^{-1})$ communication costs. In contrast, our work shows a smaller constant communication cost $O(M \sum_{k:\Delta_k>0} \log \Delta_k^{-1})$ is enough to guarantee the

[1]Constant communication cost in this paper means it is independent of time horizon T.

optimal individual and group regrets, and we prove a communication lower bound showing that this communication cost is tight in terms of all factors.

Contributions. This paper presents `DoE-bandit`, the first fully distributed algorithm that guarantees the optimal group and maximum individual regrets with optimal communication costs (see Theorem 3). Specifically, `DoE-bandit` achieves an $O(\sum_{k:\Delta_k>0} \Delta_k^{-1} \log T)$ group regret and an $O(\sum_{k:\Delta_k>0}(\Delta_k^{-1}/M) \log T)$ maximum individual regret, where Δ_k is the gap of reward means between the optimal arm k^* and arm k. Further, `DoE-bandit` achieves the constant communication cost of $O(M \sum_k \log \Delta_k^{-1})$. We also propose a novel communication lower bound $\Omega(\max\{\sum_k \log \Delta_k^{-1}, M\})$ for any `MA2B` algorithm that achieves near-optimal group and individual regrets (see Theorem 1). This lower bound shows that the cost communication cost of `DoE-bandit` is tight in terms of all factors. A summary of our results and the most relevant prior work is given in Table 1.

2 Problem Description

We introduce a basic multi-agent multi-armed bandit system model. We note that the communication policy developed in this paper is generic and could be applied to a broad range of cooperative online learning settings.

Consider a multi-agent stochastic bandit setting with a set $\mathcal{M} \coloneqq \{1, \ldots, M\}$ of independent agents existing over the entire time period, and a set $\mathcal{K} \coloneqq \{1, 2, \ldots, K\}$ of arms. Associated with arms are mutually independent sequences of i.i.d. $[0, 1]$-valued (e.g., Bernoulli) rewards with mean $0 \leq \mu(k) \leq 1$, for arm $k \in \mathcal{K}$. Agent $m \in \mathcal{M}$ has full access to the set of arms. Agents are allowed to pull and receive a reward from any arm k from $\mathcal{K}$. For ease of presentation, we focus on a basic model formulation where agents reside on a complete graph, incur no communication delays, and the communication is lossless. However, the basic model and communication policy proposed in this paper could be extended to account for these practical additions.

In bandit learning, the goal of each agent m is to learn the best arm as fast as possible with minimizing the *pseudo-regret* in $T \in \mathbb{N}^+$ decision rounds (called *regret* for short in the rest of this paper). The expected regret of an agent m is formally defined as $\mathbb{E}[R_T^{(m)}] \coloneqq \mu(k^*)T - \mathbb{E}[\sum_{t=1}^T x_t(I_t^{(m)})]$, where k^* is the optimal arm, $I_t^{(m)}$ is the action taken by agent m at round t, and $x_t(I_t^{(m)})$ is the realized reward. Also, the expectation is taken over the randomness of stochastic rewards and the algorithms. In a multi-agent setting, the total performance is measured by the total expected regret of all agents, defined as

$$\mathbb{E}\left[R_T\right] \coloneqq \sum_{m \in \mathcal{M}} \mathbb{E}\left[R_T^{(m)}\right].$$

In addition to the group regret, which characterizes overall performance, the individual performance of each agent is also important. To capture this individual performance, we measure the maximum individual regret defined as follows,

$$\mathbb{E}\left[\bar{R}_T\right] \coloneqq \mathbb{E}\left[\max_{m \in \mathcal{M}} R_T^{(m)}\right].$$

Similar to other distributed learning problems, the `MA2B` setting encourages distributed agents to cooperate with each other by sharing information through *messages*, which include reward observations, reward averages, or arm indices. We assume any message can be communicated within a single time slot. The total number of messages communicated among these agents quantifies the communication cost of an algorithm. We denote the expected total communication cost in T rounds among M agents as follows,

$$\mathbb{E}[C_T] = \sum_{t=1}^{T} \sum_{m \in \mathcal{M}} \sum_{k \in \mathcal{K}} \mathbb{E}[c_t^{(m)}(k)],$$

where $c_t^{(m)}(k) \coloneqq \mathbb{1}\{\text{agent } m \text{ communicates about arm } k \text{ at time slot } t\}$. The communication cost definition assumes that each message only contains the information of one arm, and if the agents want to share information of multiple arms, they need multiple separated messages. We choose this dentition in order to show the tightness of our communication cost analysis in arm level, and our algorithm design and theoretical analysis can be adapted to the case that one message aggregates multiple arms' information.

3 Algorithm

This section presents an algorithm that adds a Distributed Online Estimation (`DoE`) subroutine to each learning agent m and enables them to approximate the estimate of the optimal centralized algorithm having all samples when estimating the parameter of a common i.i.d. process. We introduce the details of the `DoE` algorithms in Section 3.1 and then integrate it to a bandit algorithm in Section 3.2.

3.1 Distributed Online Estimation Algorithm (`DoE`)

To facilitate the presentation of the high-level idea of `DoE`, let us focus on a simplified setting that involves only one arm k whose reward mean $\mu(k)$ is unknown to the distributed agents, where agents sample the process simultaneously in each slot. Since each agent possesses the same number of pulls, we denote $n_t(k)$ as the number of samples available to each agent up to time t. The idea of `DoE` is to synchronize the estimates of distributed agents when the local estimates deviate substantially from the centralized one with all samples. By properly configuring `DoE`, each individual agent needs to efficiently control the deviation of its local estimates with incurring low communication costs.

More specifically, during the running time, `DoE` adopts a threshold policy to decide whether to trigger a communication round for agents to synchronize their estimates with all samples in the system. To decide whether to start a communication round, each agent maintains the so-called *Common Mean* (CM) for the mean over all system-wide available samples in the last communication round, and simply compare CM with *Auxiliary Local Estimates* (ALE, details shown in (1)). The value of CM, denoted as $\hat{\mu}_{\texttt{com},t}(k)$, is calculated by averaging all samples up to the last communication round, so, its value is updated only once at each communication round and remains unchanged in the subsequent non-communication rounds. At specific time slots, each agent checks whether the gap between CM and ALE is smaller than some threshold value.

In `DoE`, all agents share a common threshold value denoted as $\texttt{ECR}_t(k)$, which can be time-varying with the number of available samples $n_t(k)$. If the gap between ALE and CM is larger than the threshold value, a new communication round is triggered to synchronize the estimates. By doing so, the sum of new samples from other agents will be collected, a new common mean is calculated, and then the agent broadcasts the new CM to all others.

The threshold value $\texttt{ECR}_t(k)$ plays a key role in controlling estimate deviations and communication overheads. Intuitively, when the ALEs of each individual agent center around the common mean, the actual estimates of all agents center around CM as well. Thus, no communication is needed. Otherwise, a communication round is triggered to synchronize the estimates of all agents. Hence, the threshold value determines how far the estimates deviate from each other during the non-communication rounds; the smaller the threshold value, the smaller the deviations, and the closer the local estimates of agents approach the global mean over all samples. On the other hand, with smaller threshold values $\texttt{ECR}_t(k)$, agents communicate more frequently with each other. Hence, the trade-off of estimation performance versus communication overheads is associated with $\texttt{ECR}_t(k)$.

Next, we present the technical details of the `DoE` algorithm and show how to construct the estimate interval for each agent by using local estimates.

Algorithm 1 DoE: an algorithm for estimating the mean of arm k by agent m, subscript t is dropped

1: **Parameters:** $\beta > 1$;
2: **Variables:** $\hat{\mu}^{(m)}_{\texttt{aux}}(k)$, $n(k) \leftarrow 0$ $\hat{\mu}_{\texttt{com}}(k) \leftarrow 0$, $\texttt{ECR}_{\texttt{last}} \leftarrow 0$; $X^{(m')}(k) \leftarrow 0$, $X^{(m')}_{\texttt{last}}(k) \leftarrow \infty$, $\forall m' \in \mathcal{M}, \texttt{ECR}_t(k) \leftarrow 0$
3: **for** each round t when the agent gets a new sample **do**
4: $n(k) \leftarrow n(k) + 1$
5: Update $X^{(m)}(k)$ with the new sample
6: **if** $\beta\texttt{ECR}(k) \leq \texttt{ECR}_{\texttt{last}}$ **then**
7: $\texttt{ECR}_{\texttt{last}} \leftarrow \texttt{ECR}(k)$
8: **if** $|\hat{\mu}^{(m)}_{\texttt{aux}}(k) - \hat{\mu}_{\texttt{com}}(k)| > \texttt{ECR}(k)$ **then**
9: //Communicate to synchronize the estimates
10: Collect $X^{(m')}(k)$ from other agents and calculate the new $\hat{\mu}_{\texttt{com}}(k)$
11: Broadcast the new $\hat{\mu}_{\texttt{com}}(k)$ to other agents
12: $X^{(m')}_{\texttt{last}}(k) \leftarrow X^{(m')}(k)$ for all $m' \in \mathcal{M}$
13: Update $\hat{\mu}^{(m)}_{\texttt{aux}}(k)$ according to (1) and $\hat{\mu}^{(m)}(k)$ according to (2)

Constructing the Auxiliary Local Estimates (ALE). At a non-communication round t, an agent only accesses partial external samples from others. Below we introduce how an agent builds up the Auxiliary Local Estimate with missing samples from others.

Note that $n_t(k)$ is the number of samples that an agent has made for arm k up to time slot t. Let $t_{\texttt{last}}$ denote the last round before t that the Condition in Line 6 holds, and $X^{(m)}_t(k)$ be the sum of rewards from $n_t(k)$ samples of agent m at time slot t for arm k. For agent m, there are $n_t(k) - n_{t_{\texttt{last}}}(k)$ missing samples from any other agents. In DoE, agent m uses local samples in the same time slot to compensate the missing samples from other agents to construct ALE, denoted by $\hat{\mu}^{(m)}_{\texttt{aux},t}(k)$. That is

$$\hat{\mu}^{(m)}_{\texttt{aux},t}(k) = \frac{\sum_{m'=1}^{M}(X^{(m')}_{t_{\texttt{last}}}(k) + X^{(m)}_t(k) - X^{(m)}_{t_{\texttt{last}}}(k))}{Mn_t(k)} \tag{1}$$

where the term $X^{(m)}_t(k) - X^{(m)}_{t_{\texttt{last}}}(k)$ serves as the compensation for the missing samples from other agents $m' \neq m$ from $t_{\texttt{last}}$ to t. In DoE, ALE mimics the estimate of the estimator, which possesses all $Mn_t(k)$ samples and serves as an index through which the agents decide when to communicate.

We weight the local estimates in ALE such that it may involve a larger estimation error. Hence, in addition to ALE, each agent m calculates the local estimate $\hat{\mu}^{(m)}_t(k)$ to be used in a bandit algorithm using the following equation.

$$\hat{\mu}^{(m)}_t(k) = \frac{(\sum_{m'=1}^{M} X^{(m')}_{t_{\texttt{last}}}(k)) + X^{(m)}_t(k) - X^{(m)}_{t_{\texttt{last}}}(k)}{Mn_{t_{\texttt{last}}}(k) + n_t(k) - n_{t_{\texttt{last}}}(k)} \tag{2}$$

Communication Policy of DoE. Now with the definition of ALE, we present the communication policy of DoE. The pseudocode of DoE is summarized in Algorithm 1. To decide a communication round, an agent m checks the values of $\hat{\mu}^{(m)}_{\texttt{aux},t}(k)$ and $\hat{\mu}_{\texttt{com},t}(k) := (\sum_{m=1}^{M} X^{(m)}_{t_{\texttt{last}}}(k))/(Mn_{t_{\texttt{last}}}(k))$ every time the specified threshold value $\texttt{ECR}(k)$ reduces to $1/\beta$ ($\beta > 1$) times of the original value $\texttt{ECR}_{\texttt{last}}$ (Lines 6, 7). In DoE, β determines how frequently the algorithm checks those values. Once the deviation of the local estimate $\hat{\mu}^{(m)}_{\texttt{aux},t}(k)$ from the common mean $\hat{\mu}_{\texttt{com},t}(k)$ is larger than $\texttt{ECR}(k)$ (Line 8), agent m calls for triggering of a new communication round. In a communication round triggered by agent m, the sum of missing samples from the last communication round $t_{\texttt{last}}$ from each other agent will be collected to calculate a new common mean. Then, this new common mean will be broadcast to all other agents.

Our analysis in Lemma 2 shows that DoE can provide a provable performance guarantee for the single-arm-estimation problem (in the form of confidence interval) with a tunable trade-off between the estimation quality and communication overheads. With a richer communication budget, the estimation performance of DoE approaches that of the optimal estimator with full access to the samples. Since DoE can provide an explicit confidence interval for the mean to be estimated, it is straightforward to plug DoE into bandit algorithms, as exemplified in the next section.

Algorithm 2 `DoE-bandit` for agent m; subscript t is dropped

1: **Parameters:** $\alpha > 0$, $\beta > 1$; $\texttt{ECR}_n$, $n = 1, 2, \ldots$
2: **Initialization:** $\hat{\mu}^{(m)}_{\texttt{com}}(k) \leftarrow 0$; $\texttt{ECR}_{\texttt{last}}(k)$; $n(k) \leftarrow 0$, $\hat{\mu}^{(m)}_{\texttt{aux}}(k)$, for $\forall i$; $\texttt{ECR}_n \leftarrow \alpha\texttt{CR}(Mn, \delta_t)$, $n = 1, 2, \ldots$
3: **for** each round t **do**
4: **if** an arm is eliminated by some other agent **then**
5: Update the candidate set
6: Pull arm k from the candidate set in round-robin manner
7: Execute Lines 4-12 of `DoE` (Algorithm 1) for communication on arm k
8: Update the candidate set via (4)
9: Notify other agents if an arm is eliminated

3.2 Integrating `DoE` to a Bandit Learning Algorithm

In this section, we present a distributed bandit algorithm named `DoE-bandit` that uses `DoE` as the underlying communication policy. We summarize the pseudocode of `DoE-bandit` in Algorithm 2.

`DoE-bandit` is based on active arm elimination, which is a classic approach to address the well-known tradeoff between exploration (acquiring new information) and exploitation (optimizing based on available information) in bandit problems. In this approach, the learner constructs a *candidate set* for the arms, which are likely to be optimal, and exploration is allowed only from the arms in the candidate set. When exploring the candidate set, the algorithm periodically pulls an arm in and dynamically eliminates the arms which are unlikely to be optimal.

To integrate `DoE` with the bandit algorithm, we initiate multiple instances of `DoE` run by `DoE-bandit`, each of which tackles the estimation of a single arm. To implement the `DoE` subroutine, each agent notifies others once an arm is eliminated (Line 9 in Algorithm 2) and pulls arms in the candidate set in a round-robin manner (Line 6), so that all agents always pull the same arm at each time slot and `DoE` is able to keep track of the total number of samples in the system by $Mn_t(k)$. The above rules imply that all agents have a common candidate set, which is denoted by $\mathcal{C}_t$.

Constructing the candidate set. To construct the candidate set, `DoE-bandit` determines an explicit confidence interval for the reward means of arms. Define $\texttt{CR}(n, \delta_t)$ as the radius of the confidence interval for the reward process with n samples and confidence level $1 - \delta_t$. If the reward process is $[0, 1]$-valued, we define

$$\texttt{CR}(n, \delta_t) = \sqrt{\frac{\log \delta_t^{-1}}{2n}}, \tag{3}$$

where δ_t specifies the violation probability that the true mean lies outside the above confidence interval. As we mentioned, the threshold value, $\texttt{ECR}_t(k)$, in `DoE` determines the deviation of the estimates in individual agents from the optimal one with all samples. Hence, in order to guarantee distributed agents to achieve the same order of the convergence rate as the optimal one, we set $\texttt{ECR}_t(k)$ according to the confidence interval with the total of $Mn_t(k)$ samples. By setting $\texttt{ECR}_t(k) = \alpha\texttt{CR}(Mn_t(k), \delta_t)$ where $\alpha > 0$, `DoE` yields a confidence interval for the mean of arm k, whose radius is $(2\alpha\beta + \beta)\texttt{CR}(Mn_t(k), \delta_t)$ (see Lemma 2 on detailed derivation). With the above result, an arm k is eliminated by agent m from the candidate set $\mathcal{C}_t$ at time t if there exist an arm $k' \in \mathcal{C}_t$ such that $\hat{\mu}^{(m)}_t(k) + (2\alpha + \beta)\texttt{CR}(Mn_t(k), \delta_t) < \hat{\mu}^{(m)}_t(k') - (2\alpha + \beta)\texttt{CR}(Mn_t(k'), \delta_t)$. That is, the candidate arm set $\mathcal{C}$ is updated as follows,

$$\mathcal{C} \leftarrow \left\{ k \in \mathcal{C} : \max_{k' \in \mathcal{C}} \hat{\mu}^{(m)}_t(k') < \hat{\mu}^{(m)}_t(k) + 2(2\alpha + \beta)\texttt{CR}(Mn_t(k), \delta_t) \right\} \tag{4}$$

4 Regret and Communication Cost Analysis

In this section, we summarize the theoretical results. We start with a novel communication lower bound for the `MA2B` model in Section 4.1. Then, we turn to analyze the `DoE-bandit` algorithm in Section 4.2. With both results, we show that `DoE-bandit` attains near-optimal guarantees in both group and individual regrets and communication costs in Section 4.3.

4.1 Communication Lower Bound

In this section, we present a communication lower bound for the MA2B model. Our focus is on investigating the necessary number of communications (lower bound) for any MA2B algorithm attaining near-optimal group and individual regrets. The result is provided in Theorem 1 as follows:

Theorem 1. *For any algorithm that achieves the near-optimal group regret $O(\sum_k \Delta_k^{-1} \log T)$ and individual regret $O((\sum_k \Delta_k^{-1}/M) \log T)$ and for any* MA2B *instance, the algorithm spends a number of communications that is lower bounded as follows,*

$$\mathbb{E}[C_T] \geq \Omega \left(\max \left\{ \sum_{k:\Delta_k>0} \log \Delta_k^{-1}, M \right\} \right).$$

4.2 Regret and Communication Cost Upper Bound

The following lemma shows the upper bound of estimation error of DoE is proportional to the radius of the confidence interval with system-wide samples. Then, we summarize the results for DoE-bandit in Theorem 3.

Lemma 2. *Assume M agents independently sample an arm with an i.i.d. reward process with unknown mean $\mu(k)$, and $n_t(k)$ is the available samples for each agent up to time slot t. With $\beta > 1$ and $\texttt{ECR}_t(k) = \alpha \texttt{CR}(Mn_t(k), \delta_t)$, where $\delta_t \in (0,1)$ is a sequence of parameters non-increasing with respect to t, then, for any t, with probability $1 - Mt\delta_t^{1/2}$, we have $|\hat{\mu}_t^{(m)}(k) - \mu(k)| \leq (2\alpha\beta + \beta)\texttt{CR}(Mn_t(k), \delta_t)$.*

Theorem 3. *Let $\texttt{CR}_{[0,1]}(n, \delta_t)$ in (3) with $1 \geq \delta_t > 0$ be the radius of the confidence interval of a $[0,1]$-valued i.i.d. process with n samples. Set $\beta > 1$ and $\texttt{ECR}_t(k) = \alpha \min\{1, \texttt{CR}_{[0,1]}(Mn_t(k), \delta_t)\}$, where $\alpha > 0$.* DoE-bandit *achieves the following performance:*

$$\text{(Group Regret)} \qquad \mathbb{E}[R_T] \leq MK + \sum_{k:\Delta_k>0} \frac{8(2\alpha\beta+\beta)^2 \log \delta_T^{-1}}{\Delta_k} + KM^3T \sum_{t \leq T} t\delta_t^{1/2},$$

$$\text{(Maximum Individual Regret)} \qquad \mathbb{E}\left[\bar{R}_T\right] \leq K + \sum_{k:\Delta_k>0} \frac{8(2\alpha\beta+\beta)^2 \log \delta_T^{-1}}{M\Delta_k} + KM^2T \sum_{t \leq T} t\delta_t^{1/2},$$

$$\text{(Communication Cost)} \qquad \mathbb{E}[C_T] \leq MK + \sum_{k:\Delta_k>0} 6M \log_\beta \left(\frac{4(2\alpha+1)}{\alpha\Delta_k} \right) + KM^3T \sum_{t \leq T} t\delta_t^{1/2}. \quad (5)$$

4.3 Discussion

We first present a special case of Theorem 3 and then discuss the significance of our results.

Corollary 4. *With the same parameters as Theorem 3 and setting $\delta_t \leftarrow 1/(K^2M^6T^2t^6)$,* DoE-bandit *achieves the following performance:*

$$\text{(Group Regret)} \qquad \mathbb{E}[R_T] \leq O\left(\sum_{k:\Delta_k>0} \frac{8(2\alpha\beta+\beta)^2 \log T}{\Delta_k} \right), \quad (6)$$

$$\text{(Maximum Individual Regret)} \qquad \mathbb{E}\left[\bar{R}_T\right] \leq O\left(\sum_{k:\Delta_k>0} \frac{8(2\alpha\beta+\beta)^2 \log T}{M\Delta_k} \right), \quad (7)$$

$$\text{(Communication Cost)} \qquad \mathbb{E}[C_T] \leq O\left(\sum_{k:\Delta_k>0} M \log_\beta \left(\frac{2\alpha+1}{\alpha\Delta_k} \right) \right). \quad (8)$$

Optimality in all three metrics. Corollary 4's (6) and (7) show that we can recover a $O(\sum_{k:\Delta_k>0}(1/\Delta_k)\log T)$ group regret and $O(\sum_{k:\Delta_k>0}(1/\Delta_k)\log T/M)$ individual regret for the distributed bandit problem, implying that the proposed algorithm attains both the (order-) optimal group and maximum individual regrets (Wang et al., 2023). In the meantime, compared with the communication lower bound $\Omega(\max\left\{\sum_{k:\Delta_k>0}\log\Delta_k^{-1}, M\right\})$ in Theorem 1, the communication upper bound of `DoE-bandit` in (8), i.e., $O(M\sum_{k:\Delta_k>0}\log\Delta_k^{-1})$, is optimal in terms of both the agent number M and the summation $\sum_{k:\Delta_k>0}\log\Delta_k^{-1}$.

Influence of α and β. Corollary 4's (8) shows that communication overheads influence the estimation quality through parameters α and β. Generally speaking, β specifies the frequency that `DoE` checks the deviation of individual estimates, directly upper bounding the communication overheads for `DoE-bandit`. Hence, β seems to have a larger influence in the communication overheads bound than α. On the other hand, α specifies the radius of the estimate interval `CR` as well as the threshold for the estimate deviation `ECR`, which triggers an actual communication demand.

Results for other i.i.d. processes. In `DoE-bandit`, the communication overheads on a suboptimal arm k is approximately $O(\log_\beta(\texttt{ECR}_1/\texttt{ECR}_T))$.The threshold value $\texttt{ECR}_t$ is set based on that of the confidence interval with all samples (up to a tunable parameter α). For a Bernoulli process, the mean always lies in $[0,1]$. Hence, we can set $\texttt{ECR}_1 = 1$, which results in $O(\log_\beta(1/\texttt{ECR}_T))$ communication overheads. By slight modification, the `DoE-bandit` algorithm can tackle other i.i.d. processes with similar results obtained. For an i.i.d. process with an unbounded mean, such as the Gaussian process, the `DoE-bandit` may choose to start a communication round only when the size of the confidence interval shrinks to $O(\sqrt{M})$. This will not degrade the regret results guaranteed in Theorem 3, since the algorithm only has to spend on average $O(\log T)$ samples in shrinking the confidence intervals of all arms, with an increase of $O(K\log T)$ regret. On the other hand, the communication overheads is only $O(\log(\sqrt{M}/\delta_t))$, since $\texttt{ECR}_1$ can be set to $O(\sqrt{M})$.

5 Conclusions

This paper presented `DoE-bandit`, a fully distributed algorithm for a cooperative multi-agent multi-armed bandits problem. `DoE-bandit` achieves the optimal group and individual regret with constant communication overhead. We also proposed a new communication lower bound that matches the constant communication overhead. This implied that our algorithm is near-optimal in all three metrics. The theoretical claims are verified by numerical experiments and show that `DoE-bandit` outperforms prior algorithms. We also conducted numerical simulations to show that `DoE-bandit` has the best communication cost among all algorithms in Appendix A.

The core communication policy proposed in this paper could be further extended in multiple directions. To address the exploitation-exploration dilemma in bandit learning, `DoE-bandit` adopts an elimination-based strategy, but there also exists other strategy, such as upper confidence bound and Thompson sampling. As Figure 1 shows UCB-like algorithm has better regret performance, it is interesting to develop an UCB/TS-based algorithm which achieves better practical performance with guaranteeing the same optimal theoretical results claimed in this work. Second, one can extend the work to capture more practical concerns, such as considering an underlying topology for agents, communication delays between agents, and lossy communication between agents.

References

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

Avito. *Avito Context Ad Clicks*, 2015. https://www.kaggle.com/c/avito-context-ad-clicks.

Yogev Bar-On and Yishay Mansour. Individual regret in cooperative nonstochastic multi-armed bandits. *Advances in Neural Information Processing Systems*, 32, 2019.

Ilai Bistritz and Nicholas Bambos. Cooperative multi-player bandit optimization. *Advances in Neural Information Processing Systems*, 33:2016–2027, 2020.

Sébastien Bubeck. *Bandits games and clustering foundations*. PhD thesis, Université des Sciences et Technologie de Lille-Lille I, 2010.

Swapna Buccapatnam, Jian Tan, and Li Zhang. Information sharing in distributed stochastic bandits. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 2605–2613. IEEE, 2015.

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

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*, pp. 3471–3481. PMLR, 2020.

Yu-Zhen Janice Chen, Lin Yang, Xuchuang Wang, Xutong Liu, Mohammad Hajiesmaili, John CS Lui, and Don Towsley. On-demand communication for asynchronous multi-agent bandits. In *International Conference on Artificial Intelligence and Statistics*, pp. 3903–3930. PMLR, 2023.

Abhimanyu Dubey et al. Cooperative multi-agent bandits with heavy tails. In *International Conference on Machine Learning*, pp. 2730–2739. PMLR, 2020.

Hossein Esfandiari, Amin Karbasi, Abbas Mehrabian, and Vahab Mirrokni. Regret bounds for batched bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7340–7348, 2021.

Raphaël Féraud, Réda Alami, and Romain Laroche. Decentralized exploration in multi-armed bandits. In *International Conference on Machine Learning*, pp. 1901–1909. PMLR, 2019.

Zijun Gao, Yanjun Han, Zhimei Ren, and Zhengqing Zhou. Batched multi-armed bandits problem. *Advances in Neural Information Processing Systems*, 32, 2019.

Aurélien Garivier, Tor Lattimore, and Emilie Kaufmann. On explore-then-commit strategies. *Advances in Neural Information Processing Systems*, 29, 2016.

Tianyuan Jin, Jing Tang, Pan Xu, Keke Huang, Xiaokui Xiao, and Quanquan Gu. Almost optimal anytime algorithm for batched multi-armed bandits. In *International Conference on Machine Learning*, pp. 5065–5073. PMLR, 2021.

Nikolai Karpov and Qin Zhang. Collaborative regret minimization in multi-armed bandits. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

Ravi Kumar Kolla, Krishna Jagannathan, and Aditya Gopalan. Collaborative learning of stochastic bandits over a social network. *IEEE/ACM Transactions on Networking*, 26(4):1782–1795, 2018.

Tze Leung Lai, Herbert Robbins, et al. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

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)*, pp. 167–172. IEEE, 2016.

Udari Madhushani and Naomi Leonard. When to call your neighbor? strategic communication in cooperative stochastic bandits. *arXiv preprint arXiv:2110.04396*, 2021.

David Martínez-Rubio, Varun Kanade, and Patrick Rebeschini. Decentralized cooperative stochastic bandits. *Advances in Neural Information Processing Systems*, 32, 2019.

Abbas Mehrabian, Etienne Boursier, Emilie Kaufmann, and Vianney Perchet. A practical algorithm for multiplayer bandits when arm means vary among players. In *International Conference on Artificial Intelligence and Statistics*, pp. 1211–1221. PMLR, 2020.

Vianney Perchet, Philippe Rigollet, Sylvain Chassang, and Erik Snowberg. Batched bandit problems. *The Annals of Statistics*, pp. 660–681, 2016.

Chengshuai Shi and Cong Shen. Federated multi-armed bandits. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

Chengshuai Shi, Cong Shen, and Jing Yang. Federated multi-armed bandits with personalization. In *International Conference on Artificial Intelligence and Statistics*, pp. 2917–2925. PMLR, 2021a.

Chengshuai Shi, Wei Xiong, Cong Shen, and Jing Yang. Heterogeneous multi-player multi-armed bandits: Closing the gap and generalization. *Advances in Neural Information Processing Systems*, 34, 2021b.

Balazs Szorenyi, Róbert Busa-Fekete, István Hegedus, Róbert Ormándi, Márk Jelasity, and Balázs Kégl. Gossip-based distributed stochastic bandit algorithms. In *International Conference on Machine Learning*, pp. 19–27. PMLR, 2013.

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*, pp. 4120–4129. PMLR, 2020a.

Xuchuang Wang, Lin Yang, Yu-zhen Janice Chen, Xutong Liu, Mohammad Hajiesmaili, John Lui, and Don Towsley. Achieve near-optimal individual regret & low communications in multi-agent bandits. In *International Conference on Learning Representations*, 2023.

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.

Yuanhao Wang, Jiachen Hu, Xiaoyu Chen, and Liwei Wang. Distributed bandit learning: Near-optimal regret with efficient communication. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020b.

Lin Yang, Yu-Zhen Janice Chen, Stephen Pasteris, Mohammad Hajiesmaili, John Lui, and Don Towsley. Cooperative stochastic bandits with asynchronous agents and constrained feedback. *Advances in Neural Information Processing Systems*, 34:8885–8897, 2021.

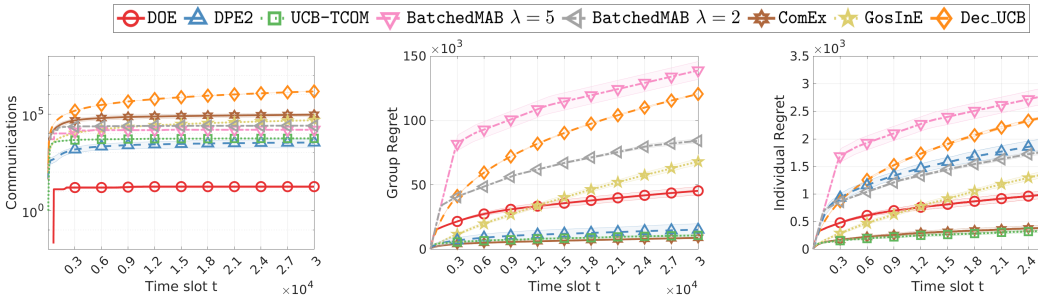
Lin Yang, Yu-Zhen Janice Chen, Mohammad H Hajiemaili, John CS Lui, and Don Towsley. Distributed bandits with heterogeneous agents. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, pp. 200–209. IEEE, 2022.

Jingxuan Zhu, Ethan Mulle, Christopher Salomon Smith, and Ji Liu. Decentralized multi-armed bandit can outperform classic upper confidence bound. *arXiv preprint arXiv:2111.10933*, 2021.

A Numerical Results

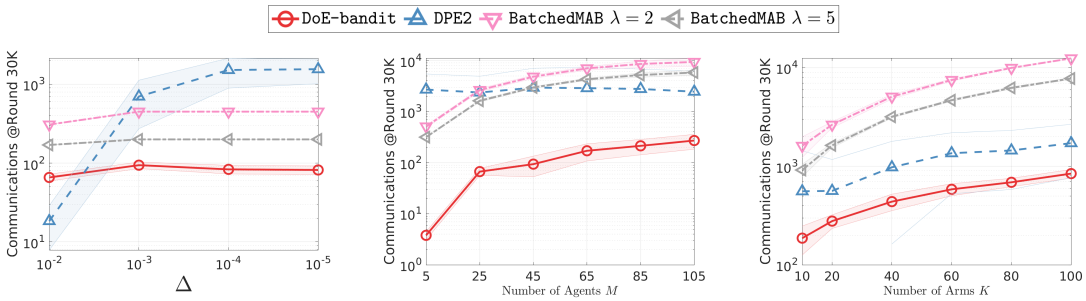
In this section, we conduct numerical experiments to corroborate the performance of the `DoE-bandit` algorithm. We aim to highlight the advantage of `DoE-bandit` in group and individual regrets and in communication costs over start-of-the-art baselines.

Setups and Baselines. In `DoE-bandit` algorithm, we set parameters $\alpha = 1, \beta = 3$ and $\delta_t = 1/T^2$. We run 50 trials of each experiment and plot the means as lines and their standard deviations as shaded regions. We compare the regret and communication costs of `DoE-bandit` with six baselines, `ComEx` (Madhushani & Leonard, 2021), `GosInE` (Chawla et al., 2020), `Dec_UCB` (Zhu et al., 2021),



(a) Communication cost (b) Group regret (c) Individual regret

Figure 1: `DoE-bandit` (this work) vs. baseline algorithms listed in Table 1



(a) Vary reward gap (b) Vary agent number (c) Vary arm number

Figure 2: Communications: `DoE-bandit` vs. `DPE2` and `BatchedMAB`

`DPE2` (Wang et al., 2020a), `UCB-TCOM` (Wang et al., 2023)) and `BatchedMAB` (Karpov & Zhang, 2024) outlined in Table 1. We note that some of the baseline algorithms are developed for a set of agents that are connected through an underlying graph topology. Hence, to make the comparison fair, we consider a complete graph for all algorithms so that any two agents can communicate. Among these baselines, the most relevant ones to ours are `BatchedMAB` and `DPE2`, as they also achieve constant communications. Especially, `Batched` has an important parameter λ (≥ 2) that is used to tune its communication frequency. To make the comparison fair, we pick both $\lambda = 2$ and $\lambda = 5$ in the experiments. All other baselines' parameters follow their default choice.

Experimental Results. Figure 1 reports the comparison results in group regret, individual regret, and communication costs. Figure 1a shows that `DoE-bandit` achieves the smallest communication costs among all algorithms. The experiments is conducted in a multi-agent bandits setting with $K = 100$ arms, $M = 50$ agents, and $T = 30K$, and each arm is associated with a Bernoulli distribution with mean randomly taken from the click-through-rate in `Ad-Clicks` Avito (2015). Figure 1b reports the group regrets of algorithms. The results show `DoE-bandit` is not as good as `DPE2`, `ComEx`, and `UCB-TCOM`. This is because `DoE-bandit` is based on the arm-elimination policy and others are UCB-like algorithms. It is known that with the same order-wise regret performance, UCB algorithms are empirically better than elimination ones in general (Garivier et al., 2016, §6). Figure 1c reports the maximum individual regrets of agents. UCB-like algorithms perform still better than others. However, `DPE2`—one of the other algorithms with constant communication cost—suffer poor individual regret since `DPE2` leverages a leader-follower structure, where the leader agent incurs high individual regret. For both $\lambda = 2$ and $\lambda = 5$ cases, `BatchedMAB`, the other baseline with constant communication, has relative bad group and individual regret performance.

Figure 2 compares the communication costs of `DoE-bandit` with the constant communication cost alternatives `DPE2` and `BatchedMAB` across various parameter settings. Three parameters are analyzed: (1) reward gap Δ between arms ($K = 10$ with mean $\mu(k) = 0.09 + k\Delta$, $M = 5$) in Figure 2a; (2) number of agents M ($K = 20$) in Figure 2b; and (3) number of arms K ($M = 25$) in Figure 2c, where

the reward means of Figures 2b and 2c are also drawn from `Ad-Clicks` Avito (2015). The log-y-axis represents cumulative communication costs at the time horizon's end. `DoE-bandit` consistently outperforms `DPE2` and `BatchedMAB` in all scenarios, except for large Δ, where `DPE2` excels. Notably, as Δ decreases, `DoE-bandit`'s communication costs remain stable, contrasting with `DPE2`, which experiences an increase. This is attributed to the superior $O(KM \log \Delta^{-1})$ communication cost of `DoE-bandit` compared to `DPE2`'s $O(K^2M^2\Delta^{-2})$. Lastly, communication costs for all algorithms rise with increasing M (Figure 2b) or K (Figure 2c), confirming their dependence on K and M as per their communication cost upper bounds.

B Proof for Regret and Communication Cost Upper Bounds (Theorem 3)

Proof for the Regret Results By running the `DoE` subroutine, the bandit learning algorithm can build up a confidence interval for the mean reward of an arm. In Lemma 2, we provide the estimation performance of `DoE` in estimating the mean of an arm.

According to the results in Lemma 2, letting $\texttt{ECR}_t(k) = \alpha \min\{1, \texttt{CR}_{[0,1]}(Mn_t(k), \delta_t)\}$, each agent can attain the order-optimal estimate (up to a constant factor $2\alpha\beta + \beta$) for the mean reward, which slightly degrades the performance of the bandit algorithm. We prove the regret of `DoE-bandit` by using the observation in Lemma 2.

In our analysis, we categorize decisions made by the agents into Type-I and Type-II decisions. Type-I corresponds to the decisions of an agent when the true mean values of all arms lie in the confidence intervals calculated by each agent, i.e., for any arm k and agent m,

$$\mu(k) \in \left[\hat{\mu}_t^{(m)}(k) - (2\alpha\beta+\beta)\texttt{CR}_{[0,1]}(Mn_t(k),\delta_t), \hat{\mu}_t^{(m)}(k) + (2\alpha\beta+\beta)\texttt{CR}_{[0,1]}(Mn_t(k),\delta_t)\right].$$

Otherwise, Type-II decision occurs, i.e., the actual mean value of some arm is not within the confidence interval calculated by some agents. Note that agents may incur high regret when wrongly eliminating the optimal arm from the candidate set at some time slot with making a Type-II decision. To prove the regret, we upper bound the probability that a Type-II decision happens and the number of pulls of suboptimal arms without any Type-II decision occurring, respectively.

We first upper bound the probability of the occurrence of a Type-II decision. Note that an agent makes a Type-II decision once the true mean of some arm is outside the confidence interval. For any time slot t, any agent m and any arm k, from Lemma 2, we have that

$$\mu(k) \notin \left[\hat{\mu}_t^{(m)}(k) - (2\alpha\beta+\beta)\texttt{CR}_{[0,1]}(Mn_t(k),\delta_t), \hat{\mu}_t^{(m)}(k) + (2\alpha\beta+\beta)\texttt{CR}_{[0,1]}(Mn_t(k),\delta_t)\right],$$

with a probability of at most $Mt\delta_t^{1/2}$. Then, an agent at any time slot makes a Type-II decision with probability at most $KMt\delta_t^{1/2}$ (there are K arms). Hence, with a union bound, the probability that a Type-II decision has happened before a time slot s can be obtained by summing up the above probabilities over investigated time slots (up to s) and agents, which is $KM^2 \sum_{t\le s} t\delta_t^{1/2}$.

Now we proceed to upper bound the number of pulls of a suboptimal arm with only Type-I decisions happening. According to the rule of the elimination-based bandit algorithm, a suboptimal arm will be removed from the candidate set without further consideration only when the radius of the confidence interval for this arm reduces to a small value with enough samples. By the following lemma, we upper bound the number of pulls of suboptimal arms by all agents, i.e., $Mn_t(k)$, when Type-I decision happens.

Lemma 5. *At any time $t \le T$, if the optimal arm lies in the candidate set and an agent makes a Type-I decision with pulling a suboptimal arm k, i.e., $I_t^{(m)} = k$, there is*

$$Mn_t(k) \le \frac{8(2\alpha+1)^2\beta^2 \log \delta_t^{-1}}{\Delta_k^2} + M.$$

Lemma 5's proof is presented at Appendix B.2. Lemma 5 holds for any arm, and, therefore, the total number of times of pulling all K arms (before elimination, in a round-robin manner) is upper bounded by $\tau \coloneqq \frac{8K(2\alpha+1)^2\beta^2 \log \delta_t^{-1}}{\Delta^2}$, where $\Delta \coloneqq \min_{k:\Delta_k>0} \Delta_k$. Hence, if there is no Type-II decision happening before τ, the optimal arm will stay in the candidate set all the time, and the regret of `DoE-bandit` in this case is

$$\sum_{k:\Delta_k>0} M n_t(k)\Delta_k = \sum_{k:\Delta_k>0} \left(\frac{8(2\alpha\beta+\beta)^2 \log \delta_t^{-1}}{\Delta_k} + M\Delta_k \right).$$

On the other hand, if there is a Type-II decision happening before τ, we can upper bound the regret of `DoE-bandit` by MT.

Last, with the regrets in the cases with/without a Type-II decision, we can upper bound the expected regret of the `DoE-bandit` algorithm.

$$\mathbb{E}\left[R_T\right] \leq \sum_{k:\Delta_k>0} \left(\frac{8(2\alpha\beta+\beta)^2 \log \delta_T^{-1}}{\Delta_k} + M\Delta_k \right) + KM^3T \sum_{t\leq\tau} t\delta_t^{1/2},$$

where the first term on the right-hand side corresponds to the regret portion when there is no Type-II decision before τ, and the second term corresponds to the other case.

Due to the same round-robin arm pulling manner in Line 6 of Algorithm 2, all agents by `DoE-bandit` pull the same arm at any time slot, and, therefore, all agents' individual regrets (rewards) are equal. So, we obtain the individual regret for each agent by dividing the above total regret upper bound equally. We summarize the above results and give the regret upper bounds for `DoE-bandit` in Theorem 3.

Proof for the Communication Costs We analyze the communication overheads of `DoE-bandit` arm by arm. If there is a Type-II decision before τ, we use MT to upper bound the communication overheads. The expected communication complexity in this case is then

$$KM^3T \sum_{t\leq\tau} t\delta_t^{1/2}. \tag{9}$$

In the following, we focus on Type-I decisions. For any suboptimal arm k (with $\Delta_k > 0$), let τ_k be the last time that `DoE-bandit` pulls the arm k. At τ_k, we have

$$4(2\alpha\beta+\beta)\texttt{CR}_{[0,1]}\left(Mn_{\tau_k}(k), \delta_t\right) \geq \Delta_k.$$

The above equation is proved in the proof of Lemma 5 (at Appendix B.2). With $\texttt{ECR}_{\tau_k}(k) = \alpha\texttt{CR}_{[0,1]}(Mn_{\tau_k}(k), \delta_t)$, and

$$4(2\alpha\beta+\beta)\frac{1}{\alpha}\texttt{ECR}_{\tau_k}(k) \geq \Delta_k.$$

Hence, up to time τ_k, the communications due to arm k is (recall $\texttt{ECR}_1(k) = 1$)

$$\log_\beta \frac{\texttt{ECR}_1(k)}{\texttt{ECR}_{\tau_k}(k)} \leq \log_\beta \left(\frac{4(2\alpha\beta+\beta)}{\alpha\Delta_k} \right).$$

The expected number of communications by suboptimal arms is at most

$$\sum_{k:\Delta_k>0} \log_\beta \left(\frac{4(2\alpha\beta+\beta)}{\alpha\Delta_k} \right).$$

For the optimal arm, the number of communications (when there is no Type-II decision) can be upper bounded by the largest communication overheads of suboptimal arms. That is, the number of

communications about the optimal arm is upper bounded by $O(\log_\beta(1/\Delta))$ where the Δ corresponds to the smallest non-zero reward gap. That is because when there are multiple arms in the candidate set, the optimal arm with others in the candidate set is pulled in a round-robin manner and incurs the same communication overheads as others in the set; and when there is only one arm left in the candidate set, the `DoE-bandit` stops communication. So, to sum up, the total communication overheads is upper bounded by

$$\sum_{k:\Delta_k>0} \log_\beta\left(\frac{4(2\alpha\beta+\beta)}{\alpha\Delta_k}\right) + \log_\beta\left(\frac{4(2\alpha\beta+\beta)}{\alpha\Delta}\right) \leq 2\cdot \sum_{k:\Delta_k>0} \log_\beta\left(\frac{4(2\alpha\beta+\beta)}{\alpha\Delta_k}\right)$$

At each communication time, agents spend totally $3M$ messages in collecting messages and synchronize the estimates in each agent. In addition, `DoE` may update the candidate set in agents when an arm is eliminated, that costs another $M(K-1)$ messages. Therefore, combined with (9)), the expected communication overheads of `DoE-bandit` (the total number of messages) is upper bounded by (5)).

B.1 A Proof of Lemma 2

We prove the lemma by analyzing the following two cases. Let s denote the last detection point, i.e., the last time slot (before t) that the condition in Line **??** of Algorithm 2 holds.

Case (1): the agent communicated at the last detection point s. In this case, the estimate $\hat{\mu}_t^{(m)}(k)$ is obtained by averaging $Mn_s(k)+n_t(k)-n_s(k)$ samples. Hence, the following equation holds with probability $1-Mt\delta_t^{1/2}$,

$$\begin{aligned}
|\hat{\mu}_t^{(m)}(k)-\mu(k)| &\overset{(a)}{\leq} \texttt{CR}_{[0,1]}(Mn_s(k)+n_t(k)-n_s(k),\delta_t)\\
&\overset{(b)}{\leq} \texttt{CR}_{[0,1]}(Mn_s(k),\delta_t)\\
&\overset{(c)}{\leq} \texttt{CR}_{[0,1]}(Mn_s(k),\delta_s)\\
&\overset{(d)}{\leq} \beta\texttt{CR}_{[0,1]}(Mn_t(k),\delta_t),
\end{aligned}$$

where the inequality (a) is proved by Hoeffding's inequality and union bound (see below), inequality (b) is due to that the confidence radius `CR` increases with a smaller number of samples, inequality (c) is because δ_t is decreasing with respect to t and $s<t$, and the inequality (d) is due to that the condition in Line **??** is false at time slot t.

Below, we present the detailed steps for proving inequality (a) as follows,

$$
\begin{aligned}
&\mathbb{P}\left(|\hat{\mu}_t^{(m)}(k)-\mu(k)| \leq \mathtt{CR}_{[0,1]}(Mn_s(k)+n_t(k)-n_s(k),\delta_t)\right)\\
&=\mathbb{P}\left(|\hat{\mu}_t^{(m)}(k)-\mu(k)| \leq \sqrt{\frac{\log \delta_t^{-1}}{2\left(Mn_s(k)+n_t(k)-n_s(k)\right)}}\right)\\
&=1-\mathbb{P}\left(|\hat{\mu}_t^{(m)}(k)-\mu(k)| > \sqrt{\frac{\log \delta_t^{-1}}{2\left(Mn_s(k)+n_t(k)-n_s(k)\right)}}\right)\\
&\overset{(a1)}{=}1-\sum_{n=1}^{M\cdot t}\mathbb{P}\left(|\hat{\mu}_t^{(m)}(k)-\mu(k)| > \sqrt{\frac{\log \delta_t^{-1}}{2n}}\,\middle|\, Mn_s(k)+n_t(k)-n_s(k)=n\right)\\
&\qquad\qquad\times\mathbb{P}\left(Mn_s(k)+n_t(k)-n_s(k)=n\right)\\
&\geq 1-\sum_{n=1}^{M\cdot t}\mathbb{P}\left(|\hat{\mu}_t^{(m)}(k)-\mu(k)| > \sqrt{\frac{\log \delta_t^{-1}}{2n}}\,\middle|\, Mn_s(k)+n_t(k)-n_s(k)=n\right)\\
&\overset{(a2)}{\geq} 1-\sum_{n=1}^{M\cdot t}\delta_t^{1/2} \geq 1-Mt\delta_t^{1/2},
\end{aligned}
$$

where the equation (a1) is due to union bound, and inequality (a2) is by applying Hoeffding's inequality.

In this case, the result in Lemma 2 holds.

Case (2): there is no communication at s. Let A be the sum of samples obtained by agent m if communication *happened* at s. We have

$$
\begin{aligned}
&\left|(Mn_s(k)+n_t(k)-n_s(k))\hat{\mu}_t^{(m)}(k)-A\right|\\
&=\left|\left(Mn_s(k)\hat{\mu}_{\mathtt{aux},s}^{(m)}(k)+\left(X_t^{(m)}(k)-X_s^{(m)}(k)\right)\right)-\left(\sum_{m'=1}^{M}X_s^{(m')}(k)+\left(X_t^{(m)}(k)-X_s^{(m)}(k)\right)\right)\right|\\
&=\left|Mn_s(k)\hat{\mu}_{\mathtt{aux},s}^{(m)}(k)-\sum_{m'=1}^{M}X_s^{(m')}(k)\right|.
\end{aligned}
$$

The above equation is based on the fact that agent always has the local samples after s no matter there is communication at s.

Hence,

$$
\begin{aligned}
\left|\hat{\mu}_t^{(m)}(k)-\frac{A}{Mn_s(k)+n_t(k)-n_s(k)}\right| &= \frac{1}{(Mn_s(k)+n_t(k)-n_s(k))}\left|Mn_s(k)\hat{\mu}_{\mathtt{aux},s}^{(m)}(k)-\sum_{m'=1}^{M}X_s^{(m')}(k)\right|\\
&\leq \left|\hat{\mu}_{\mathtt{aux},s}^{(m)}(k)-\frac{1}{Mn_s(k)}\sum_{m'=1}^{M}X_s^{(m')}(k)\right|\\
&\overset{(a)}{\leq} 2\alpha\mathtt{CR}_{[0,1]}(Mn_s(k),\delta_s(k))\\
&\leq 2\alpha\mathtt{CR}_{[0,1]}(Mn_s(k),\delta_t(k))\\
&\leq 2\alpha\beta\mathtt{CR}(Mn_t(k),\delta_t(k)),
\end{aligned}
$$

where inequality (a) is because the condition in Line **??** in Algorithm 2 does not hold at time slot s (since there is no communication at s). In this case, $\left|\hat{\mu}_{\mathtt{aux},s}^{(m)}(k)-\hat{\mu}_{\mathtt{aux},s}^{(m')}(k)\right| \leq 2\mathtt{ECR}_s(k) =$

$2\alpha\texttt{CR}_{[0,1]}(Mn_s(k),\delta_s)$ for any agents m and m'. Also, $\frac{1}{Mn_s(k)}\sum_{m'=1}^{M} X_s^{(m')}(k)$ averages all samples up to s, and hence its value lies between $\min_{m'}\hat{\mu}^{(m')}_{\texttt{aux},s}(k)$ and $\max_{m'}\hat{\mu}^{(m')}_{\texttt{aux},s}(k)$, which weight partial local samples with a factor M to replace missing ones. Combining the two facts yields inequality (a).

Since A contains the same set of samples as the $\hat{\mu}_t^{(m)}(k)$ in Case (1), the following equation also holds with probability $1-Mt\delta_t^{1/2}$:

$$\left|\frac{A}{Mn_s(k)+n_t(k)-n_s(k)}-\mu(k)\right|\leq\beta\texttt{CR}_{[0,1]}(Mn_t(k),\delta_t).$$

Combining the above two equations yields

$$\left|\hat{\mu}_t^{(m)}(k)-\mu(k)\right|\leq(2\alpha+1)\beta\texttt{CR}_{[0,1]}(Mn_t(k),\delta_t),\ \text{with probability}\ 1-Mt\delta_t^{1/2}.$$

As a result, we prove the Lemma 2.

B.2 A Proof of Lemma 5

We consider agent m running the proposed algorithm makes a Type-I decision at time t and $I_t^{(m)}=k$. First, we claim that the following holds.

$$2(2\alpha\beta+\beta)\texttt{CR}_{[0,1]}\left(Mn_t(k),\delta_t\right)+2(2\alpha\beta+\beta)\texttt{CR}_{[0,1]}\left(Mn_t(k^*),\delta_t\right)\geq\Delta_k, \tag{10}$$

where k^* is the optimal arm.

Otherwise, we have

$$\begin{aligned}
&\hat{\mu}_t^{(m)}\left(k^*\right)-(2\alpha\beta+\beta)\texttt{CR}_{[0,1]}\left(Mn_t(k^*),\delta_t\right)\\
&=\hat{\mu}_t^{(m)}\left(k^*\right)+(2\alpha\beta+\beta)\texttt{CR}_{[0,1]}\left(Mn_t(k^*),\delta_t\right)-2(2\alpha\beta+\beta)\texttt{CR}_{[0,1]}\left(Mn_t(k^*),\delta_t\right)\\
&\geq\mu\left(k^*\right)-2(2\alpha\beta+\beta)\texttt{CR}_{[0,1]}\left(Mn_t(k^*),\delta_t\right)\\
&=\mu(k)+\Delta_k-2(2\alpha\beta+\beta)\texttt{CR}_{[0,1]}\left(Mn_t(k^*),\delta_t\right)\\
&\overset{(a)}{>}\mu(k)+2(2\alpha\beta+\beta)\texttt{CR}_{[0,1]}\left(Mn_t(k),\delta_t\right)\\
&\geq\hat{\mu}_t^{(m)}(k)+(2\alpha\beta+\beta)\texttt{CR}_{[0,1]}\left(Mn_t(k),\delta_t\right),
\end{aligned}$$

where inequality (a) uses the negation of (10). It shows the fact that the lower confidence bound of arm k^* is larger than the upper confidence bound of arm k, contradicting the rules of the algorithm to pull arm k.

It follows from Equation (10) that $4(2\alpha\beta+\beta)\texttt{CR}_{[0,1]}\left(M(n_t(k)-1),\delta_t\right)\geq\Delta_k$, since the algorithm pull arms in a round robin manner. Last, we apply the confidence interval function for a Bernoulli process, and prove that the number of observations of k received by agent m is upper bounded by

$$Mn_t(k)\leq\frac{8(2\alpha+1)^2\beta^2\log\delta_t^{-1}}{\Delta_k^2}+M.$$

C Proof for Communication Lower Bound (Theorem 1)

Proof of Theorem 1. We first prove the $\Omega(\sum_{k:\Delta_k>0}\log\Delta_k^{-1})$ *communication lower bound.* Let us fix a suboptimal arm k. To facilitate the proof presentation, we denote $n_t^{(m)}(k)$ as the *accessible* number of observations for arm k of agent m at time t, where the "accessible" observations includes the agent's own local observations as well as other agents' observations received through communications (on or before time slot t). We first prove two key claims by contradiction as follows,

Claim 1. The initial communication concerning arm k must occur on or before the time slot when the accessible number of observations of arm k of all agents reach $G \log T$ for a universal constant G.

Claim 2. If the algorithm communicates regarding arm k at a specific time slot t, it must have a further communication on arm k either on or before the time slot when the expected accessible number of observations of arm k of all agents reach twice the count recorded at time slot t.

Proof of Claim 1. If at the initial communication, all agents' accessible observations of arm k are all $\Omega(G \log T)$ and noticing that all of these observations are local, the total number of pulls on arm k at time t would be $\Omega(MG \log T)$, where the M factor contradicts the near-optimal group regret (without M) that the algorithm achieves. □

Proof of Claim 2. For the communication time slot t, the accessible observations of an agent is equivalent to the global observations of all agents. So, we have $n_t^{(m)}(k) = n_t(k)$ for all agents $m \in \mathcal{M}$. We note that since the algorithm achieves near-optimal regrets, we have that $\mathbb{E}[n_t(k)] = \Omega\left(\frac{\log t}{\Delta_k^2}\right)$. After time slot t, if there is no communication till time slot t' such that $n_{t'}^{(m)}(k) \geq 2n_t(k)$ for all agent m, then all agents more respectively pull arm k for $n_{t'}^{(m)}(k) - n_t^{(m)}(k) \geq n_t(k)$. Then, the total number of pulls of arm k between time slots t and t' is at least $\Omega(Mn_t(k)) = \Omega\left(M\frac{\log t}{\Delta_k^2}\right)$. This M factor on the number of pulling times of arm k contradicts the near-optimal group regret that the algorithm achieves. Therefore, there must exist a communication on arm k between time slots t and t'. □

With Claims 1 and 2 and assuming that the algorithm makes $\mathbb{E}[C_T(k)]$ number of communications on arm k, then the total number of pulls on arm k is at most $2^{\mathbb{E}[C_T(k)]} G \log T$. Since the algorithm also achieves the near-optimal regret upper bound, we have

$$2^{\mathbb{E}[C_T(k)]} G \log T \geq \Theta\left(\frac{\log T}{\Delta_k^2}\right),$$

which yields the communication lower bound for arm k as follows,

$$\mathbb{E}[C_T(k)] \geq \Omega\left(\log\left(\frac{1}{G\Delta_k^2}\right)\right) = \Omega(\log \Delta_k^{-1}).$$

Therefore, summing over all suboptimal arms yields the overall communication lower bound $\Omega(\sum_{k:\Delta_k>0} \log \Delta_k^{-1})$.

Next, we prove another communication lower bound $\Omega(M)$, which, together with the above bound, concludes the proof. Blow, we prove the bound via contradiction. That is, we start from assuming the communication is less than cM where c is a constant.

Denote $Y^{(m)}$ as the number of integers or real numbers that agent m sends or receives throughout a run. $Y^{(m)}$ is a random variable. Since expected communication cost is less than cM, we have

$$\sum_{m=1}^{M} \mathbb{E}[Y^{(m)}] \leq cM.$$

Denote $\mathcal{S}$ as the set of $M/2$ agents with smaller $\mathbb{E}[Y^{(m)}]$. The expected communication cost of any agent $m \in \mathcal{S}$ is at most $2c$. For any agent $m \in \mathcal{S}$, we have $\mathbb{P}(Y^{(m)} \geq 1) \leq \mathbb{E}[Y^{(m)}] \leq 2c$, where the first inequality is by Markov's inequality. That is, for any of these agents, the probability of communicating with some agent is less than $2c$. Suppose that agent m is such an agent. Then, we can map the communication protocol to a single-agent algorithm by simulating the learning process of agent m.

The simulation proceeds as follows: We engage with a single-agent bandit over a time horizon of T, executing the code corresponding to agent m within the specified protocol. In the absence of any communication requirements, we advance to the subsequent line in agent m's code. However, if the code initiates message transmission or awaits a message, we conclude the execution. Throughout the remaining time steps, we employ a single-agent optimal algorithm, specifically the one employed to achieve the optimal regret upper bound, denoted as R_T^*.

Then, if agent m's code has δ probability of involving in communication, and if agent m's regret $R_T^{(m)} \leq A$ (in its original distributed algorithm design), via this reduction, we can obtain an algorithm for single-agent MAB with expected regret

$$R_T \leq A + \delta \cdot R_T^*.$$

By the regret lower bound result of Lai et al. (1985, Theorem 1), we have

$$\sum_{k>1} \frac{\Delta_k}{\mathrm{KL}(\mu_k, \mu_1)} \leq \liminf_{T\to\infty} \frac{R_T}{\log T} \leq \liminf_{T\to\infty} \frac{A + \delta \cdot R_T^*}{\log T} \leq \liminf_{T\to\infty} \frac{A}{\log T} + \delta \cdot \sum_{k>1} \frac{\Delta_k}{\mathrm{KL}(\mu_k, \mu^*)}.$$

That is,

$$\liminf_{T\to\infty} \frac{A}{\log T} \geq (1-\delta) \sum_{k>1} \frac{\Delta_k}{\mathrm{KL}(\mu_k, \mu_1)} \geq (1-2c) \sum_{k>1} \frac{\Delta_k}{\mathrm{KL}(\mu_k, \mu_1)}.$$

If setting $c = 0.0005$, then the regret of any agent m in set $\mathcal{S}$ fulfills the above lower bound. So, the total regret is at least

$$\sum_{m\in\mathcal{S}} \liminf_{T\to\infty} \frac{A}{\log T} \geq \left(\frac{1}{2} - c\right) \cdot M \sum_{k>1} \frac{\Delta_k}{\mathrm{KL}(\mu_k, \mu_1)},$$

which contradicts the near-optimal regret upper bound (without the linear dependence on M). Therefore, a `MA2B` algorithm with near-optimal regret requires at least $\Omega(M)$ communications. □