On the Combinatorial Multi-Armed Bandit Problem with Markovian Rewards

Yi Gai*, Bhaskar Krishnamachari* and Mingyan Liu‡
*Ming Hsieh Department of Electrical Engineering, University of Southern California, CA 90089, USA
‡Department of Electrical Engineering and Computer Science, University of Michigan, MI 48109, USA
Email: {ygai,bkrishna}@usc.edu; mingyan@eecs.umich.edu

***Abstract*—We consider a combinatorial generalization of the classical multi-armed bandit problem that is defined as follows. There is a given bipartite graph of M users and $N \geq M$ resources. For each user-resource pair (i, j), there is an associated state that evolves as an aperiodic irreducible finite-state Markov chain with unknown parameters, with transitions occurring each time the particular user i is allocated resource j. The user i receives a reward that depends on the corresponding state each time it is allocated the resource j. The system objective is to learn the best matching of users to resources so that the long-term sum of the rewards received by all users is maximized. This corresponds to minimizing regret, defined here as the gap between the expected total reward that can be obtained by the best-possible static matching and the expected total reward that can be achieved by a given algorithm. We present a polynomial-storage and polynomial-complexity-per-step matching-learning algorithm for this problem. We show that this algorithm can achieve a regret that is uniformly arbitrarily close to logarithmic in time and polynomial in the number of users and resources. This formulation is broadly applicable to scheduling and switching problems in communication networks including cognitive radio networks and significantly extends prior results in the area.**

I. Introduction

Multi-armed bandit problems provide a fundamental approach to learning under stochastic rewards, and find rich applications in a wide range of networking contexts, such as medium access in cognitive radio networks [1]–[3]. In the simplest, classic non-Bayesian version of the problem, studied by Lai and Robbins [4], there are K independent arms, each generating stochastic rewards that are i.i.d. over time. The player is unaware of the parameters for each arm, and must use some policy to play the arms in such a way as to maximize the cumulative expected reward over the long term. The policy's performance is measured in terms of its "regret", defined as the gap between the the expected reward that could be obtained by an omniscient user that knows the parameters for the stochastic rewards generated by each arm and the expected cumulative reward of that policy. It is of interest to characterize the growth of regret with respect to time as well as with respect to the number of arms/players. Intuitively, if the regret grows sublinearly over time, the time-averaged regret tends to zero.

There is inherently a tradeoff between exploration and exploitation in the learning process in a multi-armed bandit problem: on the one hand all arms need to be sampled periodically by the policy used, to ensure that the "true" best arm is found; on the other hand, the policy should play the arm that is considered to be the best often enough to accumulate rewards at a good pace.

In this paper, we formulate a novel combinatorial generalization of the multi-armed bandit problem that allows for Markovian rewards and propose an efficient policy for it. In particular, there is a given bipartite graph of M users and $N \geq M$ resources. For each user-resource pair (i, j), there is an associated state that evolves as an aperiodic irreducible finite-state Markov chain with unknown parameters, with transitions occurring each time the particular user i is allocated resource j. The user i receives a reward that depends on the corresponding state each time it is allocated the resource j. A key difference from the classic multi-armed bandit is that each user can potentially see a different reward process for the same resource. If we therefore view each possible matching of users to resources as an arm, then we have a super-exponential number of arms with dependent rewards. Thus, this new formulation is significantly more challenging than the traditional multi-armed bandit problems.

Because our formulation allows for user-resource matching, it could be potentially applied to a diverse range of networking settings such as switching in routers (where inputs need to be matched to outputs) or frequency scheduling in wireless networks (where nodes need to be allocated to channels) or for server assignment problems (for allocating computational resources for various processes), etc., with the objective of learning as quickly as possible so as to maximize the usage of the best options. For instance, our formulation is general enough to be applied to the channel allocation problem in cognitive radio networks considered in [1] if the rewards for each user-channel pair come from a discrete set and are i.i.d. over time (which is a special case of Markovian rewards).

Our main contribution in this work is the design of a novel policy for this problem that we refer to Matching Learning for Markovian Rewards (MLMR). Since we treat each possible matching of users to resources as an arm, the number of arms in our formulation grows super-exponentially. However, MLMR uses only polynomial storage, and requires only polynomial computation at each step. We analyze the regret for this policy with respect to the best possible static matching, and show that it is uniformly logarithmic over time under some restrictions on the underlying Markov process. We also show that when these restrictions are removed, the regret can still be made arbitrarily close to logarithmic with respect to time. In either case, the regret is polynomial in the number of users and resources.

There has been relatively less work on multi-armed bandits with Markovian rewards. Anantharam *et al.* [5] wrote one of the earliest papers with such a setting. They proposed a policy to pick m out of the N arms each time slot and prove the lower bound and the upper bound on regret. However, the rewards in their work are assumed to be generated by rested Markov chains with transition probability matrices defined by a single parameter θ with identical state spaces. Also, the result for the upper bound is achieved only asymptotically. For the case of single users and independent arms, a recent work by Tekin and Liu [6] has extended the results in [5] to the case with no requirement for a single parameter and identical state spaces across arms. They propose to use UCB1 from [7] for the multi-armed bandit problem with Markovian rewards and prove a logarithmic upper bound on the regret under some conditions on the Markov chain. We use elements of the proof from [6] in this work, which is however quite different in its combinatorial matching formulation (which allows for dependent arms).

The rest of the paper is organized as follows. In section II we present the problem formulation. In section III we present a polynomial-storage polynomial-time-per-step learning policy, which we refer to as MLMR. We analyze the regret for this policy in section IV and show that it yields a bound on the regret that is uniformly logarithmic over time and polynomial in the number of users and resources under certain conditions on the Markov chains describing the state evolution for the arms. We then show that the regret can still be arbitrarily close to logarithmic with respect to time when no knowledge is available. We present some examples and simulations in section V, and conclude with some comments and ideas for future work in section VI.

II. Problem Formulation

We consider a bipartite graph with M users and N resources predefined by some application. Time is slotted and is indexed by n. At each decision period (also referred to interchangeably as time slot), each of the M users is assigned a resource with some policy.

For each user-resource pair (i, j), there is an associated state that evolves as an aperiodic irreducible finite-state Markov chain with unknown parameters. When user i is assigned resource j, assuming there are no other conflicting users assigned this resource, i is able to receive a reward that depends on the corresponding state each time it is allocated the resource j. We denote the state space as $S_{i,j} = \{z_1, z_2, \dots, z_{|S_{i,j}|}\}$. The state of the Markov chain for each user-resource pair (i, j) evolves only when resource j is allocated to user i. We assume the Markov chains for different user-resource pairs are mutually independent. The reward got by user i while allocated resource j on state $z \in S_{i,j}$ is denoted as $\theta_z^{i,j}$, which is also unknown to the users. We denote $\mathbf{P}_{i,j} = \{p_{i,j}(z_a, z_b)\}_{z_a, z_b \in S_{i,j}}$ as the transition probability matrix for the Markov chain (i, j). Denote $\pi_z^{i,j}$ as the steady state distribution for state z. The mean reward got by user i on resource j is denoted as $\mu_{i,j}$. Then we have $\mu_{i,j} = \sum_{z \in S_{i,j}} \theta_z^{i,j} \pi_z^{i,j}$. The set of all mean rewards is denoted as $\boldsymbol{\mu} = \{\mu_{i,j}\}$.

We denote $Y_{i,j}(n)$ as the actual reward obtained by a user i if it is assigned resource j at time n. We assume that $Y_{i,j}(n) = \theta_{i,j}^{z(n)}$, if user i is the only occupant of resource j at time n where $z(n)$ is the state of Markov chain associated with (i, j) at time n. Else, if multiple users are allocated resource j, then we assume that, due to interference, at most one of the conflicting users j' gets reward $Y_{i,j'}(n) = \theta_{i,j'}^{z'(n)}$ where $z'(n)$ is the state of Markov chain associated with (i, j') at time n, while the other users on the resources $j \neq j'$ get zero reward, i.e., $Y_{i,j}(n) = 0$. This interference model covers scenarios in many networking settings.

Due to the fact that allocating more than one user to a resource is always worse than assigning each a different resource in terms of sum-throughput, we will focus on collision-free policies that assign all users distinct resources, which we will refer to as a permutation or matching. There are $P(N, M)$ such permutations when $N \geq M$, and $P(M, N)$ such permutations when $N < M$.

We formulate our problem as a combinatorial multi-armed bandit, in which each arm corresponds to a matching of the users to resources. We can represent the arm corresponding to a permutation k ($1 \leq k \leq P(N, M)$, or $1 \leq k \leq P(M, N)$) as the index set $\mathcal{A}_k = \{(i, j) : (i, j) \text{ is in permutation } k\}$. The stochastic reward for choosing arm k at time n under policy α is then given as

$$Y_{\alpha(n)}(n) = \sum_{(i,j) \in \mathcal{A}_{\alpha(n)}} Y_{i,j}(n) = \sum_{(i,j) \in \mathcal{A}_{\alpha(n)}} \theta_{i,j}^{z_{\alpha(n)}}.$$

Note that different from most prior work on multi-armed bandits, this combinatorial formulation results in dependence across arms that share common components.

A key metric of interest in evaluating a given policy for this problem is *regret*, which is defined as the difference between the expected reward that could be obtained by the best-possible static matching, and that obtained by the given policy. It can be expressed as:

$$R^\alpha(n) = n\mu^* - E^\alpha[\sum_{t=1}^{n} Y_{\alpha(t)}(t)], \tag{1}$$

where $\mu^* = \max_k \sum_{(i,j) \in \mathcal{A}_k} \mu^{i,j}$, the expected reward of the optimal arm, is the expected sum-weight of the maximum weight matching of users to resources with $\mu_{i,j}$ as the weight.

We are interested in designing policies for this combinatorial multi-armed bandit problem with Markovian rewards that perform well with respect to regret. Intuitively, we would like the regret $R^\alpha(n)$ to be as small as possible. If it is sub-linear with respect to time n, the time-averaged regret will tend to zero.

III. Matching Learning for Markovian Rewards

A straightforward idea for the combinatorial multi-armed bandit problem with Markovian rewards is to treat each matching as an arm, apply UCB1 policy (given by Auer *et al.* [7]) directly, and ignore the dependencies across the different arms. For each arm k, two variables are stored and updated: the time average of all the observation values of arm k and the number of times that arm k has been played up to the current time slot. The UCB1 policy makes decisions based on this information alone.

However, there are several problems that arise in applying UCB1 directly in the above setting. We note that UCB1 requires both the storage and computation time that are linear in the number of arms. Since the number of arms in this formulation grows as $P(N,M)$, it is highly unsatisfactory. Also, the upper-bound of regret given in [6] will not work anymore since the rewards across arms are not independent anymore and the states of an arm may involve even when this arm is not played. No existing analytical result on the upper-bound of regret can be applied directly in this setting to the best of our knowledge.

So we are motivated to propose a policy which more efficiently stores observations from correlated arms and exploits the correlations to make better decisions. Our key idea is to use two M by N matrices, $(\hat{\theta}_{i,j})_{M\times N}$ and $(n_{i,j})_{M\times N}$, to store the information for each user-resource pair, rather than for each arm as a whole. $\hat{\theta}_{i,j}$ is the average (sample mean) of all the observed values of resource j by user i up to the current time slot (obtained through potentially different sets of arms over time). $n_{i,j}$ is the number of times that resource j has been assigned to user i up to the current time slot.

At each time slot n, after an arm k is played, we get the observation of $Y_{i,j}(n)$ for all $(i,j)\in\mathcal{A}_k$. Then $(\hat{\theta}_{i,j})_{M\times N}$ and $(n_{i,j})_{M\times N}$ (both initialized to 0 at time 0) are updated as follows:

$$\hat{\theta}_{i,j}(n) = \begin{cases} \frac{\hat{\theta}_{i,j}(n-1)n_{i,j}(n-1)+Y_{i,j}(n)}{n_{i,j}(n-1)+1} & \text{, if } (i,j)\in\mathcal{A}_k \\ \hat{\theta}_{i,j}(n-1) & \text{, else} \end{cases} \tag{2}$$

$$n_{i,j}(n) = \begin{cases} n_{i,j}(n-1)+1 & \text{, if } (i,j)\in\mathcal{A}_k \\ n_{i,j}(n-1) & \text{, else} \end{cases} \tag{3}$$

Note that while we indicate the time index in the above updates for notational clarity, it is not necessary to store the matrices from previous time steps while running the algorithm.

Our proposed policy, which we refer to as Matching Learning for Markovian Rewards, is shown in Algorithm 1.

IV. Analysis of Regret

The regret of a policy for a multi-armed bandit problem is traditionally upper-bounded by analyzing the expected number of times that each non-optimal arm is played and then taking the summation over these expectation times the reward difference between an optimal arm and a non-optimal arm all non-optimal arms. Although we could use this approach to analyze the MLMR policy, we notice that the upper-bound for regret consequently obtained is quite loose, which is linear in the number of arms, $P(N,M)$. Instead, we present here a novel analysis for a tighter analysis of the MLMR policy. Our analysis shows an upper bound of the regret that is polynomial in M and N, and uniformly logarithmic over time.

Algorithm 1 Matching Learning for Markovian Rewards (MLMR)

```
// Initialization
for $p = 1$ to $M$ do
    for $q = 1$ to $N$ do
        $n = (M-1)p + q$;
        Play any permutation $k$ such that $(p,q) \in \mathcal{A}_k$;
        Update $(\hat{\theta}_{i,j})_{M\times N}$, $(n_{i,j})_{M\times N}$ accordingly.
    end for
end for
// Main loop
while 1 do
    $n = n + 1$;
    Solve the Maximum Weight Matching problem (e.g.,
    using the Hungarian algorithm [8]) on the bipartite
    graph of users and resources with edge weights
    $\left(\hat{\theta}_{i,j} + \sqrt{\frac{L\ln n}{n_{i,j}}}\right)_{M\times N}$ to play arm $k$ that maximizes
        $\sum_{(i,j)\in\mathcal{A}_k}\left(\hat{\theta}_{i,j} + \sqrt{\frac{L\ln n}{n_{i,j}}}\right)$    (4)
    where $L$ is a positive constant.
    Update $(\hat{\theta}_{i,j})_{M\times N}$, $(n_{i,j})_{M\times N}$ accordingly.
end while
```

The following lemmas are needed for our main results in Theorem 1:

Lemma 1: (Lemma 2.1 from [5]) $\{X_n, n = 1,2,\ldots\}$ is an irreducible aperiodic Markov chain with state space S, transition matrix P, a stationary distribution π_z, $\forall z \in S$, and an initial distribution $\mathbf{q}$. Denote F_t as the σ-algebra generated by $X_1, X_2, \ldots, X_t$. Let G be a σ-algebra independent of $F = \vee_{t\geq 1} F_t$. Let τ be a stopping time with respect to the increasing family of σ-algebra $G \vee F_t, t \geq 1$. Define $N(z,\tau)$ such that $N(z,\tau) = \sum_{t=1}^{\tau} I(X_t = z)$. Then,

$$|E[N(z,\tau) - \pi_z E[\tau]]| \leq A_P, \tag{5}$$

for all $\mathbf{q}$ and all τ such that $E[\tau] < \infty$. A_P is a constant that depends on P.

Lemma 2: (Corollary 1 from [6]) Let $\pi_{\min}$ be the minimum value among the stationary distribution, which is defined as $\pi_{\min} = \min_{z\in S} \pi_z$. Then $A_P \leq 1/\pi_{\min}$.

Lemma 3: For user-resource matching, if the state of reward associated with each user-resource pair (i,j) is given by a Markov chain, denoted $\{X_1^{i,j}, X_2^{i,j}, \ldots\}$, satisfying the properties of Lemma 1, then the regret under policy α is bounded by:

$$R^{\alpha}(n) \leq \sum_{k=1}^{P(N,M)} (\mu^* - \mu^k) E_{\alpha}[T_k^{\alpha}(n)] + A_{\mathbf{S},\mathbf{P},\Theta}, \quad (6)$$

where $A_{\mathbf{S},\mathbf{P},\Theta}$ is a constant that depends on all the state spaces $\{S_{i,j}\}_{1\leq i\leq M, 1\leq i\leq N}$, transition probability matrices $\{\mathbf{P}_{i,j}\}_{1\leq i\leq M, 1\leq i\leq N}$ and the rewards set $\{\theta_{i,j}^z, z \in S_{i,j}\}_{1\leq i\leq M, 1\leq i\leq N}$.

Proof: See [9]. ■

Lemma 4: (Theorem 2.1 from [10]) Let $\{X_n, n = 1, 2, \ldots\}$ be an irreducible aperiodic Markov chain with finite state space S, transition matrix $\mathbf{P}$, a stationary distribution π_z, $\forall z \in S$, and an an initial distribution $\mathbf{q}$. Let $N_{\mathbf{q}} = ||(\frac{q_z}{\pi_z}), z \in S||_2$. The eigenvalue gap ϵ is defined as $\epsilon = 1 - \lambda_2$, where λ_2 is the second largest eigenvalue of the matrix $\mathbf{P}$. $\forall A \subset S$, define $t_A(n)$ as the total number of times that all states in the set A are visited up to time n. Then $\forall \gamma \geq 0$,

$$P(t_A(n) - n\pi_A \geq \gamma) \leq (1 + \frac{\gamma\epsilon}{10n} N_{\mathbf{q}} e^{-\gamma^2\epsilon/20n}), \quad (7)$$

where $\pi_A = \sum_{z\in A} \pi_z$.

Our main results on the regret of MLMR policy are shown in Theorem 1. We show that with using a constant L which is bigger than a value determined by the minimum eigenvalue gap of the transition matrix, maximum value of the number of states, and maximum value of the rewards, our MLMR policy is guaranteed to achieve a regret that is uniformly logarithmic in time, and polynomial in the number of users and resources.

Theorem 1: When using any constant $L \geq \frac{(50+40M)\theta_{\max}^2 s_{\max}^2}{\epsilon_{min}}$, the expected regret under the MLMR policy specified in Algorithm 1 is at most

$$\left[\frac{4M^3NL\ln n}{(\Delta_{\min})^2} + MN + \right. \\ \left. M^2N\frac{s_{\max}}{\pi_{\min}}\left(1 + \frac{\epsilon_{\max}\sqrt{L}}{10s_{\min}\theta_{\min}}\right)\frac{\pi}{3}\right]\Delta_{\max} + A_{\mathbf{S},\mathbf{P},\Theta}, \quad (8)$$

where $\Delta_{\min} = \min_{k:\mu_k<\mu^*}(\mu^* - \mu_k)$, $\Delta_{\max} = \max_k(\mu^* - \mu_k)$, $\pi_{\min} = \min_{1\leq i\leq M, 1\leq j\leq N, z\in S_{i,j}} \pi_z^{i,j}$, $s_{\max} = \max_{1\leq i\leq M, 1\leq j\leq N} |S_{i,j}|$, $s_{\min} = \min_{1\leq i\leq M, 1\leq j\leq N} |S_{i,j}|$, $\theta_{\max} = \max_{1\leq i\leq M, 1\leq j\leq N, z\in S_{i,j}} \theta_z^{i,j}$, $\theta_{\min} = \min_{1\leq i\leq M, 1\leq j\leq N, z\in S_{i,j}} \theta_z^{i,j}$, $\epsilon_{\max} = \max_{1\leq i\leq M, 1\leq j\leq N} \epsilon_{i,j}$, $\epsilon_{\min} = \min_{1\leq i\leq M, 1\leq j\leq N} \epsilon_{i,j}$. $\epsilon_{i,j}$ is eigenvalue gap, defined as $1 - \lambda_2$, where λ_2 is the second largest eigenvalue of $\mathbf{P}_{i,j}$. $C_{\mathbf{S},\mathbf{P},\Theta}$ follows the definition in Lemma 3.

Proof: Below is a sketch of the proof. A detailed proof can be found in [9].

We use $*$ index indicating that a parameter is for the optimal arm. If there are multiple optimal arms, $*$ refers to any of them. We denote n_i^k as $n_{i,j}$ such that $(i,j) \in \mathcal{A}_k$ at current time slot. Denote μ^k as $\sum_{(i,j)\in\mathcal{A}_k} \mu^{i,j}$. We define $\hat{\overline{\theta}}_k(n)$ as $\hat{\overline{\theta}}_k(n) = \sum_{(i,j)\in\mathcal{A}_k} \hat{\theta}_{i,j}(n)$. It is the summation of all the average observation values in arm k at time n.

Denote $C_{t,n}$ as $\sqrt{\frac{L\ln t}{n}}$. Denote $C_{t,\mathbf{n}_{A_k}} = \sum_{(i,j)\in\mathcal{A}_k} \sqrt{\frac{L\ln t}{n_{i,j}}} = \sum_{i=1}^{M} \sqrt{\frac{L\ln t}{n_i^k}} = \sum_{i=1}^{M} C_{t,n_i^k}$. It is also denoted as $C_{t,(n_1^k,\ldots,n_M^k)}$ sometimes for a clear explanation in this proof.

We introduce $\widetilde{T}_{i,j}(n)$ as a counter after the initialization period. It is updated in the following way:

At each time slot after the initialization period, one of the two cases must happen: (1) an optimal arm is played; (2) a non-optimal arm is played. In the first case, $(\widetilde{T}_{i,j}(n))_{M\times N}$ won't be updated. When an non-optimal arm $k(n)$ is picked at time n, there must be at least one $(i,j) \in \mathcal{A}_k$ such that $n_{i,j}(n) = \min_{(i_1,j_1)\in\mathcal{A}_k} n_{i_1,j_1}$. If there is only one such arm, $\widetilde{T}_{i,j}(n)$ is increased by 1. If there are multiple such arms, we arbitrarily pick one, say (i', j'), and increment $\widetilde{T}_{i'j'}$ by 1.

Each time when a non-optimal arm is picked, exactly one element in $(\widetilde{T}_{i,j}(n))_{M\times N}$ is incremented by 1. This implies that the total number that we have played the non-optimal arms is equal to the summation of all counters in $(\widetilde{T}_{i,j}(n))_{M\times N}$. We denote $T_k(n)$ as number of times arm k has been played by MLMR in the first n time slots. Therefore, we have:

$$\sum_{k:\mu_k<\mu*} \mathbb{E}[T_k(n)] = \sum_{i=1}^{M}\sum_{j=1}^{N} \mathbb{E}[\widetilde{T}_{i,j}(n)]. \quad (9)$$

Also note for $\widetilde{T}_{i,j}(n)$, the following inequality holds: $\widetilde{T}_{i,j}(n) \leq n_{i,j}(n), \forall 1 \leq i \leq M, 1 \leq j \leq N$.

Denote by $\widetilde{I}_{i,j}(n)$ the indicator function which is equal to 1 if $\widetilde{T}_{i,j}(n)$ is added by one at time n. Let l be an arbitrary positive integer. Then: $\widetilde{T}_{i,j}(n) \leq l + \sum_{t=MN+1}^{n} \mathbb{1}\{\widetilde{I}_{i,j}(t), \widetilde{T}_{i,j}(t-1) \geq l\}$ where $\mathbb{1}(x)$ is the indicator function defined to be 1 when the predicate x is true, and 0 when it is false.

When $\widetilde{I}_{i,j}(t) = 1$, there exists some arm such that a non-optimal arm is picked for which $n_{i,j}$ is the minimum in this arm. We denote this arm as $k(t)$ since at each time that $\widetilde{I}_{i,j}(t) = 1$, we may get different arms. Then,

$$\begin{aligned}\widetilde{T}_{i,j}(n) \leq\, & l + \sum_{t=MN}^{n} \mathbb{1}\{\hat{\overline{\theta^*}}(t) + C_{t,\mathbf{n}^*(t)} \\ & \leq \hat{\overline{\theta}}_{k(t)}(t) + C_{t,\mathbf{n}_{\mathcal{A}_{k(t)}}(t)}, \widetilde{T}_{i,j}(t) \geq l\}.\end{aligned} \quad (10)$$

$l \leq \widetilde{T}_{i,j}(t)$ implies, $l \leq \widetilde{T}_{i,j}(t) \leq n_{i,j}(t) = n_i^{k(t)}$. So, $\forall 1 \leq i \leq M, n_i^{k(t)} \geq l$.

We define $\hat{\theta}_{i,n_i^k}^k = \hat{\theta}_{i,j}(n)$ such that $(i,j) \in \mathcal{A}_k$ and $n_{i,j}(n) = n_i^k$. We define $\hat{\overline{\theta}}_{k,n_1^k,\ldots,n_M^k} = \sum_{i=1}^{M} \hat{\theta}_{i,n_i^k}^k$. Then we could bound $\widetilde{T}_{i,j}(n)$ as,

$$\begin{aligned}\widetilde{T}_{i,j}(n) \leq l + \sum_{t=1}^{\infty}[\sum_{n_1^*=1}^{t} \cdots \sum_{n_M^*=1}^{t} \sum_{n_1^{k(t)}=l}^{t} \cdots \sum_{n_M^{k(t)}=l}^{t} \mathbb{1}\{\hat{\overline{\theta^*}}_{n_1^*,\ldots,n_M^*} \\ + C_{t,(n_1^*,\ldots,n_M^*)} \leq \hat{\overline{\theta}}_{k(t),n_1^{k(t)},\ldots,n_M^{k(t)}} + C_{t,(n_1^{k(t)},\ldots,n_M^{k(t)})}\}].\end{aligned}$$

$\hat{\overline{\theta^*}}_{n_1^*,\dots,n_M^*} + C_{t,(n_1^*,\dots,n_M^*)} \leq \hat{\overline{\theta}}_{k(t),n_1^{k(t)},\dots,n_M^{k(t)}} + C_{t,(n_1^{k(t)},\dots,n_M^{k(t)})}$ means that at least one of the following must be true:

$$\hat{\overline{\theta^*}}_{n_1^*,\dots,n_M^*} \leq \mu^* - C_{t,(n_1^*,\dots,n_M^*)}, \tag{11}$$

$$\hat{\overline{\theta}}_{k(t),n_1^{k(t)},\dots,n_M^{k(t)}} \geq \mu_{k(t)} + C_{t,(n_1^{k(t)},\dots,n_M^{k(t)})}, \tag{12}$$

$$\mu^* < \mu_{k(t)} + 2C_{t,(n_1^{k(t)},\dots,n_M^{k(t)})}. \tag{13}$$

Here we first find the upper bound for $Pr\{\hat{\overline{\theta^*}}_{n_1^*,\dots,n_M^*} \leq \mu^* - C_{t,(n_1^*,\dots,n_M^*)}\}$:

$$\begin{aligned} &Pr\{\hat{\overline{\theta^*}}_{n_1^*,\dots,n_M^*} \leq \mu^* - C_{t,(n_1^*,\dots,n_M^*)}\} \\ &\leq \sum_{i=1}^{M} Pr\{\hat{\theta}^*_{i,n_i^*} \leq \mu_i^* - C_{t,n_i^*}\}. \end{aligned}$$

$\forall 1 \leq i \leq M$, the following expressions can be derived:

$$\begin{aligned} &Pr\{\hat{\theta}_{i,n_i^*} \leq \mu_i^* - C_{t,n_i^*}\} \\ &\leq \sum_{z=1}^{|S_i^*|} Pr\{\sum_{l \neq z} n_i^*(l) - n_i^* \sum_{l \neq z} \pi_i^*(z) \geq \frac{n_i^*}{|S_i^*|\theta_i^*(z)} C_{t,n_i^*}\}. \end{aligned} \tag{14}$$

$\forall 1 \leq z \leq |S_i^*|$, applying Lemma 4, we could find the upper bound of each probablilty in (14) as,

$$\begin{aligned} &Pr\{\hat{\theta}_{i,n_i^*} \leq \mu_i^* - C_{t,n_i^*}\} \\ &\leq \sum_{z=1}^{|S_i^*|} \left(1 + \frac{\epsilon_{\max}\sqrt{Lt}}{10 s_{\min}\theta_{\min}}\right) N_{\mathbf{q}_{i,j}} e^{-\frac{L \ln t \epsilon_{\min}}{20 s_{\max}^2 \theta_{\max}^2}} \\ &\leq \frac{s_{\max}}{\pi_{\min}} \left(1 + \frac{\epsilon_{\max}\sqrt{L}}{10 s_{\min}\theta_{\min}}\right) t^{-\frac{L\epsilon_{\min} - 10 s_{\max}^2 \theta_{\max}^2}{20 s_{\max}^2 \theta_{\max}^2}}, \end{aligned} \tag{15}$$

where (15) holds since for any $\mathbf{q}_{i,j}$, $N_{\mathbf{q}_{i,j}} = \left\|\frac{q_z^{i,j}}{\pi_z^{i,j}}, z \in S_{i,j}\right\|_2 \leq \sum_{z=1}^{|S_{i,j}|} \left\|\frac{q_z^{i,j}}{\pi_z^{i,j}}\right\|_2 \leq \frac{1}{\pi_{\min}}$.

Thus,

$$\begin{aligned} &Pr\{\hat{\overline{\theta^*}}_{n_1^*,\dots,n_M^*} \leq \theta^* - C_{t,(n_1^*,\dots,n_M^*)}\} \\ &\leq \frac{M s_{\max}}{\pi_{\min}} \left(1 + \frac{\epsilon_{\max}\sqrt{L}}{10 s_{\min}\theta_{\min}}\right) t^{-\frac{L\epsilon_{\min} - 10 s_{\max}^2 \theta_{\max}^2}{20 s_{\max}^2 \theta_{\max}^2}}. \end{aligned} \tag{16}$$

With the similar calculation, we can also get the upper bound of the probability for (12):

$$\begin{aligned} &Pr\{\hat{\overline{\theta}}_{k(t),n_1^{k(t)},\dots,n_M^{k(t)}} \geq \mu_k + C_{t,(n_1^{k(t)},\dots,n_M^{k(t)})}\} \\ &\leq \frac{M s_{\max}}{\pi_{\min}} \left(1 + \frac{\epsilon_{\max}\sqrt{L}}{10 s_{\min}\theta_{\min}}\right) t^{-\frac{L\epsilon_{\min} - 10 s_{\max}^2 \theta_{\max}^2}{20 s_{\max}^2 \theta_{\max}^2}}. \end{aligned} \tag{17}$$

Note that for $l \geq \left\lceil \frac{4L\ln n}{\left(\frac{\Delta_{k(t)}}{M}\right)^2} \right\rceil$, the following expressions can be derived, $\mu^* - \mu_{k(t)} - 2C_{t,(n_1^{k(t)},\dots,n_M^{k(t)})} \geq 0$. This implies that condition (13) is false when $l = \left\lceil \frac{4L\ln n}{\left(\frac{\Delta_{k(t)}}{M}\right)^2} \right\rceil$. If we let $l = \left\lceil \frac{4L\ln n}{\left(\frac{\Delta_{\min}^{i,j}}{M}\right)^2} \right\rceil$, then (13) is false for all $k(t), 1 \leq t \leq \infty$ where $\Delta_{\min}^{i,j} = \min_k\{\Delta_k : (i,j) \in \mathcal{A}_k\}$.

Therefore, the upper bound of $\mathbb{E}[\widetilde{T}_{i,j}(n)]$ could be derived as,

$$\begin{aligned} \mathbb{E}[\widetilde{T}_{i,j}(n)] &\leq \frac{4M^2 L \ln n}{\left(\Delta_{\min}^{i,j}\right)^2} + 1 \\ &+ M\frac{s_{\max}}{\pi_{\min}} \left(1 + \frac{\epsilon_{\max}\sqrt{L}}{10 s_{\min}\theta_{\min}}\right) \sum_{t=1}^{\infty} 2t^{-\frac{L\epsilon_{\min} - (40M+10) s_{\max}^2 \theta_{\max}^2}{20 s_{\max}^2 \theta_{\max}^2}} \\ &\leq \frac{4M^2 L \ln n}{\left(\Delta_{\min}^{i,j}\right)^2} + 1 + M\frac{s_{\max}}{\pi_{\min}} \left(1 + \frac{\epsilon_{\max}\sqrt{L}}{10 s_{\min}\theta_{\min}}\right) \sum_{t=1}^{\infty} 2t^{-2} \\ &= \frac{4M^2 L \ln n}{\left(\Delta_{\min}^{i,j}\right)^2} + 1 + M\frac{s_{\max}}{\pi_{\min}} \left(1 + \frac{\epsilon_{\max}\sqrt{L}}{10 s_{\min}\theta_{\min}}\right) \frac{\pi}{3}, \end{aligned} \tag{18}$$

where (18) holds since $L \geq \frac{(50+40M)\theta_{\max}^2 s_{\max}^2}{\epsilon_{min}}$.

So under our MLMR policy,

$$\begin{aligned} R_\pi(n) &\leq \Delta_{\max} \sum_{i=1}^{M}\sum_{j=1}^{N} \mathbb{E}[\widetilde{T}_{i,j}(n)] + A_{\mathbf{S},\mathbf{P},\Theta} \\ &\leq \left[\frac{4M^3 N L \ln n}{(\Delta_{\min})^2} + MN \right. \\ &\left. + M^2 N \frac{s_{\max}}{\pi_{\min}} \left(1 + \frac{\epsilon_{\max}\sqrt{L}}{10 s_{\min}\theta_{\min}}\right) \frac{\pi}{3}\right] \Delta_{\max} + A_{\mathbf{S},\mathbf{P},\Theta}. \end{aligned} \tag{19}$$

■

Theorem 1 shows when we use a constant L which is large enough such that $L \geq \frac{(50+40M)\theta_{\max}^2 s_{\max}^2}{\epsilon_{min}}$, the regret of Algorithm 1 is upper-bounded uniformly over time n by a function that grows as $O(M^3 N \ln n)$. However, when $\theta_{\max}$, $s_{\max}$ or ϵ_{min} is unknown, the upper bound of regret could not be guaranteed to grow logarithmically in n.

So we extend the MLMR policy to achieve a regret that is bounded uniformly over time n by a function that grows as $O(M^3 N L(n) \ln n)$, by using any arbitrarily slowly diverging non-decreasing sequence $L(n)$ such that $L(n) \leq n$ for any n in Algorithm 1. Since $L(n)$ could grow arbitrarily slowly, the MLMR could achieve a regret arbitrarily close to the logarithmic order. We present our analysis in Theorem 2.

***Theorem** 2:* When using any arbitrarily slowly diverging non-decreasing sequence $L(n)$ (i.e., $L(n) \to \infty$ as $n \to \infty$) in (4) such that $\forall n$, $L(n) \leq n$, the expected regret under the MLMR policy specified in Algorithm 1 is at most

$$\left[\frac{4M^3NL(n)\ln n}{(\Delta_{\min})^2} + MNB_{\mathbf{S},\mathbf{P},\Theta}\right.$$
$$\left. + M^2N\frac{s_{\max}}{\pi_{\min}}\left(1+\frac{\epsilon_{\max}}{10s_{\min}\theta_{\min}}\right)\frac{\pi}{3}\right]\Delta_{\max} + A_{\mathbf{S},\mathbf{P},\Theta}, \tag{20}$$

where $B_{\mathbf{S},\mathbf{P},\Theta}$ is a constant that depends on $\theta_{\max}$, $s_{\max}$ and ϵ_{min}.

Proof: See [9]. ■

V. Examples and Simulation Results

We consider a system that consists of $M = 2$ users and $N = 4$ resources. The state of each resource evolves as an irreducible, aperiodic Markov chain with two states "0" and "1". For all the tables in this section, the element in the i-th row and j-th column represents the value for the user-resource pair (i, j). The transition probabilities are shown below:

0.5	0.4	0.7	0.3
0.2	0.9	0.9	0.7

$\mathbf{p}_{01}$

0.6	0.7	0.8	0.9
0.9	0.5	0.4	0.4

$\mathbf{p}_{10}$

The rewards on each states are:

0.6	0.5	0.2	0.4
0.3	0.7	0.8	0.3

$\boldsymbol{\theta}_0$

0.8	0.2	0.7	0.5
0.5	0.3	0.6	0.6

$\boldsymbol{\theta}_1$

For $1 \leq i \leq M$, $1 \leq j \leq N$, the stationary distribution of user-resource pair (i, j) on state "0" is calculated as $\frac{p_{10}^{i,j}}{p_{01}^{i,j}+p_{10}^{i,j}}$; the stationary distribution on state "1" is calculated as $\frac{p_{01}^{i,j}}{p_{01}^{i,j}+p_{10}^{i,j}}$. The eigenvalue gap is $\epsilon_{i,j} = p_{01}^{i,j} + p_{10}^{i,j}$. The expected reward $\mu_{i,j}$ for all the pairs can be calculated as:

0.6909	0.3909	0.4333	0.425
0.3363	0.4429	0.6615	0.4909

$\boldsymbol{\mu}$

We can see that the arm $\{(1, 1), (2, 3)\}$ is the optimal arm with greatest expected reward $\mu^* = 0.6909+0.6615 = 1.3524$. $\Delta_{\min} = 0.1706$.

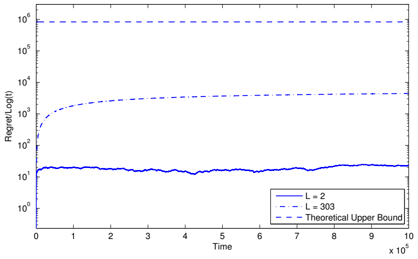

Fig. 1. Simulation Results of Example 1 with $\Delta_{\min} = 0.1706$

Figure 1 shows the simulation result of the regret (normalized with respect to the logarithm of time) for our MLMR policy for the above system with different choices of L. We also show the theoretical upper bound for comparison. The value of L to satisfy the condition in Theorem 1 is $L \geq \frac{(50+40M)R^2s_{\max}^2}{\epsilon_{min}} = 303$, so we picked $L = 303$ in the simulation.

Note that in the proof of Theorem 1, when $L < \frac{(50+40M)R^2s_{\max}^2}{\epsilon_{min}}$, $-\frac{L\epsilon_{\min}-(40M+10)s_{\max}^2\theta_{\max}^2}{20s_{\max}^2\theta_{\max}^2} > -2$. This implies $\sum_{t=1}^{\infty} 2t^{-\frac{L\epsilon_{\min}-(40M+10)s_{\max}^2\theta_{\max}^2}{20s_{\max}^2\theta_{\max}^2}}$ does not converge anymore and thus we could not bound $\mathbb{E}[\widetilde{T}_{i,j}(n)]$ any more. Empirically, however, in 1 the case when $L = 2$ also seems to yield logarithmic regret over time and the performance is in fact better than $L = 303$, since the non-optimal arms are played less when L is smaller. However, this may possibly be due to the fact that the cases when $\widetilde{T}_{i,j}(n)$ grows faster than $\ln(t)$ only happens with very small probability when $L = 2$.

VI. Conclusion

We have presented the MLMR policy for the problem of learning combinatorial matchings of users to resources when the reward process is Markovian. We showed that this policy requires only polynomial storage and computation per step, and yields a regret that grows uniformly logarithmically over time and polynomially with the number of users and resources.

In future work, we would like to also consider the case when the rewards evolve not just when a user-resource pair is selected, but rather at each discrete time. Further, we would like to investigate if it is possible to analyze regret with respect to the best non-static policy, which would be a stronger notion of regret than that considered in this paper but is much harder to analyze. Finally, exploring distributed schemes is also of interest, though likely to be highly challenging in case of limited information exchange between users.

References

[1] Y. Gai, B. Krishnamachari, and R. Jain, "Learning multiuser channel allocations in cognitive radio networks: a combinatorial multi-armed bandit formulation," *IEEE Symposium on International Dynamic Spectrum Access Networks (DySPAN)*, April, 2010.

[2] K. Liu and Q. Zhao, "Decentralized multi-armed bandit with multiple distributed players," *Information Theory and Applications Workshop (ITA)*, January, 2010.

[3] A. Anandkumar, N. Michael, and A. K. Tang. "Opportunistic spectrum access with multiple users: learning under competition," *IEEE International Conference on Computer Communications*, March, 2010.

[4] T. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4-22, 1985.

[5] V. Anantharam, P. Varaiya, and J . Walrand, "Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part II: markovian rewards", *IEEE Transactions on Automatic Control*, vol. 32, no. 11, pp. 977-982, 1987.

[6] C. Tekin, M. Liu, "Online algorithms for the multi-armed bandit problem with markovian rewards", Allerton Conference, September, 2010.

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

[8] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1, pp. 83-97, 1955.

[9] Y. Gai, B. Krishnamachari and M. Liu, "On the Combinatorial Multi-Armed Bandit Problem with Markovian Rewards," Technical Report, March, 2011. Available at http://anrg.usc.edu/www/publications/papers/CRMAB2011.pdf.

[10] D. Gillman, "A chernoff bound for random walks on expander graphs," *SIAM Journal on Computing*, vol. 27, no. 4, pp. 1203-1220, 1998