Online Learning for Combinatorial Network Optimization with Restless Markovian Rewards

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

***Abstract*—Combinatorial network optimization algorithms that compute optimal structures taking into account edge weights form the foundation for many network protocols. Examples include shortest path routing, minimal spanning tree computation, maximum weighted matching on bipartite graphs, etc. We present CLRMR, the first online learning algorithm that efficiently solves the stochastic version of these problems where the underlying edge weights vary as independent Markov chains with unknown dynamics.**

The performance of an online learning algorithm is characterized in terms of regret, defined as the cumulative difference in rewards between a suitably-defined genie, and that obtained by the given algorithm. We prove that, compared to a genie that knows the Markov transition matrices and uses the single-best structure at all times, CLRMR yields regret that is polynomial in the number of edges and nearly-logarithmic in time.

I. Introduction

The following abstract description of combinatorial network optimization covers many graph theoretic algorithms that form the basis of network protocol design in wired and wireless networks. Given a graph $G = (V, E)$, where each edge $e \in E$ is associated with a weight w_e, find a structure consisting of a collection of edges satisfying some given property (e.g., a path, a tree, a matching, or an independent set), that maximizes or minimizes the sum of the weights on the selected edges. This kind of linear network combinatorial optimization covers, for instance, shortest path and minimum spanning tree computation used in routing protocols, and maximum-weight matching used for channel scheduling and switching.

In practice, the edge weights may correspond to some link quality metric of interest such as packet reception ratio, delay, or throughput. In such a case, the edge weights are often stochastically varying with time. Moreover, the dynamics may not be known *a priori*. The solution approach to this problem that we advocate here is to combine the estimation and optimization phases jointly via an efficient online learning algorithm.

We present in this paper an online learning algorithm that is designed for the setting where the edge weights are modeled by finite-state Markov chains, with unknown transition matrices. We show that this problem can be modeled as a combinatorial multi-armed bandit problem with restless Markovian rewards.

To characterize the performance of this algorithm, following the convention in the multi-armed bandit literature, we define a notion of regret, defined as the difference in reward between a suitably defined model-aware genie and that accumulated by the given algorithm over time. Specifically, in this work, we consider a single-action regret formulation, whereby the genie is assumed to know the transition matrices for all edges, but is constrained to stick with one action (corresponding to a particular network structure) at all times[1]. We prove that our algorithm, which we refer to as CLRMR (Combinatorial Learning with Restless Markov Rewards) achieves a regret that is polynomial in the number of Markov chains (i.e., number of edges), and logarithmic with time. This implies that our learning algorithm, which does not know the transition matrices, asymptotically achieves the maximum time averaged reward possible with any single-action policy, even if that policy is given advanced knowledge of the transition matrices. By contrast, the conventional approach of estimating the mean of each edge weight and then finding the desired network structure via deterministic optimization would incur greater overhead and provide only linearly increasing regret over time, which is not asympotically optimal.

While recent work has shown how to address multi-armed bandits with restless Markovian rewards in the classic non-combinatorial setting [1], and combinatorial multi-armed bandits in the simpler settings of i.i.d. rewards [2] or rested Markovian rewards [3], this paper is the first to show how to efficiently implement online learning for stochastic combinatorial network optimization when edge weights are dynamically evolving as restless Markovian processes. We perform simulations to evaluate our new algorithm over two combinatorial network optimization problems: stochastic shortest path routing and bipartite matching for channel allocation, and show that its regret performance is substantially better than that of the algorithm presented in [1], which can handle restless Markovian rewards but does not exploit the dependence between the arms, resulting in a regret that grows exponentially

This research was sponsored in part by the U.S. Army Research Laboratory under the Network Science Collaborative Technology Alliance, Agreement Number W911NF-09-2-0053, and by the U.S. National Science Foundation under award number CNS-1049541.

[1]Although a stronger notion of regret can be defined, allowing the genie to vary the action at each time, the problem of minimizing such a stronger regret is much harder and remains open even for simpler settings than the one we consider here.

in the number of unknown variables.

The rest of the paper is organized as follows. We first provide a survey of prior work in section II. We then present a formal model of the combinatorial restless multi-armed bandit problems in section III. In section IV, we present our CLRMR policy, and show that it requires only polynomial storage. We present our novel analysis of the regret of CLRMR policy in section V. In section VI, we discuss examples and show the numerical simulation results, to show that our proposed policy is widely useful for various interesting combinatorial network optimization problems. We finally conclude our paper in section VII.

II. Related Work

We summarize below the related work, which has treated a) temporally i.i.d. rewards, b) rested Markovian rewards, and c) restless Markovian rewards.

A. Temporally i.i.d. rewards

Lai and Robbins [4] wrote one of the earliest papers on the classic non-Bayesian infinite horizon multi-armed bandit problem. They assume K independent arms, each generating rewards that are i.i.d. over time obtained from a distribution that can be characterized by a single-parameter. For this problem, they present a policy that provides an expected regret that is $O(K \log n)$, i.e. linear in the number of arms and asymptotically logarithmic in n. Anantharam *et al.* extend this work to the case when M simultaneous plays are allowed [5]. The work by Agrawal [6] presents easier to compute policies based on the sample mean that also has asymptotically logarithmic regret. The paper by Auer *et al.* [7] that considers arms with nonnegative rewards that are i.i.d. over time with an arbitrary non-parameterized distribution that has the only restriction that it have a finite support. Further, they provide a simple policy (referred to as UCB1), which achieves logarithmic regret uniformly over time, rather than only asymptotically. Our work utilizes a general Chernoff-Hoeffding-bound-based approach to regret analysis pioneered by Auer *et al.*.

Some recent work has shown the design of distributed multiuser policies for independent arms. Motivated by the problem of opportunistic access in cognitive radio networks, Liu and Zhao [8], Anandkumar *et al.* [9], [10], and Gai and Krishnamachari [11], have developed policies for the problem of M distributed players operating N independent arms.

Our work in this paper is closest to and builds on the recent work by Gai *et al.* which introduced combinatorial multi-armed bandits [2]. The formulation in [2] has the restriction that the reward process must be i.i.d. over time. A polynomial storage learning algorithm is presented in [2] that yields regret that is polynomial in users and resources and uniformly logarithmic in time for the case of i.i.d. rewards.

B. Rested Markovian rewards

There has been relatively less work on multi-armed bandits with Markovian rewards. Anantharam *et al.* [12] 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* (i.e. rewards that only evolve when the arms are selected) Markov chains with transition probability matrices defined by a single parameter θ with identical state spaces. Also, for the upper bound the result is achieved only asymptotically.

For the case of single users and independent arms, a recent work by Tekin and Liu [13] has extended the results in [12] relaxing the requirement of a single parameter and identical state spaces across arms. They propose to use UCB1 from [7] for the multi-armed bandit problem with rested Markovian rewards and prove a logarithmic upper bound on the regret under some conditions on the Markov chain.

In a recent work by Gai *et al.* [3], learning policies for combinatorial multi-armed bandits with rested Markovian rewards have been studied. Unlike [3], we adopt a model with restless Markovian rewards, which has much broader applications in many network optimization problems.

C. Restless Markovian rewards

Restless arm bandits are so named because the arms evolve at each time, changing state even when they are not selected. Work on restless Markovian rewards with single users and independent arms can be found in [1], [14]–[16]. In these papers there is no consideration of possible dependencies among arms, as in our work here.

Tekin and Liu [1] have proposed a RCA policy that achieves logarithmic single-action regret when certain knowledge about the system is known. We use elements of the policy and proof from [1] in this work, which is however quite different in its combinatorial matching formulation (which allows for dependent arms). Liu *et al.* [14], [15] adopted the same problem formulation as in [1], and proposed a policy named RUCB, achieving a logarithmic single-action regret over time when certain system knowledge is known. They also extend the RUCB policy to achieve a near-logarithmic regret asymptotically when no knowledge about the system is available.

Dai *et al.* in [16] adopt a stronger definition of regret: the difference in expected reward compared to a model-aware genie. They develop a policy that yields regret of order arbitrarily close to logarithmic for certain classes of restless bandits with a finite-option structure, such as restless MAB with two states and identical probability transition matrices.

III. Problem Formulation

We consider a system with N edges predefined by some application, where time is slotted and indexed by n. For each edge i ($1 \leq i \leq N$), there is an associated state that evolves as a discrete-time, finite-state, aperiodic, irreducible Markov

chain[2] $\{X^i(n), n \geq 0\}$ with unknown parameters[3]. We denote the state space for the i-th Markov chain by S^i. We assume these N Markov chains are mutually independent. The reward obtained from state x ($x \in S^i$) of Markov chain i is denoted as r_x^i. Denote by π_x^i the steady state distribution for state x. The mean reward obtained on Markov chain i is denoted by μ^i. Then we have $\mu^i = \sum_{z \in S_{i,j}} r_x^i \pi_x^i$. The set of all mean rewards is denoted by $\boldsymbol{\mu} = \{\mu^i\}$.

At each decision period n (also referred to interchangeably as time slot), an N-dimensional action vector $\mathbf{a}(n)$, representing an arm, is selected under a policy $\phi(n)$ from a finite set $\mathcal{F}$. We assume $a_i(n) \geq 0$ for all $1 \leq i \leq N$. When a particular $\mathbf{a}(n)$ is selected, the value of $r_{x_i(n)}^i$ is observed, only for those i with $a_i(n) \neq 0$. We denote by $\mathcal{A}_{\mathbf{a}(n)} = \{i : a_i(n) \neq 0, 1 \leq i \leq N\}$ the index set of all $a_i(n) \neq 0$ for an arm $\mathbf{a}$. We treat each $\mathbf{a}(n) \in \mathcal{F}$ as an arm. The reward is defined as:

$$R^{\mathbf{a}(n)}(n) = \sum_{i \in \mathcal{A}_{\mathbf{a}(n)}} a_i(n) r_{x_i(n)}^i \tag{1}$$

where $x_i(n)$ denotes the state of a Markov chain i at time n.

When a particular arm $\mathbf{a}(n)$ is selected, the rewards corresponding to non-zero components of $\mathbf{a}(n)$ are revealed, i.e., the value of $r_{x_i(n)}^i$ is observed for all i such that $a_i(n) \neq 0$.

The state of the Markov chain evolves *restlessly*, i.e., the state will continue to evolve independently of the actions. We denote by $P^i = (p_{x,y}^i)_{x,y \in S^i}$ the transition probability matrix for the Markov chain i. We denote by $(P^i)' = \{(p^i)'_{x,y}\}_{x,y \in S^i}$ the *adjoint* of P^i on $l_2(\pi)$, so $(p^i)'_{x,y} = p^i_{y,x}\pi^i_y/\pi^i_x$. Denote $\hat{P}^i = (P^i)'P$ as the *multiplicative symmetrization* of P^i. For aperiodic irreducible Markov chains, $\hat{P}^i$s are irreducible [17].

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

$$\begin{aligned} \mathfrak{R}^{\phi}(n) &= n\gamma^* - \mathbb{E}^{\phi}[\sum_{t=1}^{n} R^{\phi(t)}(t)] \\ &= n\gamma^* - \mathbb{E}^{\phi}[\sum_{t=1}^{n} \sum_{i \in \mathcal{A}_{\mathbf{a}(t)}} a_i(t) r_{x_i(t)}^i] \end{aligned} \tag{2}$$

where $\gamma^* = \max_{\mathbf{a} \in \mathcal{F}} \sum_{i \in \mathcal{A}_{\mathbf{a}(n)}} a_i \mu^i$ is the expected reward of the optimal arm. For the rest of the paper, we use $*$ as the index indicating that a parameter is for an optimal arm. If there is more than one optimal arm, $*$ refers to any one of them. We denote by $\gamma^{\mathbf{a}}$ the expected reward of arm $\mathbf{a}$, so $\gamma^{\mathbf{a}} = \sum_{j=1}^{|\mathcal{A}_{\mathbf{a}}|} a_{p_j} \mu^{p_j}$.

[2]We also refer Markov chain $\{X^i(n), n \geq 0\}$ and Markov chain i interchangeably.

[3]Alternatively, for Markov chain $\{X^i(n), n \geq 0\}$, it suffices to assume that the multiplicative symmetrization of the transition probability matrix is irreducible.

For this combinatorial multi-armed bandit problem with restless Markovian rewards, our goal is to design policies that perform well with respect to regret. Intuitively, we would like the regret $\mathfrak{R}^{\phi}(n)$ to be as small as possible. If it is sublinear with respect to time n, the time-averaged regret will tend to zero.

IV. Policy Design

For the above combinatorial MAB problem with restless rewards, we have two challenges here for the policy design:

(1) A straightforward idea is to apply RCA in [1], or RUCB in [14] directly and naively, and ignore the dependencies across the different arms. However, we note that RCA and RUCB both require the storage and computation time that are linear in the number of arms. Since there could be exponentially many arms in this formulation, it is highly unsatisfactory.

(2) Unlike our prior work on combinatorial MAB with rested rewards, for which the transitions only occur each time the Markov chains are observed, the policy design for the restless case is much more difficult, since the current state while starting to play a Markov chain depends not only on the transition probabilities, but also on the policy.

To deal with the first challenge, we want to design a policy which more efficiently stores observations from the correlated arms, and exploits the correlations to make better decisions. Instead of storing the information for each arm, our idea is to use two 1 by N vectors to store the information for each Markov chain. Then an *index* for each each arm is calculated, based on the information stored for underlying components. This *index* is used for choosing the arm to be played each time when a decion needs to be made.

To deal with the second challenge, for each arm $\mathbf{a}$ we note that the multidimensional Markov chain $\{X^{\mathbf{a}}(n), n \geq 0\}$ defined by underlying components as $X^{\mathbf{a}}(n) = (X^i(n))_{i \in \mathcal{A}_{\mathbf{a}}}$ is aperiodic and irreducible. Instead of utilizing the actual sample path of all observations, we only take the observations from a regenerative cycle for Markov chains and discard the rest in its estimation of the *index*.

Our proposed policy, which we refer to as Combinatorial Learning with Restless Markov Reward (CLRMR), is shown in Algorithm 1. Table I summerizes the notation we use for CLRMR algorithm. For Algorithm 1, $(x_i)_{i \in \mathcal{A}_{\mathbf{a}}} = (\zeta^i)_{i \in \mathcal{A}_{\mathbf{a}}}$ means $x_i = \zeta^i, \forall i$.

CLRMR operates in blocks. Figure 1 illustrates one possible realization of this Algorithm 1. At the beginning of each block, an arm $\mathbf{a}$ is picked and within one block, this algorithm always play the same arm. For each Markov chain $\{X^i(n)\}$, we specifiy a state ζ^i at the beginning of the algorithm as a state to mark the regenerative cycle. Then, for the multidimentional Markov chain $\{X^{\mathbf{a}}(n)\}$ associated with this arm, the state $(\zeta^i)_{i \in \mathcal{A}_{\mathbf{a}}}$ is used to define a regenerative cycle for $\{X^{\mathbf{a}}(n)\}$.

Each block is broken into three sub-blocks denoted by SB1, SB2 and SB3. In SB1, the selected arm $\mathbf{a}$ is played until the state $(\zeta^i)_{i \in \mathcal{A}_{\mathbf{a}}}$ is observed. Upon this observation we enter a

Algorithm 1 Combinatorial Learning with Restless Markov Reward (CLRMR)

1: // INITIALIZATION
2: $t = 1$, $t_2 = 1$;
3: $\forall i = 1, \cdots, N$, $m_2^i = 0$, $\bar{z}_2^i = 0$;
4: **for** $b = 1$ to N **do**
5: $t := t + 1$, $t_2 := t_2 + 1$;
6: Play any arm $\mathbf{a}$ such that $b \in \mathcal{A}_{\mathbf{a}}$; denote $(x_i)_{i\in\mathcal{A}_{\mathbf{a}}}$ as the observed state vector for arm $\mathbf{a}$;
7: $\forall i \in \mathcal{A}_{\mathbf{a}(n)}$, let ζ^i be the first state observed for Markov chain i if ζ^i has never been set; $\bar{z}_2^i := \frac{\bar{z}_2^i m_2^i + r_{x_i}^i}{m_2^i+1}$, $m_2^i := m_2^i + 1$;
8: **while** $(x_i)_{i\in\mathcal{A}_{\mathbf{a}}} \neq (\zeta^i)_{i\in\mathcal{A}_{\mathbf{a}}}$ **do**
9: $t := t + 1$, $t_2 := t_2 + 1$;
10: Play arm $\mathbf{a}$; denote $(x_i)_{i\in\mathcal{A}_{\mathbf{a}}}$ as the observed state vector;
11: $\forall i \in \mathcal{A}_{\mathbf{a}(n)}$, $\bar{z}_2^i := \frac{\bar{z}_2^i m_2^i + r_{x_i}^i}{m_2^i+1}$, $m_2^i := m_2^i + 1$;
12: **end while**
13: **end for**
14: // MAIN LOOP
15: **while** 1 **do**
16: // SB1 STARTS
17: $t := t + 1$;
18: Play an arm $\mathbf{a}$ which maximizes

$$\max_{\mathbf{a}\in\mathcal{F}} \sum_{i\in\mathcal{A}_{\mathbf{a}}} a_i \left(\bar{z}_2^i + \sqrt{\frac{L \ln t_2}{m_2^i}} \right); \tag{3}$$

where L is a constant.
19: Denote $(x_i)_{i\in\mathcal{A}_{\mathbf{a}}}$ as the observed state vector;
20: **while** $(x_i)_{i\in\mathcal{A}_{\mathbf{a}}} \neq (\zeta^i)_{i\in\mathcal{A}_{\mathbf{a}}}$ **do**
21: $t := t + 1$;
22: Play an arm $\mathbf{a}$ and denote $(x_i)_{i\in\mathcal{A}_{\mathbf{a}}}$ as the observed state vector;
23: **end while**
24: // SB2 STARTS
25: $t_2 := t_2 + 1$;
26: $\forall i \in \mathcal{A}_{\mathbf{a}(n)}$, $\bar{z}_2^i := \frac{\bar{z}_2^i m_2^i + r_{x_i}^i}{m_2^i+1}$, $m_2^i := m_2^i + 1$;
27: **while** $(x_i)_{i\in\mathcal{A}_{\mathbf{a}}} \neq (\zeta^i)_{i\in\mathcal{A}_{\mathbf{a}}}$ **do**
28: $t := t + 1$, $t_2 := t_2 + 1$;
29: Play an arm $\mathbf{a}$ and denote $(x_i)_{i\in\mathcal{A}_{\mathbf{a}}}$ as the observed state vector;
30: $\forall i \in \mathcal{A}_{\mathbf{a}(n)}$, $\bar{z}_2^i := \frac{\bar{z}_2^i m_2^i + r_{x_i}^i}{m_2^i+1}$, $m_2^i := m_2^i + 1$;
31: **end while**
32: // SB3 IS THE LAST PLAY IN THE WHILE LOOP. THEN A BLOCK COMPLETES.
33: $b := b + 1$, $t := t + 1$;
34: **end while**

N :	number of resources
$\mathbf{a}$:	vectors of coefficients, defined on set $\mathcal{F}$; we map each $\mathbf{a}$ as an arm
$\mathcal{A}_{\mathbf{a}}$:	$\{i : a_i \neq 0, 1 \leq i \leq N\}$
t:	current time slot
t_2:	number of time slots in SB2 up to the current time slot
b:	number of blocks up to the current time slot
m_2^i:	number of times that Markov chain i has been observed during SB2 up to the current time slot
$\bar{z}_2^i$:	average (sample mean) of all the observed values of Markov chain i during SB2 up to the current time slot
ζ^i:	state that determine the regenerative cycles for Markov chain i
x_i:	the observed state when Markov Chain i is played; $(x_i)_{i\in\mathcal{A}_{\mathbf{a}}}$ is the observed state vector if arm $\mathbf{a}$ is played

TABLE I
NOTATION FOR ALGORITHM 1

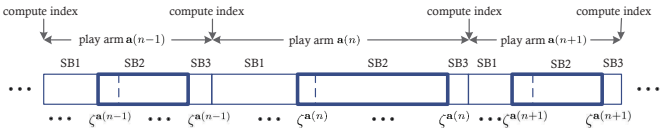

Fig. 1. An illustration of CLRMR

regenerative cycle, and continue playing the same arm untill $(\zeta^i)_{i\in\mathcal{A}_{\mathbf{a}}}$ is observed again. SB2 includes all time slots from the first visit of $(\zeta^i)_{i\in\mathcal{A}_{\mathbf{a}}}$ up to but excluding the second visit to $(\zeta^i)_{i\in\mathcal{A}_{\mathbf{a}}}$. SB3 consists a single time slot with the second visit to $(\zeta^i)_{i\in\mathcal{A}_{\mathbf{a}}}$. SB1 is empty if the first observed state is $(\zeta^i)_{i\in\mathcal{A}_{\mathbf{a}}}$. So SB2 includes the observed rewards for a regenerative cycle of the multidimentional Markov chain $\{X^{\mathbf{a}}(n)\}$ associated with arm $\mathbf{a}$, which implies that SB2 also includes the observed rewards for one or more regenerative cycles for each underlying Markov chain $\{X^i(n)\}, i \in \mathcal{A}_{\mathbf{a}}$.

The key to the algorithm 1 is to store the observations for each Markov chain instead of the whole arm, and utilize the observations only in SB2 for them, and virtually assemble them (highlighted with thick lines in Figure 1). Due to the regenerative nature of the Markov chain, by putting the observations in SB2, the sample path has exactly the same statics as given by the transition probability matrix. So the problem is tractable.

LLR policy requires storage linear in N. We use two 1 by N vectors to store the information for each Markov chain after we play the selected arm at each time slot in SB2. One is $(\bar{z}_2^i)_{1\times N}$ in which $\bar{z}_2^i$ is the average (sample mean) of observed values in SB2 up to the current time slot (obtained through potentially different sets of arms over time). The other one is $(m_2^i)_{1\times N}$ in which m_2^i is the number of times that $\{X^i(n)\}$ has been observed in SB2 up to the current time slot.

Line 1 to line 13 are the initialization, for which each Markov chain is observed at least once, and ζ^i is specified as the first state observed for $\{X^i(n)\}$.

After the initialization, at the beginning of each block,

CLRMR selects the arm which solves the maximization problem as in (3). It is a deterministic linear optimal problem with a feasible set $\mathcal{F}$ and the computation time for an arbitrary $\mathcal{F}$ may not be polynomial in N. But, as we show in Section VI, there exist many practically useful examples with polynomial computation time.

V. Analysis of Regret

We summarize some notation we use in the description and analysis of our CLRMR policy in Table II.

We first show in Theorem 1 an upper bound on the total expected number of plays of suboptimal arms.

***Theorem** 1:* When using any constant $L \geq 56(H+1)S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2/\epsilon_{\min}$, we have

$$\sum_{\mathbf{a}:\gamma^{\mathbf{a}}<\gamma^*} (\gamma^* - \gamma^{\mathbf{a}})\mathbb{E}[T^{\mathbf{a}}(n)] \leq Z_1 \ln n + Z_2$$

where

$$Z_1 = \Delta_{\max}\left(\frac{1}{\Pi_{\min}} + M_{\max} + 1\right)\frac{4NLH^2 a_{\max}^2}{\Delta_{\min}^2}$$

$$Z_2 = \Delta_{\max}\left(\frac{1}{\Pi_{\min}} + M_{\max} + 1\right)\left(N + \frac{\pi NHS_{\max}}{3\pi_{\min}}\right)$$

Proof: Below is a sketch of the proof. A detailed proof can be found in [18].

We introduce $\widetilde{B}^i(b)$ as a counter for the regret analysis to deal with the combinatorial arms. After the initialization period, $\widetilde{B}^i(b)$ is updated in the following way: at the beginning of any block when a nonoptimal arm is chosen to be played, find i such that $i = \arg\min_{j\in\mathcal{A}_{\mathbf{a}}(b)} m_2^j$. If there is only one such arm, $\widetilde{B}^i(b)$ is increased by 1. If there are multiple such arms, we arbitrarily pick one, say i', and increment $\widetilde{B}^{i'}$ by 1. Based on the above definition of $\widetilde{B}^i(b)$, each time a non-optimal arm is chosen to be played at the beginning of a block, exactly one element in $(\widetilde{B}^i(b))_{1\times N}$ is incremented by 1. So the summation of all counters in $(\widetilde{B}^i(b))_{1\times N}$ equals the total number of blocks in which we have played non-optimal arms,

$$\sum_{\mathbf{a}:\gamma^{\mathbf{a}}<\gamma^*} \mathbb{E}[B^{\mathbf{a}}(b)] = \sum_{i=1}^{N} \mathbb{E}[\widetilde{B}^i(b)] \tag{4}$$

We also have the following inequality for $\widetilde{B}^i(b)$: $\widetilde{B}^i(b) \leq m_2^i(t(b-1)), \forall 1 \leq i \leq N, \forall b$. Denote by $c_{t,s}$ $\sqrt{\frac{L\ln t}{s}}$. Denote by $\widetilde{I}^i(b)$ the indicator function which is equal to 1 if $\widetilde{B}^i(b)$ is added by one at block b. Let l be an arbitrary positive integer. Then we can get the upper bound of $\mathbb{E}[\widetilde{B}^i(b)]$ as: $\mathbb{E}[\widetilde{B}^i(b)] = \sum_{\beta=N+1}^{b} \mathbb{P}\{\widetilde{I}^i(\beta) = 1\} \leq l + \sum_{\beta=N+1}^{b} \mathbb{P}\{\sum_{k\in\mathcal{A}_{\mathbf{a}^*}} a_k^* g_{t_2(\beta-1),m_2^k(t(\beta-1))}^k \leq \sum_{j\in\mathcal{A}_{\mathbf{a}(h)}} a_j(b) g_{t_2(\beta-1),m_2^j(t(\beta-1))}^j, \widetilde{B}^i(\beta-1) \geq l\}$, where $g_{t,s}^i = \bar{z}_2^i(s) + c_{t,s}$ and $\mathbf{a}(\beta)$ is defined as a non-optimal arm picked at block β when $\widetilde{I}^i(\beta) = 1$. Note that

H :	$\max_{\mathbf{a}} \lvert\mathcal{A}_{\mathbf{a}}\rvert$. Note that $H \leq N$
$\mathbf{a}(\tau)$:	the arm played in time τ
$b(n)$:	number of completed blocks up to time n
$t(b)$:	time at the end of block b
$t_2(b)$:	total number of time slots spent in SB2 up to block b
$B^{\mathbf{a}}(b)$:	total number of blocks within the first b blocks in which arm $\mathbf{a}$ is played
$m_2^i(t_2(b))$:	total number of time slots Markov chain i is observed during SB2 up to block b
$\bar{z}_2^i(s)$:	the mean reward from Markov chain i when it is observed for the s-th time of only those times played during SB2
$T(n)$:	time at the end of the last completed block
$T^{\mathbf{a}}(n)$:	total number of time slots arm $\mathbf{a}$ is palyed up to time $T(n)$
$m_x^i(s)$:	number of times that state x occured when Markov chain i has been observed s times
$Y_1^i(j)$:	vector of observed states from SB1 of the j-th block for playing Markov chain i
$Y_2^i(j)$:	vector of observed states from SB2 of the j-th block for playing Markov chain i
$Y^i(j)$:	vector of observed states from the j-th block for playing Markov chain i
$\hat{\pi}_x^i$:	$\max\{\pi_x^i, 1-\pi_x^i\}$
$\hat{\pi}_{\max}$:	$\max_{i,x\in S^i} \hat{\pi}_x^i$
$\pi_{\min}$:	$\min_{i,x\in S^i} \pi_x^i$
$\pi_{\max}$:	$\max_{i,x\in S^i} \pi_x^i$
ϵ^i:	eigenvalue gap, defined as $1-\lambda_2$, where λ_2 is the second largest eigenvalue of the multiplicative symmetrization of P^i
$\epsilon_{\min}$:	$\min_i \epsilon^i$
$S_{\max}$:	$\max_i \lvert S^i\rvert$
$r_{\max}$:	$\max_{i,x\in S^i} r_x^i$
$a_{\max}$:	$\max_{i\in\mathcal{A}_{\mathbf{a}},\mathbf{a}\in\mathcal{F}} a_i$
$\Delta_{\mathbf{a}}$:	$\gamma^* - \gamma^{\mathbf{a}}$
$\Delta_{\min}$:	$\min_{\gamma^{\mathbf{a}}\leq\gamma^*} \Delta_{\mathbf{a}}$
$\Delta_{\max}$:	$\max_{\gamma^{\mathbf{a}}\leq\gamma^*} \Delta_{\mathbf{a}}$
$\{X^{\mathbf{a}}(n)\}$:	multidimentional Markov chain defined by $X^{\mathbf{a}}(n) = (X^i(n))_{i\in\mathcal{A}_{\mathbf{a}}}$
$\zeta^{\mathbf{a}}$:	$(\zeta^i)_{i\in\mathcal{A}_{\mathbf{a}}}$, state vector that determines the regenerative cycles for $\{X^{\mathbf{a}}(n)\}$
$\Pi_z^{\mathbf{a}}$:	steady state distribution for state z of $\{X^{\mathbf{a}}(n)\}$
$\Pi_{\min}^{\mathbf{a}}$:	$\min_{z\in S^{\mathbf{a}}} \Pi_z^{\mathbf{a}}$
$\Pi_{\min}$:	$\min_{\mathbf{a},z\in S^{\mathbf{a}}} \Pi_z^{\mathbf{a}}$
$M_{z_1,z_2}^{\mathbf{a}}$:	mean hitting time of state z_2 starting from an initial state z_1 for $\{X^{\mathbf{a}}(n)\}$
$M_{\max}^{\mathbf{a}}$:	$\max_{z_1,z_2\in S^{\mathbf{a}}} M_{z_1,z_2}^{\mathbf{a}}$
$\gamma'_{\max}$:	$\max_{\gamma^{\mathbf{a}}\leq\gamma^*} \gamma^{\mathbf{a}}$

TABLE II
Notation for Regret Analysis

$m_2^i = \min_j\{m_2^j : \forall j \in \mathcal{A}_{\mathbf{a}(\beta)}\}$. We denote this arm by $\mathbf{a}(\beta)$ since at each block that $\widetilde{I}^i(\beta) = 1$, we could get different arms.

Note that $l \leq \widetilde{B}^i(\beta-1)$ implies, $l \leq \widetilde{B}^i(\beta-1) \leq m_2^i(t(\beta-1)), \forall j \in \mathcal{A}_{\mathbf{a}(\beta)}$. So we can further derive the upper bound of $\mathbb{E}[\widetilde{B}^i(b)]$ shown in (5), where h_j $(1 \leq j \leq |\mathcal{A}_{\mathbf{a}*}|)$ represents the j-th element in $\mathcal{A}_{\mathbf{a}*}$; p_j $(1 \leq j \leq |\mathcal{A}_{\mathbf{a}(\beta)}|)$ represents the j-th element in $\mathcal{A}_{\mathbf{a}(\beta)}$ or $\mathcal{A}_{\mathbf{a}(t)}$. $\mathcal{A}_{\mathbf{a}(\tau)}$ represents the arm played in the τ-th time slots counting only in SB2. Note that

$$\begin{aligned}
&\mathbb{P}\{\sum_{j=1}^{|\mathcal{A}_{\mathbf{a}*}|} a_{h_j}^* g_{\tau,s_{h_j}}^{h_j} \leq \sum_{j=1}^{|\mathcal{A}_{\mathbf{a}(\tau)}|} a_{p_j}(t) g_{\tau,s_{p_j}}^{p_j}\} \\
&= \mathbb{P}\{\sum_{j=1}^{|\mathcal{A}_{\mathbf{a}*}|} a_{h_j}^* (\bar{z}_2^{h_j}(s_{h_j}) + c_{\tau,s_{h_j}}) \\
&\quad \leq \sum_{j=1}^{|\mathcal{A}_{\mathbf{a}(\tau)}|} a_{p_j}(\tau)(\bar{z}_2^{p_j}(s_{p_j}) + c_{\tau,s_{p_j}})\} \\
&= \mathbb{P}\{\text{At least one of the following must hold:} \\
&\sum_{j=1}^{|\mathcal{A}_{\mathbf{a}*}|} a_{h_j}^* \bar{z}_2^{h_j}(s_{h_j}) \leq \gamma^* - \sum_{j=1}^{|\mathcal{A}_{\mathbf{a}*}|} a_{h_j}^* c_{\tau,s_{h_j}}, \qquad (6) \\
&\sum_{j=1}^{|\mathcal{A}_{\mathbf{a}(\tau)}|} a_{p_j}(\tau)\bar{z}_2^{p_j}(s_{p_j}) \geq \gamma^{\mathbf{a}(\tau)} + \sum_{j=1}^{|\mathcal{A}_{\mathbf{a}(\tau)}|} a_{p_j}(\tau) c_{\tau,s_{p_j}}, \qquad (7) \\
&\gamma^* < \gamma^{\mathbf{a}(\tau)} + 2\sum_{j=1}^{|\mathcal{A}_{\mathbf{a}(\tau)}|} a_{p_j}(\tau) c_{\tau,s_{p_j}}\} \qquad (8)
\end{aligned}$$

Now we show the upper bound on the probabilities of inequalities (6), (7) and (8) separately. We first find an upper bound on the probability of (6). We note that $\mathbb{P}\{\sum_{j=1}^{|\mathcal{A}_{\mathbf{a}*}|} a_{h_j}^* \bar{z}_2^{h_j}(s_{h_j}) \leq \gamma^* - \sum_{j=1}^{|\mathcal{A}_{\mathbf{a}*}|} a_{h_j}^* c_{\tau,s_{h_j}}\} = \sum_{j=1}^{|\mathcal{A}_{\mathbf{a}*}|} \mathbb{P}\{\bar{z}_2^{h_j}(s_{h_j}) \leq \mu^{h_j} - c_{\tau,s_{h_j}}\}$. $\forall 1 \leq j \leq |\mathcal{A}_{\mathbf{a}*}|$, the following expressions can be derived,

$$\mathbb{P}\{\bar{z}_2^{h_j}(s_{h_j}) \leq \mu^{h_j} - c_{\tau,s_{h_j}}\} \leq \frac{|S^{h_j}|}{\pi_{\min}} \tau^{-\frac{L\epsilon_{\min}}{28S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2}} \qquad (9)$$

Note that all the quantities in computing the indices and the probabilities above come from SB2. Connecting these SB2 intervals together we form a continuous sample path which can be viewed as a sample path generated by a multidimensional Markov chain with transition matrix identical to the original arm. This is the reason why we can have (9) for this Markov chain.

Therefore, $\mathbb{P}\{\sum_{j=1}^{|\mathcal{A}_{\mathbf{a}*}|} a_{h_j}^* \bar{z}_2^{h_j}(s_{h_j}) \leq \gamma^* - \sum_{j=1}^{|\mathcal{A}_{\mathbf{a}*}|} a_{h_j}^* c_{\tau,s_{h_j}}\} \leq \frac{HS_{\max}}{\pi_{\min}} \tau^{-\frac{L\epsilon_{\min}}{28S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2}}$. With a similar derivation, we have $\mathbb{P}\{\sum_{j=1}^{|\mathcal{A}_{\mathbf{a}(\tau)}|} a_{p_j}(\tau)\bar{z}_2^{p_j}(s_{p_j}) \geq \gamma^{\mathbf{a}(\tau)} + \sum_{j=1}^{|\mathcal{A}_{\mathbf{a}(\tau)}|} a_{p_j}(\tau) c_{\tau,s_{p_j}}\} \leq \frac{HS_{\max}}{\pi_{\min}} \tau^{-\frac{L\epsilon_{\min}}{28S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2}}$.

Note that when $l \geq \left\lceil \frac{4L\ln t_2(b)}{\left(\frac{\Delta_{\mathbf{a}(\tau)}}{Ha_{\max}}\right)^2} \right\rceil$, (8) is false for τ, which gives, $\gamma^* - \gamma^{\mathbf{a}(\tau)} - 2\sum_{j=1}^{|\mathcal{A}_{\mathbf{a}(\tau)}|} a_{p_j}(\tau) c_{\tau,s_{p_j}} \geq \gamma^* - \gamma^{\mathbf{a}(\tau)} - \Delta_{\mathbf{a}(\tau)} = 0$. Hence, when we let $l \geq \left\lceil \frac{4LH^2 a_{\max}^2 \ln t_2(b)}{\Delta_{\min}^2} \right\rceil$, (8) is false for all $\mathbf{a}(\tau)$. Therefore, we have (10). So

$$\begin{aligned}
\mathbb{E}[\widetilde{B}^i(b)] &\leq \frac{4LH^2 a_{\max}^2 \ln n}{\Delta_{\min}^2} + 1 \\
&+ \frac{HS_{\max}}{\pi_{\min}} \sum_{\tau=1}^{\infty} 2\tau^{-\frac{L\epsilon_{\min} - 56HS_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2}{28S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2}} \qquad (11) \\
&= \frac{4LH^2 a_{\max}^2 \ln n}{\Delta_{\min}^2} + 1 + \frac{HS_{\max}}{\pi_{\min}} \sum_{\tau=1}^{\infty} 2\tau^{-2} \qquad (12) \\
&= \frac{4LH^2 a_{\max}^2 \ln n}{\Delta_{\min}^2} + 1 + \frac{\pi HS_{\max}}{3\pi_{\min}}
\end{aligned}$$

(12) follows since $L \geq 56(H+1)S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2/\epsilon_{\min}$.

According to (4), $\sum_{\mathbf{a}:\gamma^{\mathbf{a}}<\gamma^*} \mathbb{E}[B^{\mathbf{a}}(b)] = \sum_{i=1}^{N} \mathbb{E}[\widetilde{B}^i(b)] \leq \frac{4NLH^2 a_{\max}^2 \ln n}{\Delta_{\min}^2} + N + \frac{\pi NHS_{\max}}{3\pi_{\min}}$.

Note that the total number of plays of arm $\mathbf{a}$ at the end of block $b(n)$ is equal to the total number of plays of arm $\mathbf{a}$ during SB2s (the regenerative cycles of visiting state $\zeta^{\mathbf{a}}$) plus the total number of plays before entering the regenerative cycles plus one more play resulting from the last play of the block which is state $\zeta^{\mathbf{a}}$. So we have

$$E[T^{\mathbf{a}}(n)] \leq \left(\frac{1}{\Pi_{\min}^{\mathbf{a}}} + M_{\max}^{\mathbf{a}} + 1\right) E[B^{\mathbf{a}}(b(n))]$$

Therefore

$$\begin{aligned}
&\sum_{\mathbf{a}:\gamma^{\mathbf{a}}<\gamma^*} (\gamma^* - \gamma^{\mathbf{a}}) \mathbb{E}[T^{\mathbf{a}}(n)] \\
&\leq \Delta_{\max} \sum_{\mathbf{a}:\gamma^{\mathbf{a}}<\gamma^*} \left(\frac{1}{\Pi_{\min}^{\mathbf{a}}} + M_{\max}^{\mathbf{a}} + 1\right) E[B^{\mathbf{a}}(b(n))] \\
&\leq \Delta_{\max} \left(\frac{1}{\Pi_{\min}} + M_{\max} + 1\right) \sum_{\mathbf{a}:\gamma^{\mathbf{a}}<\gamma^*} E[B^{\mathbf{a}}(b(n))] \\
&\leq Z_1 \ln n + Z_2
\end{aligned}$$

where

$$\begin{aligned}
Z_1 &= \Delta_{\max} \left(\frac{1}{\Pi_{\min}} + M_{\max} + 1\right) \frac{4NLH^2 a_{\max}^2}{\Delta_{\min}^2} \\
Z_2 &= \Delta_{\max} \left(\frac{1}{\Pi_{\min}} + M_{\max} + 1\right) \left(N + \frac{\pi NHS_{\max}}{3\pi_{\min}}\right)
\end{aligned}$$

■

Now we show our main results on the regret of CLRMR policy as in Theorem 2.

***Theorem** 2:* When using any constant $L \geq 56(H+1)S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2/\epsilon_{\min}$, the regret of CLRMR can be upper bounded uniformly over time by the following,

$$\mathfrak{R}^{CLRMR}(n) \leq Z_3 \ln n + Z_4 \qquad (13)$$

$$\mathbb{E}[\widetilde{B}^i(b)] \leq l + \sum_{\tau=1}^{t_2(b)} \sum_{s_{h_1}=1}^{\tau-1} \cdots \sum_{s_{h_{|\mathcal{A}^*|}}=1}^{\tau-1} \sum_{s_{p_1}=l}^{\tau-1} \cdots \sum_{s_{p_{|\mathcal{A}_{\mathbf{a}(\beta)}|}}=l}^{\tau-1} \mathbb{P}\{\sum_{j=1}^{|\mathcal{A}_{\mathbf{a}^*}|} a^*_{h_j} g^{h_j}_{\tau,s_{h_j}} \leq \sum_{j=1}^{|\mathcal{A}_{\mathbf{a}(\tau)}|} a_{p_j}(\tau) g^{p_j}_{\tau,s_{p_j}}\} \tag{5}$$

$$\mathbb{E}[\widetilde{B}^i(b)] \leq \left\lceil \frac{4LH^2 a^2_{\max} \ln t_2(b)}{\Delta^2_{\min}} \right\rceil + \sum_{\tau=1}^{t_2(b)} \sum_{s_{h_1}=1}^{\tau-1} \cdots \sum_{s_{h_{|\mathcal{A}^*|}}=1}^{\tau-1} \sum_{s_{p_1}=l}^{\tau-1} \cdots \sum_{s_{p_{|\mathcal{A}_{\mathbf{a}(\beta)}|}}=l}^{\tau-1} \frac{2HS_{\max}}{\pi_{\min}} \tau^{-\frac{L\epsilon^2_{\min}}{28S^2_{\max} r^2_{\max} \bar{\pi}^2_{\max}}} \tag{10}$$

where

$$Z_3 = Z_1 + Z_5 \frac{4NLH^2 a^2_{\max}}{\Delta^2_{\min}}$$

$$Z_4 = Z_2 + \gamma^*(\frac{1}{\pi_{\min}} + M_{\max} + 1) + Z_5(N + \frac{\pi NHS_{\max}}{3\pi_{\min}})$$

and

$$Z_5 = \gamma'_{\max}(\frac{1}{\Pi_{\min}} + M_{\max} + 1 - \frac{1}{\pi_{\max}}) + \gamma^* M^*_{\max}$$

Proof: We show below a sketch of the proof. A detailed proof can be found in [18].

Denote the expectations with respect to policy CLRMR given ζ by E_ζ. Then following from Theorem 1 and $\mathbb{E}_\zeta[n - T(n)] \leq \frac{1}{\Pi_{\min}} + M_{\max} + 1$, the regret can be bounded as

$$\begin{aligned}
\mathfrak{R}^{CLRMR}_\zeta(n) &= \gamma^* \mathbb{E}_\zeta[T(n)] - \mathbb{E}_\zeta[\sum_{t=1}^{T(n)} \sum_{i \in \mathcal{A}_{\mathbf{a}(t)}} a_i(t) r^i_{x_i(t)}] \\
&\quad + \gamma^* \mathbb{E}_\zeta[n - T(n)] - \mathbb{E}_\zeta[\sum_{t=T(n)+1}^{n} \sum_{i \in \mathcal{A}_{\mathbf{a}(t)}} a_i(t) r^i_{x_i(t)}] \\
&\leq Z_1 \ln n + Z_2 + \gamma^*(\frac{1}{\Pi_{\min}} + M_{\max} + 1) \\
&\quad + \left(\sum_{\mathbf{a}} \gamma^{\mathbf{a}} \mathbb{E}_\zeta[T^{\mathbf{a}}(n)] - \mathbb{E}_\zeta[\sum_{t=1}^{T(n)} \sum_{i \in \mathcal{A}_{\mathbf{a}(t)}} a_i(t) r^i_{x_i(t)}] \right)
\end{aligned}$$

Note that the following expressions can be derived,

$$\begin{aligned}
&\sum_{\mathbf{a}} \gamma^{\mathbf{a}} \mathbb{E}_\zeta[T^{\mathbf{a}}(n)] - \mathbb{E}_\zeta[\sum_{t=1}^{T(n)} \sum_{i \in \mathcal{A}_{\mathbf{a}(t)}} a_i(t) r^i_{x_i(t)}] \\
&\leq Q^*(n) \\
&\quad + \sum_{\mathbf{a}: \gamma^{\mathbf{a}} < \gamma^*} \gamma^{\mathbf{a}} (\frac{1}{\Pi^{\mathbf{a}}_{\min}} + M^{\mathbf{a}}_{\max} + 1 - \frac{1}{\pi_{\max}}) \mathbb{E}_\zeta[B^{\mathbf{a}}(b(n))]
\end{aligned}$$

where

$$\begin{aligned}
Q^*(n) &= \gamma^* \mathbb{E}_\zeta[T^*(n)] \\
&\quad - \sum_{i \in \mathcal{A}_{\mathbf{a}^*}} \sum_{y \in S^i} a^*_i r^i_y \mathbb{E}_\zeta[\sum_{j}^{B^*(b(n))} \sum_{Y^i_t \in Y^i(j)} \mathbb{1}(Y^i_t = y)]
\end{aligned} \tag{14}$$

We now consider the upper bound for $Q^*(n)$. We note that the total number of time slots for playing all suboptimal arms is at most logarithmic, so the number of time slots in which the optimal arm is not played is at most logarithmic. We could then combine the successive blocks in which the best arm is played, and denote by $\bar{Y}^*(j)$ the j-th combined block. Denote $\bar{b}^*$ as the total number of combined blocks up to block b. Each combined block $\bar{Y}^*$ starts after discontinuity in playing the optimal arm, so $\bar{b}^*(n)$ is less than or equal to total number of completed blocks in which the best arm is not played up to time n. Thus, $\mathbb{E}_\zeta[\bar{b}^*(n)] \leq \sum_{\mathbf{a}: \gamma^{\mathbf{a}} < \gamma^*} \mathbb{E}_\zeta[B^{\mathbf{a}}(b(n))]$.

Each combined block $\bar{Y}^*$ consists of two sub-blocks: $\bar{Y}^*_1$ which contains the state vectors for the optimal arm visited from beginning of $\bar{Y}^*$ (empty if the first state is ζ^*) to the state right before hitting ζ^* and sub-block $\bar{Y}^*_2$ which contains the rest of $\bar{Y}^*$ (a random number of regenerative cycles). Denote the length of $\bar{Y}^*_1$ by $|\bar{Y}^*_1|$ and the length of $\bar{Y}^*_2$ by $|\bar{Y}^*_2|$. We denote by $\bar{Y}^i_2(j)$ the states for Markov chain i for all $i \in \mathcal{A}_{\mathbf{a}^*}$ in $\bar{Y}^*_2$.

Therefore we get the upper bound for $Q^*(n)$ as

$$\begin{aligned}
Q^*(n) &\leq \sum_{i \in \mathcal{A}_{\mathbf{a}^*}} \sum_{y \in S^i} a^*_i r^i_y \pi^i_y \mathbb{E}_\zeta[\sum_{j=1}^{\bar{b}^*(n)} |\bar{Y}^*_2|] \\
&\quad - \sum_{i \in \mathcal{A}_{\mathbf{a}^*}} \sum_{y \in S^i} a^*_i r^i_y \mathbb{E}_\zeta[\sum_{j=1}^{\bar{b}^*(n)} \sum_{Y^i_t \in \bar{Y}^i_2(j)} \mathbb{1}(Y^i_t = y)] \\
&\quad + \sum_{i \in \mathcal{A}_{\mathbf{a}^*}} \sum_{y \in S^i} \gamma^* \mathbb{E}_\zeta[\sum_{j=1}^{\bar{b}^*(n)} |\bar{Y}^*_1|] \\
&\leq \gamma^* M^*_{\max} \sum_{\mathbf{a}: \gamma^{\mathbf{a}} < \gamma^*} \mathbb{E}_\zeta[B^{\mathbf{a}}(b(n))]
\end{aligned} \tag{15}$$

where the inequality in (15) comes from counting only the rewards obtained in sub-block $\bar{Y}^i_2$ in (14). Hence $\forall \zeta$,

$$\begin{aligned}
&\mathfrak{R}^{CLRMR}_\zeta(n) \leq Z_1 \ln n + Z_2 + \gamma^*(\frac{1}{\pi_{\min}} + M_{\max} + 1) + \\
&(\gamma'_{\max}(\frac{1}{\Pi_{\min}} + M_{\max} + 1 - \frac{1}{\pi_{\max}}) + \gamma^* M^*_{\max}) \mathbb{E}_\zeta[B^{\mathbf{a}}(b(n))] \\
&\leq Z_3 \ln n + Z_4
\end{aligned} \tag{16}$$

where (16) follows from Theorem 1, and $Z_3 = Z_1 + Z_5 \frac{4NLH^2 a^2_{\max}}{\Delta^2_{\min}}$, $Z_4 = Z_2 + \gamma^*(\frac{1}{\Pi_{\min}} + M_{\max} + 1) + Z_5(N + \frac{\pi NHS_{\max}}{3\pi_{\min}})$. Z_5 is defined as $Z_5 = \gamma'_{\max}(\frac{1}{\Pi_{\min}} + M_{\max} + 1 - \frac{1}{\pi_{\max}}) + \gamma^* M^*_{\max}$. ■

Theorem 2 shows when we use a constant $L \geq 56(H+1)S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2/\epsilon_{\min}$, the regret of Algorithm 1 is upper-bounded uniformly over time n by a function that grows as $O(N^3 \ln n)$. However, when $S_{\max}$, $r_{\max}$, $\hat{\pi}_{\max}$ or $\epsilon_{\min}$ (or the bound of them) are unknown, the upper bound of regret can not be guaranteed to grow logarithmically in n.

When no knowledge about the system is available, we extend the CLRMR policy to achieve a regret bounded uniformly over time n by a function that grows as $O(N^3 L(n) \ln n)$, using any arbitrarily slowly diverging non-decreasing sequence $L(n)$ in Algorithm 1. Since $L(n)$ could grow arbitrarily slowly, this modified version of CLRMR, named CLRMR-LN, could achieve a regret arbitrarily close to the logarithmic order. We present our analysis in Theorem 3.

***Theorem** 3:* When using any arbitrarily slowly diverging non-decreasing sequence $L(n)$ (i.e., $L(n) \to \infty$ as $n \to \infty$), and replacing (3) in Algorithm 1 accordingly with

$$\max_{\mathbf{a} \in \mathcal{F}} a_i \left(\bar{z}_2^i + \sqrt{\frac{L(n(t_2)) \ln t_2}{m_2^i}} \right) \tag{17}$$

where $n(t_2)$ is the time when total number of time slots spent in SB2 is t_2, the expected regret under this modified version of CLRMR, named CLRMR-LN policy, is at most

$$\mathfrak{R}^{CLRMR-LN}(n) \leq Z_6 L(n) \ln n + Z_7 \tag{18}$$

where Z_6 and Z_7 are constants.

Proof: Replacing $c_{t,s}$ with $\sqrt{\frac{L(n(t)) \ln t}{s}}$, and replacing L with $L(n(t_2(b)))$ or $L(n(\tau))$ accordingly in the proof of Theorem 1, (4) to (11) still stand.

$L(n(\tau))$ is a diverging non-decreasing sequence, so there exists a constant τ' such that for all $\tau \geq \tau'$, $L(n(\tau)) \geq 56(H+1)S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2/\epsilon^{\min}$, which implies $\tau^{-\frac{L(n(\tau))\epsilon^{\min} - 56HS_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2}{28 S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2}} \leq \tau^{-2}$.

Thus, we have $\mathbb{E}[\widetilde{B}^i(b)] \leq \frac{4L(n)H^2 a_{\max}^2 \ln n}{\Delta_{\min}^2} + 1 + \frac{\pi H S_{\max}}{3\pi_{\min}} + Z_8$, where $Z_8 = \frac{HS_{\max}}{\pi_{\min}} \sum_{\tau=1}^{\tau'-1} 2\tau^{-\frac{L\epsilon^{\min} - 56HS_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2}{28 S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2}}$.

Then we can according have $\sum_{\mathbf{a}:\gamma^{\mathbf{a}} < \gamma^*} (\gamma^* - \gamma^{\mathbf{a}}) \mathbb{E}[T^{\mathbf{a}}(n)] \leq Z_9 L(n) \ln n + Z_2 + \Delta_{\max} \left(\frac{1}{\Pi_{\min}} + M_{\max} + 1 \right) N Z_8$, where $Z_9 = \Delta_{\max} \left(\frac{1}{\Pi_{\min}} + M_{\max} + 1 \right) \frac{4NH^2 a_{\max}^2}{\Delta_{\min}^2}$.

So $\mathfrak{R}^{CLRMR-LN}(n) \leq Z_6 L(n) \ln n + Z_7$, where $Z_6 = Z_9 + Z_5 \frac{4NH^2 a_{\max}^2}{\Delta_{\min}^2}$, $Z_7 = Z_2 + \gamma^* (\frac{1}{\Pi_{\min}} + M_{\max} + 1) + \Delta_{\max} \left(\frac{1}{\Pi_{\min}} + M_{\max} + 1 \right) N Z_7 + Z_5 (N + \frac{\pi N H S_{\max}}{3\pi_{\min}} + N Z_7)$. ∎

VI. Applications and Simulation Results

We now present an evaluation of our policy over stochastic versions of two combinatorial network optimization problems of practical interest: stochastic shortest path (for routing), and stochastic bipartite matching (for channel allocation).

A. Stochastic Shortest Path

In the stochastic shortest path problem, given a graph $G = (V, E)$, with edge weights (D_{ij}) stochastically varying with time as restless Markov chains with unknown dynamics, we seek to find a path between a given source s and destination t with minimum expected delay. We can apply the CLRMR policy to this problem, with some very minor modifications to the policy and the corresponding regret definition to be applicable to a minimization problem instead of maximization.

For the stochastic shortest path problems, each path between s and t is mapped to an arm. Although the number of paths could grow exponentially with the number of Markov chains, $|E|$. CLRMR efficiently solves this problem with polynomial storage $|E|$ and regret scaling as $O(|E|^3 \log n)$.

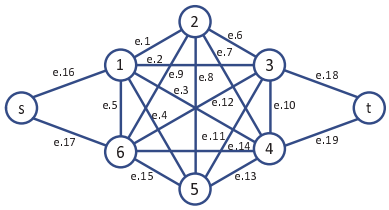

Fig. 2. A graph with 19 links and 260 acyclic paths between s and t for stochastic shortest path routing.

We show the numerical simulation results for the graph in Figure 2. We assume each link has two states with the delay 0.1 on good links, and 1 on bad links. Table III summarizes the transition probabilities on each link.

Link	p_{01}, p_{10}	Link	p_{01}, p_{10}	Link	p_{01}, p_{10}
e.1	0.2, 0.8	e.8	0.3, 0.8	e.15	0.1, 0.8
e.2	0.3, 0.9	e.9	0.1, 0.9	e.16	0.8, 0.1
e.3	0.2, 0.7	e.10	0.9, 0.1	e.17	0.2, 0.7
e.4	0.7, 0.1	e.11	0.3, 0.8	e.18	0.9, 0.1
e.5	0.3, 0.9	e.12	0.2, 0.7	e.19	0.3, 0.8
e.6	0.2, 0.7	e.13	0.8, 0.1		
e.7	0.2, 0.8	e.14	0.4, 0.8		

TABLE III
TRANSITION PROBABILITIES

Figure 3 shows the simulation results. We see that our proposed CLRMR performs better than RCA, the algorithm presented in [1] for all L values considered. If we let $L = 1512$ in this problem, we have that $L \geq 56(H+1)S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2/\epsilon_{\min}$. For lower values of L it is not guaranteed by the analysis that the algorithms should yield logarithmic regret. However, numerically, we find that both policies seem to achieve logarithmic regret, and yield much better regret performance, even for much smaller L values. It is unclear whether this can be proved rigorously or whether it is due low probability events not captured in the simulations.

B. Stochastic Bipartite Matching for Channel Allocation

As a second application, we consider an application in a cognitive radio networks where M secondary users interfering with each other need to be allocated to Q non-conflicting orthogonal channels. We assume that, due to geographic

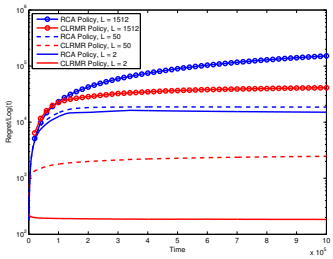

Fig. 3. Comparison of normalized regret $\frac{\Re(n)}{\ln n}$ vs. n time slots for the stochastic shortest path problem.

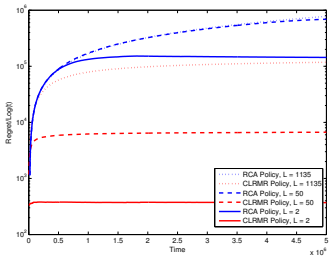

Fig. 4. Comparison of normalized regret $\frac{\Re(n)}{\ln n}$ vs. n time slots for Stochastic Bipartite Matching / Channel Allocation Problem.

dispersion, each user may see different primary user occupancy behavior in each channel. The availability of spectrum opportunities on each user-channel combination (i,j) over a decision period is modeled as a restless two-state Markov chain. It is easy to show that applying CLRMR to this problem yields storage linear in MQ, and a regret bound that scales as $O(\min\{M,Q\}^2 MQ \log n)$, following Theorem 2.

We show simulation results comparing CLRMR again with RCA for a system consisting of 9 orthogonal channels, and 5 secondary users. The transition probability matrix used for these scenarios is presented in table IV.

	ch.1	ch.2	ch.3	ch.4	ch.5	ch.6	ch.7	ch.8	ch.9
u.1	0.5,0.6	0.2,0.7	0.2,0.9	0.8,0.1	0.2,0.7	0.3,0.7	0.2,0.9	0.2,0.7	0.1,0.9
u.2	0.3,0.8	0.1,0.9	0.2,0.8	0.3,0.7	0.3,0.6	0.2,0.8	0.4,0.7	0.2,0.8	0.9,0.2
u.3	0.8,0.1	0.2,0.7	0.3,0.7	0.2,0.8	0.5,0.6	0.2,0.7	0.2,0.7	0.2,0.8	0.1,0.9
u.4	0.3,0.9	0.2,0.8	0.2,0.9	0.4,0.6	0.9,0.2	0.2,0.9	0.2,0.9	0.2,0.9	0.2,0.9
u.5	0.5,0.6	0.2,0.7	0.3,0.9	0.2,0.7	0.5,0.5	0.2,0.7	0.8,0.1	0.3,0.9	0.3,0.9

TABLE IV
TRANSITION PROBABILITIES p_{01}, p_{10} FOR EACH USER-CHANNEL PAIR

The simulation results are shown in Figure 4. As in the stochastic shortest path problem, we find that CLRMR consistently outperforms RCA, for all values of L. Here $L = 1135$ corresponds to ensuring that $L \geq 56(H+1)S_{\max}^2 r_{\max}^2 \hat{\pi}_{\max}^2/\epsilon_{\min}$, which is when the logarithmic regret is guaranteed in theory. However, again, we see that the performance seems to improve in practice with smaller L values, even if it is not be theoretically guaranteed.

VII. CONCLUSION

We have presented CLRMR, a provably efficient online learning policy for stochastic combinatorial network optimization with restless Markovian rewards. This algorithm is widely applicable to many networking problems of interest, as illustrated by our simulation based evaluation of the policy over two different problems: stochastic shortest path and stochastic maximum weight bipartite matching.

One shortcoming of this work is that our focus has been on designing and evaluating the policy with respect to the best single-action policy. However, in general, with restless Markovian rewards, it is possible to further improve performance by developing an algorithm that dynamically switches between different actions over time as the underlying Markov chains evolve. Although this problem is much harder and remains unsolved except in a special case [16], we hope to investigate it further in our future work.

REFERENCES

[1] C. Tekin and M. Liu, "Online learning in opportunistic spectrum access: a restless bandit approach," *IEEE INFOCOM*, Shanghai, April, 2011.
[2] Y. Gai, B. Krishnamachari and R. Jain, "Combinatorial Network Optimization with Unknown Variables: Multi-Armed Bandits with Linear Rewards and Individual Observations," IEEE/ACM Transactions on Networking, vol. 20, no. 5, 2012.
[3] Y. Gai, B. Krishnamachari and M. Liu, "On the combinatorial multi-armed bandit problem with markovian rewards," *IEEE GLOBECOM*, Houston, December, 2011.
[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 I: IID rewards," *IEEE Transactions on Automatic Control*, vol. 32, no. 11, pp. 968-976, 1987.
[6] R. Agrawal, "Sample mean based index policies with O(log n) regret for the multi-armed bandit problem," *Advances in Applied Probability*, vol. 27, no. 4, pp. 1054-1078, 1995.
[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] K. Liu and Q. Zhao, "Distributed learning in multi-armed bandit with multiple players," *IEEE Transactions on Signal Processing*, vol. 58, no.11, pp. 5667-5681, 2010.
[9] A. Anandkumar, N. Michael, and A.K. Tang, "Opportunistic spectrum access with multiple users: learning under competition," *IEEE INFOCOM*, San Diego, March, 2010.
[10] A. Anandkumar, N. Michael, A. Tang, and A. Swami, "Distributed learning and allocation of cognitive users with logarithmic regret," *IEEE JSAC on Advances in Cognitive Radio Networking and Communications*, vol. 29, no. 4, pp. 781-745, 2011.
[11] Y. Gai and B. Krishnamachari, "Decentralized online learning algorithms for opportunistic spectrum access," *IEEE GLOBECOM*, Houston, December, 2011.
[12] 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.
[13] C. Tekin and M. Liu, "Online algorithms for the multi-armed bandit problem with markovian rewards," *Allerton*, Monticello, September, 2010.
[14] H. Liu, K. Liu and Q. Zhao, "Logrithmic weak regret of non-bayesian restless multi-armed bandit,"*IEEE ICASSP*, Prague, May, 2011.
[15] H. Liu, K. Liu, and Q. Zhao,"Learning and sharing in a changing world: non-bayesian restless bandit with multiple players," *ITA*, San Diego, January, 2011.
[16] W. Dai, Y. Gai, B. Krishnamachari and Q. Zhao, "The Non-Bayesian Restless Multi-Armed Bandit: a Case of Near-Logarithmic Regret," *IEEE ICASSP*, Prague, May, 2011.
[17] P. Diaconis and L. Saloff-Coste, "Nash inequalities for finite markov chains," *Journal of Theoretical Probability*, vol. 9, no. 2, pp. 459-510, 1996.
[18] Y. Gai, B. Krishnamachari and M. Liu, "Online learning for combinatorial network optimization with restless markovian rewards," arXiv:1109.1606.