Decentralized Learning for Multiplayer Multiarmed Bandits

Dileep Kalathil, *Student Member, IEEE*, Naumaan Nayyar, *Student Member, IEEE*, and Rahul Jain, *Member, IEEE*

***Abstract*— We consider the problem of distributed online learning with multiple players in multiarmed bandit (MAB) models. Each player can pick among multiple arms. When a player picks an arm, it gets a reward. We consider both independent identically distributed (i.i.d.) reward model and Markovian reward model. In the i.i.d. model, each arm is modeled as an i.i.d. process with an unknown distribution with an unknown mean. In the Markovian model, each arm is modeled as a finite, irreducible, aperiodic and reversible Markov chain with an unknown probability transition matrix and stationary distribution. The arms give different rewards to different players. If two players pick the same arm, there is a collision, and neither of them get any reward. There is no dedicated control channel for coordination or communication among the players. Any other communication between the users is costly and will add to the regret. We propose an online index-based distributed learning policy called $\mathtt{dUCB_4}$ algorithm that trades off exploration versus exploitation in the right way, and achieves expected regret that grows at most as near-$O(\log^2 T)$. The motivation comes from opportunistic spectrum access by multiple secondary users in cognitive radio networks wherein they must pick among various wireless channels that look different to different users. This is the first distributed learning algorithm for multiplayer MABs with heterogeneous players (that have player-dependent rewards) to the best of our knowledge.**

***Index Terms*— Distributed adaptive control, multi-armed bandit, online learning, multi-agent systems.**

I. Introduction

In [1], Lai and Robbins introduced the classical non-Bayesian multi-armed bandit model. Such models capture the essence of the learning problem that players face in an unknown environment, where the players must not only *explore* to learn but also *exploit* in choosing the best arm. Specifically, suppose a player can choose between N arms. Upon choosing an arm i, it gets a reward from a distribution with density $f(x, \theta_i)$. Time is slotted, and players do not know the distributions (nor any statistics about them). The problem is to find a learning policy that minimizes the expected *regret* over some time horizon T. It was shown by Lai and Robbins [1] that there exists an index-type policy that achieves expected regret that grows asymptotically as $\log T$, and this is order-optimal, i.e., there exists no causal policy that can do better. This was generalized by Anantharam, *et al.* [2] to the case of multiple plays, i.e., when the player can pick multiple arms at the same time. In [3], Agrawal proposed a sample mean based index policy which achieves $\log T$ regret asymptotically. Assuming that the rewards are coming from a distribution of bounded support, Auer, *et al.* [4] proposed a much simpler sample mean based index policy, called $\mathtt{UCB_1}$, which achieves $\log T$ uniformly over time, not only asymptotically. Also, unlike the policy in [3], the index doesn't depend on the specific family of distributions that the rewards come from.

In [5], Anantharam, *et al.* proposed a policy to the case where the arms are modelled as Markovian, not i.i.d. The rewards are assumed to come from a finite, irreducible and aperiodic Markov chain represented by a single parameter probability transition matrix. The state of each arm evolves according to an underlying transition probability matrix when the arm is played and remains frozen when passive. Such problems are called *rested Markovian bandit problems* (where *rested* refers to no state evolution until the arm is played). In [6], Tekin and Liu extended the $\mathtt{UCB_1}$ policy to the case of rested Markovian bandit problems. If some non-trivial bounds on the underlying Markov chains are known a priori, they showed that the policy achieves $\log T$ regret uniformly over time. Also, if no information about the underlying Markov chains is available, the policy can easily be modified to get a *near-*$O(\log T)$ regret asymptotically. The models in which the state of an arm continues to evolve even when it is not played are called *restless Markovian bandit problems*. Restless models are considerably more difficult than the rested models and have been shown to be P-SPACE hard [7]. This is because the optimal policy no longer will be to "play the arm with the highest mean reward". [8] employs a weaker notion of regret (*weak regret*) which compares the reward of a policy to that of a policy which always plays the arm with the highest mean reward. They propose a policy which achieves $\log T$ (weak) regret uniformly over time if certain bounds on the underlying Markov model are known a priori and achieves a near-$O(\log T)$ (weak) regret asymptotically when no such knowledge is available. [9] proposes another simpler policy which achieves the same bounds for weak regret. [10] proposes a policy based on deterministic sequence of exploration and exploitation and achieves the same bounds for weak regret.

Manuscript received April 29, 2012; revised August 27, 2013; accepted December 5, 2013. Date of current version March 13, 2014. This work was supported in part by AFOSR under Grant FA9550-10-1-0307 and in part by NSF CAREER under Award CNS-0954116.

The authors are with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: manisser@usc.edu; nnayyar@usc.edu; rahul.jain@usc.edu).

Communicated by N. Cesa-Bianchi, Associate Editor for Pattern Recognition, Statistical Learning, and Inference.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2014.2302471

In [11], the authors consider the notion of *strong regret* and propose a policy which achieves near-log T (strong) regret for some special cases of the restless model.

Recently, there is an increasing interest in multi-armed bandit models, partly because of opportunistic spectrum access problems. Consider a user who must choose between N wireless channels. Yet, it knows nothing about the channel statistics, i.e., has no idea of how good or bad the channels are, and what rate it may expect to get from each channel. The rates could be learnt by exploring various channels. Thus, these have been formulated as multi-armed bandit problems, and index-type policies have been proposed for choosing spectrum channels. In many scenarios, there are multiple users accessing the channels at the same time. Each of these users must be matched to a different channel. These have been formulated as a *combinatorial multi-armed bandit problem* [12], [13], and it was shown that an "index-matching" algorithm that at each instant determines a matching by solving a sum-index maximization problem achieves $O(\log T)$ regret uniformly over time, and this is indeed order-optimal.

In other settings, the users cannot coordinate, and the problem must be solved in a decentralized manner. Thus, settings where all channels (arms) are identical for all users with i.i.d. rewards have been considered, and index-type policies that can achieve coordination have been proposed that get $O(\log T)$ regret uniformly over time [10], [14]–[16]. A similar result for Markovian reward model with weak regret has been shown by [10], assuming some non-trivial bounds on the underlying Markov chains are known a priori. The regret scales only polynomially in the number of users and channels. Surprisingly, the lack of coordination between the players asymptotically imposes no additional cost or regret.

In this paper, we consider the decentralized multi-armed bandit problem with distinct arms for each player. We consider both the i.i.d. reward model and the *rested* Markovian reward model. All players together must discover the best arms to play as a team. However, since they are all trying to learn at the same time, they may collide when two or more pick the same arm. We propose an index-type policy $\texttt{dUCB}_4$ based on a variation of the $\texttt{UCB}_1$ index. At its heart is a distributed bipartite matching algorithm such as Bertsekas' auction algorithm [17]. This algorithm operates in rounds, and in each round prices for various arms are determined based on bid-values. This imposes communication (and computation) cost on the algorithm that must be accounted for. Nevertheless, we show that when certain non-trivial bounds on the model parameters are known a priori, the $\texttt{dUCB}_4$ algorithm that we introduce achieves (at most) *near-*$O(\log^2 T)$ growth non-asymptotically in expected regret. If no such information about the model parameters are available, $\texttt{dUCB}_4$ algorithm still achieves (at most) *near-*$O(\log^2 T)$ regret asymptotically. A lower bound, however, is not known at this point, and is a work in progress.

The paper is organized as follows. In Section II, we present the model and problem formulation. In section III and IV we present some variations on single player MAB with i.i.d. rewards and Markovian rewards respectively. In section V, we introduce the decentralized MAB problem with i.i.d. rewards. We then extend the results to the decentralized cases with Markovian rewards in section VI. In section VII we present the distributed bipartite matching algorithm which is used in our main algorithm for decentralized MAB.

II. Model and Problem Formulation

A. Arms With i.i.d. Rewards

We consider an N-armed bandit with M players. In a wireless cognitive radio setting [18], each arm could correspond to a channel, and each player to a user who wants to use a channel. Time is slotted, and at each instant each player picks an arm. There is no dedicated control channel for coordination among the players. So, potentially more than one player can pick the same arm at the same instant. We will regard that as a *collision*. Player i playing arm k at time t yields i.i.d. reward $S_{ik}(t)$ with density function $f_{ik}(s)$. We will assume that the rewards are bounded, and without loss of generality lie in $[0, 1]$. Let $\mu_{i,k}$ denote the mean of $S_{ik}(t)$ w.r.t. the pdf $f_{ik}(s)$. We assume that the players have no information about the mean, the distributions or any other statistics about the rewards from various arms other than what they observe while playing. We also assume that each player can only observe the rewards that they get. When there is a collision, we will assume that all players that choose the arm on which there is a collision get zero reward. This could be relaxed where the players share the reward in some manner though the results do not change appreciably.

Let $X_{ij}(t)$ be the reward that player i gets from arm j at time t. Thus, if player i plays arm k at time t (and there is no collision), $X_{ik}(t) = S_{ik}(t)$, and $X_{ij}(t) = 0,\ j \neq k$. Denote the action of player i at time t by $a_i(t) \in \mathcal{A} := \{1, \ldots, N\}$. Then, the *history* seen by player i at time t is $\mathcal{H}_i(t) = \{(a_i(1), X_{i,a_i(1)}(1)), \ldots, (a_i(t), X_{i,a_i(t)}(t))\}$ 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 given the history seen by the player. Let $\mathcal{P}(N)$ be the set of vectors such that

$$\mathcal{P}(N) := \{\mathbf{a} = (a_1, \ldots, a_M) : a_i \in \mathcal{A}, a_i \neq a_j, \text{for} \quad i \neq j\}.$$

The players have a *team objective*: namely over a time horizon T, 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. If the parameters $\mu_{i,j}$ are known, this could easily be achieved by picking a bipartite matching

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

i.e., the optimal bipartite matching with expected reward from each match. Note that this may not be unique. Since the expected rewards, $\mu_{i,j}$, are unknown, the players must pick learning policies that minimize the *expected regret*, defined for policies $\alpha = (\alpha_i, 1 \leq i \leq M)$ as

$$\mathcal{R}_\alpha(T) = T \sum_i \mu_{i,k_i^{**}} - \mathbb{E}_\alpha \left[\sum_{t=1}^{T} \sum_{i=1}^{M} X_{i,\alpha_i(t)}(t) \right]. \tag{2}$$

Our goal is to find a decentralized algorithm that players can use such that together they minimize the expected regret.

B. Arms With Markovian Rewards

Here we follow the model formulation introduced in the previous subsection, with the exception that the rewards are now considered Markovian. The reward that player i gets from arm j (when there is no collision) X_{ij}, is modelled as an irreducible, aperiodic, reversible Markov chain on a finite state space $\mathcal{X}^{i,j}$ and represented by a transition probability matrix $P^{i,j} := \left(p^{i,j}_{x,x'} : x, x' \in \mathcal{X}^{i,j}\right)$. We assume that rewards are bounded and strictly positive, and without loss of generality lie in (0, 1]. Let $\pi^{i,j} := \left(\pi^{i,j}_x, x \in \mathcal{X}^{i,j}\right)$ be the stationary distribution of the Markov chain $P^{i,j}$. The mean reward from arm j for player i is defined as $\mu_{i,j} := \sum_{x \in \mathcal{X}^{i,j}} x \pi^{i,j}_x$. Note that the Markov chain represented by $P^{i,j}$ makes a state transition only when player i plays arm j. Otherwise it remains *rested.*

We note that although we use the 'big O' notation to emphasis the regret order, unless otherwise noted results are non-asymptotic.

III. Some Variations on Single Player Multi-Armed Bandit With i.i.d. Rewards

We first present some variations on the single player non-Bayesian multi-armed bandit model. These will prove useful later for the multi-player problem though they should also be of independent interest.

A. UCB_1 With Index Recomputation Every L Slots

Consider the classical single player non-Bayesian N-armed bandit problem. At each time t, the player picks a particular arm, say j, and gets a random reward $X_j(t)$. The rewards $X_j(t), 1 \leq t \leq T$, are independent and identically distributed according to some unknown probability measure with an unknown expectation μ_j. Without loss of generality, assume that $\mu_1 > \mu_i > \mu_N$, for $i = 2, \ldots N-1$. Let $n_j(t)$ denotes the number of times arm j has been played by time t. Denote $\Delta_j := \mu_1 - \mu_j$, $\Delta_{min} := \min_{j, j \neq 1} \Delta_j$ and $\Delta_{max} := \max_j \Delta_j$. The regret for any policy α is

$$\mathcal{R}_\alpha(T) := \mu_1 T - \sum_{j>1}^{N} \mu_j \mathbb{E}_\alpha[n_j(T)]. \quad (3)$$

UCB_1 index [4] is defined as

$$g_j(t) := \overline{X}_j(t) + \sqrt{\frac{2\log(t)}{n_j(t)}}, \quad (4)$$

where $\overline{X}_j(t)$ is the average reward obtained by playing arm j by time t. It is defined as $\overline{X}_j(t) = \sum_{m=1}^{t} r_j(m)/n_j(t)$, where $r_j(m)$ is the reward obtained from arm j at time m. If arm j is played at time t then $r_j(m) = X_j(m)$ and otherwise $r_j(t) = 0$. Now, an index-based policy called UCB_1 [4] is to pick the arm that has the highest index at each instant. It can be shown that this algorithm achieves regret that grows logarithmically in T non-asymptotically.

An easy variation of the above algorithm which will be useful in our analysis of subsequent algorithms is the following. Suppose the index is re-computed only once every L slots. In that case, it is easy to establish the following.

Theorem 1: Under the UCB_1 algorithm with recomputation of the index once every L slots, the expected regret by time T is given by

$$\mathcal{R}_{\text{UCB}_1}(T) \leq \sum_{j>1}^{N} \frac{8L\log T}{\Delta_j} + L\left(1 + \frac{\pi^2}{3}\right) \sum_{j>1}^{N} \Delta_j. \quad (5)$$

The proof follows [4] and taking into account the fact that every time a suboptimal arm is selected, it is played for the next L time slots. We omit it due to space consideration.

B. UCB_4 Algorithm When Index Computation is Costly

Often, learning algorithms pay a penalty or cost for computation. This is particularly the case when the algorithms must solve combinatorial optimization problems that are NP-hard. Such costs also arise in decentralized settings wherein algorithms pay a communication cost for coordination between the decentralized players. This is indeed the case, as we shall see later when we present an algorithm to solve the decentralized multi-armed bandit problem. Here, however, we will just consider an "abstract" communication or computation cost. The problem we formulate below can be solved with better regret bounds than what we present. At this time though we are unable to design algorithms with better regret bounds, that also help in decentralization.

Consider a computation cost every time the index is recomputed. Let the cost be C units. Let $m(t)$ denote the number of times the index is computed by time t. Then, under policy α the expected regret is now given by

$$\tilde{\mathcal{R}}_\alpha(T) := \mu_1 T - \sum_{j=1}^{N} \mu_j \mathbb{E}_\alpha[n_j(T)] + C\mathbb{E}_\alpha[m(T)]. \quad (6)$$

It is easy to argue that the UCB_1 algorithm will give a regret $\Omega(T)$ for this problem. We present an alternative algorithm called UCB_4 algorithm, that gives sub-linear regret. Define the UCB_4 index

$$g_j(t) := \overline{X}_j(t) + \sqrt{\frac{3\log(t)}{n_j(t)}}. \quad (7)$$

We define an arm $j^*(t)$ to be the *best arm* if $j^*(t) \in \arg\max_{1 \leq i \leq N} g_i(t)$.

We will use the following concentration inequality.

Fact 1: Chernoff-Hoeffding inequality [19]
Let $X_1, \ldots, X_t$ be random variables with a common range such that $\mathbb{E}[X_t | X_1, \ldots, X_{t-1}] = \mu$. Let $S_t = \sum_{i=1}^{t} X_i$. Then for all $a \geq 0$,

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

Theorem 2: The expected regret for the single player multi-armed bandit problem with per computation cost C using the UCB_4 algorithm is given by

$$\tilde{\mathcal{R}}_{\text{UCB}_4}(T) \leq (\Delta_{max} + C(1 + \log T)) \cdot \left(\sum_{j>1}^{N} \frac{12\log T}{\Delta_j^2} + 2N\right).$$

Thus, $\tilde{\mathcal{R}}_{\text{UCB}_4}(T) = O(\log^2 T)$.

Proof: We prove this in two steps. First, we compute the expected number of times a suboptimal arm is played and then the expected number of times we recompute the index.

Consider any suboptimal arm $j > 1$. Denote $c_{t,s} = \sqrt{3 \log t/s}$ and the indicator function of the event A by $I\{A\}$. Let $\tau_{j,m}$ be the time at which the player makes the mth transition to arm j from another arm and $\tau^{'}_{j,m}$ be the time at which the player makes the mth transition from arm j to another arm. Let $\tilde{\tau}^{'}_{j,m} = \min\{\tau^{'}_{j,m}, T\}$. Then,

$$
\begin{aligned}
& n_j(T) \\
&\quad \leq 1 + \sum_{m=1}^{T} |(\tilde{\tau}^{'}_{j,m} - \tau_{j,m})| \\
&\qquad\quad I\{\text{Arm } j \text{ is picked at time } \tau_{j,m}, \tau_{j,m} \leq T\} \\
&\quad \leq 1 + \sum_{m=1}^{T} |(\tilde{\tau}^{'}_{j,m} - \tau_{j,m})| \\
&\qquad\quad I\{g_j(\tau_{j,m} - 1) \geq g_1(\tau_{j,m} - 1), \tau_{j,m} \leq T\} \\
&\quad \leq l + \sum_{m=1}^{T} |(\tilde{\tau}^{'}_{j,m} - \tau_{j,m})| I\{g_j(\tau_{j,m} - 1) \geq \\
&\qquad\quad g_1(\tau_{j,m} - 1), \tau_{j,m} \leq T, n_j(\tau_{j,m} - 1) \geq l\} \\
&\quad \overset{(a)}{\leq} l + \sum_{m=1}^{T} \sum_{p=0}^{\infty} 2^p I\{g_j(\tau_{j,m} + 2^p - 2) \\
&\quad \geq g_1(\tau_{j,m} + 2^p - 2), \tau_{j,m} + 2^p \leq T, n_j(\tau_{j,m} - 1) \geq l\} \\
&\quad \overset{(b)}{\leq} l + \sum_{m=2}^{T} \sum_{p=0}^{\infty} 2^p I\{g_j(m + 2^p - 2) \\
&\quad \geq g_1(m + 2^p - 2), m + 2^p \leq T, n_j(m - 1) \geq l\} \\
&\quad \leq l + \sum_{m=1}^{T} \sum_{p \geq 0, m+2^p \leq T} 2^p \\
&\qquad I\{\overline{X}_j(m + 2^p - 1) + c_{m+2^p-1, n_j(m+2^p-1)} \\
&\quad \geq \overline{X}_1(m + 2^p - 1) + c_{m+2^p-1, n_1(m+2^p-1)}, n_j(m - 1) \geq l\} \\
&\quad \leq l + \sum_{m=1}^{T} \sum_{p \geq 0, m+2^p \leq T} 2^p \\
&\qquad I\{\max_{l \leq s_j < m+2^p} \overline{X}_j(m + 2^p - 1) + c_{m+2^p-1, s_j} \geq \\
&\qquad\quad \min_{1 \leq s_1 < m+2^p} \overline{X}_1(m + 2^p - 1) + c_{m+2^p-1, s_1}\} \\
&\quad \leq l + \sum_{m=1}^{\infty} \sum_{p \geq 0, m+2^p \leq T} 2^p \sum_{s_1=1}^{m+2^p} \sum_{s_j=l}^{m+2^p} \\
&\qquad I\{\overline{X}_j(m + 2^p) + c_{m+2^p, s_j} \geq \overline{X}_1(m + 2^p) + c_{m+2^p, s_1}\}.
\end{aligned} \tag{8}
$$

In Algorithm 1 (UCB$_4$), if an arm is for the pth time consecutively (without switching to any other arms in between), it will be played for the next 2^p slots. Inequality (a) uses this fact. In the inequality (b), we replace $\tau_{j,m}$ by m which is clearly an upper bound. Now, observe that the event $\{\overline{X}_j(m + 2^p) + c_{m+2^p,s_j} \geq \overline{X}_1(m + 2^p) + c_{m+2^p,s_1}\}$ implies at least one of the following events,

Algorithm 1 : UCB$_4$

```
Initialization: Select each arm $j$ once for $t \leq N$. Update the UCB$_4$ indices. Set $\eta = 1$.
while ($t \leq T$) do
  if ($\eta = 2^p$ for some $p = 0, 1, 2, \ldots$) then
    Update the index vector $g(t)$;
    Compute the best arm $j^*(t)$;
    if ($j^*(t) \neq j^*(t-1)$) then
      Reset $\eta = 1$;
    end if
  else
    $j^*(t) = j^*(t-1)$;
  end if
  Play arm $j^*(t)$;
  Increment counter $\eta = \eta + 1$; $t = t + 1$;
end while
```

$$
\begin{aligned}
A &:= \{\overline{X}_1(m + 2^p) \leq \mu_1 - c_{m+2^p,s_1}\} \\
B &:= \{\overline{X}_j(m + 2^p) \geq \mu_j + c_{m+2^p,s_j}\}, \text{ or} \\
C &:= \{\mu_1 < \mu_j + 2c_{m+2^p,s_j}\}.
\end{aligned} \tag{9}
$$

Now, using the Chernoff-Hoeffding bound, we get

$$
\begin{aligned}
\mathbb{P}\left(\overline{X}_1(m + 2^p) \leq \mu_1 - c_{m+2^p,s_1}\right) &\leq (m + 2^p)^{-6}, \\
\mathbb{P}\left(\overline{X}_j(m + 2^p) \geq \mu_j + c_{m+2^p,s_j}\right) &\leq (m + 2^p)^{-6}.
\end{aligned}
$$

For $l = \left\lceil \frac{12 \log T}{\Delta_j^2} \right\rceil$, the last event in (9) is false. In fact, for $s_j \geq \left\lceil 12 \log T/\Delta_j^2 \right\rceil$,

$$
\begin{aligned}
\mu_1 - \mu_j - 2c_{m+2^p,s_j} &= \mu_1 - \mu_j - 2\sqrt{3 \log(m + 2^p)/s_j} \\
&\geq \mu_1 - \mu_j - \Delta_j = 0.
\end{aligned}
$$

So, taking expectation on both sides, we get $\mathbb{E}[n_j(T)]$

$$
\begin{aligned}
&\leq \left\lceil 12 \log T/\Delta_j^2 \right\rceil + \sum_{m=1}^{\infty} \sum_{p=0}^{\infty} 2^p \sum_{s_1=1}^{m+2^p} \sum_{s_j=1}^{m+2^p} 2(m + 2^p)^{-6} \\
&\leq \left\lceil 12 \log T/\Delta_j^2 \right\rceil + 2 \sum_{m=1}^{\infty} \sum_{p=0}^{\infty} 2^p (m + 2^p)^{-4} \\
&\leq \frac{12 \log T}{\Delta_j^2} + 2.
\end{aligned} \tag{10}
$$

Next, we upper-bound the expectation of $m(T)$, the number of index computations performed by time T. We can write $m(T) = m_1(T) + m_2(T)$, where $m_1(T)$ is the number of index updates that result in an optimal allocation, and $m_2(T)$ is the number of index updates that result in a suboptimal allocation. Clearly, the number of updates resulting in a suboptimal allocation is less than the number of times a suboptimal arm is played. Thus,

$$
\mathbb{E}[m_2(T)] \leq \sum_{j>1}^{N} \mathbb{E}[n_j(T)]. \tag{11}
$$

To bound $\mathbb{E}[m_1(T)]$, let τ_l be the time at which the player makes the lth transition to an optimal arm from a suboptimal arm and $\tau_l^{'}$ be the time at which the player makes the lth transition from an optimal arm to a suboptimal arm.

Then, $m_1(T) \leq \sum_{l=1}^{n_{sub}(T)} \log|\tau_l - \tau_l'|$, where $n_{sub}(T)$ is the total number of such transitions by time T. Clearly, $n_{sub}(T)$ is upper-bounded by the total number of times the player picks a sub-optimal arm. Also, $\log|\tau_l - \tau_l'| \leq \log T$. So,

$$\mathbb{E}[m_1(T)] \leq \sum_{j>1}^{N} \mathbb{E}[n_j(T)] \cdot \log T. \tag{12}$$

Thus, from bounds (11) and (12), we get

$$\mathbb{E}[m(T)] \leq \sum_{j>1}^{N} \mathbb{E}[n_j(T)] \cdot (1+\log T). \tag{13}$$

Now, using equation (6), the expected regret is

$$\begin{aligned}\tilde{\mathcal{R}}_{\mathrm{UCB}_4}(T) &= \sum_{j>1}^{N} \mathbb{E}[n_j(T)] \cdot \Delta_j + C\mathbb{E}[m(T)] \\ &\leq \Delta_{max} \sum_{j>1}^{N} \mathbb{E}[n_j(T)] + C\mathbb{E}[m(T)] \\ &\leq (\Delta_{max} + C(1+\log T)) \sum_{j>1}^{N} \mathbb{E}[n_j(T)].\end{aligned}$$

by using (13). Now, by bound (10), we get the desired bound on the expected regret. ■

Remarks. 1. It is easy to show that the lower bound for the single player MAB problem with computation costs is $\Omega(\log T)$. This can be achieved by the UCB_2 algorithm [4]. To see this, note that the number of times the player selects a suboptimal arm when using UCB_2 is $O(\log T)$. Since $\mathbb{E}[n_j(T)] = O(\log T)$, we get $\mathbb{E}[\sum_{j>1}^{N} n_j(T)] = O(\log T)$, and also $\mathbb{E}[m_2(T)] = O(\log T)$. Now, since the epochs are not getting reset after every switch and are exponentially spaced, the number of updates that result in the optimal allocation, $m_1(T) \leq \log T$. These together yield

$$\tilde{\mathcal{R}}_{\mathrm{UCB}_2}(T) \leq \sum_{j>1}^{N} \mathbb{E}[n_j(T)] \cdot \Delta_j + C\mathbb{E}[m(T)] = O(\log T).$$

2. However, it is unknown at this time if UCB_2 can be decentralized. This is the main reason for introducing the UCB_4 algorithm.

C. Algorithms With Finite Precision Indices

Often, the indices might be known only upto a certain precision. This can happen when either in an implementation where only a finite number of bits are available to represent indices, or when there is a cost to compute the indices to greater precision. For example, as we will see in the decentralized setting, indices must be communicated to other players. Since only a finite number of bits can be communicated in finite time, the algorithm should be able to work with indices with limited precision. Thus, when indices are known up to some ϵ precision, it may not be possible to tell which of the two indices is greater if they are within ϵ of each other. This becomes important because it can empirically be observed (see Fig. 2) that indices track each other closely in index based algorithms. The question then is, how are the performances of various index-based policies affected if there are limits on index resolution, and only an arm with an ϵ-highest index can be picked.

We first show that if Δ_{min} is known, we can fix a precision $0 < \epsilon < \Delta_{min}$, so that UCB_4 algorithm will achieve order log-squared regret growth with T. If Δ_{min} is not known, we can pick a positive monotone sequence $\{\epsilon_t\}$ such that $\epsilon_t \to 0$, as $t \to \infty$. Denote the cost of computation for ϵ-precision be $C(\epsilon)$. We assume that $C(\epsilon) \to \infty$ monotonically as $\epsilon \to 0$.

Theorem 3: (i) If Δ_{min} is known, choose an $0 < \epsilon < \Delta_{min}$. Then, the expected regret of the UCB_4 algorithm with ϵ-precise computations is given by

$$\tilde{\mathcal{R}}_{\mathrm{UCB}_4}(T) \leq (\Delta_{max} + C(\epsilon)(1+\log T)) \cdot \left(\sum_{j>1}^{N} \frac{12\log T}{(\Delta_j - \epsilon)^2} + 2N\right).$$

Thus, $\tilde{\mathcal{R}}_{\mathrm{UCB}_4}(T) = O(\log^2 T)$.

(ii) If Δ_{min} is unknown, denote $\epsilon_{min} = \Delta_{min}/2$ and choose a positive monotone sequence $\{\epsilon_t\}$ such that $\epsilon_t \to 0$ as $t \to \infty$. Then, there exists a $t_0 > 0$ such that for all $T > t_0$,

$$\tilde{\mathcal{R}}_{\mathrm{UCB}_4}(T) \leq (\Delta_{max}+C(\epsilon_{min}))\, t_0 + (\Delta_{max}+C(\epsilon_T)(1+\log T)) \cdot \left(\sum_{j>1}^{N} \frac{12\log T}{(\Delta_j - \epsilon_{min})^2} + 2N\right)$$

where t_0 is the smallest t such that $\epsilon_{t_0} < \epsilon_{min}$. Thus by choosing an arbitrarily slowly decreasing sequence $\{\epsilon_t\}$, we can make the regret arbitrarily close to $O(\log^2 T)$ asymptotically.

Proof: (i) The proof is only a slight modification of the proof given in Theorem 2. Due to the ϵ precision, the player will pick a suboptimal arm if the event $\{\overline{X}_j(m+2^p) + c_{m+2^p,s_j} + \epsilon \geq \overline{X}_1(m+2^p) + c_{m+2^p,s_1}\}$ occurs. Thus equation (8) becomes,

$$n_j(T) \leq l + \sum_{m=1}^{\infty} \sum_{p \geq 0, m+2^p \leq T} 2^p \sum_{s_1=1}^{m+2^p} \sum_{s_j=l}^{m+2^p} I\{\overline{X}_j(m+2^p) + c_{m+2^p,s_j} + \epsilon \geq \overline{X}_1(m+2^p) + c_{m+2^p,s_1}\}.$$

Now, the event $\{\overline{X}_j(m+2^p) + c_{m+2^p,s_j} + \epsilon \geq \overline{X}_1(m+2^p) + c_{m+2^p,s_1}\}$ implies that at least one of the following events must occur:

$$\begin{aligned} A &:= \left\{\overline{X}_1(m+2^p) \leq \mu_1 - c_{m+2^p,s_1}\right\}, \\ B &:= \left\{\overline{X}_j(m+2^p) \geq \mu_j + \epsilon + c_{m+2^p,s_j}\right\}, \\ C &:= \left\{\mu_1 < \mu_j + \epsilon + 2c_{m+2^p,s_j}\right\}, \text{ or,} \\ D &:= \left\{\mu_1 < \mu_j + \epsilon\right\}. \end{aligned} \tag{14}$$

Since $\{\overline{X}_j(m+2^p) \geq \mu_j + \epsilon + c_{m+2^p,s_j}\} \subseteq \{\overline{X}_j(m+2^p) \geq \mu_j + c_{m+2^p,s_j}\}$, we have

$$\begin{aligned}&\mathbb{P}(\{\overline{X}_j(m+2^p) \geq \mu_j + \epsilon + c_{m+2^p,s_j}\}) \\ &\qquad \leq \mathbb{P}(\{\overline{X}_j(m+2^p) \geq \mu_j + c_{m+2^p,s_j}\}).\end{aligned}$$

Also, for $l = \lceil 12 \log T/(\Delta_j - \epsilon)^2 \rceil$, the event C cannot happen. In fact, for $s_j \geq \lceil 12 \log T/(\Delta_j - \epsilon)^2 \rceil$,

$$\mu_1 - \mu_j - \epsilon - 2c_{t+2^p,s_j} = \mu_1 - \mu_j - \epsilon - 2\sqrt{\frac{3\log(t+2^p)}{s_j}}$$
$$\geq \mu_1 - \mu_j - \epsilon - (\Delta_j - \epsilon) = 0,$$

If $\epsilon < \Delta_{min}$, the last event (D) in equation (14) is also not true. Thus, for $0 < \epsilon < \Delta_{min}$, we get

$$\mathbb{E}[n_j(T)] \leq \frac{12\log(n)}{(\Delta_j - \epsilon)^2} + 2. \tag{15}$$

The rest of the proof is the same as in Theorem 2. Now, if Δ_{min} is known, we can choose $0 < \epsilon < \Delta_{min}$ and by Theorem 2 and bound (15), we get the desired result.

(ii) If Δ_{min} is unknown, we can choose a positive monotone sequence $\{\epsilon_t\}$ such that $\epsilon_t \to 0$ as $t \to \infty$. Thus, there exists a t_0 such that for $t > t_0$, $\epsilon_t < \epsilon_{min}$. We may get a linear regret upto time t_0 but after that the analysis follows as that in the proof of Theorem 2, and regret grows only sub-linearly. Since $C(\cdot)$ is monotone, $C(\epsilon_T) > C(\epsilon_t)$ for all $t < T$. The last part can now be trivially established using the obtained bound on the expected regret. ■

IV. Single Player Multi-Armed Bandit With Markovian Rewards

Now, we consider the scenario where the rewards obtained from an arm are not i.i.d. but come from a Markov chain. Reward from each arm is modeled as an irreducible, aperiodic, Markov chain on a finite state space $\mathcal{X}^i$ and represented by a transition probability matrix $P^i := \left(p^i_{x,x^{'}} : x, x^{'} \in \mathcal{X}^i\right)$. Assume that the reward space $\mathcal{X}^i \subseteq (0, 1]$. Let $X_i(1), X_i(2), \ldots$ denote the successive rewards from arm i. All arms are mutually independent. Let $\pi^{\mathbf{i}} := \left(\pi^i_x, x \in \mathcal{X}^i\right)$ be the stationary distribution of the Markov chain P^i. Since the Markov chains are ergodic under these assumptions, the mean reward from arm i is given by $\mu_i := \sum_{x \in \mathcal{X}^i} x\pi^i_x$. Without loss of generality, assume that $\mu_1 > \mu_i > \mu_N$, for $i = 2, \ldots N-1$. As before, $n_j(t)$ denotes the number of times arm j has been played by time t. Denote $\Delta_j := \mu_1 - \mu_j$, $\Delta_{min} := \min_{j, j \neq 1} \Delta_j$ and $\Delta_{max} := \max_j \Delta_j$. Denote $\pi_{min} := \min_{1 \leq i \leq N, x \in \mathcal{X}^i} \pi^i_x$, $x_{max} := \max_{1 \leq i \leq N, x \in \mathcal{X}^i} x$ and $x_{min} := \min_{1 \leq i \leq N, x \in \mathcal{X}^i} x$. Let $\hat{\pi}^i_x := \max\{\pi^i_x, 1 - \pi^i_x\}$ and $\hat{\pi}_{max} := \max_{1 \leq i \leq N, x \in \mathcal{X}^i} \hat{\pi}^i_x$. Let $|\mathcal{X}^i|$ denote the cardinality of the state space $\mathcal{X}^i$, $|\mathcal{X}|_{max} := \max_{1 \leq i \leq N} |\mathcal{X}^i|$.

Let $P^{i'}$ be the adjoint of P^i on $l_2(\pi^i)$, defined as,

$$P^{i'}_{x,y} := \pi_y P^i{}_{y,x}/\pi_x. \tag{16}$$

Then, $\tilde{P}^i := P^{i'}P^i$, denotes the multiplicative symmetrization of P^i. We assume that P^is are such that $\tilde{P}^i$s are irreducible. Clearly, this is a less stringent assumption than the reversibility of the matrices P^is. Also, if $P^i_{x,x} > 0, \forall x \in \mathcal{X}^i$, then it is easy to show that $\tilde{P}^i$s are reversible. Let ρ^i be the eigenvalue gap, $1 - \lambda_2$, where λ_2 is the second largest eigenvalue of the matrix $\tilde{P}^i$. Denote $\rho_{max} := \max_{1 \leq i \leq N} \rho^i$ and $\rho_{min} := \min_{1 \leq i \leq N} \rho^i$, where ρ^i is the eigenvalue gap of the ith arm.

The total reward obtained by the time T is then given by $S_T = \sum_{j=1}^N \sum_{s=1}^{n_j(T)} X_j(s)$. The regret for any policy α is,

$$\tilde{\mathcal{R}}_{M,\alpha}(T) := \mu_1 T - \mathbb{E}_\alpha \sum_{j=1}^{N} \sum_{s=1}^{n_j(T)} X_j(s) + C\mathbb{E}_\alpha[m(T)] \tag{17}$$

where C is the cost per computation and $m(T)$ is the number of times the index is computed by time T. Define the index

$$g_j(t) := \overline{X}_j(t) + \sqrt{\frac{\kappa \log(t)}{n_j(t)}}, \tag{18}$$

where $\overline{X}_j(t)$ is the average reward obtained by playing arm j by time t, as defined in the previous section, and κ can be any constant satisfying $\kappa > 168|\mathcal{X}|^2_{max}/\rho_{min}$.

We introduce one more notation here. If $\mathcal{F}$ and $\mathcal{G}$ are two σ-algebras, then $\mathcal{F} \vee \mathcal{G}$ denotes the smallest σ-algebra containing $\mathcal{F}$ and $\mathcal{G}$. Similarly, if $\{\mathcal{F}_t, t = 1, 2, \ldots\}$ is a collection of σ-algebras, then $\vee_{t \geq 1} F_t$ denotes the smallest $\sigma-$algebra containing $\mathcal{F}_1, \mathcal{F}_2, \ldots$.

We use the following results on Markov chains (Lemma 1, Lemma 2) to prove an intermediate result in Lemma 3.

Lemma 1: [5] Let $(X_t, t = 1, 2, \ldots)$ be an irreducible, aperiodic Markov chain on a finite state space $\mathcal{X}$ with transition probability matrix P, an initial distribution that is non-zero in all states and a stationary distribution π. Let $\mathcal{F}_t$ be the σ-algebra generated by $(X_1, X_2, \ldots, X_t)$. Let $\mathcal{G}$ be a σ-algebra independent of $\mathcal{F} = \vee_{t \geq 1} F_t$. Let τ be a stopping time of $\mathcal{F}_t \vee \mathcal{G}$. Let $N(x, \tau) := \sum_{t=1}^{\tau} I\{X_t = x\}$. Then, $|\mathbb{E}[N(x, \tau)] - \pi_x \mathbb{E}[\tau]| \leq K$, where $K \leq 1/\pi_{min}$ and $\pi_{min} = \min_{x \in \mathcal{X}} \pi_x$. K depends on P.

Lemma 2: [20] Let $(X_t, t = 1, 2, \ldots)$ be an irreducible, aperiodic Markov chain on a finite state space $\mathcal{X}$ with transition probability matrix P, an initial distribution λ and a stationary distribution π. Denote $N_\lambda = \left\| \left(\frac{\lambda_x}{\pi_x}, x \in \mathcal{X}\right) \right\|_2$. Let ρ be the eigenvalue gap, $1 - \lambda_2$, where λ_2 is the second largest eigenvalue of the matrix $\tilde{P}$ where $\tilde{P}$ is the multiplicative symmetrization of the transition matrix P. Let $f : \mathcal{X} \to \mathbf{R}$ be such that $\sum_{x \in \mathcal{X}} \pi_x f(x) = 0$, $\|f\|_\infty \leq 1$, $\|f\|_2^2 \leq 1$. If $\tilde{P}$ is irreducible, then for any $\gamma > 0$, $\mathbb{P}\left(\sum_{a=1}^t f(X_a)/t \geq \gamma\right) \leq N_\lambda e^{-t\rho\gamma^2/28}$.

Lemma 3: If the reward of each arm is given by a Markov chain satisfying the hypothesis of Lemma 1, then under any policy α we have

$$\tilde{\mathcal{R}}_{M,\alpha}(T) \leq \sum_{j=2}^{N} \Delta_j \mathbb{E}_\alpha[n_j(T)] + K_{\mathcal{X},P} + C\mathbb{E}_\alpha[m(T)] \tag{19}$$

where $K_{\mathcal{X},P} = \sum_{j=1}^N \sum_{x \in \mathcal{X}^j} x/\pi^j_{min}$, $\pi^j_{min} = \min_{x \in \mathcal{X}^j} \pi^j_x$.

Proof: Let $X_j(1), X_j(2), \ldots$ denote the successive rewards from arm j. Let $\mathcal{F}^j_t$ denotes the σ-algebra generated by $\left(X_j(1), \ldots, X_j(t)\right)$. Let $\mathcal{F}^j = \vee_{t \geq 1} \mathcal{F}^j_t$ and $\mathcal{G}^j = \vee_{i \neq j} F^i$. Since arms are independent, $\mathcal{G}^j$ is independent of $\mathcal{F}^j$. Clearly, $n_j(T)$ is a stopping time with respect to $\mathcal{G}^j \vee \mathcal{F}^j_T$. The total reward is $S_T = \sum_{j=1}^N \sum_{s=1}^{n_j(T)} X_j(s) = \sum_{j=1}^N \sum_{x \in \mathcal{X}^j} x N(x, n_j(T))$ where

$N(x, n_j(T)) := \sum_{t=1}^{n_j(T)} I\{X_j(t) = x\}$. Taking the expectation and using the Lemma 1, we have

$$\left|\mathbb{E}[S_T] - \sum_{j=1}^{N} \sum_{x \in \mathcal{X}^j} x\pi_x^j \mathbb{E}[n_j(T)]\right| \leq \sum_{j=1}^{N} \sum_{x \in \mathcal{X}^j} x/\pi_{min}^j,$$

which implies $\left|\mathbb{E}[S_T] - \sum_{j=1}^{N} \mu_j \mathbb{E}[n_j(T)]\right| \leq K_{\mathcal{X},P}$, where $K_{\mathcal{X},P} = \sum_{j=1}^{N} \sum_{x \in \mathcal{X}^j} x/\pi_{min}^j$. Since regret $\tilde{\mathcal{R}}_{M,\alpha}(T) = \mu_1 T - \mathbb{E}_\alpha \sum_{j=1}^{N} \sum_{t=1}^{n_j(T)} X_j(t) + C\mathbb{E}_\alpha[m(T)]$ (c.f. equation (17)), we get

$$\left|\tilde{\mathcal{R}}_{M,\alpha}(T) - \left(\mu_1 T - \sum_{j=1}^{N} \mu_j \mathbb{E}[n_j(T)] + C\mathbb{E}_\alpha[m(T)]\right)\right| \leq K_{\mathcal{X},P}.$$

■

Theorem 4: (i) If $|\mathcal{X}|_{max}$ and ρ_{min} are known, choose $\kappa > 168|\mathcal{X}|_{max}^2/\rho_{min}$. Then, the expected regret using the UCB_4 algorithm with the index defined as in (18) for the single player multi-armed bandit problem with Markovian rewards and per computation cost C is given by

$$\tilde{\mathcal{R}}_{M,\text{UCB}_4}(T) \leq (\Delta_{max} + C(1 + \log T)) \cdot \left(\sum_{j>1}^{N} \frac{4\kappa \log T}{\Delta_j^2} + N(2D+1)\right) + K_{\mathcal{X},P}$$

where $D = \frac{|\mathcal{X}|_{max}}{\pi_{min}}$. Thus, $\tilde{\mathcal{R}}_{M,\text{UCB}_4}(T) = O(\log^2 T)$.
(ii) If $|\mathcal{X}|_{max}$ and ρ_{min} are not known, choose a positive monotone sequence $\{\kappa_t\}$ such that $\kappa_t \to \infty$ as $t \to \infty$ and $\kappa_t \leq t$. Then, $\tilde{\mathcal{R}}_{M,\text{UCB}_4}(T) = O(\kappa_T \log^2 T)$. Thus, by choosing an arbitrarily slowly increasing sequence $\{\kappa_t\}$ we can make the regret arbitrarily close to $\log^2 T$.

Proof: (i) Consider any suboptimal arm $j > 1$. Denote $c_{t,s} = \sqrt{\kappa \log t/s}$. As in the proof of Theorem 2, we start by bounding $n_j(T)$. The initial steps are the same as in the proof of Theorem 2. So, we skip those steps and start from the inequality (8) there.

$$n_j(T) \leq l + \sum_{m=1}^{\infty} \sum_{p \geq 0, m+2^p \leq T} 2^p \sum_{s_1=1}^{m+2^p} \sum_{s_j=l}^{m+2^p} I\{\overline{X}_j(m+2^p) + c_{m+2^p,s_j} \geq \overline{X}_1(m+2^p) + c_{m+2^p,s_1}\}.$$

The event $\{\overline{X}_j(m+2^p) + c_{m+2^p,s_j} \geq \overline{X}_1(m+2^p) + c_{m+2^p,s_1}\}$ is true only if at least one of the events shown in display (9) are true. We note that, for any initial distribution λ^j for arm j,

$$N_{\lambda^j} = \left\|\left(\frac{\lambda_x^j}{\pi_x^j}, x \in \mathcal{X}^j\right)\right\|_2 \leq \sum_{x \in \mathcal{X}^j} \left\|\left(\frac{\lambda_x^j}{\pi_x^j}\right)\right\|_2 \leq \frac{1}{\pi_{min}}. \quad (20)$$

Also, $x_{max} \leq 1$. Let $n_x^j(s_j)$ be the number of times the state x is observed when arm j is pulled s_j times. Then, the probability of the first event in (9),

$$\begin{aligned}
&\mathbb{P}(\overline{X}_j(m+2^p) \geq \mu_j + c_{m+2^p,s_j}) \\
&= \mathbb{P}\left(\sum_{x \in \mathcal{X}^j} x n_x^j(s_j) \geq s_j \sum_{x \in \mathcal{X}^j} x\pi_x^j + s_j c_{m+2^p,s_j}\right) \\
&= \mathbb{P}\left(\sum_{x \in \mathcal{X}^j} x(n_x^j(s_j) - s_j\pi_x^j) \geq s_j c_{m+2^p,s_j}\right) \\
&\overset{(a)}{\leq} \sum_{x \in \mathcal{X}^j} \mathbb{P}\left(n_x^j(s_j) - s_j\pi_x^j \geq \frac{s_j c_{m+2^p}}{x|\mathcal{X}^j|}\right) \\
&= \sum_{x \in \mathcal{X}^j} \mathbb{P}\left(\frac{\sum_{t=1}^{s_j} I\{X_j(t) = x\} - s_j\pi_x^j}{s_j\hat{\pi}_x^j} \geq \frac{c_{m+2^p,s_j}}{x|\mathcal{X}^j|\hat{\pi}_x^j}\right) \\
&\overset{(b)}{\leq} \sum_{x \in \mathcal{X}^j} N_{\lambda^j}(m+2^p)^{-\kappa\rho^i/28x^2|\mathcal{X}^j|^2(\hat{\pi}_x^j)^2} \\
&\overset{(c)}{\leq} \frac{|\mathcal{X}|_{max}}{\pi_{min}}(m+2^p)^{-\kappa\rho_{min}/28|\mathcal{X}|_{max}^2}.
\end{aligned}$$

The inequality (a) follows after some simple algebra, which we skip due to space limitations. The inequality (b) follows by defining the function $f(X_j(t)) = (I\{X_j(t) = x\} - \pi_x^j)/\hat{\pi}_x^j$ and using the Lemma 2. For inequality (c) we used the facts that $N_{\lambda^j} \leq 1/\pi_{min}$, $x_{max} \leq 1$ and $\hat{\pi}_{max} \leq 1$. Thus,

$$\begin{aligned}
\mathbb{P}(\overline{X}_j(m+2^p) &\geq \mu_j + c_{m+2^p,s_j}) \\
&\leq D(m+2^p)^{-\kappa\rho_{min}/28|\mathcal{X}|_{max}|^2} \quad (21)
\end{aligned}$$

where $D = \frac{|\mathcal{X}|_{max}}{\pi_{min}}$. Similarly we can get,

$$\begin{aligned}
\mathbb{P}(\overline{X}_1(m+2^p) &\leq \mu_1 - c_{m+2^p,s_1}) \\
&\leq D(m+2^p)^{-\kappa\rho_{min}/28|\mathcal{X}|_{max}|^2} \quad (22)
\end{aligned}$$

For $l = \left\lceil 4\kappa \log T/\Delta_j^2 \right\rceil$, the last event in (9) is false. In fact, for $s_j \geq \left\lceil 4\kappa \log T/\Delta_j^2 \right\rceil$,

$$\begin{aligned}
\mu_1 - \mu_j - 2c_{m+2^p,s_j} &= \mu_1 - \mu_j - 2\sqrt{\kappa \log(m+2^p)/s_j} \\
&\geq \mu_1 - \mu_j - \Delta_j = 0
\end{aligned}$$

Thus, $\mathbb{E}[n_j(T)] \leq$

$$\begin{aligned}
&\left\lceil \frac{4\kappa \log T}{\Delta_j^2} \right\rceil + \sum_{m=1}^{\infty} \sum_{p=0}^{\infty} 2^p \sum_{s_1=1}^{m+2^p} \sum_{s_j=1}^{m+2^p} 2D(m+2^p)^{-\frac{\kappa\rho_{min}}{28|\mathcal{X}|_{max}|^2}} \\
&= \left\lceil \frac{4\kappa \log T}{\Delta_j^2} \right\rceil + 2D \sum_{m=1}^{\infty} \sum_{p=0}^{\infty} 2^p (m+2^p)^{-\frac{\kappa\rho_{min} - 56|\mathcal{X}|_{max}^2}{28|\mathcal{X}|_{max}^2}}.
\end{aligned} \quad (23)$$

When $\kappa > 168|\mathcal{X}|_{max}^2/\rho_{min}$, the above summation converges to a value less that 1 and we get

$$\mathbb{E}[n_j(T)] \leq \frac{4\kappa \log T}{\Delta_j^2} + (2D+1). \quad (24)$$

Now, from the proof of Theorem 2 (equation (13)),

$$\mathbb{E}[m(T)] \leq \sum_{j>1}^{N} \mathbb{E}[n_j(T)] \cdot (1 + \log T). \quad (25)$$

Now, using inequality (19), the expected regret $\tilde{\mathcal{R}}_{M,\mathtt{UCB}_4}(T)$

$$\begin{aligned}
&= \sum_{j>1}^{N} \mathbb{E}[n_j(T)]\cdot \Delta_j + C\mathbb{E}[m(T)] + K_{\mathcal{X},P} \\
&\leq \Delta_{max} \sum_{j>1}^{N} \mathbb{E}[n_j(T)] + C\mathbb{E}[m(T)] + K_{\mathcal{X},P} \\
&\leq (\Delta_{max} + C(1+\log T)) \sum_{j>1}^{N} \mathbb{E}[n_j(T)] + K_{\mathcal{X},P}.
\end{aligned}$$

by using (25). Now, by bound (24), we get the desired bound on the expected regret.

(ii) Replacing κ with κ_t, equation (23) becomes

$$\begin{aligned}
\mathbb{E}[n_j(T)] \leq &\left\lceil \frac{4\kappa_T \log T}{\Delta_j^2} \right\rceil + \\
&2D \sum_{m=1}^{\infty} \sum_{p=0}^{\infty} 2^p (m+2^p)^{-\frac{\kappa_{m+2^p}\rho_{min} - 56|\mathcal{X}|^2_{max}}{28|\mathcal{X}|^2_{max}}}
\end{aligned}$$

Since, $\kappa_t \to \infty$ as $t \to \infty$, the exponent $-\frac{\kappa_{m+2^p}\rho_{min} - 56|\mathcal{X}|^2_{max}}{28|\mathcal{X}|^2_{max}}$ becomes smaller that -4 for sufficiently large m and p, and the above summation converges, yielding the desired result. ■

We note that the results for Markovian reward just presented extend easily even with finite precision indices. As before, suppose the cost of computation for ϵ-precision is $C(\epsilon)$. We assume that $C(\epsilon) \to \infty$ monotonically as $\epsilon \to 0$. We will use this result in Section VI.

Theorem 5: (i) If Δ_{min}, $|\mathcal{X}|_{max}$ and ρ_{min} are known, choose an $0 < \epsilon < \Delta_{min}$, and a $\kappa > 168|\mathcal{X}|^2_{max}/\rho_{min}$. Then, the expected regret using the $\mathtt{UCB}_4$ algorithm with the index defined as in (18) for the single player multi-armed bandit problem with Markovian rewards with ϵ-precise computations is given by

$$\begin{aligned}
\tilde{\mathcal{R}}_{M,\mathtt{UCB}_4}(T) \leq &(\Delta_{max} + C(\epsilon)(1+\log T)) \\
&\cdot \left(\sum_{j>1}^{N} \frac{4\kappa \log T}{(\Delta_j - \epsilon)^2} + N(2D+1) \right).
\end{aligned}$$

where $D = \frac{|\mathcal{X}|_{max}}{\pi_{min}}$. Thus, $\tilde{\mathcal{R}}_{M,\mathtt{UCB}_4}(T) = O(\log^2 T)$.

(ii) If Δ_{min}, $|\mathcal{X}|_{max}$ and ρ_{min} are unknown, choose a positive monotone sequences $\{\epsilon_t\}$ and $\{\kappa_t\}$ such that $\kappa_t \leq t$, $\epsilon_t \to 0$ and $\kappa_t \to \infty$ as $t \to \infty$. Then, $\tilde{\mathcal{R}}_{M,\mathtt{UCB}_4}(T) = O(C(\epsilon_T)\kappa_T \log^2 T)$. We can choose $\{\epsilon_t\}$ and $\{\kappa_t\}$ as two arbitrarily slowly decreasing and increasing sequences respectively, and the regret can be made arbitrarily close to $\log^2(T)$. The proof follows by a combination of the proof of the theorems 3 and 4, and is omitted.

V. The Decentralized MAB Problem With i.i.d. Rewards

We now consider the decentralized multi-armed bandit problem with i.i.d. rewards wherein multiple players play at the same time. Players have no information about means or distribution of rewards from various arms. If two or more players pick the same arm, we assume that neither gets any reward. This is an online learning problem of distributed bipartite matching.

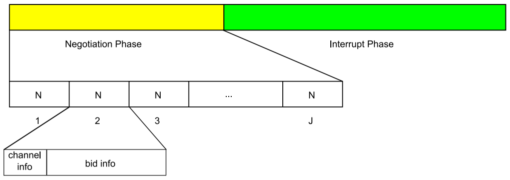

Fig. 1. Structure of the decision frame.

Communication requirement: We assume that there are no *dedicated control channels* for coordination or communication between the players. However, we do allow for players to communicate with each other by pulling arms in a specific order. For example, a player could communicate a bit sequence to other players by picking arm to indicate 'bit 1' and any other arm for 'bit 0'. However, such a communication overhead would add to regret and affect the learning rate.

Distributed algorithms for bipartite matching algorithms are known [21], [22] which determine an ϵ-optimal matching with a 'minimum' amount of information exchange and computation. However, every run of this distributed bipartite matching algorithm incurs a cost due to computation and communication necessary to exchange some information for decentralization. Let C be the cost per run, and $m(t)$ denote the number of times the distributed bipartite matching algorithm is run by time t. Then, under policy α the expected regret is

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

where $\mathbf{k}^{**}$ is the optimal matching as defined in equation (1) in section II-A.

Temporal Structure. We divide time into frames. Each frame is one of two kinds: a *decision frame*, and an *exploitation frame*. In the decision frame, the index is recomputed, and the distributed bipartite matching algorithm run again to determine the new matching. The length of such a frame can be seen as cost of the algorithm. We further divide the decision frame into two phases, a *negotiation phase* and an *interrupt phase* (see Fig. 1). The information exchange needed to compute an ϵ-optimal matching is done in the *negotiation phase*. In the *interrupt phase*, a player signals to other players if his allocation has changed. In the exploitation frame, the current matching is *exploited* without updating the indices. Later, we will allow the frame lengths to increase with time.

We now present the $\mathtt{dUCB}_4$ algorithm, a decentralized version of $\mathtt{UCB}_4$. For each player i and each arm j, we define a $\mathtt{dUCB}_4$ index at the end of frame t as

$$g_{i,j}(t) := \overline{X}_{i,j}(t) + \sqrt{\frac{(M+2)\log n_i(t)}{n_{i,j}(t)}}, \tag{26}$$

where $n_i(t)$ is the number of successful plays (without collisions) of player i by frame t, $n_{i,j}(t)$ is the number of times player i picks arm j successfully by frame t. $\overline{X}_{i,j}(t)$ is the sample mean of rewards from arm j for player i

Algorithm 2 dUCB$_4$ for User i

```
Initialization: Play a set of matchings so that each player plays each arm at least once. Set counter $\eta = 1$.
while ($t \leq T$) do
  if ($\eta = 2^p$ for some $p = 0, 1, 2, \ldots$) then
    //Decision frame:
    Update $g(t)$;
    Participate in the dBM$_\epsilon(g(t))$ algorithm to obtain a match $k_i^*(t)$;
    if ($k_i^*(t) \neq k_i^*(t-1)$) then
      Use interrupt phase to signal an INTERRUPT to all other players about changed allocation;
      Reset $\eta = 1$;
    end if
    if (Received an INTERRUPT) then
      Reset $\eta = 1$;
    end if
  else
    // Exploitation frame:
    $k_i^*(t) = k_i^*(t-1)$;
  end if
  Play arm $k_i^*(t)$;
  Increment counter $\eta = \eta + 1$, $t = t + 1$;
end while
```

from $n_{i,j}(t)$ samples. Let $g(t)$ denote the vector $(g_{i,j}(t), 1 \leq i \leq M, 1 \leq j \leq N)$. We will refer to an ϵ-optimal distributed bipartite matching algorithm as dBM$_\epsilon(g(t))$ that yields a solution $\mathbf{k}^*(t) := (k_1^*(t), \ldots, k_M^*(t)) \in \mathcal{P}(N)$ such that $\sum_{i=1}^M g_{i,k_i^*(t)}(t) \geq \sum_{i=1}^M g_{i,k_i}(t)) - \epsilon$, $\forall \mathbf{k} \in \mathcal{P}(N)$, $\mathbf{k} \neq \mathbf{k}^*$. There exists an optimal bipartite matching $\mathbf{k}^{**} \in \mathcal{P}(N)$ 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 that $\Delta_{min} > 0$.

In the dUCB$_4$ algorithm, at the end of every decision frame, the dBM$_\epsilon(g(t))$ will give a legitimate matching with no two players colliding on any arm. Thus, the regret accrues either if the matching $\mathbf{k}(t)$ is not the optimal matching $\mathbf{k}^{**}$, or if a decision frame is employed by the players to recompute the matching. Every time a frame is a decision frame, it adds a cost C to the regret. 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 example when indices must be communicated to other players in 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 denote the cost as $C(\epsilon)$.

We first show that if Δ_{min} is known, we can choose an $\epsilon < \Delta_{min}/(M+1)$, so that dUCB$_4$ algorithm will achieve order log-squared regret growth with T. If Δ_{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 in the decision frames. It, thus, is some monotonically decreasing function $\epsilon = f(L)$ of their length L such that $\epsilon \to 0$ as $L \to \infty$. Thus, we must pick a positive monotone sequence $\{L_t\}$ such that $L_t \to \infty$. Clearly, $C(f(L_t)) \to \infty$ as $t \to \infty$. This can happen arbitrarily slowly.

Theorem 6: (i) Let $\epsilon > 0$ be the precision of the bipartite matching algorithm and the precision of the index representation. If Δ_{min} is known, choose $\epsilon > 0$ such that $\epsilon < \Delta_{min}/(M+1)$. Let L be the length of a frame. Then, the expected regret of the dUCB$_4$ algorithm is

$$\tilde{\mathcal{R}}_{\mathtt{dUCB}_4}(T) \leq (L\Delta_{max} + C(f(L))(1 + \log T)) \cdot \left(\frac{4M^3(M+2)N \log T}{(\Delta_{min} - ((M+1)\epsilon)^2} + NM(2M+1) \right).$$

Thus, $\tilde{\mathcal{R}}_{\mathtt{dUCB}_4}(T) = O(\log^2 T)$.
(ii) When Δ_{min} is unknown, denote $\epsilon_{min} = \Delta_{min}/(2(M+1))$ and let $L_t \to \infty$ as $t \to \infty$. Then, there exists a $t_0 > 0$ such that for all $T > t_0$,

$$\begin{aligned}\tilde{\mathcal{R}}_{\mathtt{dUCB}_4}(T) \leq (&L_{t_0}\Delta_{max} + C(f(L_{t_0}))t_0 \\ &+ (L_T\Delta_{max} + C(f(L_T))(1 + \log T)) \\ &\cdot \left(\frac{4M^3(M+2)N \log T}{(\Delta_{min} - \epsilon_{min})^2} + NM(2M+1) \right),\end{aligned}$$

where t_0 is the smallest t such that $f(L_{t_0}) < \epsilon_{min}$. Thus by choosing an arbitrarily slowly increasing sequence $\{L_t\}$ we can make the regret arbitrarily close to $\log^2 T$.

Proof: (i) First, we obtain a bound for $L = 1$. Then, appeal to a result like Theorem 1 to obtain the result for general L. The implicit dependence between ϵ and L through the function $f(\cdot)$ does not affect this part of the analysis. Details are omitted due to space limitations.

We first upper bound the number of sub-optimal plays. We define $\tilde{n}_{i,j}(t), 1 \leq i \leq M, 1 \leq j \leq N$ as follows: Whenever the dBM$_\epsilon(g(t))$ algorithm gives a non-optimal matching $\mathbf{k}(t)$, $\tilde{n}_{i,j}(t)$ is increased by one for some $(i, j) \in \arg\min_{1 \leq i \leq M, 1 \leq j \leq N} n_{i,j}(t)$. Let $\tilde{n}(T)$ denote the total number of suboptimal plays. Then, clearly, $\tilde{n}(T) = \sum_{i=1}^M \sum_{j=1}^N \tilde{n}_{i,j}(T)$. So, in order to get a bound on $\tilde{n}(T)$ we first get a bound on $\tilde{n}_{i,j}(T)$.

Let $\tilde{I}_{i,j}(t)$ be the indicator function which is equal to 1 if $\tilde{n}_{i,j}(t)$ is incremented by one, at time t. Thus $\tilde{I}_{i,j}(t)$ is equal to 1 whenever there is a non-optimal matching. Let this non-optimal matching be $\mathbf{k}(t)$, where $\mathbf{k}(t) \neq \mathbf{k}^{**}$. In the following, we denote it as $\mathbf{k}$, omitting the time index. A non-optimal matching $\mathbf{k}$ is selected if the event $\left\{ \sum_{i=1}^M g_{i,k_i^{**}}(m + 2^p - 1) \leq (M+1)\epsilon + \sum_{i=1}^M g_{i,k_i}(m + 2^p - 1) \right\}$ happens. If each index has an error of at most ϵ, the sum of M terms may introduce an error of atmost $M\epsilon$. In addition, the distributed bipartite matching algorithm dBM$_\epsilon$ itself yields only an ϵ-optimal matching. This accounts for the term $(M+1)\epsilon$ above. The initial proof steps are similar to that in Theorem 2. We define $\tau_{ij,m}, \tau'_{ij,m}$ in the same way as in the proof of

Theorem 2 and let $\tilde{\tau}^{'}_{j,m} = \min\{\tau^{'}_{ij,m}, T\}$. Let $\mathbf{k}$ be the matching specified before. Then we get $\tilde{n}_{i,j}(T) \leq$

$$\begin{aligned}
1 &+ \sum_{m=1}^{T} |(\tilde{\tau}^{'}_{ij,m} - \tau_{ij,m})| I\Bigg\{ \sum_{i=1}^{M} g_{i,k_i^{**}}(\tau_{ij,m} - 1) \\
&\leq (M+1)\epsilon + \sum_{i=1}^{M} g_{i,k_i}(\tau_{ij,m} - 1) \Bigg\} \\
&\leq l + \sum_{m=1}^{T} |(\tilde{\tau}^{'}_{ij,m} - \tau_{ij,m})| I\Bigg\{ \sum_{i=1}^{M} g_{i,k_i^{**}}(\tau_{j,m} - 1) \\
&\leq (M+1)\epsilon + \sum_{i=1}^{M} g_{i,k_i}(\tau_{j,m} - 1), \tilde{n}_{i,j}(\tau_{ij,m} - 1) \geq l \Bigg\} \\
&\leq l + \sum_{m=1}^{T} \sum_{p=0}^{\infty} 2^p I\Bigg\{ \sum_{i=1}^{M} g_{i,k_i^{**}}(\tau_{ij,m} + 2^p - 1) \\
&\leq (M+1)\epsilon + \sum_{i=1}^{M} g_{i,k_i}(\tau_{ij,m} + 2^p - 1), \tilde{n}_{i,j}(\tau_{ij,m} - 1) \geq l \Bigg\} \\
&\leq l + \sum_{m=1}^{T} \sum_{p=0}^{\infty} 2^p I\Bigg\{ \sum_{i=1}^{M} g_{i,k_i^{**}}(m + 2^p - 1) \\
&\leq (M+1)\epsilon + \sum_{i=1}^{M} g_{i,k_i}(m + 2^p - 1), \tilde{n}_{i,j}(m - 1) \geq l \Bigg\} \\
&\leq l + \sum_{m=1}^{T} \sum_{p=0}^{\infty} 2^p I\Bigg\{ \sum_{i=1}^{M} \\
&\quad \left(\overline{X}_{i,k_i^{**}}(m + 2^p - 1) + c_{m+2^p-1, n_{i,k_i^{**}}(m+2^p-1)}\right) \\
&\leq (M+1)\epsilon + \sum_{i=1}^{M} \Bigg(\overline{X}_{i,k_i}(m + 2^p - 1) \\
&\quad + c_{m+2^p-1, n_{i,k_i}(m+2^p-1)}\Bigg), \tilde{n}_{i,j}(m - 1) \geq l \Bigg\} \\
&\leq l + \sum_{m=1}^{T} \sum_{p=0}^{\infty} 2^p I\Bigg\{ \min_{1 \leq s_{1,k_1^{**}}, \ldots, s_{M,k_M^{**}} < m+2^p} \\
&\quad \sum_{i=1}^{M} \left(\overline{X}_{i,k_i^{**}}(m + 2^p - 1) + c_{m+2^p-1, s_{i,k_i^{**}}}\right) \\
&\leq (M+1)\epsilon + \max_{l \leq s^{'}_{1,k_1}, \ldots, s^{'}_{M,k_M} < m+2^p} \\
&\quad \sum_{i=1}^{M} \left(\overline{X}_{i,k_i}(m + 2^p - 1) + c_{m+2^p-1, s^{'}_{i,k_i}}\right) \Bigg\} \\
&\leq l + \sum_{m=1}^{\infty} \sum_{p=0}^{\infty} 2^p \sum_{s_{1,k_1^{**}}=1}^{m+2^p} \cdots \sum_{s_{M,k_M^{**}}=1}^{m+2^p} \sum_{s^{'}_{1,k_1}=1}^{m+2^p} \cdots \sum_{s^{'}_{M,k_M}=1}^{m+2^p} \\
&\quad I\Bigg\{ \sum_{i=1}^{M} \left(\overline{X}_{i,k_i^{**}}(m + 2^p) + c_{m+2^p, s_{i,k_i^{**}}}\right) \\
&\leq (M+1)\epsilon + \sum_{i=1}^{M} \left(\overline{X}_{i,k_i}(m + 2^p) + c_{m+2^p, s^{'}_{i,k_i}}\right) \Bigg\}. \qquad (27)
\end{aligned}$$

Now, it is easy to observe that the event

$$\begin{aligned}
\Bigg\{ \sum_{i=1}^{M} &\left(\overline{X}_{i,k_i^{**}}(m + 2^p) + c_{m+2^p, s_{i,k_i^{**}}}\right) \leq (M+1)\epsilon \\
&+ \sum_{i=1}^{M} \left(\overline{X}_{i,k_i}(m + 2^p) + c_{m+2^p, s^{'}_{i,k_i}}\right) \Bigg\}
\end{aligned}$$

implies at least one of the following events:

$$\begin{aligned}
A_i &:= \left\{ \overline{X}_{i,k_i^{**}}(m + 2^p) \leq \mu_{i,k_i^{**}} - c_{m+2^p, s_{i,k_i^{**}}} \right\}, \\
B_i &:= \left\{ \overline{X}_{i,k_i}(m + 2^p) \geq \mu_{i,k_i} + c_{m+2^p, s^{'}_{i,k_i}} \right\}, \quad 1 \leq i \leq M, \\
C &:= \Bigg\{ \sum_{i=1}^{M} \mu_{i,k_i^{**}} < (M+1)\epsilon \\
&\quad + \sum_{i=1}^{M} \mu_{i,k_i} + 2 \sum_{i=1}^{M} c_{m+2^p, s^{'}_{i,k_i}} \Bigg\} \qquad (28)
\end{aligned}$$

Using the Chernoff-Hoeffding inequality, we get

$$\begin{aligned}
\mathbb{P}(A_i) &\leq (m + 2^p)^{-2(M+2)}, \\
\mathbb{P}(B_i) &\leq (m + 2^p)^{-2(M+2)}, \ 1 \leq i \leq M.
\end{aligned}$$

For $l \geq \left\lceil \frac{4M^2(M+2)\log T}{(\Delta_{min} - (M+1)\epsilon)^2} \right\rceil$, we get

$$\begin{aligned}
&\sum_{i=1}^{M} \mu_{i,k_i^{**}} - \sum_{i=1}^{M} \mu_{i,k_i} - (M+1)\epsilon - 2\sum_{i=1}^{M} c_{m+2^p, s^{'}_{i,k_i}} \\
&\geq \sum_{i=1}^{M} \mu_{i,k_i^{**}} - \sum_{i=1}^{M} \mu_{i,k_i} - (M+1)\epsilon \\
&\qquad -2M\sqrt{\frac{(M+2)\log(m + 2^p)}{l}} \\
&\geq \sum_{i=1}^{M} \mu_{i,k_i^{**}} - \sum_{i=1}^{M} \mu_{i,k_i} - (M+1)\epsilon - (\Delta_{min} - (M+1)\epsilon) \\
&\geq 0 \qquad (29)
\end{aligned}$$

where we used the fact that $(\Delta_{min} - (M+1)\epsilon) > 0$ by assumption. So, we get,

$$\begin{aligned}
&\mathbb{E}[\tilde{n}_{i,j}(T)] \\
&\leq \left\lceil \frac{4M^2(M+2)\log T}{(\Delta_{min} - (M+1)\epsilon)^2} \right\rceil \\
&\quad + \sum_{m=1}^{\infty} \sum_{p=0}^{\infty} 2^p \sum_{s_{1,k_1^{**}}=1}^{m+2^p} \cdots \sum_{s_{M,k_M^{**}}=1}^{m+2^p} \sum_{s^{'}_{1,k_1}=1}^{m+2^p} \cdots \sum_{s^{'}_{M,k_M}=1}^{m+2^p} \\
&\qquad 2M(m + 2^p)^{-2(M+2)} \\
&\leq \left\lceil \frac{4M^2(M+2)\log T}{(\Delta_{min} - (M+1)\epsilon)^2} \right\rceil + 2M \sum_{m=1}^{\infty} \sum_{p=0}^{\infty} 2^p (m + 2^p)^{-4} \\
&\leq \frac{4M^2(M+2)\log T}{(\Delta_{min} - (M+1)\epsilon)^2} + (2M + 1). \qquad (30)
\end{aligned}$$

Now, putting it all together, we get

$$\mathbb{E}[\tilde{n}(T)] = \sum_{i=1}^{M}\sum_{j=1}^{N}\mathbb{E}[\tilde{n}_{i,j}(T)]$$
$$\leq \frac{4M^3(M+2)N\log T}{(\Delta_{min}-(M+1)\epsilon)^2} + (2M+1)MN. \qquad (31)$$

Now, by the proof of Theorem 2 (c.f. equation(13),

$$\mathbb{E}[m(T)] \leq \mathbb{E}[\tilde{n}(T)](1+\log T).$$

We can now bound the regret,

$$\tilde{\mathcal{R}}_{\text{dUCB}_4}(T) = \sum_{k\in\mathcal{P}(N),k\neq k^{**}} \Delta_k \sum_{i=1}^{M}\mathbb{E}[\tilde{n}_{i,k_i}(T)] + C\mathbb{E}[m(T)]$$
$$\leq \Delta_{max} \sum_{k\in\mathcal{P}(N),k\neq k^{**}} \sum_{i=1}^{M}\mathbb{E}[\tilde{n}_{i,k_i}(T)] + C\mathbb{E}[m(T)]$$
$$= \Delta_{max}\mathbb{E}[\tilde{n}(T)] + C\mathbb{E}[m(T)].$$

For a general L, by Theorem 1 we get

$$\tilde{\mathcal{R}}_{\text{dUCB}_4}(T) \leq L\Delta_{max}\mathbb{E}[\tilde{n}(T)] + C(f(L))\mathbb{E}[m(T)]$$
$$\leq (L\Delta_{max} + C(f(L))(1+\log T))\mathbb{E}[\tilde{n}(T)].$$

Now, using the bound (31), we get the desired upper bound on the expected regret.

(ii) Since $\epsilon_t = f(L_t)$ is a monotonically decreasing function of L_t such that $\epsilon_t \to 0$ as $L_t \to \infty$, there exists a t_0 such that for $t > t_0$, $\epsilon_t < \epsilon_{min}$. We may get a linear regret upto time t_0 but after that by the analysis of Theorem 2, regret grows only sub-linearly. Since $C(\cdot)$ is monotonically increasing, $C(f(L_T)) \geq C(f(L_t)), \forall t \leq T$, we get the desired result. The last part is illustrative and can be trivially established using the obtained bound on the regret in (ii). ■

Remarks. 1. The UCB$_2$ algorithm described in [4] performs computations only at exponentially spaced time epochs. So, it is natural to imagine that a decentralized algorithm based on it could be developed, and get a better regret bound. Unfortunately, the single player UCB$_2$ algorithm has an obvious weakness: regret is linear in the number of arms. Thus, the decentralized/combinatorial extension of UCB$_2$ would yield regret growing exponentially in the number of players and arms. We use a similar index but a different scheme, allowing us to achieve *poly-log* regret growth and a linear memory requirement for each player.

VI. The Decentralized MAB Problem With Markovian Rewards

Now, we consider the decentralized MAB problem with M players and N arms where the rewards obtained each time when an arm is pulled are not i.i.d. but come from a Markov chain. The reward that player i gets from arm j (when there is no collision) X_{ij}, is modelled as an irreducible, aperiodic, reversible Markov chain on a finite state space $\mathcal{X}^{i,j}$ and represented by a transition probability matrix $P^{i,j} := \left(p^{i,j}_{x,x'} : x, x' \in \mathcal{X}^{i,j}\right)$. Assume that $\mathcal{X}^{i,j} \in (0,1]$. Let $X_{i,j}(1), X_{i,j}(2), \ldots$ denote the successive rewards from arm j for player i. All arms are mutually independent for all players. Let $\pi^{i,j} := \left(\pi^{i,j}_x, x \in \mathcal{X}^{i,j}\right)$ be the stationary distribution of the Markov chain $P^{i,j}$. The mean reward from arm j for player i is defined as $\mu_{i,j} := \sum_{x\in\mathcal{X}^{i,j}} x\pi^{i,j}_x$. Note that the Markov chain represented by $P^{i,j}$ makes a state transition only when player i plays arm j. Otherwise, it remains *rested.* As described in the previous section, $n_i(t)$ is the number of successful plays (without collisions) of player i by frame t, $n_{i,j}(t)$ is the number of times player i picks arm j successfully by frame t and $\overline{X}_{i,j}(t)$ is the sample mean of rewards from arm j for player i from $n_{i,j}(t)$ samples. Denote $\Delta_{min} := \min_{\mathbf{k}\in\mathcal{P}(N),\mathbf{k}\neq\mathbf{k}^{**}} \Delta_{\mathbf{k}}$ and $\Delta_{max} := \max_{\mathbf{k}\in\mathcal{P}(N)} \Delta_{\mathbf{k}}$. Denote $\pi_{min} := \min_{1\leq i\leq M,1\leq j\leq N,x\in\mathcal{X}^{i,j}} \pi^{i,j}_x$, $x_{max} := \max_{1\leq i\leq M,1\leq j\leq N,x\in\mathcal{X}^{i,j}} x$ and $x_{min} := \min_{1\leq i\leq M,1\leq j\leq N,x\in\mathcal{X}^{i,j}} x$. Let $\hat{\pi}^{i,j}_x := \max\{\pi^{i,j}_x, 1-\pi^{i,j}_x\}$ and $\hat{\pi}_{max} := \max_{1\leq i\leq M,1\leq j\leq N,x\in\mathcal{X}^{i,j}} \hat{\pi}^{i,j}_x$. Let $|\mathcal{X}^{i,j}|$ denote the cardinality of the state space $\mathcal{X}^{i,j}$, $|\mathcal{X}|_{max} := \max_{1\leq i\leq M,1\leq j\leq N} |\mathcal{X}^{i,j}|$. Let $\rho^{i,j}$ be the eigenvalue gap, $1-\lambda_2$, where λ_2 is the second largest eigenvalue of the matrix ${P^{i,j}}^2$. Denote $\rho_{max} := \max_{1\leq i\leq M,1\leq j\leq N} \rho^{i,j}$ and $\rho_{min} := \min_{1\leq i\leq M,1\leq j\leq N} \rho^{i,j}$.

The total reward obtained by time T is

$$S_T = \sum_{j=1}^{N}\sum_{i=1}^{M}\sum_{s=1}^{n_{i,j}(T)} X_{i,j}(s)$$

and the regret is

$$\tilde{\mathcal{R}}_{M,\alpha}(T) := T\sum_{i=1}^{M}\mu_{i,k_i^{**}} - \mathbb{E}_\alpha\left[\sum_{j=1}^{N}\sum_{i=1}^{M}\sum_{s=1}^{n_{i,j}(T)} X_{i,j}(s)\right] + C\mathbb{E}[m(T)]. \quad (32)$$

Define the index

$$g_{i,j}(t) := \overline{X}_{i,j}(t) + \sqrt{\frac{\kappa\log n_i(t)}{n_{i,j}(t)}} \qquad (33)$$

where κ be any constant such that $\kappa > (112 + 56M)|\mathcal{X}|^2_{max}/\rho_{min}$.

We need the following lemma to prove the regret bound.

Lemma 4: If the reward of each player-arm pair (i,j) is given by a Markov chain, satisfying the properties of Lemma 1, then under any policy α

$$\tilde{\mathcal{R}}_{M,\alpha}(T) \leq \sum_{k\in\mathcal{P}(N),k\neq k^{**}} \Delta_k\mathbb{E}[n^k(T)] + C\mathbb{E}[m(T)] + \tilde{K}_{\mathcal{X},P}$$

where $n^k(T)$ is the number of times that the matching k occurred by the time T and $\tilde{K}_{\mathcal{X},P}$ is defined as $\tilde{K}_{\mathcal{X},P} = \sum_{j=1}^{N}\sum_{i=1}^{M}\sum_{x\in\mathcal{X}^{i,j}} x/\pi^j_{min}$.

Proof: Let $(X_{i,j}(1), X_{i,j}(2), \ldots)$ denote the successive rewards for player i from arm j. Let $\mathcal{F}^{i,j}_t$ denote the σ-algebra generated by $(X_{i,j}(1), \ldots, X_{i,j}(t))$, $\mathcal{F}^{i,j} = \vee_{t\geq 1}\mathcal{F}^{i,j}_t$ and $\mathcal{G}^{i,j} = \vee_{(k,l)\neq(i,j)} F^{k,l}$. Since arms are independent, $\mathcal{G}^{i,j}$ is independent of $\mathcal{F}^{i,j}$. Clearly, $n_{i,j}(T)$ is a stopping time with respect to $\mathcal{F}^{i,j} \vee \mathcal{G}^{i,j}_T$. The total reward is

$$S_T = \sum_{j=1}^{N}\sum_{i=1}^{M}\sum_{t=1}^{n_{i,j}(T)} X_{i,j}(t) = \sum_{j=1}^{N}\sum_{i=1}^{M}\sum_{x\in\mathcal{X}^{i,j}} xN(x, n_{i,j}(T))$$

where $N(x, n_{i,j}(T)) := \sum_{t=1}^{n_{i,j}(T)} I\{X_{i,j}(t) = x\}$. Taking expectations and using the Lemma 1,

$$\left|\mathbb{E}[S_T] - \sum_{j=1}^{N}\sum_{i=1}^{M}\sum_{x\in\mathcal{X}^{i,j}} x\pi_x^{i,j}\mathbb{E}[n_{i,j}(T)]\right| \le \sum_{j=1}^{N}\sum_{i=1}^{M}\sum_{x\in\mathcal{X}^{i,j}} x/\pi_{min}^{i,j}$$

which implies,

$$\left|\mathbb{E}[S_T] - \sum_{j=1}^{N}\sum_{i=1}^{M}\mu_{i,j}\mathbb{E}[n_{i,j}(T)]\right| \le \tilde{K}_{\mathcal{X},P}$$

where $\tilde{K}_{\mathcal{X},P} = \sum_{j=1}^{N}\sum_{i=1}^{M}\sum_{x\in\mathcal{X}^{i,j}} x/\pi_{min}^{i,j}$. Now,

$$\begin{aligned}\sum_{j=1}^{N}\sum_{i=1}^{M} \quad & \mu_{i,j}\mathbb{E}[n_{i,j}(T)] \\ &= \sum_{j=1}^{N}\sum_{i=1}^{M}\sum_{k\in\mathcal{P}(N),(i,j)\in k}\mu_{i,k_i}\mathbb{E}[n_{i,k_i}(T)] \\ &= \sum_{k\in\mathcal{P}(N)}\sum_{i=1}^{M}\mu_{i,k_i}\mathbb{E}[n_{i,k_i}(T)] \\ &= \sum_{k\in\mathcal{P}(N)}\mu^k\mathbb{E}[n^k(T)]\end{aligned}$$

where $\mu^k = \sum_{i=1}^{M}\mu_{i,k_i}$. Since regret is defined as in the equation (32),

$$\begin{aligned}\Bigg|\tilde{\mathcal{R}}_{M,\alpha}(T) - (T\mu^{**} - &\sum_{k\in\mathcal{P}(N),(i,j)\in k}\mu_{i,k_i}\mathbb{E}[n_{i,k_i}(T)] \\ &+C\mathbb{E}_\alpha[m(T)])\Bigg| \le \tilde{K}_{\mathcal{X},P}. \qquad (34)\end{aligned}$$

■

The main result of this section is the following.

Theorem 7: (i) Let $\epsilon > 0$ be the precision of the bipartite matching algorithm and the precision of the index representation. If Δ_{min}, $|\mathcal{X}|_{max}$ and ρ_{min} are known, choose $\epsilon > 0$ such that $\epsilon < \Delta_{min}/(M+1)$ and $\kappa > (112+56M)|\mathcal{X}|_{max}^2/\rho_{min}$. Let L be the length of a frame. Then, the expected regret of the dUCB$_4$ algorithm with index (33) for the decentralized MAB problem with Markovian rewards and per computation cost C is given by

$$\begin{aligned}\tilde{\mathcal{R}}_{M,\mathtt{dUCB}_4}(T) &\le (L\Delta_{max} + C(f(L))(1+\log T)) \\ &\cdot\left(\frac{4M^3\kappa N\log T}{(\Delta_{min}-(M+1)\epsilon)^2} + (2MD+1)MN\right) + \tilde{K}_{\mathcal{X},P}.\end{aligned}$$

Thus, $\tilde{\mathcal{R}}_{M,\mathtt{dUCB}_4}(T) = O(\log^2 T)$.

(ii) If Δ_{min}, $|\mathcal{X}|_{max}$ and ρ_{min} are unknown, denote $\epsilon_{min} = \Delta_{min}/(2(M+1))$ and let $L_t \to \infty$ as $t \to \infty$. Also, choose a positive monotone sequence $\{\kappa_t\}$ such that $\kappa_t \to \infty$ as $t \to \infty$ and $\kappa_t \le t$. Then, $\tilde{\mathcal{R}}_{M,\mathtt{dUCB}_4}(T) = O(C(f(L_T))\kappa_T\log^2 T)$. Thus by choosing an arbitrarily slowly increasing sequences, we can make the regret arbitrarily close to $\log^2 T$.

Proof: (i) We skip the initial steps as they are same as in the proof of Theorem 6. We start by bounding $\tilde{n}_{i,j}(T)$ as defined in the proof of Theorem 6. Then, from equation (27), we get $\tilde{n}_{i,j}(T)$

$$\begin{aligned}&\le l + \sum_{m=1}^{\infty}\sum_{p=0}^{\infty}2^p\sum_{s_{1,k_1^{**}}=1}^{m+2^p}\cdots\sum_{s_{M,k_M^{**}}=1}^{m+2^p}\sum_{s'_{1,k_1}=1}^{m+2^p}\cdots \\ &\sum_{s'_{M,k_M}=1}^{m+2^p} I\Bigg\{\sum_{i=1}^{M}\left(\overline{X}_{i,k_i^{**}}(m+2^p) + c_{m+2^p,s_{i,k_i^{**}}}\right) \\ &\le (M+1)\epsilon + \sum_{i=1}^{M}\left(\overline{X}_{i,k_i}(m+2^p) + c_{m+2^p,s'_{i,k_i}}\right)\Bigg\}\end{aligned}$$

Now, the event in the parenthesis $\{\cdot\}$ above implies at least one of the events (A_i, B_i, C, D) given in the display (28). From the proof of Theorem 4 (equations (21, 22),

$$\begin{aligned}\mathbb{P}(A_i) &\le D(m+2^p)^{-\frac{\kappa\rho_{min}}{28|\mathcal{X}|_{max}|^2}}, \\ \mathbb{P}(B_i) &\le D(m+2^p)^{-\frac{\kappa\rho_{min}}{28|\mathcal{X}|_{max}|^2}}, \quad 1 \le i \le M.\end{aligned}$$

Similar to the steps in display (29), we can show that the event C is false. Also, the event D is false by assumption. So, similar to the proof of the Theorem 6 (c.f. display (30) we get,

$$\begin{aligned}\mathbb{E}[\tilde{n}_{i,j}(T)] &\le \left\lceil\frac{4M^2\kappa\log T}{(\Delta_{min}-(M+1)\epsilon)^2}\right\rceil \\ &+\sum_{m=1}^{\infty}\sum_{p=0}^{\infty}2^p\sum_{s_{1,k_1^{**}}=1}^{m+2^p}\cdots\sum_{s_{M,k_M^{**}}=1}^{m+2^p} \\ &\sum_{s'_{1,k_1}=1}^{m+2^p}\cdots\sum_{s'_{M,k_M}=1}^{m+2^p} 2MD(m+2^p)^{-\frac{\kappa\rho_{min}}{28|\mathcal{X}|_{max}|^2}} \\ &\le \left\lceil\frac{4M^2\kappa\log T}{(\Delta_{min}-(M+1)\epsilon)^2}\right\rceil \\ &+2MD\sum_{m=1}^{\infty}\sum_{p=0}^{\infty}2^p(m+2^p)^{-\frac{\kappa\rho_{min}-56M|\mathcal{X}|_{max}^2}{28|\mathcal{X}|_{max}^2}} \\ &\le \frac{4M^2\kappa\log T}{(\Delta_{min}-(M+1)\epsilon)^2} + (2MD+1).\end{aligned}$$

when $\kappa > (112+56M)|\mathcal{X}|_{max}^2/\rho_{min}$. Now, putting it all together, we get

$$\begin{aligned}\mathbb{E}[\tilde{n}(T)] &= \sum_{i=1}^{M}\sum_{j=1}^{N}\mathbb{E}[\tilde{n}_{i,j}(T)] \\ &\le \frac{4M^3\kappa N\log T}{(\Delta_{min}-(M+1)\epsilon)^2} + (2MD+1)MN. \qquad (35)\end{aligned}$$

Now, by proof of the Theorem 2 (equation (13)),

$$\mathbb{E}[m(T)] \le \mathbb{E}[\tilde{n}(T)](1+\log T).$$

We can now bound the regret,

$$\begin{aligned}\tilde{\mathcal{R}}_{M,\texttt{dUCB}_4}(T) &= \sum_{k\in\mathcal{P}(N),k\neq k^{**}} \Delta_k \sum_{i=1}^{M} \mathbb{E}[\tilde{n}_{i,k_i}(T)] \\ &\quad + C\mathbb{E}[m(T)] + \tilde{K}_{\mathcal{X},P} \\ &\leq \Delta_{max} \sum_{k\in\mathcal{P}(N),k\neq k^{**}} \sum_{i=1}^{M} \mathbb{E}[\tilde{n}_{i,k_i}(T)] \\ &\quad + C\mathbb{E}[m(T)] + \tilde{K}_{\mathcal{X},P} \\ &= \Delta_{max}\mathbb{E}[\tilde{n}(T)] + C\mathbb{E}[m(T)] + \tilde{K}_{\mathcal{X},P}.\end{aligned}$$

For a general L, by Theorem 1

$$\begin{aligned}\tilde{\mathcal{R}}_{M,\texttt{dUCB}_4}(T) &\leq L\Delta_{max}\mathbb{E}[\tilde{n}(T)] + C(f(L))\mathbb{E}[m(T)] \\ &\quad + \tilde{K}_{\mathcal{X},P}. \\ &\leq (L\Delta_{max} + C(f(L))(1+\log T))\mathbb{E}[\tilde{n}(T)] + \tilde{K}_{\mathcal{X},P}.\end{aligned}$$

Now, using the bound (35), we get the desired upper bound on the expected regret.

(ii) This can now easily be obtained using the above and following Theorem 6. ■

VII. Distributed Bipartite Matching: Algorithm and Implementation

In the previous section, we referred to an unspecified distributed algorithm for bipartite matching dBM, that is used by the $\texttt{dUCB}_4$ algorithm. We now present one such algorithm, namely, Bertsekas' auction algorithm [17], and its distributed implementation. We note that the presented algorithm is not the only one that can be used. The $\texttt{dUCB}_4$ 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 $\sum_i \mu_{i,k^{**}(i)} - \sum_i \mu_{i,k^*(i)} \leq \epsilon$.

Here, $2\max_j$ 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 [17] establishes that Bertsekas' auction algorithm will find the ϵ-optimal matching in a finite number of steps.

Lemma 5: [17] Given $\epsilon > 0$, Algorithm 3 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.

The temporal structure of the $\texttt{dUCB}_4$ algorithm is such that time is divided into frames of length L. Each frame is either a *decision* frame, or an *exploitation* frame. In the exploitation frame, each player plays the arm it was allocated in the last decision frame. The distributed bipartite matching algorithm (e.g. based on Algorithm 3), is run in the decision frame. The decision frame has an interrupt phase of length M and negotiation phase of length $L-M$. We now describe an implementation structure for these phases in the decision frame.

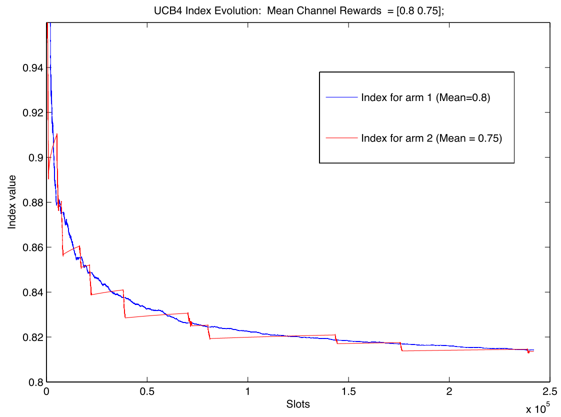

Fig. 2. Indices track each other closely in $\texttt{UCB}_4$.

Algorithm 3 $\texttt{dBM}_\epsilon$ (Bertsekas Auction Algorithm)

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

Interrupt Phase: The interrupt phase can be implemented very easily. It has length M time slots. On a pre-determined channel, each player by turn transmits a '1' if the arm with which it is now matched has changed, '0' otherwise. If any user transmits a '1', everyone knows that the matching has changed, and they reset their counter $\eta = 1$.

Negotiation Phase: The information needed to be exchanged to compute an ϵ-optimal matching is done in the negotiation phase. We first provide a *packetized implementation* of the negotiation phase. The negotiation phase consists of J subframes of length M each (See Fig. 1). In each subframe, the users transmit a packet by turn. The packet contains bid information: (channel number, bid value). Since all users transmit by turn, all the users know the bid values by the end of the subframe, and can compute the new allocation, and the prices independently. The length of the subframe J determines the precision ϵ of the distributed bipartite matching algorithm. Note that in the packetized implementation, $\epsilon_1 = 0$, i.e., bid values can be computed exactly, and for a given ϵ_2, we can determine J, the number of rounds the dBM algorithm 3 runs for, and returns an ϵ_2-optimal matching.

If a packetized implementation is not possible, we can give a *physical implementation*. 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. The whole negotiation phase is again divided into J sub-frames. In each sub-frame, each user transmits by turn. It simply transmits $\lceil \log M \rceil$ bits to indicate a channel number, and then $\lceil \log 1/\epsilon_1 \rceil$ bits to indicate its bid value to precision ϵ_1. The number of such sub-frames J is again chosen so that the dBM algorithm (based on Algorithm 3) returns an ϵ_2-optimal matching.

VIII. Conclusion

We have proposed a dUCB$_4$ algorithm for decentralized learning in multi-armed bandit problems that achieves a regret of near-$O(\log^2(T))$. The motivation for this came from opportunistic spectrum access for cognitive radios where users do not have a dedicated control channel for coordination.

The key contributions of thus paper is to lay down a framework in which to think about decentralized learning with heterogeneous players. Namely, think of this as an online decentralized bipartite matching problem in which players exchange "bids". However, since there is no dedicated channel for message exchange, only "quantized" bids can be exchanged which affects both the optimality of the bipartite matching, as well as learning. Moreover, this message exchange comes at a "cost". Modularizing the problem in this manner is what made it tractable. Within this framework, we proposed using the dUCB$_4$ schedule for exploration-exploitation in which case $O(\log^2 T)$ bound on expected regret was obtained. However, it may be possible to use other schedules. In particular, the deterministic schedule such as proposed for the DSEE algorithm [23] may be used and may result in an improved $O(\log T)$ bound on expected regret.

A very important and fundamental question is whether a "fully" decentralized learning algorithm (without any coordination) even exists, and if so, what would its lower bound be. This has very interesting connections with distributed universal coding, which itself is a difficult problem. However, should this be achievable, it will yield very important insight about whether explicit communication/side information is necessary for distributed learning, or not, as in distributed source coding via Slepian-Wolff coding, which established that side information is not needed. In future work, we hope to be able to provide a lower bound by considering a more restricted, stylized setting by, for example, assuming availability of side information. Other directions for future work would be extension to distributions with heavy tails [24] and restless bandits [8], [10], [14].

Acknowledgment

The authors benefited greatly from discussions with Yi Gai and Bhaskar Krishnamachari of the University of Southern California, Tsachy Weismann of Stanford University and Alon Orlitsky and Young-Han Kim of UC San Diego.

References

[1] T. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Adv. Appl. Math.*, vol. 6, no. 1, pp. 4–22, 1985.

[2] 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, Nov. 1987.

[3] R. Agrawal, "Sample mean based index policies with $(O(\log n))$ regret for the multi-armed bandit problem," *Adv. Appl. Probab.*, vol. 27, no. 4, pp. 1054–1078, 1995.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2, pp. 235–256, 2002.

[5] V. Anantharam, P. Varaiya, and J. Walrand, "Asymptotically efficient allocation rules for the multi-armed bandit problem with multiple plays—Part II: Markovian rewards," *IEEE Trans. Autom. Control*, vol. 32, no. 11, pp. 977–982, Nov. 1987.

[6] C. Tekin and M. Liu, "Online algorithms for the multi-armed bandit problem with Markovian rewards," in *Proc. 48th Annu. Allerton Conf. Commun., Control, Comput.*, 2010, pp. 1675–1682.

[7] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of optimal queuing network control," *Math. Oper. Res.*, vol. 24, no. 2, pp. 293–305, 1999.

[8] C. Tekin and M. Liu, "Online learning of rested and restless bandits," *IEEE Trans. Inf. Theory*, vol. 58, no. 8, pp. 5588–5611, Aug. 2012.

[9] W. Dai, Y. Gai, and B. Krishnamachari, "Efficient online learning for opportunistic spectrum access," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 3086–3090.

[10] H. Liu, K. Liu, and Q. Zhao, "Learning in a changing world: Restless multi-armed bandit with unknown dynamics," *IEEE Trans. Inf. Theory*, vol. 59, no. 3, pp. 1902–1916, Mar. 2013.

[11] W. Dai, Y. Gai, B. Krishnamachari, and Q. Zhao, "The non-Bayesian restless multi-armed bandit: A case of near-logarithmic regret," in *Proc. IEEE ICASSP*, Sep. 2011, pp. 2940–2943.

[12] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1466–1478, Oct. 2012.

[13] Y. Gai, B. Krishnamachari, and M. Liu, "On the combinatorial multi-armed bandit problem with Markovian rewards," in *Proc. IEEE GLOBECOM*, Dec. 2011, pp. 1–6.

[14] K. Liu and Q. Zhao, "Indexability of restless bandit problems and optimality of whittle index for dynamic multichannel access," *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5547–5567, Nov. 2010.

[15] A. Anandkumar, N. Michael, A. Tang, and A. Swami, "Distributed algorithms for learning and cognitive medium access with logarithmic regret," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 731–745, Apr. 2011.

[16] Y. Gai and B. Krishnamachari, "Decentralized online learning algorithms for opportunistic spectrum access," in *Proc. IEEE GLOBECOM*, Dec. 2011, pp. 1–6.

[17] D. P. Bertsekas, "Auction algorithms for network flow problems: A tutorial introduction," *Comput. Optim. Appl.*, vol. 1, no. 1, pp. 7–66, 1992.

[18] E. Hossain and V. K. Bhargava, *Cognitive Wireless Communication Networks*. New York, NY, USA: Springer-Verlag, 2007.

[19] S. Boucheron, G. Lugosi, and P. Massart, *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford, U.K., Oxford Univ. Press, 2013.

[20] P. Lezaud, "Chernoff-type bound for finite Markov chains," *Ann. Appl. Probab.*, vol. 8, no. 3, pp. 849–867, 1998.

[21] 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.

[22] M. M. Zavlanos, L. Spesivtsev, and G. J. Pappas, "A distributed auction algorithm for the assignment problem," in *Proc. 47th IEEE CDC*, Dec. 2008, pp. 1212–1217.

[23] S. Vakili, K. Liu, and Q. Zhao, "Deterministic sequencing of exploration and exploitation for multi-armed bandit problems," *IEEE J. Sel. Topics Signal Process.*, vol. 59, no. 3, pp. 1902–1916, Oct. 2013.

[24] K. Liu and Q. Zhao, "Multi-armed bandit problems with heavy-tailed reward distributions," in *Proc. Allerton Conf. Commun., Control, Comput.*, 2011, pp. 485–492.

Dileep Kalathil received a B. Tech in ECE from the National Institute of Technology, Calicut, India in 2006 and an M.Tech from IIT Madras, India in 2008. He worked as a wireless research engineer in the Center of Excellence in Wireless Technology for an year. Since 2009, he is a PhD candidate at USC where he has been an Annenberg Fellow from 2009-11. His research interests include game theory, decentralized control, online optimization and applied probability.

Naumaan Nayyar received a B.Tech in Electrical Engineering from the Indian Institute of Technology, Bombay, India in 2010. He subsequently began his doctoral studies at the University of Southern California as a Provost Fellow. His research interests are in the areas of learning, online optimization, statistical inference and stochastic control theory.

Rahul Jain is an associate professor and the K. C. Dahlberg Early Career Chair in the EE department at the University of Southern California. He received his PhD in EECS and an MA in Statistics from the University of California, Berkeley, his B.Tech from IIT Kanpur. He is winner of numerous awards including the NSF CAREER award, an IBM Faculty award and the ONR Young Investigator award. His research interests span wireless communications, network economics and game theory, queueing theory, power systems and stochastic control theory.