On Regret-Optimal Learning in Decentralized Multi-player Multi-armed Bandits

Naumaan Nayyar, Dileep Kalathil and Rahul Jain

Abstract—We consider the problem of learning in single-player and multiplayer multi-armed bandit models. Bandit problems are classes of online learning problems that capture exploration versus exploitation tradeoffs. In a multi-armed bandit model, players can pick among many arms, and each play of an arm generates an i.i.d. reward from an unknown distribution. The objective is to design a policy that maximizes the expected reward over a time horizon for a single player setting and the sum of expected rewards for the multiplayer setting. In the multiplayer setting, arms may give different rewards to different players. There is no separate channel for coordination among the players. Any attempt at communication is costly and adds to regret. We propose two decentralizable policies, E^3 (E-cubed) and E^3-TS, that can be used in both single player and multiplayer settings. These policies are shown to yield expected regret that grows at most as $O(\log^{1+\delta} T)$ (and $O(\log T)$ under some assumption). It is well known that $O(\log T)$ is the lower bound on the rate of growth of regret even in a centralized case. The proposed algorithms improve on prior work where regret grew at $O(\log^2 T)$. More fundamentally, these policies address the question of additional cost incurred in decentralized online learning, suggesting that there is at most an δ-factor cost in terms of order of regret. This solves a problem of relevance in many domains and had been open for a while.

Index Terms—Learning algorithms; Decision making; Decentralized systems; Optimization problems; Cognitive systems.

I. Introduction

Multi-armed bandit models represent an exploration versus exploitation trade-off where the player must choose between *exploring* the environment to find better options, and *exploiting* based on her current knowledge to maximize her utility. These models are widely applicable in many application like display advertisements, sensor networks, route planning and spectrum sharing. The model can be understood through a simple game of choosing between two coins with unknown biases. The coins are tossed repeatedly and one of them is chosen at each instant. If at a given instance, the chosen coin turns up heads, we get a reward of \$1, otherwise we get no reward. It is known that one of the two coins has a better bias, but the identity of the coin is not known. The question is, what is the optimal 'learning' policy that helps maximize the expected reward, i.e., to discover which coin has a better bias and at the same time maximize the cumulative reward as the game is played. Note that the player doesn't know the value of the biases as well as she has no prior probability distribution on these values. This

The first and third authors are with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089. The second author is with the Department of EECS, University of California, Berkeley. Emails: (nnayyar,rahul.jain)@usc.edu, dileep.kalathil@berkeley.edu.

motivates the *non-Bayesian* setting. The formulation where the player has a prior distribution on the parameters is called Bayesian multi-armed bandits.

The idea of multi-armed bandit models dates back to Thompson [1] and the first rigorous formulation is due to Robbins [2]. The single player multi-armed bandit problem in a non-Bayesian setting was first formulated by Lai and Robbins [3]. Any bandit policy that makes the best choice more than a constant fraction of the time is said to have sub-linear *regret*. Regret measures the performance of any strategy formally against the best policy that could be employed if the distribution parameters were known. It was shown in [3] that there is no learning policy that asymptotically has expected regret growing slower than $O(\log T)$. A learning scheme was also constructed that asymptotically achieved this lower bound.

This model was subsequently studied and generalized by many researchers. In [4], Anantharam et al. generalized it to the case of multiple plays, i.e., the player can pick multiple arms (or coins) when there are more than 2 arms. In [5], Agrawal proposed a sample mean based index policy that asymptotically achieved $O(\log T)$ regret. For the special case of bounded support for rewards, Auer et al. [6] introduced a simple index-based policy, UCB_1 that achieved logarithmic expected regret over finite time horizons. UCB_1 has since become the benchmark to compare new algorithms against because of its power and simplicity.

Recently, policies based on Thompson Sampling (TS) [1] have experienced a surge of interest due to their much better empirical performance [7]. It is a probability matching policy which, unlike the UCB-class of policies that use a deterministic confidence bound, draws samples from a distribution to determine which arm to play based on the probability of its being optimal. The logarithmic regret performance of the policy was not proved until very recently [8]. [9] introduced the Bayes-UCB algorithm which also uses use a Bayesian approach for analyzing the regret bound for stochastic bandit problems.

Deterministic sequencing algorithms which have separate exploration and exploitation phases have also appeared in the literature as an alternative to the joint exploration and exploitation approaches of UCB-like and probability matching algorithms. Noteworthy among these are the Phased Exploration and Greedy Exploitation policy for linear bandits [10] that achieves $O(\sqrt{T})$ regret in general and $O(\log(T))$ regret for finitely many linearly parametrized arms. Other noteworthy algorithms include the logarithmic regret achieving deterministic sequencing of exploration and exploitation policy, [11] with i.i.d. setting and [12] with Markovian setting. Single-player bandit problems have also been looked at in the PAC

framework, for instance, in [13]. However, we restrict our attention to performance in the expected sense in this work.

In addition to single player bandits, there has been growing interest in multiplayer learning in multi-armed bandits, motivated by distributed sensor networks, wireless spectrum sharing and in particular cognitive radio networks. Suppose there are two wireless users trying to choose between two wireless channels. Each wireless channel is random, and looks different to each user. If channel statistics were known, we would try to determine a matching wherein the expected sum-rate of the two users is maximized. But the channel statistics are unknown, and they must be learnt by sampling the channels. Moreover, the two users have to do this independently and cannot share their observations as there is no dedicated communication channel between them. They, however, may communicate implicitly for coordination but this would come at the expense of reduced opportunities for rewards or benefits, and thus would add to regret. One can easily imagine a more general network setting with M users and $N > M$ channels. This immediately gives rise to two questions. First, what is the lower bound for decentralized learning? That is, is there an inherent cost of decentralization in such network? And second, can we design a simple learning algorithm with provably optimal performance guarantees, in the context of such a decentralized network problem?

Policies for decentralized learning with sublinear regret have appeared in the literature for various models. When arms were restricted to have the same rewards for different users, Anandkumar et al. [14] showed that logarithmic regret was achievable as the problem reduces to a ranking problem that can be solved in constant time in a decentralized manner. Similar works have also appeared for i.i.d. [11] [15] and Markovian [12] [16] arm reward settings. Relaxing this assumption makes the problem more complicated as it now becomes a bipartite matching problem and no decentralized algorithm performs quick enough. In our previous work [17], we proposed a policy, dUCB$_4$ that achieved $O(\log^2 T)$ regret through a recurrent negotiation mechanism between players. However, the answers to the two questions above remained unknown. In a similar work [18], authors address the problem of decentralized multi-armed bandits. While they address the same problem as ours, the emphasis is on the stability of this decentralized setting with minimum possible communication. Also, they don't provide any optimality guarantees as compared to the optimal centralized learning problem. However, our paper assumes that players in the system remains the same. In [18] users can arrive and leave at random times. Landgren et. al. [19] uses a multi-armed bandit model for cooperative decision making problem in the context of running a consensus algorithm. Their setting is very different from the problem considered in this paper.

In this paper, we do not present an information theoretic lower bound on decentralized learning in a multiplayer multi-armed bandit problems. Such a result would be very interesting as it will also yield insight into the exact role of information sharing between players for a decentralized policy to work without an increase in expected regret. However, we managed to partially answer both questions above through two new decentralizable policies, E^3 and E^3-TS, where E^3 stands for Exponentially-spaced Exploration and Exploitation policy, which we also call as E-cubed.

Both policies yield expected regret of the order $O(\log^{1+\delta} T)$ ($O(\log T)$ under some assumptions) in both single and multiplayer settings. The policies are based on exploration and exploitation in pre-determined phases such that over a long time horizon T, there are only logarithmically many slots in the exploration phases. It is well known that the optimal order of regret that can be achieved is $O(\log T)$ [3]. These policies suggest an answer to the fundamental question of inherent cost to decentralize, that there is no cost to the order optimality, at least up to an $\log^\delta T$ factor. An asymptotic lower bound for the decentralized MAB problem (similar to that of the centralized MAB in [3]) is an important future research question.

The policies introduced in this paper, and the corresponding results hold even when the rewards are Markovian, i.e., each arm is a *rested* arm but the reward sequence from it forms a Markov chain [20]. However, we only present the i.i.d. case here and refer readers to our earlier paper [17] for ideas on extensions to the Markovian setting. Extensive simulations were conducted to evaluate the empirical performances of these policies and compared to prior work in the literature, including the classical UCB$_1$ and TS policies. The decentralized policies dE^3 and dE^3-TS are compared with the previously known dUCB$_4$ policy.

The rest of the paper is organized as follows. Section II describes the model and problem formulations for single and multiplayer bandits. Section III describes relevant prior work in the area. The new policies E^3 and E^3-TS, and their multiplayer counterparts, dE^3 and dE^3-TS are described and studied in Section IV. Section V presents empirical performances of new and previous policies.

II. MODEL AND PROBLEM FORMULATION

In this section, we describe problem formulations for single and multiplayer bandits. The single player formulation has been well-studied in literature, for instance, by Auer et al. [6] and others. The multiplayer formulation is much newer, and has appeared in our previous work where the dUCB$_4$ algorithm was presented [17].

A. Single player model

We consider an N-armed bandit problem. At each instant t, an arm k is chosen, and a reward $X_k(t)$ is generated, from an independent and identically distributed (i.i.d) random process with a fixed but unknown distribution. The processes are assumed to have bounded support, without loss of generality, in $[0, 1]$. Arm reward distributions have means μ_k that are unknown.

When choosing an arm, the player has access to the entire history of rewards and actions, $\mathcal{H}(t)$, with $\mathcal{H}(0) := \emptyset$. Denote the arm chosen at time t by $a(t) \in \mathcal{A} := \{1, ..., N\}$. A *policy* α is a sequence of maps $\alpha(t) : \mathcal{H}(t) \to \mathcal{A}$ that specifies the arm chosen at time t. The player's objective is to choose a policy that maximizes the expected reward over a finite time horizon T.

If the mean rewards of the arms were known, the problem is trivially solved by always playing the arm with the highest mean reward, i.e., $\tilde{\alpha}(t) = \arg\max_{1 \leq i \leq N} \mu_i$, $\forall t$. When the mean rewards are not known, a commonly used notion of *regret* is used to compare policies. Regret is the difference in cumulative rewards over a time horizon between that obtained by a policy α and when playing the most rewarding arm all the time. Formally, the player's objective is to minimize the expected regret over all causal policies α as defined above, which is given by,

$$\mathcal{R}_\alpha(T) = T\mu_1 - \mathbb{E}_\alpha\left[\sum_{t=1}^{T} X_{\alpha(t)}(t)\right], \qquad (1)$$

where arm 1 is taken to have the greatest mean w.l.o.g.

In practical implementations of bandit algorithms in low-power settings such as sensor networks where the implementation of any learning/control policy should consume minimum amount of energy, it will be useful to include a computation cost as well. This is particularly the case when the algorithms must solve combinatorial optimization problems that are NP-hard. Such costs arise in decentralized settings in particular, where algorithms pay a communication cost for coordination between the decentralized players. For example, as we shall see later in our decentralized learning algorithm, the players may have to spend many time slots for coming up with a bipartite matching. We model it as a constant C units of cost each time an index is computed by the policy. With this refinement, the regret of a policy α that computes its indices $m(T)$ times over a time horizon T is,

$$\tilde{\mathcal{R}}_\alpha(T) := \mu_1 T - \sum_{j=1}^{N} \mu_j \mathbb{E}_\alpha[n_j(T)] + C\mathbb{E}_\alpha[m(T)], \qquad (2)$$

where $n_j(T)$ is the number of times arm j is played.

B. Multiplayer model

We now describe the generalization of the single player, where we consider an N-armed bandit with M players. We will refer to arms as channels interchangeably. There is no dedicated communication channel for coordination among the players. However, we do allow players to communicate with one another by playing arms in a certain way, e.g., arm 1 signals a bit '0', arm 2 can signal a bit '1'. This of course will add to regret, and hence such communication comes at a cost. We assume that $N \geq M$.

At any instant t, each player choose one arm from the set of N arms or takes no action (i.e., selects no arm). If more than one player picks the same arm, we regard it as a *collision* and this interference results in zero reward for those players. The rest of the model is similar to the single player case. Arm k chosen by player i generates an i.i.d. reward $S_{i,k}(t)$ from an unknown distribution, which has bounded support, w.l.o.g., in $[0, 1]$. Let $\mu_{i,k}$ denote the unknown mean of $S_{i,k}(t)$.

Let $X_{i,k}(t)$ be the reward that player i gets from playing arm k at time t. Thus, if there is no collision, $X_{i,k}(t) = S_{i,k}(t)$. Denote the action of player i at time t by $a_i(t) \in$

$\mathcal{A} := \{1, \ldots, N\}$. Let $Y_i(t)$ be the communication message from player i at time t and $Y_{-i}(t)$ be the messages from all the other players except player i at time t. Then, the *history* seen by player i at time t is $\mathcal{H}_i(t) = \{(a_i(1), X_{i,a_i(1)}(1), Y_{-i}(1)), \cdots, (a_i(t-1), X_{i,a_i(t-1)}(t-1), Y_{-i}(t-1))\}$ with $\mathcal{H}_i(0) = \emptyset$. A *policy* $\alpha_i = (\alpha_i(t))_{t=1}^{\infty}$ for player i is a sequence of maps $\alpha_i(t) : \mathcal{H}_i(t) \to \mathcal{A}$ that specifies the arm to be played at time t.

The players have a *team objective*: they want to maximize the expected sum of rewards $\mathbb{E}[\sum_{t=1}^{T} \sum_{i=1}^{M} X_{i,a_i(t)}(t)]$ over some time horizon T.

Let $\mathcal{P}(N)$ denote the set of possible permutations of the N arms. If $\mu_{i,j}$ were known, the optimal policy is clearly to pick the optimal bipartite matching between arms and players (which may not be unique),

$$\mathbf{k}^{**} \in \arg\max_{\mathbf{k} \in \mathcal{P}(N)} \sum_{i=1}^{M} \mu_{i,k_i}. \qquad (3)$$

When expected rewards are not known, players must pick learning policies that minimize the *expected regret*, defined for policies $\alpha = (\alpha_i, 1 \leq i \leq M)$ as,

$$\mathcal{R}_\alpha(T) = T\sum_i \mu_{i,k_i^{**}} - \mathbb{E}_\alpha\left[\sum_{t=1}^{T}\sum_{i=1}^{M} X_{i,\alpha_i(t)}(t)\right]. \qquad (4)$$

As in the single player model, we consider a refinement of the regret to factor in computational or communication costs. Communication costs are justified because known distributed algorithms for bipartite matching [21], [22] require a certain amount of information exchange over multiple time slots. This cost will depend on the specific algorithm. Here, however, we will just consider an 'abstract' cost C.

Let C units of cost be incurred each time this occurs, and let $m(t)$ be the number of times it happens in time t. Then, the expected regret for policy α to be minimized is,

$$\mathcal{R}_\alpha(T) = T\sum_{i=1}^{M} \mu_{i,k_i^{**}} - \mathbb{E}_\alpha\left[\sum_{t=1}^{T}\sum_{i=1}^{M} X_{i,\alpha_i(t)}(t)\right] + C\mathbb{E}_\alpha[m(T)]. \qquad (5)$$

where $\mathbf{k}^{**}$ is the optimal matching as defined in (3).

III. PRIOR WORK

We now briefly describe key features and results of existing single and multiplayer bandit policies.

A. Single player policies

Of the several known policies for single-player multi-armed bandit problems, we focus on three that capture different classes of policies.

In [6], Auer et al. proposed an index based policy, UCB$_1$ which achieves logarithmic regret. It worked by playing the arm with the largest value of sample mean plus a confidence bound. The interval shrank deterministically as the arm got played more often and traded-off exploration and exploitation.

It was shown in [6] that the expected regret incurred by the policy over a horizon T is bounded by,

$$\mathcal{R}_{UCB_1}(T) \leq 8\log(T)\sum_{j>1}^{N}\frac{1}{\Delta_j} + (1+\frac{\pi^2}{3})\sum_{j>1}^{N}\Delta_j, \quad (6)$$

where $\Delta_j := \mu_1 - \mu_j$.

Thompson Sampling (TS) is a probability-matching policy that has been around for quite some time in the literature [1] although it was not well-studied in the context of bandit problems until quite recently [7], [8]. Arms are played randomly according to the probability of their being optimal. As an arm gets played more often, its sampling distribution become narrower. Unlike a fully Bayesian method such as Gittins Index [23], TS can be implemented efficiently in bandit problems. The regret of the policy was shown in [8] to be bounded by,

$$\mathcal{R}_{TS}(T) \leq O\left(\left(\sum_{j>1}^{N}\frac{1}{\Delta_j^2}\right)^2 \log(T)\right), \quad (7)$$

where the constants have been omitted for brevity. A stronger upper bound for the case of Bernoulli rewards that matches the asymptotic rate lower bound in Lai and Robbins [3] is given in [24]. Numerically, TS has been found to empirically outperform UCB_1 in most settings [7], [24].

UCB_4 is another confidence-bound based index policy that was proposed recently to overcome some of the shortcomings of the UCB_1 policy, namely its reliance on index computation in each time step and the difficulty in extending the algorithm to a multiplayer setting. It works by cleverly choosing a sequence of times to compute the UCB_1 index. The expected regret was shown in [17] to be bounded by,

$$\mathcal{R}_{UCB_4}(T) \leq \Delta_{\max}\left(\sum_{j>1}^{N}\frac{12\log(T)}{\Delta_j^2} + 2N\right), \quad (8)$$

where $\Delta_{\max} = \max \Delta_j$.

It can be shown that UCB_1 and TS incur linear regret if computation cost is included in the model. The expected regret of the UCB_4 algorithm over a time horizon T with computation cost C is bounded by [17],

$$\tilde{\mathcal{R}}_{UCB_4}(T) \leq \left(\Delta_{\max}+C(1+\log(T))\right)\left(\sum_{j>1}^{N}\frac{12\log(T)}{\Delta_j^2}+2N\right). \quad (9)$$

Thus, expected regret is $O(\log^2(T))$.

It is worth mentioning here that the UCB_2 policy [6] also achieves sub-linear expected regret with computation cost, actually outperforming UCB_4 with a logarithmic regret growth.

B. Multiplayer policies

The major issues that are encountered in decentralizing bandit policies are coordination among players and finite precision of indices being communicated. The $dUCB_4$ policy [17] was the first such policy that did not assume identical channel rewards for different players. The policy is a natural decentralization of UCB_4 that uses Bertsekas' auction algorithm [25] for distributed bipartite matching.

If $\Delta_{\min}$ is known, the expected regret of $dUCB_4$ is,

$$\tilde{\mathcal{R}}_{dUCB_4}(T) \leq (L\Delta_{\max} + C(L)(1 + \log(T)))\times$$
$$\left(\frac{4M^3(M+2)N\log(T)}{(\Delta_{\min} - ((M+1)\epsilon)^2} + NM(2M+1)\right),$$

where L is the frame length, $\Delta_{\min}$ is the minimum difference between the optimal and the next best permutations, $\Delta_{\max}$ is the maximum difference in rewards between permutations, and ϵ is the precision input of the distributed bipartite matching algorithm. Also, $C(L)$ indicates that the cost of communication and computation is a function of the frame length L. Thus, $\tilde{\mathcal{R}}_{dUCB_4}(T) = O(\log^2(T))$. A slight modification to the policy with increasing frame length addresses the case when $\Delta_{\min}$ is unknown [17].

IV. NEW (NEAR-)LOGARITHMIC BANDIT POLICIES

In this section, we present our work in developing two closely related policies for single player bandit problems and their generalizations to multiplayer settings.

A. Single player policies: E^3 and E^3-TS

E^3 and E^3-TS are phased policies detailed in Algorithms 1 and 2 respectively. Their key difference from the previous policies is that they have deterministic exploration and exploitation phases. In the following, an epoch is defined to comprise of one exploration phase and one exploitation phase.

Exploration phase: During an exploration phase, the player tries out different arms in a round-robin fashion and computes indices for each arm. At the end of the phase, the player chooses the arm with the maximum value of the index. The index computation differs for E^3 and E^3-TS policies.

Exploitation phase: In this phase, the player plays the arm that was chosen at the end of the previous exploration phase. No index computation happens during the exploitation phase and the player sticks to her decisions during this phase. The length of the exploitation phase doubles each successive epoch.

Algorithm 1 : Exponentially-spaced Exploration and Exploitation policy (E^3)

1: **Initialization:** Set $t = 0$ and $l = 1$;
2: **while** $(t \leq T)$ **do**
3:     **Exploration Phase:** Play each arm $j, 1 \leq j \leq N$, γ number of times;
4:     Update the sample mean $\overline{X}_j(l)$, $1 \leq j \leq N$;
5:     Compute the best arm $j^*(l) := \arg\max_{1 \leq j \leq N} \overline{X}_j(l)$;
6:     **Exploitation Phase:** Play arm $j^*(l)$ for 2^l time slots;
7:     Update $t \leftarrow t + N\gamma + 2^l$, $l \leftarrow l + 1$;
8: **end while**

E^3 and E^3-TS, while largely similar, differ in how they choose the arm to play during the exploitation phase. While E^3 uses the simple sample mean value, E^3-TS draws from a β-distribution in a manner similar to the TS policy.

The β-distribution is chosen in E^3-TS due to convenient posterior form after Bernoulli observations. A $\beta(a, b)$-distribution prior results in a posterior of $\beta(a+1, b)$ or $\beta(a, b+1)$ depending on success or failure of the Bernoulli trial, respectively.

Algorithm 2 : Exponentially-spaced Exploration and Exploitation algorithm-TS (E³-TS)

1: **Initialization:** Set $l = 1$ and $t = 0$. For each arm $i = 1, 2, ..., N$, set $S_i = 0, F_i = 0$;
2: **while** $(t \leq T)$ **do**
3:    **Exploration Phase:** Play each arm $j, 1 \leq j \leq N, \gamma$ number of times;
4:    For each play of each arm i, store reward as $\tilde{r}_i(t)$;
5:    Perform a Bernoulli trial with success probability $\tilde{r}_i(t)$ and observe output $r_i(t)$;
6:    If $r_i(t) = 1$, then set $S_i = S_i + 1$, else $F_i = F_i + 1$;
7:    Sample $\theta_i(l)$ from $Beta(S_i + 1, F_i + 1)$ distribution;
8:    Compute the *best arm* $j^*(l) := \arg\max_{1 \leq j \leq N} \theta_j(l)$;
9:    **Exploitation Phase:** Play arm $j^*(l)$ for 2^l time slots;
10:    Update $t \leftarrow t + N\gamma + 2^l, l \leftarrow l + 1$;
11: **end while**

We now give the performance bounds for the policies with an index computation cost C in the main result of this section. Both algorithms will be analyzed concurrently as their proof techniques are largely similar.

The following concentration inequality will be used in the analysis and is introduced here for the reader's ease.

Fact 1: Chernoff-Hoeffding inequality [26].
Let $X_1, \ldots, X_t$ be a sequence of real-valued random variables, such that, for each $i \in \{1, \ldots, t\}, 0 \leq X_i \leq 1$ and $\mathbb{E}[X_i | \mathcal{F}_{i-1}] = \mu$, where $\mathcal{F}_i = \sigma(X_1, \ldots, X_i)$. Let $S_t = \sum_{i=1}^{t} X_i$. Then for all $a \geq 0$,

$$\mathbb{P}\left(\frac{S_t}{t} \geq \mu + a\right) \leq e^{-2a^2 t}, \quad \mathbb{P}\left(\frac{S_t}{t} \leq \mu - a\right) \leq e^{-2a^2 t}.$$

We now give the main result of this section.

Theorem 1. *(Regret bounds for E³ and E³-TS policies)*
Let $\Delta_{\min}$ and $\Delta_{\max}$ denote the differences between the mean rewards of the optimal arm, and the second best and worst arms, respectively.
(i) If $\Delta_{\min}$ is known, set $\gamma = \lceil \frac{2}{\Delta_{\min}^2} \rceil$ and $\gamma_\beta = \lceil \frac{8}{\Delta_{\min}^2} \rceil$. Then, the expected regret of the E³ and E³-TS policies with computation cost C is,

$$\tilde{\mathcal{R}}_{E^3}(T) \leq N\Delta_{\max}\gamma \log(T) + NC\log(T) + 8N\Delta_{\max} \quad (10)$$

$$\tilde{\mathcal{R}}_{E^3-TS}(T) \leq N\Delta_{\max}\gamma_\beta \log(T) + NC\log(T) + 16N\Delta_{\max} \quad (11)$$

(ii) If Δ_{min} is not known, choose $\gamma = \gamma_t$, where $\{\gamma_t\}$ is a positive sequence such that $\gamma_t \to \infty$ as $t \to \infty$. Then,

$$\tilde{\mathcal{R}}(T) \leq N\Delta_{\max}\gamma_T \log(T) + NC\log(T) + N\Delta_{\max}B \quad (12)$$

where $\tilde{\mathcal{R}}(T) = \max\{\tilde{\mathcal{R}}_{E^3}(T), \tilde{\mathcal{R}}_{E^3-TS}(T)\}$ and B is a constant independent of T. In particular, for $\gamma_t = \log^\delta t, \delta \in (0, 1)$,

$$\tilde{\mathcal{R}}(T) \leq N\Delta_{\max}\log^{(1+\delta)}(T) + NC\log(T) + N\Delta_{\max}B(\delta)$$

where $B(\delta) = 2^{l(\delta)}, l(\delta) = (\Delta_{\min}^2/4)^{-1/\delta}$.

Proof is given in Appendix A.

Remark 1. (i) For the sake of clarity, we will assume that γ_t changes at the beginning of every exploration phase. (ii)

Part 1 of the above theorem assumes the knowledge $\Delta_{\min}$ in order to define γ. In fact we only need to know a lower bound on $\Delta_{\min}$. If $\Delta_{LB} \leq \Delta_{\min}$, we can fix $\gamma = \lceil \frac{2}{\Delta_{LB}^2} \rceil$. It is straightforward to show that, with a slight modification of the proof, the theorem still holds. Obviously, a tighter lower bound on $\Delta_{\min}$ results in a tighter bound on the regret.

Although the bounds of E³ and E³-TS are poorer than UCB₁ and TS, they lend themselves to easy decentralization and can be extended to multiplayer bandit problems with minimal effort. Performances of all single player algorithms are compared in Section V-A.

B. Multiplayer policies: dE³ and dE³-TS

In this section, we present multiplayer generalizations of the E³ and E³-TS policies that were described in the previous section. They are detailed in Algorithms 3 and 4, respectively. As in the single player policies, they are divided into exploration and exploitation phases.

Exploration phase: During exploration phases, players take turns to explore arms in a round-robin fashion. This setup is illustrated in Figure 1, where each subdivision of an exploration phase represents a player's turn.

At the end of an exploration phase, the players update their index values (either $g_{i,j}$, or $\theta_{i,j}$) to the other players via a protocol for which a certain set of time-slots are set aside. For example, each player in turn could choose arm 1 for bit '0' and arm 2 for '1' to convey a certain number of bits to all the other players. These could be "bids" that a distributed bipartite matching algorithm needs. All this, of course, comes at a cost and contributes to regret.

This communication and the distributed bipartite matching process is compressed into line 5 in Algorithm 3 and line 8 in Algorithm 4 as a call to dBM.

Distributed bipartite matching (dBM): Let $g(t)$ $(g_{i,j}(t), 1 \leq i \leq M, 1 \leq j \leq N)$ denote a vector of indices. In both algorithms, dBM_ϵ $(g(t))$ refers to an ϵ-optimal distributed bipartite matching algorithm, such as Bertsekas' auction algorithm [25], that yields a matching $k^*(t) = (k_1^*(t), \ldots, k_M^*(t)) \in \mathcal{P}(N)$ such that $\sum_{i=1}^{M} g_{i,k_i^*(t)}(t) \geq \sum_{i=1}^{M} g_{i,k_i}(t)) - \epsilon, \forall \mathbf{k} \in \mathcal{P}(N), \mathbf{k} \neq \mathbf{k}^*$. Players are made aware of their allocations at the end of the matching process. The details of dBM implementation is described in Section IV-D

Exploitation phase: In this phase, players stick to the allocation given to them at the end of the distributed bipartite matching process. No index computation is carried out in this phase. The length of the exploitation phase doubles in each successive epoch.

Fig. 1: Figure showing exploration and exploitation phases of the dE³ policy.

Algorithm 3 : $\mathtt{dE^3}$

1: **Initialization:** Set $t = N\gamma$ and $l = 1$.
2: **while** $(t \leq T)$ **do**
3:    **Exploration Phase:** Each player $i, 1 \leq i \leq M$, plays each arm $j, 1 \leq j \leq N$, γ number of times;
4:    Update the index $g_{i,j}(l) = \overline{X}_{i,j}(l)$;
5:    Participate in the $\mathtt{dBM}_\epsilon(g(l))$ algorithm to obtain a match $k^*(l)$;
6:    **Exploitation Phase:** Each player i plays arm $k_i^*(l)$ for 2^l time slots;
7:    $t \leftarrow t + MN\gamma + 2^l, l \leftarrow l + 1$;
8: **end while**

Algorithm 4 : $\mathtt{dE^3}$-TS

1: **Initialization:** Set $t = N\gamma$ and $l = 1$. For each arm $j = 1, 2, ..., N$ and player $i = 1, ..., M$, set $S_{i,j} = 0$, $F_{i,j} = 0$;
2: **while** $(t \leq T)$ **do**
3:    **Exploration Phase:** Each player $i, 1 \leq i \leq M$, plays each arm $j, 1 \leq j \leq N$, γ number of times;
4:    For each play of each arm j, store reward as $\tilde{r}_{i,j}(t)$;
5:    Perform a Bernoulli trial with success probability $\tilde{r}_{i,j}(t)$ and observe output $r_{ij}(t)$;
6:    If $r_{ij}(t) = 1$, set $S_{i,j} = S_{i,j} + 1$, else $F_{i,j} = F_{i,j} + 1$.
7:    Sample $\theta_{i,j}$ from $Beta(S_{i,j} + 1, F_{i,j} + 1)$ distribution.
8:    Participate in the $\mathtt{dBM}_\epsilon(\theta)$ algorithm to obtain a match $k^*(l)$;
9:    **Exploitation Phase:** Each player i plays arm $k_i^*(l)$ for 2^l time slots;
10:    $t = t + MN\gamma + 2^l, l = l + 1$;
11: **end while**

C. Regret analysis

In both these algorithms, the total regret can be thought as the sum of three different regret terms. The time slots spent in exploration are considered to contribute to regret as the first term, $\tilde{\mathcal{R}}^O(T)$. At the end of every exploration phase, a bipartite matching algorithm is run and each run adds cost C to the second term of regret $\tilde{\mathcal{R}}^C(T)$. The cost C depends on two parameters: (a) the precision of the bipartite matching algorithm $\epsilon_1 > 0$, and (b) the precision of the index representation $\epsilon_2 > 0$. A bipartite matching algorithm has an ϵ_1-precision if it gives an ϵ_1-optimal matching. This would happen, for example, when such an algorithm is run only for a finite number of rounds. The index has an ϵ_2-precision if any two indices are not distinguishable if they are closer than ϵ_2. This can happen, for instance, when indices must be communicated to other players with a finite number of bits. Thus, the cost C is a function of ϵ_1 and ϵ_2, and can be denoted as $C(\epsilon_1, \epsilon_2)$, with $C(\epsilon_1, \epsilon_2) \to \infty$ as ϵ_1 or $\epsilon_2 \to 0$. Since, ϵ_1 and ϵ_2 are the parameters that are fixed *a priori*, we consider $\epsilon = \min(\epsilon_1, \epsilon_2)$ to specify both precisions. We shall denote this computation and communication cost by $C(\epsilon)$ as different communication methods and implementations of distributed bipartitie matching will give different costs.

The third term in the regret expression, $\tilde{\mathcal{R}}^I(T)$, comes from non-optimal matchings in the exploitation phase, i.e., if the matching $\mathbf{k}^*(l)$ is not the optimal matching $\mathbf{k}^{**}$. Thus, we have the total expected regret of the $\mathtt{dE^3}$ and $\mathtt{dE^3}$-TS policies to be given by,

$$\tilde{\mathcal{R}}(T) = \tilde{\mathcal{R}}^O(T) + \tilde{\mathcal{R}}^I(T) + \tilde{\mathcal{R}}^C(T). \tag{13}$$

As before, $\tilde{\mathcal{R}}$ will be used to denote regret for either policy when both are within context. We now prove performance bounds for $\mathtt{dE^3}$ and $\mathtt{dE^3}$-TS policies in the main result of this section.

Theorem 2. *(1) Let $\epsilon > 0$ be the precision of the bipartite matching algorithm and the precision of the index representation. If $\Delta_{\min}$ is known, choose ϵ such that $0 < \epsilon < \Delta_{\min}/(M + 1)$, set $\gamma = \lceil 2M^2/(\Delta_{\min} - (M + 1)\epsilon)^2 \rceil$ and $\gamma_\beta = \lceil 8M^2/(\Delta_{\min} - (M + 1)\epsilon)^2 \rceil$. Then, the expected regrets of the $\mathtt{dE^3}$ and $\mathtt{dE^3}$-TS policies are,*

$$\tilde{\mathcal{R}}_{\mathtt{dE^3}}(T) \leq MN\Delta_{\max}\gamma \log(T) + MN\, C(\epsilon) \log(T) + 8MN\Delta_{\max}, \tag{14}$$

$$\tilde{\mathcal{R}}_{\mathtt{E^3-TS}}(T) \leq MN\Delta_{\max}\gamma_\beta \log(T) + MN\, C(\epsilon) \log(T) + 16MN\Delta_{\max}. \tag{15}$$

Note that, in the above expressions, ϵ is a chosen constant. Thus, $\tilde{\mathcal{R}}(T) = O(\log(T))$ for both policies.
(ii) If $\Delta_{\min}$ is not known, choose $\gamma = \gamma_t$, where $\{\gamma_t\}$ is a positive sequence such that $\gamma_t \to \infty$ as $t \to \infty$. Also choose $\epsilon = \epsilon_t$, where $\{\epsilon_t\}$ is a positive sequence such that $\epsilon_t \to 0$ as $t \to \infty$. Then,

$$\tilde{\mathcal{R}}_{\mathtt{d}}(T) \leq MN\Delta_{\max}\gamma_T \log(T) + MNC(\epsilon_T) \log(T) + MNB \tag{16}$$

where $\tilde{\mathcal{R}}_{\mathtt{d}}(T) = \max\{\tilde{\mathcal{R}}_{\mathtt{dE^3}}(T), \tilde{\mathcal{R}}_{\mathtt{dE^3-TS}}(T)\}$ and B is a constant independent of T. In particular, for $C(\epsilon) = \epsilon^{-1}$, choose $\gamma_t = \log^\delta t, \epsilon_t = \log^{-\delta} t, \delta \in (0, 1)$, and we get

$$\tilde{\mathcal{R}}_{\mathtt{d}}(T) \leq MN\Delta_{\max}\log^{(1+\delta)}(T) + MN\log^{(1+\delta)}(T) + MNB(\delta) \tag{17}$$

where $B(\delta) = b_0 2^{l(\delta)}, l(\delta) = (\Delta_{\min}^2/4)^{-1/\delta}$ and b_0 is a constant independent of δ.

Proof is given in Appendix B.

D. Distributed Bipartite Matching: Algorithm and Implementation

Both $\mathtt{dE^3}$ algorithm and $\mathtt{dE^3}$-TS algorithm use the distributed bipartite matching algorithm as a subroutine. In Section IV-B we have given an abstract description of this distributed bipartite matching algorithm. We now present one such algorithm, namely, Bertsekas' auction algorithm [21], and its distributed implementation. We note that the presented algorithm is not the only one that can be used. Both $\mathtt{dE^3}$ algorithm and $\mathtt{dE^3}$-TS algorithm will work with a distributed implementation of any bipartite matching algorithm, e.g. algorithms given in [22].

Consider a bipartite graph with M players on one side, and N arms on the other, and $M \leq N$. Each player i has a value $\mu_{i,j}$ for each arm j. Each player knows only his own values. Let us denote by k^{**}, a matching that maximizes the

matching surplus $\sum_{i,j} \mu_{i,j} x_{i,j}$, where the variable $x_{i,j}$ is 1 if i is matched with j, and 0 otherwise. Note that $\sum_i x_{i,j} \leq 1, \forall j$, and $\sum_j x_{i,j} \leq 1, \forall i$. Our goal is to find an ϵ-optimal matching. We call any matching k^* to be ϵ-optimal if $\sum_i \mu_{i,k^{**}(i)} - \sum_i \mu_{i,k^*(i)} \leq \epsilon$.

Algorithm 5 : dBM_ϵ (Bertsekas Auction Algorithm)

1: All players i initialize prices $p_j = 0, \forall$ channels j;
2: **while** (prices change) **do**
3:      Player i communicates his preferred arm j_i^* and bid $b_i = \max_j(\mu_{ij} - p_j) - \text{second.max}_j(\mu_{ij} - p_j) + \frac{\epsilon}{M}$ to all other players.
4:      Each player determines on his own if he is the *winner* i_j^* on arm j;
5:      All players set prices $p_j = \mu_{i_j^*,j}$;
6: **end while**

Here, second.max_j is the second highest maximum over all j. The *best* arm for a player i is arm $j_i^* = \arg\max_j(\mu_{i,j} - p_j)$. The *winner* i_j^* on an arm j is the one with the highest bid.

The following lemma in [21] establishes that Bertsekas' auction algorithm will find the ϵ-optimal matching in a finite number of steps.

Lemma 1. *[21] Given $\epsilon > 0$, Algorithm 5 with rewards $\mu_{i,j}$, for player i playing the jth arm, converges to a matching k^* such that $\sum_i \mu_{i,k^{**}(i)} - \sum_i \mu_{i,k^*(i)} \leq \epsilon$ where k^{**} is an optimal matching. Furthermore, this convergence occurs in less than $(M^2 \max_{i,j}\{\mu_{i,j}\})/\epsilon$ iterations.*

Our only assumption here is going to be that each user can observe a channel, and determine if there was a successful transmission on it, a collision, or no transmission, in a given time slot. This consists of J rounds. In each round, users transmit in a round robin fashion, where she can signal her channel preferences using $\lceil \log M \rceil$ bits and bid values (difference of top two indices) using $\lceil \log 1/\epsilon_1 \rceil$ bits. The number of rounds J is chosen so that the dBM algorithm (based on Algorithm 5) returns an ϵ_2-optimal matching. More details on this implementation is given in [17].

V. SIMULATIONS

We conducted extensive simulations comparing the performances of the proposed policies with prior work. The results are presented in the respective sections below.

A. Single player bandit policies

For the single player setting, we considered a four-armed bandit problem with rewards for arms drawn independently from Bernoulli distributions with means 0.1, 0.5, 0.6, 0.9. The scenario was simulated over a fixed time horizon $T = 2,000,000$ timeslots and the performance of the proposed single-player policies was evaluated. The performance of each policy was averaged over 10 sample runs and the results presented here. Different true means and distributions were also considered and they gave similar rankings for the algorithms. In the interest of space, those scenarios are not presented.

In Figure 2, the single player policies proposed in this paper, E^3 and E^3-TS, are compared with the benchmark UCB_1 policy.

$\Delta_{\min}$ is assumed to be known (0.1) and, consequently, γ is fixed. The bound for E^3-TS is also shown with the dashed line. It can be observed that, although, all three policies have logarithmic order of regret performance in time, the new E^3 and E^3-TS policies perform slightly worse than the UCB_1 policy. This is attributable to the deterministic exploration phase length which must take into account the worst-case scenario. However, as we shall see in the next section, this gives us a significant performance advantage in the multiplayer setting.
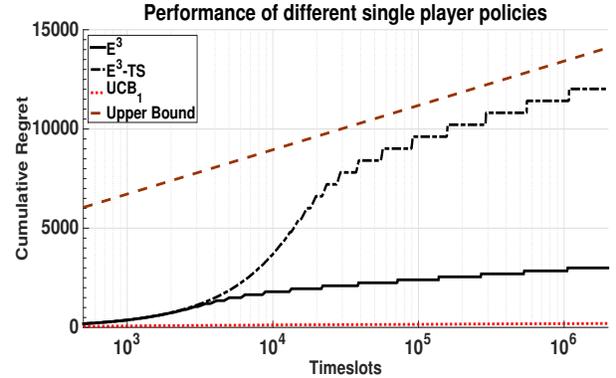

Fig. 2: Figure showing growth of cumulative regret of the E^3, E^3-TS and UCB_1 algorithms for a four-armed single-player bandit problem with true means $[0.1, 0.5, 0.6, 0.9]$ (no computation cost), with time plotted on log scale.

Note that in Figure 2, computation cost is assumed to be zero. If computation cost were included, E^3 and E^3-TS would retain their logarithmic regret performance. However, the cumulative regret of UCB_1 would grow linearly, just as with TS [17].

B. Multiplayer bandit policies

We now present the empirical performance of the proposed dE^3 and dE^3-TS policies. We consider a three-player, three-armed bandit setting. Rewards for each arm are generated independently from a Bernoulli distribution with means 0.2, 0.25, 0.3 for player 1, 0.4, 0.6, 0.5 for player 2 and 0.7, 0.9, 0.8 for player 3. A time horizon spanning 20 epochs (about 2 million timeslots) was considered. $\epsilon = 0.001$ was used as the tolerance for the bipartite matching algorithm, which was done using dBM_ϵ, a distributed implementation of Bertsekas' auction algorithm. The performance of each policy was averaged over 10 sample runs. γ was set equal to 100 for dE^3 and 400 for dE^3-TS (see analysis for the reason for differing γ's). A fixed per unit cost each time the distributed bipartite matching algorithm dBM is run, is included in the setting to model communication cost in the decentralized setting.

The plot of the growth of cumulative regret with time of dE^3, dE^3-TS and dUCB_4 is shown in Figure 3. We can see that the logarithmic regret performance of dE^3 and dE^3-TS clearly outperforms the $\log^2 T$-regret performance of our earlier dUCB_4 policy [17]. The dashed line curve is the theoretical upper bound on the performance of dE^3-TS.
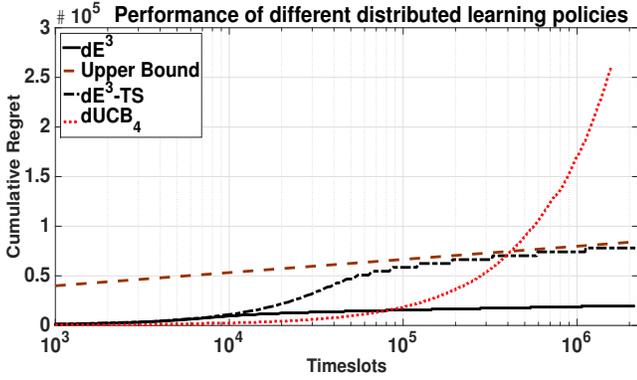
Fig. 3: Figure showing growth of cumulative regret of the dE^3 and dE^3-TS algorithms for a three-player, three-armed bandit setting with true means $[0.20.250.3; 0.40.60.5; 0.70.90.8]$ (communication cost included), with time plotted on log scale.

VI. Conclusion

We designed two closely related single player and multiplayer bandit policies that achieve logarithmic or near-logarithmic regret performance depending on the assumptions of the model. Both policies have deterministic exploration and exploitation phases, which make them well-suited to decentralization for use in the multiplayer setting.

Performances of these policies were compared to prior work in the literature. They were shown to outperform previous policies for multiplayer bandits, but not for the single player model due to the deterministic phases of these new policies. While we have approached logarithmic regret performance under certain assumptions in the multiplayer model, the question of whether a policy under truly general conditions can achieve fully logarithmic regret remains open.

References

[1] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3/4, pp. 285–294, 1933.
[2] H. Robbins, "Some aspects of the sequential design of experiments," in *Herbert Robbins Selected Papers*. Springer, 1985, pp. 169–177.
[3] T. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Adv. Appll. Math.*, vol. 6, no. 1, pp. 4–22, 1985.
[4] V. Anantharam, P. Varaiya, and J. Walrand, "Asymptotically efficient allocation rules for the multi-armed bandit problem with multiple plays - part i: i.i.d. rewards," *IEEE Trans. Autom. Control*, vol. 32, no. 11, pp. 968-975, November, 1987.
[5] R. Agrawal, "Sample mean based index policies with ($O(\log n)$) regret for the multi-armed bandit problem," *Adv. Appl. Probability*, Vol. 27, No. 4, pp. 1054-1078, 1995.
[6] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multi-armed bandit problem," *Machine learning*, vol. 47, no. 2, pp. 235–256, 2002.
[7] O. Chapelle and L. Li, "An empirical evaluation of Thompson Sampling," vol. 24, 2012.
[8] S. Agrawal and N. Goyal, "Analysis of Thompson Sampling for the multi-armed bandit problem," in *Proc. COLT*, 2012.
[9] E. Kaufmann, O. Cappé, and A. Garivier, "On Bayesian upper confidence bounds for bandit problems." in *AISTATS*, 2012, pp. 592–600.
[10] P. Rusmevichientong and J. N. Tsitsiklis, "Linearly parameterized bandits," *Math. Oper. Res.*, vol. 35, no. 2, pp. 395–411, May 2010.
[11] S. Vakili, K. Liu, and Q. Zhao, "Deterministic sequencing of exploration and exploitation for multi-armed bandit problems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 59, no. 3, pp. 1902–1916, 2013.
[12] H. Liu, K. Liu, and Q. Zhao, "Learning in a changing world: Restless multiarmed bandit with unknown dynamics," *IEEE Transactions on Information Theory*, vol. 59, no. 3, pp. 1902–1916, 2013.
[13] E. Even-dar, S. Mannor, and Y. Mansour, "PAC bounds for multi-armed bandit and Markov decision processes," in *Fifteenth Annual Conference on Computational Learning Theory (COLT)*, 2002, pp. 255–270.
[14] A. Anandkumar, N. Michael, A. K. Tang, and A. Swami, "Distributed algorithms for learning and cognitive medium access with logarithmic regret," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 4, pp. 731–745, April 2011.
[15] K. Liu and Q. Zhao, "Distributed learning in multi-armed bandit with multiple players," *Signal Processing, IEEE Transactions on*, vol. 58, no. 11, pp. 5667–5681, Nov 2010.
[16] L. Lai, H. Jiang, and H. V. Poor, "Medium access in cognitive radio networks: A competitive multi-armed bandit framework," in *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, Oct 2008, pp. 98–102.
[17] D. Kalathil, N. Nayyar, and R. Jain, "Decentralized learning for multi-player multi-armed bandits," *IEEE Trans. Inf. Theory*, vol. 60, no. 4, pp. 2331–2345, 2014.
[18] O. Avner and S. Mannor, "Multi-user lax communications: a multi-armed bandit approach," *arXiv preprint arXiv:1504.08167*, 2015.
[19] P. Landgren, V. Srivastava, and N. E. Leonard, "On distributed cooperative decision-making in multiarmed bandits," *arXiv preprint arXiv:1512.06888*, 2015.
[20] C. Tekin and M. Liu, "Online learning of rested and restless bandits," *IEEE Trans. Inf. Theory*, vol. 58, no. 8, pp. 5588–5611, 2012.
[21] D. P. Bertsekas, "Auction algorithms for network flow problems: A tutorial introduction," *Comput. Optim. Appl.*, vol. 1, no. 1, pp. 7–66, 1992.
[22] M. M. Zavlanos, L. Spesivtsev, and G. J. Pappas, "A distributed auction algorithm for the assignment problem," in *Proc. IEEE Conf. Decision and Control (CDC)*, December, 2008, pp. 1212–1217.
[23] J. C. Gittins, "Bandit processes and dynamic allocation indices," *J. R. Statist. Soc. B*, vol. 41, pp. 148–164, 1979.
[24] E. Kaufmann, N. Korda, and R. Munos, "Thompson Sampling: An asymptotically optimal finite-time analysis," in *Proceedings of the 23rd International Conference on Algorithmic Learning Theory*, ser. ALT'12. Springer-Verlag, 2012, pp. 199–213.
[25] D. P. Bertsekas, "The auction algorithm: A distributed relaxation method for the assignment problem," *Ann. Oper. Res.*, vol. 14, no. 1, pp. 105–123, 1988.
[26] S. Boucheron, G. Lugosi, and P. Massart, *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

Appendix A
Proof of Theorem 1

In the following proof, the subscript β for γ_β will be omitted where the context refers to both E^3 and E^3-TS together. Also, $\mathcal{R}$ will be used to denote regret for either policy when both are within context.

A. $\Delta_{\min}$ is known:

We denote the expected regrets incurred in the exploration phases with $\tilde{\mathcal{R}}^O(T)$, exploitation phases with $\tilde{\mathcal{R}}^I(T)$ and due to computation with $\tilde{\mathcal{R}}^C(T)$. Then,

$$\tilde{\mathcal{R}}(T) = \tilde{\mathcal{R}}^O(T) + \tilde{\mathcal{R}}^C(T) + \tilde{\mathcal{R}}^I(T), \quad (18)$$

for both E^3 and E^3-TS policies.

Let T be in the l_0-th exploitation epoch. By construction, $T \geq N\gamma l_0 + 2^{l_0} - 2$. Thus, $\log T \geq l_0$ and,

$$\tilde{\mathcal{R}}^O(T) = \gamma l_0 \sum_{j=2}^N \Delta_j \leq N\gamma l_0 \Delta_{\max} \leq N\gamma \log(T)\Delta_{\max},$$

$$(19)$$

where $\Delta_j = \mu_1 - \mu_j$. Also, using the definition of computation cost,

$$\tilde{\mathcal{R}}^C(T) = NCl_0 \leq NC \log(T). \quad (20)$$

Now, $\tilde{\mathcal{R}}^I(T) = \mathbb{E}\left[\sum_{j=2}^N \Delta_j \tilde{n}_j(T)\right]$ where $\tilde{n}_j(T)$ is the number of times arm j has been played during the exploitation phases. For E^3,

$$\tilde{n}_j(T) = \sum_{l=1}^{l_0} 2^l \mathbb{I}\{\overline{X}_j(l) > \max_{1 \le i \le N} \overline{X}_i(l)\}$$

$$\le \sum_{l=1}^{l_0} 2^l \mathbb{I}\{\overline{X}_j(l) > \overline{X}_1(l)\} \qquad (21)$$

Similarly, for E^3-TS, $\tilde{n}_j(T) \le \sum_{l=1}^{l_0} 2^l \mathbb{I}\{\theta_j(l) > \theta_1(l)\}$. Thus,

$$\tilde{\mathcal{R}}_{\mathrm{E}^3}^I(T) \le \Delta_{\max} \sum_{l=1}^{l_0} \sum_{j=2}^N 2^l \mathbb{P}(\overline{X}_1(l) < \overline{X}_j(l)), \qquad (22)$$

and, $\tilde{\mathcal{R}}_{\mathrm{E}^3-\mathrm{TS}}^I(T) \le \Delta_{\max} \sum_{l=1}^{l_0} \sum_{j=2}^N 2^l \mathbb{P}(\theta_1(l) < \theta_j(l)). \quad (23)$

The following two lemmas bound the event probabilities above for the E^3 and E^3-TS policies.

Lemma 2. *For* E^3, *with* $\gamma = \lceil \frac{2}{\Delta_{\min}^2} \rceil$,

$$\mathbb{P}(\overline{X}_1(l) < \overline{X}_j(l)) \le 2e^{-l}, \ \forall j > 1. \qquad (24)$$

Proof. The event $\{\overline{X}_1(l) < \overline{X}_j(l)\}$ implies at least one of the following events:

$A_j := \{\overline{X}_j(l) - \mu_j > \Delta_j/2\}, B_j := \{\overline{X}_1(l) - \mu_1 < -\Delta_j/2\}.$

Using the Chernoff-Hoeffding bound and choosing $\gamma = \lceil \frac{2}{\Delta_{min}^2} \rceil$, we get,

$$\mathbb{P}(A_j) \le e^{-2l\gamma \Delta_j^2/4} \le e^{-l}, \text{ and similarly, } \mathbb{P}(B_j) \le e^{-l}. \qquad (25)$$

By the union bound, we get $\mathbb{P}(\overline{X}_1(l) < \overline{X}_j(l)) \le 2e^{-l}$.
$\square$

Lemma 3. *For* E^3-TS, *with* $\gamma_\beta = \lceil \frac{8}{\Delta_{\min}^2} \rceil$,

$$\mathbb{P}(\theta_1(l) < \theta_j(l)) \le 4e^{-l}, \ \forall j > 1. \qquad (26)$$

Proof. Without loss of generality, we will assume the underlying reward distributions of the arms to have a Bernoulli distribution to simplify the analysis. This eliminates the need for line 5 in the E^3-TS policy illustrated in Algorithm 2. However, this assumption can be relaxed without any change to the results.

As in Lemma 2, the event $\{\theta_1(l) < \theta_j(l)\}$ implies at least one of the events:

$A_j := \{\theta_j(l) - \mu_j > \Delta_j/2\}, B_j := \{\theta_1(l) - \mu_1 < -\Delta_j/2\}. \qquad (27)$

Let $m_j(l)$ denote the number of plays of arm j during the exploration phases after the l-th exploration epoch, and let $s_j(l)$ be the number of successes ($r = 1$) in these plays. Then, $\theta_j(l)$ is sampled from a $\beta(s_j(l)+1, m_j(l) - s_j(l)+1)$ distribution.

Additionally, let $A(l)$ denote the event $\{\frac{s_j(l)}{m_j(l)} < \mu_j + \frac{\Delta_j}{4}\}$. Then,

$$\mathbb{P}(\theta_j(l) \ge \mu_j + \frac{\Delta_j}{2}) \le \mathbb{P}(\overline{A}(l)) + \mathbb{P}(\theta_j(l) \ge \mu_j + \frac{\Delta_j}{2}, A(l)). \qquad (28)$$

The first term in the expression,

$$\mathbb{P}(\overline{A}(l)) = \mathbb{P}\left(\frac{s_j(l)}{m_j(l)} \ge \mu_j + \frac{\Delta_j}{4}\right) \le \exp\left(\frac{-2\gamma_\beta l \Delta_j^2}{16}\right),$$

where the last inequality comes from the Chernoff-Hoeffding inequality and by noting that $\frac{s_j(l)}{m_j(l)}$ is a random variable with mean μ_j. Also, $m_j(l) = \gamma_\beta l$.

The second term,

$$\mathbb{P}(\theta_j(l) \ge \mu_j + \frac{\Delta_j}{2}, A(l))$$

$$= \mathbb{P}\left(\theta_j(l) \ge \mu_j + \frac{\Delta}{2}, \frac{s_j(l)}{m_j(l)} < \mu_j + \frac{\Delta_j}{4}\right)$$

$$\le \mathbb{P}\left(\theta_j(l) > \frac{s_j(l)}{m_j(l)} + \frac{\Delta_j}{4}\right)$$

$$= \mathbb{P}\left(\beta(s_j(l)+1, m_j(l) - s_j(l)+1) > \frac{s_j(l)}{m_j(l)} + \frac{\Delta_j}{4}\right)$$

$$= \mathbb{E}\left[F^B_{m_j(l)+1, \frac{s_j(l)}{m_j(l)} + \frac{\Delta_j}{4}}(s_j(l))\right]$$

$$\le \mathbb{E}\left[F^B_{m_j(l), \frac{s_j(l)}{m_j(l)} + \frac{\Delta_j}{4}}(s_j(l))\right]. \qquad (29)$$

Here, $F^B_{n,p}(x)$ is the cdf of the $\mathtt{binomial(n,p)}$ distribution. The equality in the second-to-last line comes from the fact that $F^\beta_{a,b}(x) = 1 - F^B_{a+b-1,x}(a-1)$, where $F^\beta_{a,b}(x)$ is the cdf of the $\beta(\mathtt{a,b})$ distribution [8]. The inequality on the last line is a standard inequality for binomial distributions.

But, by the Chernoff-Hoeffding inequality, it can be seen that $F^B_{n,p}(np - n\delta) \le \exp(-2n\delta^2)$. Thus,

$$\mathbb{P}(\theta_j(l) \ge \mu_j + \frac{\Delta}{2}, A(l)) \le \exp\left(\frac{-2\gamma_\beta l \Delta_j^2}{16}\right) \qquad (30)$$

Setting $\gamma_\beta := \lceil \frac{8}{\Delta_{\min}^2} \rceil$ in (29) and (30), we get,

$$\mathbb{P}(\theta_j(l) \ge \mu_j + \frac{\Delta_j}{2}) \le 2e^{-l}. \qquad (31)$$

Similarly, $\mathbb{P}(\theta_1(l) \le \mu_1 - \frac{\Delta_j}{2}) \le 2e^{-l}$, and the claim of the lemma follows from the union bound.
$\square$

Continuing with the proof of Theorem 1, thus,

$$\tilde{\mathcal{R}}_{\mathrm{E}^3}^I(T) \le \Delta_{\max} \sum_{l=1}^{l_0} \sum_{j=2}^N 2^l 2e^{-l} \le 2N\Delta_{\max} \sum_{l=0}^\infty (2/e)^l$$

$$\le 2N\Delta_{\max}/(1 - (2/e)) < 8N\Delta_{\max}, \qquad (32)$$

$$\tilde{\mathcal{R}}_{\mathrm{E}^3-\mathrm{TS}}^I(T) \le \Delta_{\max} \sum_{l=1}^{l_0} \sum_{j=2}^N 2^l 4e^{-l} \le 4N\Delta_{\max} \sum_{l=0}^\infty (2/e)^l$$

$$\le 4N\Delta_{\max}/(1 - (2/e)) < 16N\Delta_{\max}, \qquad (33)$$

Now, combining all the terms, we get

$$\tilde{\mathcal{R}}_{\mathrm{E}^3}(T) \le N\gamma \log T \Delta_{\max} + C\log(T) + 8N\Delta_{\max}. \qquad (34)$$

$$\tilde{\mathcal{R}}_{\mathrm{E}^3-\mathrm{TS}}(T) \le N\gamma_\beta \log T \Delta_{\max} + C\log(T) + 16N\Delta_{\max}. \qquad (35)$$

B. $\Delta_{\min}$ *is unknown:*

Suppose t_l be the time t at which lth exploration phase begins. For the clarity of explanation, we assume that γ changes only in the beginning of an exploration phase. So, in the lth exploration phase, each arms is played γ_{t_l} times in a round robin manner.

As in the proof given in the previous subsection, let T be in the l_0th exploitation epoch. By construction, $T \geq N \sum_{l=1}^{l_0} \gamma_{t_l} + 2^{l_0} - 2$. Thus, $\log T \geq l_0$ and,

$$\tilde{\mathcal{R}}^O(T) = \sum_{l=1}^{l_0} \gamma_{t_l} \sum_{j=2}^{N} \Delta_j \leq N\Delta_{\max}\gamma_T l_0 \leq N\Delta_{\max}\gamma_T \log(T).$$
(36)

The second inequality is from the fact that γ_t is a monotone increasing sequence.

The computation cost is same as before, i.e.,

$$\tilde{\mathcal{R}}^C(T) = NCl_0 \leq NC \log(T).$$

Using (25), $\tilde{\mathcal{R}}^I_{E^3}(T) \leq \Delta_{\max} \sum_{l=1}^{l_0} \sum_{j=2}^{N} 2^l \mathbb{P}(\overline{X}_1(l) < \overline{X}_j(l))$

$$\leq N\Delta_{\max} \sum_{l=1}^{\infty} 2^l e^{-b_1 \sum_{k=1}^{l} \gamma_{t_k}}$$

where $b_1 = \Delta_{\min}^2/2$. Since $\gamma_t \to \infty$ monotonically (and $\gamma_{t_k} \geq 1$), there exists an l' such that $b_1 \sum_{k=1}^{l} \gamma_{t_k} \geq l, \forall l > l'$. Then,

$$\tilde{\mathcal{R}}^I_{E^3}(T) \leq N\Delta_{\max} \left(\sum_{l=1}^{l'-1} 2^l e^{-b_1 \sum_{k=1}^{l} \gamma_{t_k}} + \sum_{l=l'}^{\infty} (2/e)^l \right)$$

$$\leq N\Delta_{\max} B$$

where B is a finite constant, independent of T.

When $\gamma_{t_l} = \log^\delta t_l$ for $\delta \in (0,1)$, it is easy to see that $\gamma_{t_l} \geq l^\delta, \forall l$. Then, $\sum_{k=1}^{l} \gamma_{t_k} \geq \sum_{k=1}^{l} k^\delta \geq \int_{x=1}^{l+1}(x-1)^\delta dx \geq 0.5 l^{(1+\delta)}$. From this, $l' = (2/b_1)^{1/\delta}$. Then, we can get $B = B(\delta) = 2^{l'}$.

APPENDIX B
PROOF OF THEOREM 2

We first show that if $\Delta_{\min}$ is known, we can choose an $\epsilon < \Delta_{\min}/(M+1)$, such that dE^3 and dE^3-TS algorithms achieve a logarithmic regret growth with T. If $\Delta_{\min}$ is not known, we can pick a positive monotone sequence $\{\epsilon_t\}$ such that $\epsilon_t \to 0$, as $t \to \infty$. In a decentralized bipartite matching algorithm, the precision ϵ will depend on the amount of information exchanged.

The proof will be illustrated here only for the dE^3 policy since the differences between it and the analysis of the dE^3-TS policy are similar to those found in Theorem 1.

Let us denote the optimal bipartite matching with $\mathbf{k}^{**} \in \mathcal{P}(N)$ such that $\mathbf{k}^{**} \in \arg\max_{\mathbf{k} \in \mathcal{P}(N)} \sum_{i=1}^{M} \mu_{i,\mathbf{k}_i}$. Denote $\mu^{**} := \sum_{i=1}^{M} \mu_{i,\mathbf{k}_i^{**}}$, and define $\Delta_{\mathbf{k}} := \mu^{**} - \sum_{i=1}^{M} \mu_{i,\mathbf{k}_i}$, $\mathbf{k} \in \mathcal{P}(N)$.

Let $\Delta_{\min} = \min_{\mathbf{k} \in \mathcal{P}(N), \mathbf{k} \neq \mathbf{k}^{**}} \Delta_{\mathbf{k}}$ and $\Delta_{\max} = \max_{\mathbf{k} \in \mathcal{P}(N)} \Delta_{\mathbf{k}}$. We assume $\Delta_{\min} > 0$.

A. $\Delta_{\min}$ *is known:*

Let T be in the l_0th exploitation epoch. It follows that, $T \geq MN\gamma l_0 + 2^{l_0} - 2$ and, hence, $\log T \geq l_0$. Then,

$$\tilde{\mathcal{R}}^O_{dE^3}(T) = MN\gamma l_0 \Delta_{\max} \leq MN\gamma \log(T)\Delta_{\max}.$$
(37)

Also, by definition,

$$\tilde{\mathcal{R}}^C_{dE^3}(T) = MNC(\epsilon)l_0 \leq MNC(\epsilon) \log(T).$$
(38)

A suboptimal matching occurs in the l-th exploitation epoch if the event $\{\sum_{i=1}^{M} \overline{X}_{i,k_i^{**}}(l) < (M+1)\epsilon + \sum_{i=1}^{M} \overline{X}_{i,k_i^*}(l)\}$ occurs. If each index has an error of at most ϵ, the sum of M terms may introduce an error of at most $M\epsilon$. In addition, the distributed bipartite matching algorithm dBM_ϵ itself yields only an ϵ-optimal matching. This accounts for the term $(M+1)\epsilon$ above.

Clearly, as in the single player case,

$$\tilde{\mathcal{R}}^I_{dE^3}(T) \leq \Delta_{\max} \sum_{l=1}^{l_0} 2^l \mathbb{P}\Big(\sum_{i=1}^{M} \overline{X}_{i,k_i^{**}}(l) < (M+1)\epsilon$$

$$+ \sum_{i=1}^{M} \overline{X}_{i,k_i^*}(l) \Big).$$
(39)

The event $\left\{ \sum_{i=1}^{M} \overline{X}_{i,k_i^{**}}(l) < (M+1)\epsilon + \sum_{i=1}^{M} \overline{X}_{i,k_i^*}(l) \right\}$ implies at least one of the following events

$$A_{i,j} := \{|\overline{X}_{i,j}(l) - \mu_{i,j}| > (\Delta_{\min} - (M+1)\epsilon)/2M\}, \quad (40)$$

for $1 \leq i \leq M, 1 \leq j \leq N$. By the Chernoff-Hoeffding bound, and then using the fact that $\gamma = \lceil 2M^2/(\Delta_{\min} - (M+1)\epsilon)^2 \rceil$,

$$\mathbb{P}(A_{i,j}) \leq 2e^{-2l\gamma(\Delta_{\min} - (M+1)\epsilon)^2/4M^2} \leq 2e^{-l}.$$
(41)

Then, by using the union bound,

$$\tilde{\mathcal{R}}^I_{E^3}(T) \leq \Delta_{\max} \sum_{l=1}^{l_0} 2^l \sum_{i=1}^{M} \sum_{j=1}^{N} \mathbb{P}(A_{i,j})$$

$$\leq \Delta_{\max} 2MN \sum_{l=0}^{\infty} (2/e)^l = 2\Delta_{\max} MN/(1-(2/e))$$

$$< 8MN\Delta_{\max}.$$
(42)

Combining all the terms, we get

$$\tilde{\mathcal{R}}_{dE^3}(T) \leq MN\gamma \log(T)\Delta_{\max} + MNC(\epsilon) \log(T)$$
$$+ 8MN\Delta_{\max}.$$
(43)

In a similar manner,

$$\tilde{\mathcal{R}}_{E^3-TS}(T) \leq MN\gamma_\beta \log(T)\Delta_{\max} + MNC(\epsilon) \log(T)$$
$$+ 16MN\Delta_{\max},$$
(44)

where $\gamma_\beta = \lceil 8M^2/(\Delta_{\min} - (M+1)\epsilon)^2 \rceil$.

B. $\Delta_{\min}$ *is unknown:*

The proof is similar to the proof of the analogous case of Theorem 1, and is omitted.